



POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

**Sistemi di Autenticazione e
Autorizzazione: analisi ed
implementazione di un prototipo
originale usabile**

Relatori

prof. Antonio Lioy

ing. Andrea Atzeni

Candidato

Giovanni VAGLICA

ANNO ACCADEMICO 2018-2019

Ai miei genitori

Sommario

L'obiettivo di questa tesi consiste nella realizzazione di un sistema di autenticazione originale, che si discosta dalle tecniche comunemente utilizzate e che sia in grado di fornire un alto livello di sicurezza, senza andare a compromettere l'usabilità dell'utente finale.

La prima parte del lavoro si focalizza sullo studio dello stato dell'arte dei sistemi di autenticazione ed autorizzazione nel panorama attuale. Questo tipo di analisi risulta di fondamentale importanza per capire in che direzione si stanno muovendo le varie tecnologie informatiche, evidenziando eventuali punti di forza e criticità. In parallelo a questa visione puramente tecnica viene poi contrapposto uno studio sull'impatto che gli stessi meccanismi hanno da un punto di vista dell'usabilità. Da questa prospettiva nasce quindi la possibilità di approfondire il rapporto che intercorre tra uomo e macchina, tentando di evincere quali siano i giusti strumenti volti al generale miglioramento della user experience dell'utente finale.

“Sicurezza” ed “usabilità” sono quindi due termini che emergeranno frequentemente all'interno di questo elaborato e che, tramite l'analisi appena affrontata, contribuiranno alla realizzazione del sistema di autenticazione finale. La ricerca termina con lo studio di alcuni casi d'uso, che pongono l'accento sulle possibili problematiche e sui miglioramenti dei sistemi di autenticazione ed autorizzazione analizzati in precedenza, all'interno di scenari realistici.

Il prototipo finale è realizzato in Java e si basa prevalentemente su due meccanismi:

1. Un sistema di autenticazione primario, che viene utilizzato per proteggere i dati e le risorse di un dispositivo. Esso si basa sull'utilizzo di una password grafica, di più facile memorizzazione rispetto ad una stringa alfanumerica. Il sistema, inoltre, risulta ulteriormente irrobustito da un differente metodo di immissione durante il login, atto ad evitare problematiche legate a shoulder surfing e smudge attack.
2. Un sistema di keystroke dynamics, che viene attivato nell'istante successivo all'autenticazione primaria e che permette di fornire una continuous authentication, in grado di riconoscere l'utente in base alla sua scrittura¹. Questo

¹È da precisare che il meccanismo di keystroke dynamics si basa sulla valutazione di parametri comportamentali oggettivi ed unici, che vengono rilevati durante la scrittura e che sono totalmente indipendenti dal contenuto del testo scritto.

tipo di analisi è completamente trasparente per l'utente finale, a cui non verrà chiesto alcun intervento particolare se non lo svolgimento delle sue normali attività. Al contrario, un malintenzionato che tenterà di attuare azioni illecite provocherà il blocco completo del sistema, il cui sblocco potrà avvenire unicamente tramite il sistema di autenticazione primario.

La valutazione del sistema di autenticazione è stata fatta mediante la costruzione di un modello che racchiude metodologie oggettive e soggettive. Le prime utilizzano dei meccanismi, scritti direttamente all'interno del codice del prototipo, che permettono di ricavare informazioni come il tempo, il numero di click e gli eventuali fallimenti rilevati all'interno di ogni singola attività. Viene costruito, in aggiunta, un log dettagliato che descrive tutte le azioni intraprese dal partecipante durante la sua simulazione². Per quanto riguarda invece la metodologia soggettiva, essa è composta da un primo breve test con il quale vengono valutate le capacità informatiche del candidato, e da un questionario finale che viene usato dall'utente per esprimere il livello di sicurezza ed usabilità percepiti durante la prova.

Queste metodologie di valutazione sono state applicate su un totale di 22 partecipanti, con un diverso livello di competenza all'interno del mondo IT. Tra i risultati rilevati è emerso che 19 di loro utilizzerebbero questa tipologia di sistema per proteggere i propri dati sensibili, andando quindi a riporre fiducia in un sistema nuovo che si discosta fortemente dai meccanismi di username/password a cui sono abituati. Un altro dato importante riguarda il grado di sicurezza e di usabilità percepiti, entrambi su livelli molto alti.

Nell'ultima parte della tesi è stato infine realizzato un prototipo di autenticazione avanzato, che si inserisce all'interno di un'attività più realistica e meno prototipale. In particolare, vengono sfruttati i principi di continuous authentication e la tecnica di keystroke dynamics all'interno di una simulazione di scrittura di una mail, in cui viene quindi data piena libertà all'utente finale. Quest'ultimo sistema non ha subito un processo di valutazione, come accaduto con il prototipo originale, in quanto esso viene visto come un punto di partenza per lavori e studi futuri che potranno essere sviluppati attorno a questi concetti.

²Tutti i valori ricavati dalla valutazione oggettiva vengono incapsulati all'interno di una struttura JSON e spediti automaticamente, tramite mail, al termine della simulazione.

Indice

1	Introduzione	1
1.1	Sicurezza e Usabilità	2
1.2	User-Interface e User-Experience	3
1.3	Non Usabilità	4
1.3.1	Digital Rights Management	4
1.3.2	iTunes	5
1.3.3	Windows Update	6
1.4	Conclusione	7
2	Autenticazione	9
2.1	Introduzione	9
2.2	Sistemi di autenticazione password-based	9
2.2.1	Repeatable Textual Password	10
2.2.2	One-Time Password	11
2.2.3	Time-based One-Time Password	11
2.2.4	S/KEY	12
2.3	Sistemi di autenticazione Graphical-based	12
2.4	Shoulder surfing Graphical Password	13
2.4.1	Blonder scheme	13
2.4.2	Cued-click Points	14
2.4.3	Persuasive Cued-Click Points	14
2.5	Shoulder surfing resistant Graphical Password	15
2.5.1	PassMatrix	15
2.5.2	Alignment based Graphical Password authentication system	16
2.5.3	Image Based Registration and Authentication System	18
2.6	Continuous authentication	19
2.7	Physiological Biometric Systems	20

2.7.1	Low FAR & FRR	21
2.7.2	High FAR & FRR	23
2.8	Behavioral Biometric Systems	25
2.8.1	Keystroke dynamics	25
2.8.2	Keystroke dynamics: metodologie avanzate	27
2.8.3	Mobile keystroke dynamics	28
2.8.4	Authentication through an Hidden Markov Model Behavior	29
2.8.5	Authentication by web browsing history	30
2.8.6	Authentication using In-Air Handwriting Style Verification	32
2.8.7	Context-Aware Security by Hierarchical Multilevel Architectures	33
2.8.8	Behavioral biometric system: uno sguardo al futuro	35
2.9	Sistemi multipli in una continuous authentication	36
2.10	Altri sistemi di autenticazione	37
2.10.1	Graphical-based password keystroke dynamics	37
2.10.2	Systems Out-Of-Band-Based	39
2.10.3	Multi-Factor Systems	41
2.10.4	Zero-Trust Authentication	42
2.11	Sistemi federati	43
2.11.1	OpenID Single Sign-On (SSO)	43
2.11.2	Security Assertion Markup Language	45
2.11.3	Open AuTHentication	46
2.12	Framework usati per il management dei sistemi di autenticazione	49
2.12.1	Vault	49
2.12.2	Entrust IdentityGuard Adaptive Authentication	50
2.12.3	IBM Security Access Manager	50
2.13	Conclusione	51
3	Autorizzazione	52
3.1	Politiche e controlli sugli accessi	52
3.1.1	Access Control tramite Path Condition	53
3.1.2	Context-Based Access Control in sistemi mobile	54
3.1.3	Attribute-Based Access Control	56
3.2	eXtensible Access Control Markup Language	57
3.2.1	XACML: tecniche avanzate	58

3.2.2	SAFAX	65
3.3	Open Authorization	67
3.4	Gestione e creazioni delle policy in Vault	69
3.5	Conclusione	69
4	Usabilità	71
4.1	Usabilità nelle Textual Password	71
4.2	Usabilità nelle Graphical Password	73
4.3	Usabilità in una continuous authentication	76
4.4	Usabilità negli altri sistemi di autenticazione	80
4.5	Usabilità nei sistemi di autorizzazione	81
4.5.1	Definizione di una policy	82
4.5.2	Conflitti	83
4.5.3	Design	84
4.6	Valutazione comparativa sull'usabilità	88
4.7	Conclusione	90
5	Scenari e casi d'uso	91
5.1	Scenario #1: acquisto online	92
5.2	Scenario #2: dispositivo personale	95
5.3	Scenario #3: account	98
5.4	Scenario #4: condivisione di risorse presenti su cloud, attraverso l'utilizzo di Temporary Constraint	101
5.5	Scenario #5: condivisione di risorse presenti su cloud, attraverso l'utilizzo di Setting Per User Permission	104
5.5.1	Nozioni teoriche	104
5.6	Scenario #6: continuous authentication	107
5.7	Implementazione prototipale	110
6	Metodologie di valutazione del prototipo	111
6.1	Struttura	111
6.2	Analisi oggettiva	112
6.2.1	Processo di registrazione	112
6.2.2	Processo di autenticazione	112
6.2.3	Completamento di specifici task	112
6.2.4	Completamento della simulazione continuous authentication	113

6.2.5	Numero di fallimenti	113
6.2.6	Log dettagliato	113
6.2.7	Logout	113
6.2.8	Interpretazione dei risultati	113
6.3	Analisi soggettiva	115
6.3.1	Parte 1: Spiegazione	116
6.3.2	Parte 2: Pre-test	116
6.3.3	Parte 3: Questionario finale	117
7	Prototipo	120
7.1	Analisi dei Tool	120
7.2	Tool 1: Mutating Graphical Password System (graphical password)	120
7.2.1	Creazione della password	122
7.2.2	Verifica della password	122
7.2.3	Home	122
7.3	Tool 2: typingDNA (keystroke dynamics)	123
7.3.1	API	123
7.3.2	Analisi del pattern inserito	124
7.4	Costruzione del prototipo	125
7.5	Miglioramenti applicati	126
7.6	Conclusione	128
8	Risultati	129
8.1	Analisi dei dati oggettivi	129
8.1.1	Registrazione	130
8.1.2	Login	131
8.1.3	Attività secondarie	133
8.1.4	Simulazione di keystroke dynamics	134
8.2	Analisi dei dati soggettivi	135
8.3	Conclusione	136
9	Conclusione finale	139

A	Manuale del programmatore	141
A.1	Ambiente di sviluppo e struttura generale del prototipo	141
A.1.1	External library	141
A.1.2	Files	142
A.1.3	Project Screenshots e README.md	143
A.1.4	Sounds	143
A.1.5	UI	143
A.2	Classi	143
A.3	HomePage package	144
A.3.1	Home.java	144
A.3.2	HomePn.java	147
A.4	SignUp package	147
A.4.1	SignUpPanel.java	148
A.4.2	SignUp.java	155
A.4.3	Languages.java	155
A.4.4	BtPanel.java	157
A.4.5	AnimPanel.java	158
A.5	kd package	158
A.5.1	AWT.java	159
A.5.2	MyCipher.java	162
A.6	kdAlgorithm package	162
A.7	Error package	162
A.7.1	ErrorPn.java	163
A.8	Login package	164
A.8.1	LoginPanel.java	165
A.8.2	Login.java	171
A.8.3	Grid.java	172
A.8.4	RandomNum.java	174
A.9	Welcome package	175
A.9.1	WelcomePanel.java	175
A.9.2	Welcome.java	177
A.10	User package	177
A.10.1	UserPn.java	177
A.10.2	UserFrame.java	185

A.10.3 ChgPassPn.java	185
A.10.4 ChgUsr.java	187
A.10.5 LogoutPn.java	191
A.10.6 ShowKd.java	193
A.10.7 UserDel.java	194
A.10.8 UsrHome.java	197
A.11 kdPost package	197
A.11.1 AWTPost.java	198
A.11.2 Request.java	198
A.12 test package	200
A.12.1 WriteTest.java	200
A.12.2 SendDataResult.java	202
A.13 Creazione del file eseguibile	202
B Manuale utente	203
B.1 Requisiti minimi di sistema	203
B.2 Installazione dell'ambiente	203
B.3 Simulazione	205
B.3.1 Pre-test	205
B.3.2 Sistema di autenticazione	205
B.3.3 Test finale	207
C Applicazione del prototipo all'interno di uno scenario avanzato	208
C.1 Funzionamento generale	208
C.2 Manuale del programmatore del prototipo avanzato	210
C.2.1 Registrazione, login ed attività secondarie	210
C.2.2 Simulazione di keystroke dynamics avanzata	211
C.2.3 Schermata di blocco	213
C.2.4 Conferma di avvenuto invio della mail	215
D Risultati individuali grezzi	217
Bibliografia	264

Capitolo 1

Introduzione

Il numero di dispositivi connessi in rete sta aumentando sempre di più negli ultimi anni, incrementando di conseguenza il numero di attacchi informatici. Qualche decina di anni fa, infatti, l'utente si connetteva alla rete unicamente tramite il proprio laptop, rappresentando a tutti gli effetti l'unico mezzo tramite il quale i malintenzionati avevano la possibilità di attaccarlo. Come si evince da un'analisi fatta da RSA nel 2016 [1],

“From mobile devices to routers, hackers are always in search of non-PC platforms (and likely less protected) to compromise”.

Questo viene confermato da Symantec [2] che riscontra un aumento del 214% delle nuove vulnerabilità presenti nel 2015 sui dispositivi mobile rispetto a quelle del 2014. Lo smartphone rappresenta infatti in assoluto uno dei device più utilizzati, portandoci a racchiudere in esso tutta la nostra vita e tutti i nostri dati più sensibili e con il quale ci stiamo abituando a fare sempre più operazioni. Un'analisi fatta da CISCO nel Visual Networking Index di febbraio 2017 [3], stima che il numero di smartphone connessi in rete nel 2021 sarà di circa 11.6 miliardi, incrementando quindi la cifra dei dispositivi potenzialmente coinvolti. Questi dati ci fanno capire come la sicurezza non deve più essere vista dai progettisti come un elemento accessorio, ma come una necessità.

Arrivati a questo punto, è chiaro come la problematica di fondo sia quella di riuscire ad insegnare, comunicare e trasmettere “sicurezza” ad un pubblico che non è abituato a farlo. I fattori umani hanno infatti tradizionalmente avuto un ruolo limitato nella ricerca di sicurezza e nello sviluppo di sistemi sicuri, portando così i progettisti ad ignorare le problematiche di usabilità, sia perché spesso non ne veniva riconosciuta l'importanza e sia perché mancava l'esperienza per affrontarla. Ma questo non rappresenta l'unico problema, come ad esempio dimostra “Apple Pay” [4], un sistema di autenticazione fruibile direttamente dal proprio iPhone, tramite il quale è possibile autorizzare transazioni commerciali. In questo caso non si ha una percezione di minor usabilità del prodotto, che al contrario risulta più semplice ed immediato da utilizzare rispetto ai meccanismi di pagamento tradizionali, ma al contrario di poca sicurezza, causata proprio dall'elevata facilità d'utilizzo, come si denota dall'indagine proposta da Jun Ho Huh [4] su 860 fruitori del sistema.

Analizzando la “IOS Security Guide” [5], si evince come in realtà il sistema implementato da Apple per trasmettere le informazioni riguardanti le carte di credito, in fase di registrazione verso i server, avvenga tramite TLS, mentre il sistema usato per autorizzare la transazione viene fatta attraverso sistemi biometrici, rendendo nel complesso il meccanismo di pagamento più sicuro dei sistemi tradizionali, dove al contrario possono verificarsi una serie di problematiche come quella di shoulder surfing¹. Questo dimostra come una non adeguata comunicazione al pubblico, abbia reso un sistema altamente usabile e sicuro, come quello di Apple, nel complesso poco utilizzato.

1.1 Sicurezza e Usabilità

Ma cosa sono “Usabilità” e “Sicurezza”? E qual è il rapporto che intercorre tra questi due termini? L’ISO 9241 [6], definisce usabile

“Il grado in cui un prodotto può essere usato da particolari utenti per raggiungere certi obiettivi con efficacia, efficienza e soddisfazione, in uno specifico contesto d’uso”.

In aggiunta nel 2003 Shneiderman [7] identifica 5 elementi che permettono di descrivere l’usabilità all’interno di un prodotto:

- Learnability, il tempo richiesto da un utente per imparare le azioni fondamentali.
- Efficiency, il tempo che impiega un utente per eseguire un’azione.
- Errors, il tasso di errori che l’utente compie nello svolgere le azioni.
- Memorability, come l’utente può conservare le sue conoscenze nel tempo.
- Subjective Satisfaction, la soddisfazione provata dall’utente nell’utilizzare quel prodotto.

L’usabilità è quindi un aspetto che si concentra prevalentemente sull’utente fruitore del prodotto e non sul prodotto in sé (nonostante il processo di produzione del prodotto giochi un ruolo fondamentale nello sviluppo della sua usabilità finale), il cui compito principale è quello di ridurre al minimo lo sforzo cognitivo dell’utente.

L’ISO/IEC 2012 [8], seguendo il modello CIA, definisce la sicurezza come l’insieme dei prodotti e dei sistemi in grado di conservare la riservatezza, l’integrità e la disponibilità delle informazioni; in aggiunta al modello CIA, sono state poi accorpate in un secondo momento le proprietà di autenticità, responsabilità, non ripudio ed affidabilità. Matt Bishop [9] espande ulteriormente il concetto di sicurezza definendolo come

¹Tecnica utilizzata per ottenere informazioni come dati personali, password o PIN, osservando l’utente mentre le digita

“Un processo tramite il quale poter identificare i requisiti di sicurezza, sviluppare le norme per garantire quei requisiti e progettare i meccanismi tramite cui implementare tali norme”.

Dopo avere visto questi aspetti da un punto di vista generale, possiamo passare ad analizzarne la loro interazione. “Sicurezza” e “Usabilità” sembrano essere a primo acchito due elementi in disaccordo: la prima ha lo scopo di rendere le azioni dei malintenzionati più difficili da realizzare, mentre la seconda di rendere le azioni svolte dall’utente più semplici da compiere. Nel testo “A Roadmap for Cybersecurity Research” [10], viene analizzato come nei sistemi moderni, a causa proprio dell’aumento del numero di attacchi informatici, si tenda ad incrementare la sicurezza e la complessità di un sistema, riducendo di conseguenza l’usabilità che l’utente avrà sul sistema stesso. Tutto questo porta all’introduzione di confusione e frustrazione nell’utente finale, che non riuscirà a sfruttare al meglio il sistema di sicurezza, portandolo ad effettuare scelte azzardate come ad esempio disabilitare il sistema stesso, magari a causa dell’elevato numero di finestre pop-up che compaiono sullo schermo. Questo rappresenta a tutti gli effetti un esempio concreto in cui la mancanza di usabilità in un prodotto, determina il fallimento del prodotto stesso.

È sbagliato quindi vedere “Sicurezza” e “Usabilità” come due elementi indipendenti, in quanto sono due aspetti che concorrono ad inseguire il medesimo obiettivo: proteggere l’utente nel modo più efficace e semplice possibile, in cui il miglioramento di uno dei due porta anche al miglioramento dell’altro. È infatti vero che spesso l’attaccante di un sistema informatico viene individuato tramite la figura dell’hacker. Ma è anche vero che molte volte l’attaccante di un sistema, è l’utente stesso del sistema.

1.2 User-Interface e User-Experience

Abbiamo visto finora come le scelte intraprese da un utente possono determinare il successo o il fallimento di un prodotto, indipendentemente dalla sua qualità effettiva. Spesso questi errori vengono causati da un’interfaccia utente poco chiara e non intuibile, che può causare confusione e frustrazione all’utilizzatore del sistema. Risulta quindi inutile realizzare un applicativo che fa largo uso della crittografia, e che persegue un rigoroso controllo dei protocolli, se poi l’utente che lo usa si dimentica di premere il relativo pulsante nel momento in cui necessita di privacy, magari perché l’icona non esprime in modo adeguato le sue funzionalità, oppure perché il pulsante stesso risulta difficilmente individuabile all’interno del proprio schermo. Questa tipologia di errori risulta nel complesso gravissima: infatti già vent’anni fa le ricerche riguardanti questa tematica [11] rilevavano che il 90% delle falle di sicurezza che si verificavano in un sistema, erano causate proprio da un’errata progettazione dell’interfaccia utente. Il processo di realizzazione di un’interfaccia utente deve essere quindi parte integrante del processo costruttivo del sistema stesso, e non un suo elemento accessorio. Per questa ragione i progettisti devono anticipare il comportamento umano, andando a realizzare contromisure che siano in grado di prevenire eventuali azioni pericolose intraprese dall’utente. Per fare questo possono fare affidamento a dei modelli, come ad esempio il modello GOMS [12] (Goals, Operators,

Methods, Selection rules) tramite il quale è possibile predire le performance degli utenti, mettendole in relazione con applicativi e dispositivi differenti.

Oltre all'User Interface (UI), che abbiamo visto rappresentare l'insieme degli elementi grafici tramite il quale l'utente interagisce col sistema, esiste anche un altro aspetto che concorre nel definire la relazione che l'utente avrà col sistema: la User Experience (UX). L'ISO FDIS 9241-210 [6] definisce l'user experience come

“L'insieme di percezioni, di comportamenti, di emozioni e di traguardi che l'utente raggiunge prima, dopo e durante l'interazione con un sistema”.

Ma se l'esperienza utente racchiude tutti questi comportamenti, presumibilmente racchiuderà anche l'efficacia e l'efficienza raggiunta dall'utente [13], sottolineando in questo modo l'analogia che intercorre tra usabilità e user experience. D'altro canto, alcuni ricercatori [14] ritengono che l'esperienza utente sia un aspetto puramente soggettivo che si distacca completamente dall'usabilità percepita dall'utente finale. Nonostante questo UI e UX rimangono due elementi cruciali di cui bisogna tenere conto durante la fase di progettazione di un sistema sicuro, dal momento che hanno un'elevata influenza sull'usabilità di quello stesso prodotto.

1.3 Non Usabilità

Analizziamo ora tre esempi concreti in cui la “poca usabilità” ha creato gravi problemi nell'interazione tra utente e sistema (DRM e iTunes) o può creare grossi punti di vulnerabilità in un sistema informatico (Windows Update).

1.3.1 Digital Rights Management

Il NIST, l'Istituto Nazionale della Scienza e della Tecnologia degli Stati Uniti, definisce il DRM (Digital Rights Management) come

“Un sistema di componenti e servizi basati su tecnologie dell'informazione, che, unitamente alle corrispondenti leggi, politiche e modelli di business hanno lo scopo di distribuire e controllare la proprietà intellettuale o i diritti relativi”.

L'obiettivo primario del DRM è quindi quello di proteggere i contenuti digitali, evitando in questo modo la loro diffusione illegale o impedendone la fruizione su più dispositivi. Questo meccanismo se da una parte tutela i diritti dell'autore del contenuto, dall'altra parte potrebbe costituire un problema sull'usabilità dell'utente finale che lo ha acquisito legalmente, ma che potrebbe riscontrare delle limitazioni nell'utilizzo del prodotto digitale.

Fairplay [15] è stato in passato il DRM utilizzato da Apple per proteggere i brani venduti su iTunes, attraverso la combinazione dell'algoritmo di Rijndael e MD5. Il

processo tramite il quale viene gestita la decriptazione del file è affidata ad una coppia di chiavi: la prima, chiamata “Master Key” è presente nel file audio stesso, mentre la seconda, chiamata “User Key”, è necessaria per decriptare la “Master Key”, con la quale si potrà poi accedere al contenuto associato a quella chiave. Ogni volta che viene acquistato un contenuto su iTunes, viene generata e rilasciata all’utente una chiave casuale che è in grado di decriptare la “Master Key” di quel contenuto. Fairplay oltre ad impedire la divulgazione dei contenuti che protegge, impone anche una limitazione sul numero di dispositivi sul quale è possibile fruire di quel contenuto. A partire dall’analisi fatta si può quindi evincere come questo tipo di DRM, impone una serie di vincoli sull’utente finale, limitandone fortemente l’usabilità. Il limite maggiore rimane infatti il controllo circoscritto che l’utente ha di quei contenuti; difatti l’iPod rimane a tutti gli effetti un dispositivo passivo, che non ha alcun potere decisionale, dal momento che tutti i contenuti presenti su di esso, saranno verificati e copiati da iTunes in fase di sincronizzazione. iTunes è stato quindi il grosso vincolo che l’utente ha dovuto pagare per sottostare alle politiche restrittive messe in atto dal DRM di Apple. Dal 2007 Apple ha distribuito il catalogo completo dei brani musicali senza il DRM, con però l’introduzione di un supplemento rispetto al prezzo normale.

In passato l’utilizzo del DRM è stato anche al centro di un’ulteriore polemica, causando dei grossi problemi di sicurezza a livello mondiale [16]. Nel 2005 infatti, Sony decide di mettere in commercio circa 22 milioni di CD in cui le politiche DRM venivano gestite da un rootkit, ovvero una collezione di software tramite il quale è possibile ottenere i diritti di root sul dispositivo installato, senza che la vittima ne abbia dato l’autorizzazione (difatti è uno strumento che viene spesso utilizzato dai cracker). L’utilizzo di questo DRM creava una backdoor che poteva poi essere utilizzata dai malintenzionati per accedere a parti sensibili del dispositivo su cui il rootkit era installato. Ci furono dei grossi problemi legati alla rimozione di questo rootkit, in quanto si scoprì che il suo uninstaller, rilasciato dalla stessa Sony, oltre a rimuoverlo, creava delle grosse falle a livello del sistema operativo, esponendo nuovamente l’utente ad enormi rischi.

In base agli esempi appena trattati si capisce come il DRM sia sicuramente uno strumento utile, ma la cui efficacia può variare a seconda degli ambiti in cui viene utilizzato. Proprio per questa ragione, nell’implementazione del DRM lo sbaglio si Sony è stato quello di voler esercitare un forte controllo sul contenuto senza avere riguardo per le conseguenze che si riflettono sugli utilizzatori del servizio. Un differente impiego del DRM è invece quello di Apple, che al fine di tutelare i propri contenuti e gli utenti che li hanno a disposizione, ha apportato delle significative limitazioni nelle fruizioni dei relativi contenuti digitali. Il punto d’incontro tra le alternative appena viste sarebbe quello di fornire una protezione sui contenuti senza però dimenticare dell’usabilità degli utenti finali, che acquistano quei prodotti e che dovrebbero poter fruire di essi in totale libertà e sicurezza.

1.3.2 iTunes

Dopo avere esaminato qual è l’impatto del DRM sull’esperienza utente, andiamo ora ad analizzare alcuni parametri che evidenziano una bassa usabilità nell’utilizzo

di iTunes, il software impiegato da Apple per gestire, riprodurre e sincronizzare i file multimediali:

- Iniziamo con il valutare qual è il tempo richiesto dall'utente per imparare il funzionamento del software. Rispetto a questo parametro, iTunes rappresenta (nella maggior parte dei casi) un esempio di non usabilità, dal momento che esso si interfaccia più facilmente con macchine Mac rispetto a quelle Windows, dove invece le operazioni risultano più macchinose da eseguire. Questo rappresenta uno svantaggio dal momento che l'utente, per interagire col software, non potrà usufruire dell'esperienza pregressa che ha maturato col tempo, ma dovrà costruirsi un insieme nuovo di competenze.
- Analizzando le operazioni di sincronizzazione tra iTunes ed i dispositivi Apple, è evidente come esse risultino tediose e complesse da eseguire, innalzando di conseguenza lo sforzo cognitivo richiesto dall'utente per usufruire del software. Infatti se i file precedentemente caricati sul device non si trovano sullo stesso dispositivo o nella stessa posizione all'interno del file system, si rischia di subire una perdita parziale o totale dei rispettivi dati. Questo, oltre a rappresentare un forte esempio di non usabilità, è indice di una forte limitazione da parte di Apple nei confronti della libertà dell'utente.
- Sempre per i motivi evidenziati nei punti precedenti, il sistema si presenta poco trasparente nei confronti dell'utente, che non potrà fare uso di tecniche basilari e poco invasive come il "drag & drop".
- Lo sforzo cognitivo richiesto dall'utente per eseguire alcune operazioni, viene spesso compromesso a causa di un'interfaccia piatta e poco consistente, che non è in grado di evidenziare a primo acchito le funzionalità principale del sistema.

1.3.3 Windows Update

Il termine "0-day", definisce l'insieme delle vulnerabilità non ancora pubblicamente note all'interno del panorama informatico che, tramite un "exploit", possono mettere in atto delle vere e proprie minacce nei confronti di tutti quegli utenti che risultano vulnerabili a quegli attacchi. Nell'istante in cui viene scoperta una nuova criticità, l'utente viene inevitabilmente esposto ad un rischio che potrà essere risolto solo nel momento in cui verrà rilasciata la relativa patch. In realtà il grado di attacco di una vulnerabilità appena scoperta non dipende esclusivamente dalla sua portata e dalla sua pericolosità, ma viene sensibilmente influenzato dal suo ciclo di vita, come viene dimostrato nell'articolo "Probability of Attack Based on System Vulnerability Life Cycle" [17], dove vengono studiate 440 vulnerabilità suddivise tra Windows e Linux. In particolare nel testo vengono rilevate cinque differenti cicli di vita:

- Zero-day attack (ZDA), dove in generale una black-hat scopre una vulnerabilità ma essa non viene resa pubblica.

- Pseudo zero-day attack (PZDA), simile alla ZDA, ma in cui gli amministratori di sistema non installano la patch, nonostante essa sia stata già rilasciata tempo prima.
- Potential for pseudo zero-day attack (PPZDA), in cui viene pubblicamente rilasciata la patch.
- Potential for attack (POA), solitamente si verifica dopo la ZDA; in questa fase sono stati realizzati i software in grado di sfruttare quelle vulnerabilità, mentre i venditori non sono in grado di produrre e distribuire la patch su ampia scala.
- Passive, in cui non sono ancora stati prodotti i codici di exploit.

Il ciclo di vita PPZDA, merita un'ulteriore analisi, in quanto la pubblicazione della patch relativa ad una vulnerabilità presente nel sistema, ha un duplice effetto: da una parte risolvere quella minaccia, dall'altra parte rendere pubblica la minaccia stessa. Un esempio è rappresentato dalla politica utilizzata da Microsoft per aggiornare il suo sistema operativo: difatti, nonostante vengano scoperte quotidianamente nuove minacce, per evitare di sommergere l'utente di continui aggiornamenti, si è deciso di rilasciare un'unica patch, il secondo martedì del mese, con il quale risolvere tutte le vulnerabilità riscontrate nel mese precedente; questo giorno è chiamato anche "Patch Tuesday". Nell'istante in cui Microsoft rilascia la patch mensile, tutti i malintenzionati vengono a conoscenza delle vulnerabilità presenti sul sistema e possono concentrarsi nel creare l'exploit che sia in grado di sfruttare quelle vulnerabilità; il giorno successivo al "Patch Tuesday" è infatti chiamato "Exploit Wednesday". Il motivo per il quale i cracker realizzano un'exploit su una vulnerabilità di cui esiste già la patch è semplice: non tutti gli utenti installano quella patch il giorno del suo rilascio. Essi infatti non percepiscono Windows Update come un qualcosa di trasparente, ma al contrario viene visto come un motivo di frustrazione e interruzione del loro flusso lavorativo. Il comportamento medio dell'utente risulterà quindi quello di rimandare la data dell'aggiornamento, estendendo di conseguenza l'arco temporale d'esposizione alle vulnerabilità.

Si è quindi visto come la creazione e il rilascio di una patch non sia sufficiente a rendere un sistema sicuro. Ancora una volta l'attenzione viene spostata sull'utente finale: bisogna sì fare sicurezza, ma deve essere fatta in base ai bisogni, le necessità e le esigenze di quello stesso utente, senza andare in alcun modo a scalfire la sua usabilità; solo in questo modo infatti l'utente potrà usufruire nel modo corretto di quel servizio.

1.4 Conclusione

Gli esempi visti sopra rappresentano solo una piccola parte delle problematiche presenti ogni giorno nei sistemi informatici. L'obiettivo di questa tesi sarà proprio quello di ricercare ed implementare un sistema che risulti facile e allo stesso tempo sicuro per l'utente finale. Per arrivare a questo punto è però necessaria un'analisi critica e comparativa dei sistemi presenti attualmente sul mercato ed è quello che faremo nel capitolo successivo, tramite un confronto dei meccanismi di autenticazione

e autorizzazione, a partire da quelli più classici e ormai conosciuti, fino ad arrivare a quelli più innovativi e non ancora presenti sui dispositivi odierni.

Capitolo 2

Autenticazione

Andremo a trattare in questo capitolo lo stato dell'arte inerente ai sistemi di autenticazione. L'analisi riguarderà prettamente gli aspetti più tecnici, con l'obiettivo di rilevare le mancanze o i punti di forza delle varie categorie e dei sistemi contenuti al loro interno. Le considerazioni inerenti all'usabilità verranno invece trattate successivamente nel capitolo [4](#).

2.1 Introduzione

L'ISO IEC 27000 [\[8\]](#) definisce l'autenticazione come

“Il processo tramite il quale viene confermata la veridicità di un attributo parziale o di un'informazione che sostiene di essere vera da parte di un'entità”.

L'autenticazione rappresenta quindi il primo elemento che un utente si trova a dover affrontare nel momento in cui interagisce con un sistema informatico. Realizzare un sistema di autenticazione che sia usabile e allo stesso tempo sicuro è un problema complesso da affrontare, a causa della vastità di dispositivi e di servizi che può essere affiancato ad uno specifico applicativo; d'altro canto, dare poca importanza a questo fattore determinerebbe inevitabilmente lo sgretolamento dell'intero sistema di sicurezza. Per tutti questi motivi, in questa prima parte del secondo capitolo, andremo ad analizzare ciò che rappresenta lo stato dell'arte per quanto riguarda l'autenticazione nel mondo odierno: questo è fondamentale perché ci permetterà di definire qual è il punto di partenza e quale potrebbe essere la strada giusta da intraprendere per raggiungere l'obiettivo finale di questa tesi.

2.2 Sistemi di autenticazione password-based

Andremo ora ad effettuare un'analisi su una serie di sistemi di autenticazione che si fondano sul principio “password-based”, cercando di capire quali possano essere i limiti e le criticità legati a questa tipologia di meccanismi.

2.2.1 Repeatable Textual Password

Come primo elemento di analisi, andiamo a considerare uno dei sistemi più conosciuti e discussi al mondo: le password testuali ripetibili. Esse rappresentano una chiave d'accesso (che è possibile utilizzare più volte), formata da caratteri alfanumerici ed utilizzata per accedere in modo esclusivo ad una risorsa informatica. Generalmente, risulta in assoluto il sistema di autenticazione più utilizzato, grazie allo sforzo minimo richiesto da parte dell'utente finale, il cui unico compito sarà quello di memorizzare una stringa alfanumerica. Il reale problema di questo sistema risiede però proprio nella figura stessa dell'uomo. La mente umana infatti non è incline a memorizzare stringhe alfanumeriche complesse, tendendo ad utilizzare parole o numeri a lui familiari. Il modo in cui viene utilizzata normalmente la password, la rende sicuramente un sistema altamente usabile ma poco sicuro, in quanto predisposto a dictionary¹ o shoulder surfing attack, trovandosi quindi in accordo con quello detto da Schneier [18]:

“Unfortunately the system of user name and password works less well than people believe”.

Per rendere sicura una password bisognerebbe utilizzare delle stringhe alfanumeriche di almeno 20/30 caratteri, chiamate passphrase, che risultano meno vulnerabili brute-force attack rispetto alle più brevi password tradizionali; inoltre, il contenuto di esse, deve evitare l'utilizzo di termini comuni o di dati anagrafici dell'individuo in quanto sono informazioni che potrebbero essere ricavate attraverso dictionary o social engineering attack². L'utilizzo di passphrase che non usa termini comuni, renderebbe però la password un sistema poco usabile e difficilmente memorizzabile, che potrebbe portare l'utente a situazioni drastiche, come appuntarsi la stringa di caratteri su un foglio di carta o sul PC stesso, aumentando in questo modo notevolmente i rischi. Inoltre, uno studio condotto da Bonneau [19] ha rilevato che gli utenti tendono ad utilizzare un set limitato di stringhe per formare la passphrase, rendendolo nel complesso la sua creazione meno efficace di quanto ci si possa aspettare da un insieme finito di parole casuali. Un meccanismo che si potrebbe introdurre per limitare in parte il problema, potrebbe essere quello di adottare delle blacklist che vietino l'utilizzo di determinate sottostringhe “comuni” all'interno delle password scelte, in modo da ridurre la riuscita di eventuali dictionary attack.

Consideriamo ora il canale trasmissivo all'interno del quale vengono spediti i dati (che deve essere sicuro per evitare attacchi replay) ed il modo in cui il server li memorizza: un'acquisizione in chiaro o tramite digest senza l'ausilio di un “sale”, metterebbe a serio rischio le credenziali di tutti gli utenti presenti in quel server. Un'ultima problematica risiede nel riutilizzo che spesso viene fatto della stessa password su servizi differenti, generando il cosiddetto “effetto domino” [20]; se un malintenzionato riuscisse infatti a ricavare la password associata ad una determinata

¹Tecnica a forza bruta utilizzata per rompere un sistema cifrato, utilizzando un set di codici.

²Tecnica psicologica utilizzata dall'attaccante con l'obiettivo di ricavare informazioni personali e comportamentali della vittima.

risorsa, potrebbe poi riutilizzarla per accedere ad altre, estendendo di conseguenza il suo attacco (magari andando a colpire inizialmente proprio il sito che implementa minor sicurezza [21]).

È evidente come questo sistema di autenticazione abbia molti punti deboli, nonostante sia il meccanismo più utilizzato dalla massa. Per questo motivo lo studio che andrò ad affrontare, sia per quanto riguarda l'analisi tecnica, sia per quanto riguarda la realizzazione del sistema prototipale, si discosterà nettamente da esso, andandosi a concentrare invece su sistemi più sicuri e più in linea con i ragionamenti e la psiche umana.

2.2.2 One-Time Password

Uno dei problemi principali delle password testuali risiede nella vulnerabilità intrinseca nel segreto definito in fase di registrazione dall'utente; basta infatti ricavare quel segreto (sia in chiaro che tramite digest), per compromettere definitivamente l'intero sistema di autenticazione. OTP [22] riesce a risanare questa lacuna attraverso un meccanismo di password testuali la cui validità è legata al suo utilizzo: dopo che sarà adoperata, essa non potrà più essere riutilizzata rendendo quindi del tutto inefficace replay e sniffing attack. Per quanto riguarda il sistema di generazione delle password, esso viene stabilito dal server che ne creerà un elenco a partire da un segreto condiviso con l'utente. È importante specificare che il server non avrà mai l'elenco completo delle password, ma esse saranno di volta in volta calcolate su richiesta del client a partire dal segreto condiviso.

Un problema di OTP riguarda il mezzo trasmissivo tramite il quale il server deve inviare i dati al client, che deve necessariamente avvenire out-of-band, ma di cui discuteremo i dettagli nel paragrafo 2.10.2.

2.2.3 Time-based One-Time Password

TOTP [23] riprende i concetti visti nel paragrafo precedente ma si differenzia dal fatto che la password viene calcolata sul momento dal server e dal client a partire dalla data e dall'ora corrente, assieme al segreto tra loro condiviso. Tutto ciò viene tipicamente realizzato tramite la costruzione di un hardware, posseduto poi dall'utente finale, che è in grado di generare un token code ogni 60 secondi. Affinché il sistema funzioni è necessario che client e server siano perfettamente sincronizzati tra loro: se da un lato il server può utilizzare il protocollo NTP per farlo in modo preciso e costante, dall'altro l'hardware del client subisce un degrado temporale inevitabile, motivo per cui dopo una certa durata il dispositivo risulterà inutilizzabile e dovrà essere sostituito. Inoltre, data l'aggiunta di un secondo elemento di confronto, ovvero il tempo, il sistema potrebbe essere violato da attacchi DoS con il preciso compito di bloccare il client a seguito di un certo numero di tentativi invalidi. È da precisare anche la pericolosità di attacchi basati su tecniche di social engineering, tramite il quale si potrebbe assumere l'identità dell'utente per farsi inizializzare e poi spedire un nuovo token.

2.2.4 S/KEY

Come abbiamo visto, il problema principale di OTP e TOTP risiede nel fatto che le password vengano generate dal server e pertanto deve necessariamente esistere un meccanismo che sia in grado di trasmetterle al client in modo sicuro. S/KEY [24], invece, riesce partendo da un segreto generato dall'utente, a ricavare una serie di One-Time Password che sono univoche ed allo stesso tempo non richiedono alcun intervento di calcolo da parte del server, evitando in questo modo qualsiasi problematica a livello trasmissivo. Il meccanismo funziona nel seguente modo:

1. In una prima fase l'utente sceglie il segreto da cui viene ricavato un elenco di password tramite l'utilizzo di una funzione di hash (generalmente viene utilizzata la funzione MD4), in modo tale che la i -esima password generata sia l'hash della password $i-1$. Il segreto è composto solitamente da una passphrase composta da alcune parole, in lingua inglese, di massimo 4 caratteri l'una. Questo permette, tramite l'utilizzo di un dizionario di 2048 parole, di ridurre il numero di bit che verranno poi trasmessi al server. Infine viene concatenato al segreto un seed, per permettere in questo modo ad un server di memorizzare segreti uguali associati a client differenti.
2. Dopo aver generato le N stringhe, il client comunica al server l'ultima password generata. È da specificare che il server non conoscerà né il segreto in chiaro e né l'intero elenco delle password generate dall'utente. La password N -esima rappresenta però un'informazione di overhead e in quanto tale non può essere usata come mezzo diretto per autenticarsi.
3. Terminata la fase di inizializzazione, si può ora passare a quella di autenticazione; il server invia all'utente un numero i generato casualmente tra 1 e $N-1$. Il client a questo punto si ricaverà sul momento la stringa richiesta, applicando i volte la funzione di hash sul segreto, e inviando al server l'informazione appena ottenuta. Il server, per verificare la veridicità della password, andrà ad applicare $N-i$ volte la funzione di hash sulla stringa inviata dal client il cui risultato verrà confrontato con la password da lui memorizzata: se le due stringhe combaciano, l'utente viene regolarmente autenticato.

È da precisare che il meccanismo S/KEY può funzionare solo grazie all'alta difficoltà che occorre nell'invertire la funzione di hash, caratteristica che gli permette di essere praticamente quasi esente da sniffing e replay attack.

2.3 Sistemi di autenticazione Graphical-based

Analizzeremo ora in questa parte una serie di tecniche che utilizzano tutte, come mezzo di autenticazione primario, le immagini. L'analisi verrà condotta su due categorie distinte, formate dai sistemi che risultano sensibili alle problematiche di shoulder surfing e da quelli che ne sono invece esenti.

2.4 Shoulder surfing Graphical Password

Vedremo in questo paragrafo tre sistemi che si basano tutti sul tocco di aree specifiche all'interno di un'immagine e che, per questo motivo, soffrono della visione da parte di un malintenzionato della sequenza tracciata dall'utente sullo schermo.

2.4.1 Blonder scheme

Lo schema di Blonder [25] prevede che l'utente si autentichi andando a premere in sequenza delle aree all'interno di un'immagine. I punti e il numero di aree vengono scelti in fase di registrazione dall'utente, che dovrà ripetere correttamente la sequenza prestabilita per almeno due volte (figura 2.1).

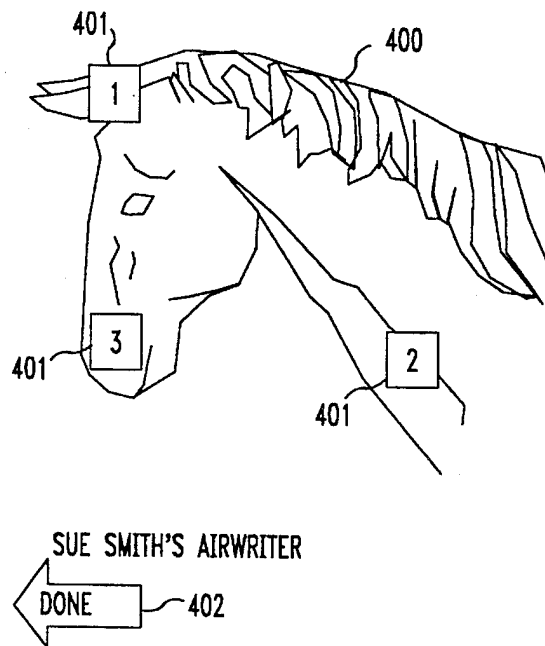


Figura 2.1. Blonder Scheme (fonte: [25]).

Il sistema soffre di un forte problema legato a shoulder surfing attack, in cui il malintenzionato potrebbe osservare la sequenza di input inserita dall'utente mentre la sta digitando ed utilizzarla per accedervi in un secondo momento, e di smudge attack³ se l'input viene immesso tramite uno schermo touchscreen. L'impiego di questo sistema è stato adottato per la prima volta da Microsoft a partire da Windows 8.

³Metodo utilizzato per scoprire un pattern digitato su uno schermo touchscreen, andando ad osservare le impronte lasciate sullo schermo stesso.

2.4.2 Cued-click Points

Il sistema CCP [26] può essere visto come l'evoluzione dello schema di Blonder dal momento che si basa sempre sull'inserimento di input, che però questa dovranno essere associati all'interno di una sequenza finita di immagini. Nel caso in cui l'autenticazione fallisca, verrà prodotto un messaggio di errore solo dopo l'inserimento dell'ultimo input, prevenendo in questo modo attacchi di tipo guessing. Inoltre, nel caso in cui venga cliccata un'area sbagliata all'interno dell'immagine, verrà fornita una figura successiva non corretta, dando in questo modo un feedback negativo tramite cui l'utente verrà avvertito dell'errore eventualmente commesso (figura 2.2).

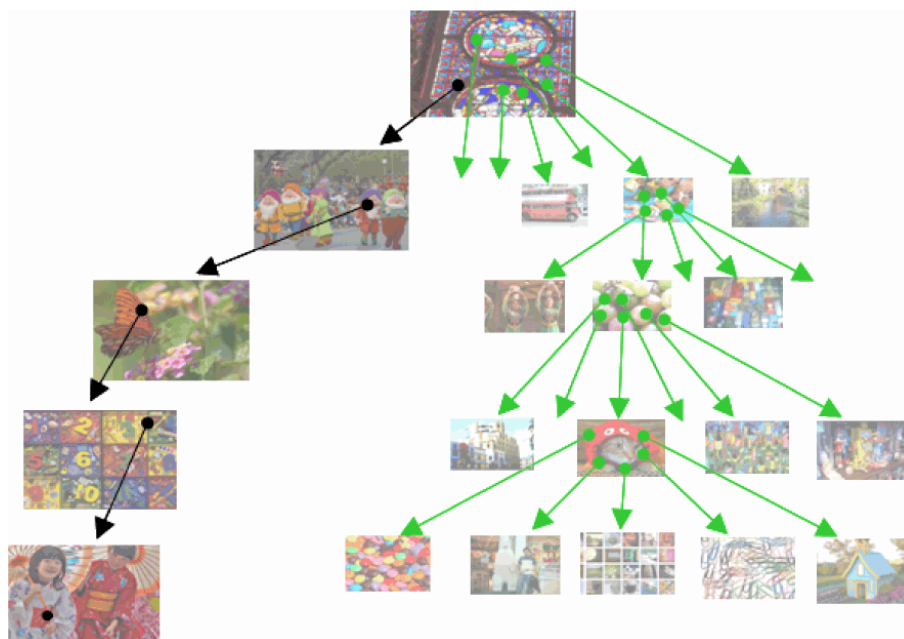


Figura 2.2. CCP Scheme (fonte: [26]).

Tramite l'utilizzo di immagini multiple, viene ridotto il numero di hotspots utili per gli attaccanti rispetto al modello precedente, non riuscendo però ad eliminarli del tutto e rimanendo ancora sensibile a shoulder surfing e smudge attack.

2.4.3 Persuasive Cued-Click Points

L'utilizzo dello schema di Blonder e della tecnica CCP portano con un sé un grosso problema, derivante dalla predicibilità delle aree scelte in fase di registrazione dall'utente. Trattandosi di una scelta effettuata dall'uomo e non da una macchina, la scelta delle aree all'interno di un'immagine sarà quasi sempre legata ad un set predefinito, riducendo in modo involontario l'efficienza dell'algoritmo. L'idea proposta dal meccanismo PCCP consiste nell'introdurre un elemento persuasivo durante la fase di scelta dell'area, andando in questo modo a condizionare la scelta dell'utente. Un esempio di applicazione consiste nel creare delle zone d'ombra nell'immagine per spingere l'utente a selezionare l'area non ombreggiata, evitando in questo modo

zone di preferenza all'interno di un'immagine. In fase di autenticazione le immagini verranno proposte nella loro forma originale, senza rappresentare alcun tipo di alterazione (figura 2.3).



Figura 2.3. PCCP Scheme (fonte: [26]).

Nonostante in termini di sicurezza si possano rilevare dei miglioramenti rispetto alle tecniche precedenti, grazie alla possibilità di evitare guess attack, rimangono ancora presenti le problematiche di shoulder surfing.

2.5 Shoulder surfing resistant Graphical Password

I sistemi visti finora presentano tutti lo stesso problema, ovvero il fatto di essere facilmente violabili tramite l'osservazione diretta durante la digitazione, non esistendo infatti alcun meccanismo per occultare la sequenza come invece accade con gli asterischi per le password testuali. Per questo motivo sono necessari dei sistemi di autenticazione che siano in grado di superare questi limiti, riuscendo allo stesso tempo ad implementare i benefici e l'alta usabilità di una password grafica. Quelli di cui andremo a discutere ora sono solo alcuni dei sistemi di autenticazione immuni alle problematiche di shoulder surfing.

2.5.1 PassMatrix

Per risolvere il problema dello shoulder surfing, PassMatrix [27] prende spunto dal modello CCP, a cui viene applicato un “login indicator” dinamico per ogni immagine (chiamata in questo sistema pass-immage), e che scade al termine della sessione in corso, rendendo in questo modo del tutto inutile il suo utilizzo per accessi futuri. PassMatrix è composto da sei elementi:

- Image Discretization Module, la divisione dell'immagine in quadrati, il cui numero dipende dalla risoluzione dello schermo su cui il sistema viene applicato.

- Login Indicator Generator Module, ovvero il modulo che si occupa di generare il login indicator per ogni immagine, utilizzando ad esempio colori, numeri o lettere generati e disposti in modo casuale. Dopo la sua generazione, l'indicatore deve essere comunicato in modo sicuro all'utente, tramite ad esempio un segnale acustico trasmesso attraverso auricolari, oppure visivamente utilizzando tecniche che permettono l'occultamento da parte di malintenzionati.
- Horizontal and Vertical Axis Control Module, che consiste nelle due barre necessarie per selezionare la pass-square corretta all'interno dell'immagine.
- Communication Module, che permette di trasferire in modo sicuro tramite SSL i dati dal client al server
- Password Verification Module, modulo che verifica i dati inseriti dall'utente; l'autenticazione avrà successo solo se tutte le pass-square attribuite ad ogni immagine sono corrette.
- Database, contenente le credenziali di autenticazione degli utenti registrati.

Per quanto riguarda il meccanismo di autenticazione, PassMatrix è composto da una prima fase di registrazione in cui l'utente inserisce le proprie credenziali e decide il numero di immagini di cui sarà composto il sistema; questo rappresenta una sostanziale differenza rispetto al CCP, in cui invece il numero di immagini aveva una lunghezza fissa e non modificabile a priori. La scelta del valore N di immagini dovrà essere sensata e dovrà rispettare il giusto tradeoff tra usabilità e sicurezza. La seconda fase invece riguarda l'autenticazione vera e propria in cui l'utente, dopo aver inserito l'username deciso in fase di registrazione, riceverà dal sistema il feedback segreto necessario per portare a termine il riconoscimento. Un esempio di feedback visivo potrebbe essere quello di congiungere il palmo della mano a cerchio: in questo modo all'utente basterà guardare all'interno di questo foro (figura 2.4) per vedere il feedback fornito dal sistema.

Per quanto riguarda il discorso sulla sicurezza, il sistema risulta essere immune a shoulder surfing e smudge attack, mentre attacchi a forza bruta hanno poche possibilità di successo, a causa dell'elevato numero di combinazioni che si vanno a generare, se si vanno a considerare la quantità di quadrati presenti all'interno delle N immagini. In conclusione, possiamo considerare questo sistema di autenticazione un ibrido rispetto a quelli menzionati in precedenza: vengono infatti risolti i problemi tipici delle password grafiche, ma per arrivare a questo si perde l'immediatezza tipica delle password testuali, rendendolo in definitiva un sistema difficilmente apprezzabile dall'utente comune, che ricerca immediatezza e praticità, ma più consigliabile ad un pubblico di nicchia.

2.5.2 Alignment based Graphical Password authentication system

Questo sistema di autenticazione si fonda su due fasi distinte [28]:



Figura 2.4. Esempio di feedback visivo (fonte: [27]).

- Selection and Training phase, in cui l'utente deve decidere un set di almeno 3 immagini, che dovrà poi ricordarsi tramite un'apposita fase di apprendimento. Il numero di immagini determinerà la robustezza dell'algoritmo e l'usabilità che l'utente avrà nel memorizzare la sequenza.
- Identification phase, che consiste nella fase di autenticazione vera e propria. Lo schema che viene proposto consiste in una serie di anelli concentrici, contenenti ciascuno una delle immagini scelte nella fase precedente (avremo quindi un numero di anelli pari alla lunghezza di cui è composta la nostra sequenza di immagini). Rispetto ai modelli visti in precedenza, non dovremmo andare a selezionare le immagini all'interno della griglia, ma bensì ruotare, tramite appositi pulsanti, i vari anelli fino a quando non si avrà allineato la sequenza corretta (figura 2.5).

Con questo tipo di meccanismo, un malintenzionato non riuscirà mai a capire con un primo sguardo la sequenza corretta di immagini, in quanto ne saranno presenti più combinazioni contemporaneamente. Tuttavia la minaccia di shoulder surfing non viene sventata completamente, in quanto l'attaccante potrebbe adottare brute-force attack in base alle combinazioni in suo possesso o fare un confronto tra i differenti schemi inseriti da un utente per accedere alla stessa risorsa. L'utilizzo di più immagini all'interno di un anello e di più anelli all'interno dello stesso schema, sono tecniche che possono essere utilizzate per irrobustire l'algoritmo, ma che dovranno essere sempre rapportate con l'usabilità che l'utente finale avrà nel fruire dell'algoritmo stesso.



Figura 2.5. Alignment based Graphical Password authentication system (fonte: [28]).

2.5.3 Image Based Registration and Authentication System

IBRAS [29] è un sistema di autenticazione basato su java che utilizza la scelta di un'immagine personale come mezzo tramite il quale autenticarsi. Il meccanismo è composto da una prima fase di registrazione in cui vengono inseriti i dati personali dell'utente e in cui viene aggiunta un'immagine personale; durante la fase di autenticazione verrà poi richiesto l'user ID e l'immagine inserita in fase di registrazione. Questa tipologia di meccanismo permette di avere un grosso vantaggio che risiede nello sforzo cognitivo che dovrà affrontare l'utente, il quale sarà di gran lunga inferiore rispetto al doversi memorizzare una password o un PIN. Non tutta l'immagine viene memorizzata e utilizzata come metodo di confronto da parte dei server, in quanto questo richiederebbe dei database ad elevate capacità per la loro memorizzazione, ed inoltre, nelle reti più lente, potrebbe rallentare la fase di autenticazione a causa dell'elevato bit-rate che si andrebbe a generare. La modalità con il quale invece viene memorizzata e poi spedita l'immagine è tramite una funzione di hash SHA-1 che produrrà in output una stringa di 20 byte, risolvendo in questo modo i problemi relativi a dictionary attack.

Photo Response Non Uniformity

Se da una parte IBRAS permette di semplificare il lavoro dell'utente, dall'altra potrebbe creare delle situazioni pericolose in cui un potenziale attaccante tenta di

acquisire digitalmente l'immagine originale per riutilizzarla poi in un secondo momento. Questo tipo di problematica può però essere risolta grazie ad una caratteristica intrinseca nel sensore fotografico chiamata "Photo Response Non Uniformity". Questa tecnica permette di trasformare i fotoni acquisiti dal sensore in elettroni ed è unica e indipendente dall'hardware, rendendo a tutti gli effetti la PRNU una sorta di caratteristica biometrica tramite il quale distinguere un sensore ottico da un altro. Le informazioni acquisite vengono poi elaborate attraverso una tecnica chiamata RAW, che permette di ricavare l'immagine digitale definitiva.

PRNU-based Physical Unclonable Functions [30] è un meccanismo con il quale vengono estratte le componenti legate alle alte frequenze in una foto RAW al fine di realizzare un modello in grado di discriminare sensori ottici differenti e allo stesso tempo di individuare discrepanze rispetto ai dati che possono essere estratti da un'eventuale copia. Le alte frequenze ricavate, sono informazioni biometriche e in quanto tali non vengono memorizzate dal server; in questo modo l'attaccante non ha nessun mezzo per accedere tramite la sua immagine non originale alla risorsa informatica protetta dalle informazioni ricavate dal RAW originale. In conclusione la tecnica qui proposta permette a tutti gli effetti di rendere più robusto il sistema IBRAS precedentemente trattato.

2.6 Continuous authentication

Abbiamo visto come i sistemi resistenti allo shoulder surfing abbiano sicuramente un'elevata potenzialità, permettendo a tutti gli effetti di mantenere un'adeguata sicurezza senza andare ad intaccare l'usabilità complessiva dell'utente, cosa che invece non accadeva con i sistemi password-based. Nonostante questo, anche i meccanismi appena trattati presentano alcune problematiche, causate da una copertura non ottimale in seguito alla fase di autenticazione primaria. In particolare, possono scaturire tutta una serie di potenziali attacchi come il social engineering attraverso tecniche di baiting⁴, furto del dispositivo dopo l'autenticazione, oppure tramite un'autenticazione sotto minaccia da parte di un malintenzionato. L'autenticazione classica viene anche definita come un processo "one-shot" [31], dal momento che verifica l'identità dell'utente un'unica volta: appena esso viene identificato, le risorse vengono rese disponibili per un determinato periodo tempo, o finché l'utente non fa esplicitamente il logout, dando quindi libero accesso a chiunque di poter fruire liberamente e senza alcun controllo informatico su quelle risorse. Il problema più grave, infine, si presenta nell'istante in cui gli utenti disabilitano il sistema di autenticazione primaria sui propri dispositivi: questa pratica è più usata del previsto e coinvolgerebbe, secondo un'analisi condotta da Egelman [32], il 34% degli utenti.

Per tutti questi motivi è necessario introdurre una serie di meccanismi che siano in grado di verificare l'identità dell'utente in modo trasparente e continuativo,

⁴Il baiting è una tecnica che consiste nel depositare in luoghi pubblici dispositivi hardware contenente malware, facendo leva sulla curiosità e sull'avidità di una potenziale vittima di impossessarsi e verificare il contenuto di tale oggetto.

realizzando così una protezione totale nei confronti dell'utente finale. Per realizzare una continuous authentication si possono utilizzare due approcci differenti:

- Physiological Biometric Systems
- Behavioral Biometric Systems

Esaminando la figura 2.6, si può rilevare il pattern decisionale che viene rilevato su un sistema in cui è presente la continuous authentication. In una prima fase, vengono raccolti i dati relativi all'utente mediante il sistema biometrico opportuno; se il sistema rileva una risposta negativa, entrerà in gioco il sistema di autenticazione tradizionale che decreterà il rilascio delle risorse o il blocco del dispositivo. Tutto il sistema di raccolta dati avviene in real time e senza interventi espliciti dell'utente (a meno che non subentrino dei dubbi sull'identità di quest'ultimo).

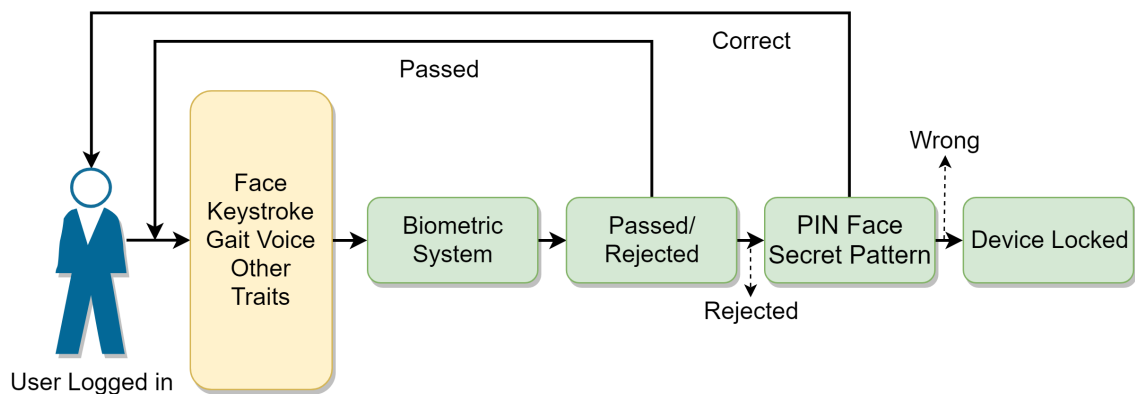


Figura 2.6. Funzionamento di un sistema di continuous authentication.

2.7 Physiological Biometric Systems

I sistemi biometrici rappresentano un cambio di direzione rispetto ai meccanismi analizzati fino ad ora, in quanto sfruttano le caratteristiche fisiche e comportamentali di un individuo come mezzo univoco per permettere loro l'autenticazione. Esistono vari tipi di sistemi biometrici come ad esempio l'impronta digitale, la scansione dell'iride o la vascolarizzazione. Il principale difetto di questa tecnologia risiede nell'hardware aggiuntivo richiesto per effettuare questo tipo di autenticazione che, per questo motivo, lo rende dipendente dal dispositivo da cui ci si autentica e non alla risorsa che si va ad autenticare; d'altro canto c'è anche da dire che negli ultimi anni il costo di realizzazione ed implementazione di queste tecnologie è diminuito parecchio. Le performance di questo meccanismo di autenticazione hanno un forte legame con l'usabilità; infatti un sensore poco preciso ed inaffidabile, genererà frustrazione nell'utente finale, che probabilmente dopo qualche tentativo si ritroverà a rinunciare ad esso, spostandosi su sistemi alternativi. I parametri tramite il quale analizzare la qualità di un sistema biometrico sono:

- Il False Acceptance Rate (FAR), che indica la frequenza con il quale vengono fatti accessi non autorizzati.
- Il False Rejection Rate (FRR), che indica la frequenza con il quale vengono rifiutati degli accessi autorizzati.
- Il False Identification Rate (FIR), che indica la frequenza con il quale il sistema riconosce l'utente, ma lo associa ad un'altra identità.

FAR e FRR sono fortemente connesse tra di loro in quanto l'incremento di uno porta al decremento dell'altro e viceversa. Si può a tal proposito definire un quarto elemento, l'Equal Error Rate (EER), che rappresenta il punto di equilibrio in cui FAR e FRR sono uguali (figura 2.7).

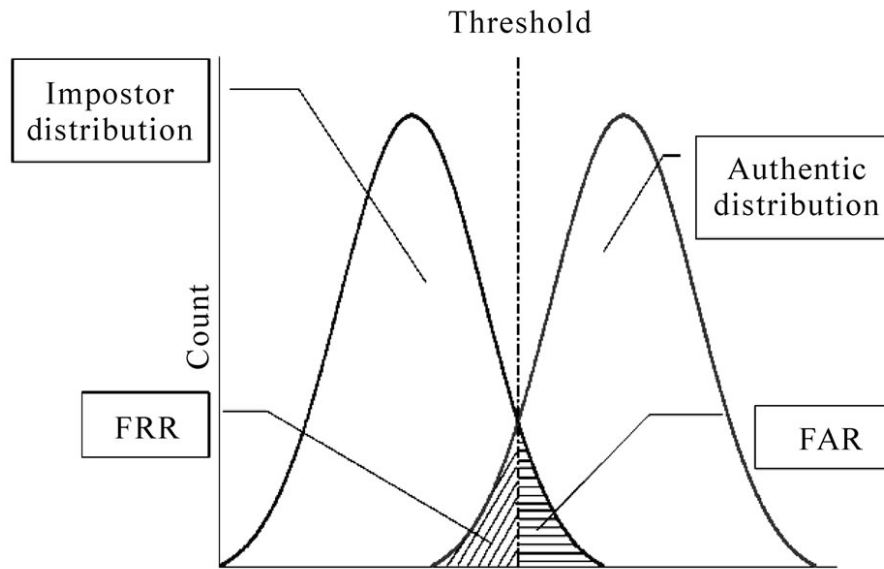


Figura 2.7. Grafico che mostra la distribuzione del FAR e del FRR (fonte: [Scientific Research](#)).

Infine bisogna considerare anche tutti quegli scenari in cui il sistema biometrico non può essere usato che, secondo un'analisi condotta da Coventry [33], coinvolge il 10% delle persone per quanto riguarda le impronte digitali e lo 0.005% per la scansione dell'iride. Analizzeremo ora più approfonditamente i sistemi biometrici a partire dalla loro attendibilità, ovvero tramite l'analisi dei valori FAR e FRR.

2.7.1 Low FAR & FRR

In questa parte andremo a considerare tutti quei fattori biometrici presenti nel corpo umano, la cui rilevazione può essere fatta attraverso una precisione tale da permettere l'implementazione e la commercializzazione. In particolare analizzeremo nel dettaglio l'iride, il volto e l'impronta digitale, andando a considerare non solo il livello di usabilità, ma anche come vengono implementati nei dispositivi di tutti i giorni.

Iride

L'iride è una membrana muscolare dell'occhio il cui colore, tessitura e pattern hanno un elevato grado di individualità tale da rendere quasi impossibile la presenza di due iridi uguali in soggetti diversi. Il metodo di acquisizione del pattern e di confronto in fase di autenticazione avviene in modo non invasivo, non generando quindi problemi a livello di usabilità; inoltre l'iride è una caratteristica che rimane tendenzialmente stabile nel tempo, a meno di incidenti legati all'occhio o a malattie particolari. Il metodo di acquisizione dell'iride avviene attraverso dispositivi molto costosi (il cui prezzo risulta essere di gran lunga maggiore rispetto a quello usato per il rilevamento delle impronte digitali), rendendolo per questo motivo un sistema biometrico di nicchia. Solo negli ultimi anni questa tecnologia viene implementata anche nei laptop e negli smartphone, anche se in quest'ultimi è presente una difficoltà maggiore nel rilevamento a causa dell'errore spaziale introdotto dal sensore nel riconoscere l'occhio dell'utente; questo non accade invece su dispositivi "fermi" rispetto all'interlocutore, come può essere il computer o un sensore apposto su un ATM o all'interno di un aeroporto. Infine è bene precisare che i sistemi di scansione dell'iride presenti su dispositivi portatili non hanno nulla a che vedere in termini di precisione con i sistemi professionali: è stato infatti dimostrato come il sistema di scansione dell'iride posto su uno dei migliori smartphone del 2017, il Samsung s8, sia facilmente bypassabile tramite l'utilizzo di una fotografia [34]. Questa è un'ulteriore conferma di come per ora questa tecnologia non sia ancora pronta per sbarcare nei dispositivi di massa, sia a causa della bassa usabilità, che dall'alta facilità con il quale il sistema può essere violato.

Volto

La scansione del volto, assieme a quella dell'iride, rappresenta uno dei sistemi biometrici meno invasivi presenti sul mercato, dal momento che non avviene un contatto diretto tra il sensore e l'utente; la tracciatura del volto avviene infatti tramite la semplice ripresa da parte di un sensore che trasforma l'immagine tridimensionale in un frame rettangolare chiamato *binary mask*. Successivamente il campione prelevato viene trasformato mediante una tecnica di "elastic graph matching", in cui essenzialmente avviene il confronto con i sample presenti nel database per verificare la provenienza di quel volto. Come accade per l'iride, anche il volto richiede tecniche costose per il rilevamento, tali da non permetterne solitamente la commercializzazione su dispositivi di massa (nonostante a settembre 2017, Apple abbia rilasciato un iPhone che utilizza proprio questa tecnologia come sistema di autenticazione). D'altra parte tecniche di rilevamento del volto vengono impiegate per monitorare costantemente posti affollati, per sventare ad esempio l'insorgere di eventuali attacchi terroristici.

Impronte

Le impronte digitali costituiscono l'informazione biometrica più studiata storicamente; essa si fonda sull'analisi delle piccole rughe presenti sul polpastrello, che rappresentano una caratteristica unica nell'essere umano. A differenza dei modelli

visti prima, il rilevamento dell'impronta avviene tramite un'interazione diretta tra dito e sensore: questo determina quindi un'analisi bidimensionale delle informazioni e non più spaziali come accadeva per l'iride e per il volto. Per questo motivo la realizzazione del relativo sensore è meno complessa e costosa, e può essere realizzata ed implementata nei dispositivi di massa. Di fatto attualmente cresce sempre di più il numero di smartphone con integrato il sensore di impronte digitali, rappresentando a tutti gli effetti un'alternativa più sicura, affidabile e usabile dei sistemi di sblocco tradizionali. Inoltre, essendo aumentati esponenzialmente i servizi fruibili da smartphone e di conseguenza i servizi applicativi su cui è necessario autenticarsi, è possibile andare a sostituire parzialmente o totalmente l'autenticazione tramite il sensore integrato nel dispositivo. Approfondiremo poi più avanti alcune tecniche implementative che aziende come Qualcomm stanno mettendo in atto per rendere ancora più usabile e funzionale il riconoscimento di questo fattore biometrico.

2.7.2 High FAR & FRR

Le tecniche viste sopra, seppur implementate in ambiti differenti, hanno tutte un utilizzo commerciale, grazie alla loro precisione ed affidabilità. Tuttavia vi sono alcuni parametri biometrici che seppur mantenendo l'unicità della caratteristica studiata, presentano alte probabilità di incorrere in falsi positivi o falsi negativi. Il motivo per il quale andiamo ugualmente ad analizzare queste tecniche è per considerare i progressi fatti negli ultimi anni, che ne hanno fatto riconsiderare l'utilità.

Sguardo

Il tracciamento del movimento degli occhi è una caratteristica interessante da analizzare: si può infatti pensare di creare un modello basato su challenge-response, in cui viene rilevato il tempo impiegato dall'occhio per visualizzare un determinato oggetto rappresentato sullo schermo. Tuttavia, nonostante i tempi brevissimi di reazione della cornea umana e l'alta usabilità del meccanismo, che non richiede nessuna interazione particolare con la macchina, questo tipo di autenticazione presenta grosse problematiche legate sia ai lunghi tempi di attesa necessari per la verifica (si parla di almeno 20 secondi) e sia alla poca precisione del meccanismo, con un EER di circa il 15% [35].

Quello che vedremo ora invece è un'evoluzione della tecnica appena presentata, in cui vengono studiate le reazioni naturali dell'occhio quando viene sottoposto a particolari stimoli esterni [36]. Nel momento in cui un osservatore fissa un oggetto, i suoi occhi non rimangono mai completamente fermi, ma eseguono centinaia di micromovimenti al secondo, alternando continuamente tra stato fisso e di saccade. Quando l'occhio è fisso significa che sta mantenendo il focus sull'oggetto attraverso un unico stimolo, mentre la saccade, uno dei movimenti più veloci del nostro corpo, viene adoperato per interrompere momentaneamente la visione per evitare il disturbo causato dal movimento dell'immagine sulla retina. Il modello proposto sfrutta l'analisi di tutti questi comportamenti involontari al fine di creare un sample legato all'osservatore. Per rendere quanto più conformi le informazioni ricavate, viene utilizzato uno stimolo ben preciso, che verrà poi riproposto nella fase di autenticazione,

ossia un puntino rosso su sfondo nero che cambia ripetutamente posizione. L'analisi degli stimoli prodotti dai nostri occhi secondo i parametri visti sopra, determinerà quindi il modello finale. Nella fase di autenticazione avremo quindi due canali visivi differenti (figura 2.8):

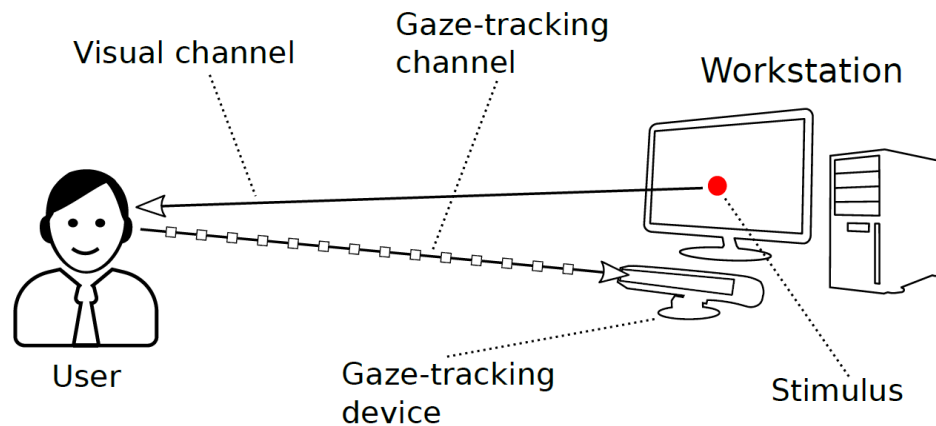


Figura 2.8. Rappresentazione del modello di tracciamento dello sguardo in risposta ad uno stimolo} (fonte: [36]).

- Visual channel, che corrisponde allo spazio che intercorre tra l'utente e la workstation su cui viene riprodotto lo stimolo luminoso.
- Gaze tracking, ovvero il canale che unisce l'osservatore con il dispositivo che dovrà acquisire i vari parametri.

Questa tecnica biometrica riesce a migliorare i limiti del modello precedente, portando ad un tempo medio di autenticazione di soli 5 secondi e un valore di EER che si attesta al 6.3%.

Voce

Le differenti caratteristiche fisiche del tratto vocale, così come le diverse abitudini di pronuncia dei fonemi, rendono la voce una proprietà biometrica ideale. L'implementazione di un sistema in grado di riconoscere la voce prodotta dall'utente può essere facilmente implementata tramite l'utilizzo di microfoni e di sensori presenti sulla maggior parte degli smartphone moderni, non richiedendo quindi l'acquisto di hardware ulteriore. Inoltre, è possibile utilizzare la voce attraverso due meccanismi differenti: il primo in base a come vengono pronunciati i fonemi, senza però considerarne il significato, e il secondo che invece analizza entrambi gli aspetti. Il problema principale di questo sistema di autenticazione risiede però nell'alta vulnerabilità ad attacchi di tipo replay [37], dal momento che si possono ricavare i sample della vittima semplicemente registrando e montando appositamente la sua voce.

VoiceLive [38] è un sistema di rilevamento della voce che, per determinare l'identità del soggetto, sfrutta i parametri ricavati da una particolare tecnica chiamata

“time-difference-of-arrival” (TDoA), che è in grado di misurare la differenza di arrivo di una sequenza di fonemi da parte di un numero di sensori dislocati nello spazio (in questo caso i 2 microfoni dello smartphone). Questo meccanismo è quindi in grado di rilevare eventuali replay attack in base al TDoA dinamico, una caratteristica unica nell’utente che non può essere presente in sample registrati. In conclusione VoiceLive rappresenta un’ottima alternativa, grazie alla completa assenza di costi hardware e l’accuratezza del sistema, che si porta su valori di EER attorno all’1%.

2.8 Behavioral Biometric Systems

Se i sistemi Physiological richiedevano uno sforzo seppur minimo di interazione con la macchina per essere autenticati, vediamo ora una serie di meccanismi che non solo non richiedono hardware aggiuntivo, ma sono completamente trasparenti all’utente finale. Essi sfruttano sempre caratteristiche uniche presenti nell’essere umano, non più però legati a dei suoi fattori fisici ma bensì ad elementi comportamentali e psicologici.

2.8.1 Keystroke dynamics

I sistemi biometrici visti finora hanno una grossa limitazione dettata dal sistema hardware a cui sono inevitabilmente legati. Il meccanismo di autenticazione “keystroke dynamics” è un sistema che appartiene sempre alle tecnologie biometriche, ma che supera il limite di cui si è parlato sopra, attraverso l’implementazione di una caratteristica propria dell’individuo, ma che allo stesso tempo non necessita di alcun hardware aggiuntivo. Quello che fa l’algoritmo di keystroke dynamics è quello di autenticare l’utente in modo automatico e totalmente trasparente in base alla sua modalità di scrittura. Un sistema che utilizza questo algoritmo, richiede come prima cosa un processo di registrazione, che si divide in:

- Una prima fase, in cui l’utente dovrà scegliere ed inserire delle credenziali (composte, ad esempio, da una delle tecniche mostrate in precedenza). In questo modo, se il meccanismo di KD non è in grado di riconoscere l’utente, subentrerà il sistema di autenticazione primario scelto in fase di registrazione.
- Nella seconda parte, l’utente dovrà invece digitare una o più frasi, al fine di ricreare un sample, che verrà poi associato al suo pattern di digitazione.

Il tipo di analisi che può essere condotta per determinare il pattern di KD, può basarsi su diverse caratteristiche comportamentali, come ad esempio [39]:

- P-P, il tempo che intercorre tra la pressione di due tasti in sequenza.
- P-R, l’intervallo di tempo tra la pressione di un tasto e il suo rilascio.
- R-P, l’intervallo di tempo tra il rilascio di un tasto e la pressione di un tasto successivo.

- R-R, il tempo che intercorre tra il rilascio di due tasti in sequenza.

Lo studio di questi parametri è associato ad un vettore temporale che tiene traccia delle pressioni e dei rilasci di un set specifico di tasti (esempio figura 2.9).

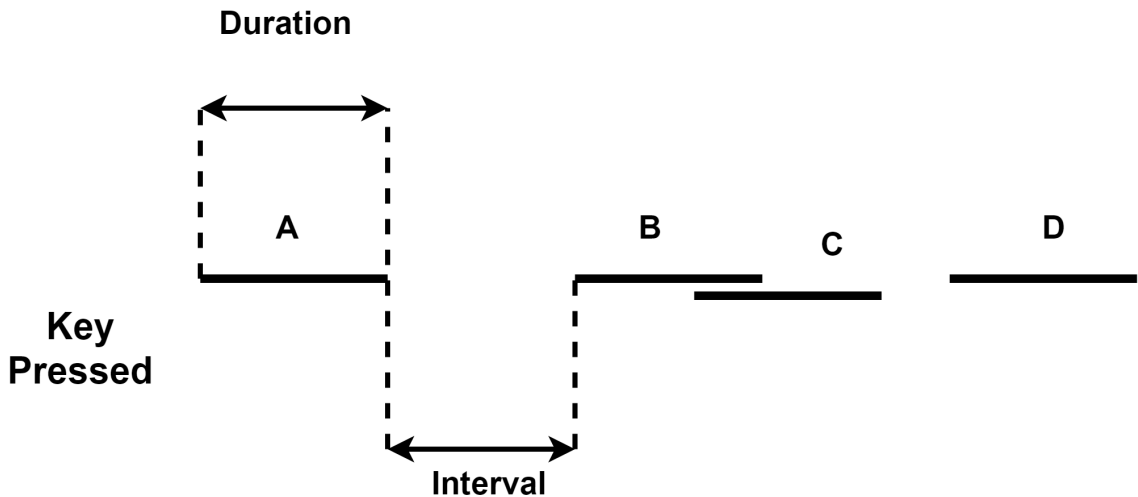


Figura 2.9. Esempio di digitazione dei caratteri “ABCD” con relativo vettore temporale {30; 60; 70;-35; 60; 35; 75; 40; 55} (fonte: [40]).

Esistono inoltre diverse modalità che l’algoritmo può utilizzare per interagire e riconoscere l’utente, tra cui le principali sono:

- Static approach: in cui viene l’utente viene analizzato solo nell’istante in cui si è autenticato, basandosi sul sample creato in fase di registrazione. In questo modo viene mantenuta inalterata la privacy dell’utente, a discapito di una minor affinamento del modello realizzato in precedenza.
- Dynamic approach: l’algoritmo analizza costantemente quello che l’utente digita tramite tastiera, indipendentemente dall’azione che sta svolgendo. Questo potrebbe causare grossi problemi a livello di privacy, ma garantisce un servizio ottimale, sia perché si hanno più dati a disposizione tramite il quale avere un confronto più accurato e sia per affinare il sample creato in fase di registrazione⁵.

Un’ultima considerazione riguarda i fattori FAR e FFR, il cui valore si attesta per entrambi i casi intorno al 5%, allineandosi con gli altri sistemi biometrici, e variando sensibilmente in base al tipo di approccio che si intende utilizzare.

In conclusione questo sistema di autenticazione unisce l’affidabilità e la sicurezza dei sistemi biometrici tradizionali ad un’elevata usabilità, che si traduce nella maggior parte dei casi in uno sforzo nullo da parte dell’utente finale. Il difetto reale di questo algoritmo, assieme a tutte le tecniche biometriche viste in precedenza, risiede però proprio nella non completa affidabilità del sistema, che lo costringe a dover essere sempre affiancato da un meccanismo di autenticazione alternativo, che subentra in caso di un eventuale fallimento.

⁵In questa metodologia, vengono utilizzate tecniche di machine learning per affinare la tecnica.

2.8.2 Keystroke dynamics: metodologie avanzate

Andiamo ora ad analizzare una serie di metodologie che hanno il compito di migliorare e rendere più preciso il meccanismo del keystroke dynamics e il cui studio è fondamentale al fine di costruire il profilo psicologico e comportamentale dell'individuo.

Semi-Timing Measurement

Uno dei primi parametri che vengono analizzati in questo set di tecniche è il cosiddetto Word-per-Minute (WPM) che viene calcolato come il rapporto tra il numero di parole immesse e il numero totale di tasti premuti, e che permette, tramite una prima analisi, di capire se l'utente ha una scrittura veloce o meno. In aggiunta al WPM, un altro valore interessante è il Negative Up-Down (negUD), che si analizza tramite la tendenza, soprattutto di chi ha una scrittura rapida, di rilasciare un tasto solo dopo avere premuto il successivo creando una sorta di sovrapposizione (figura 2.10 a).

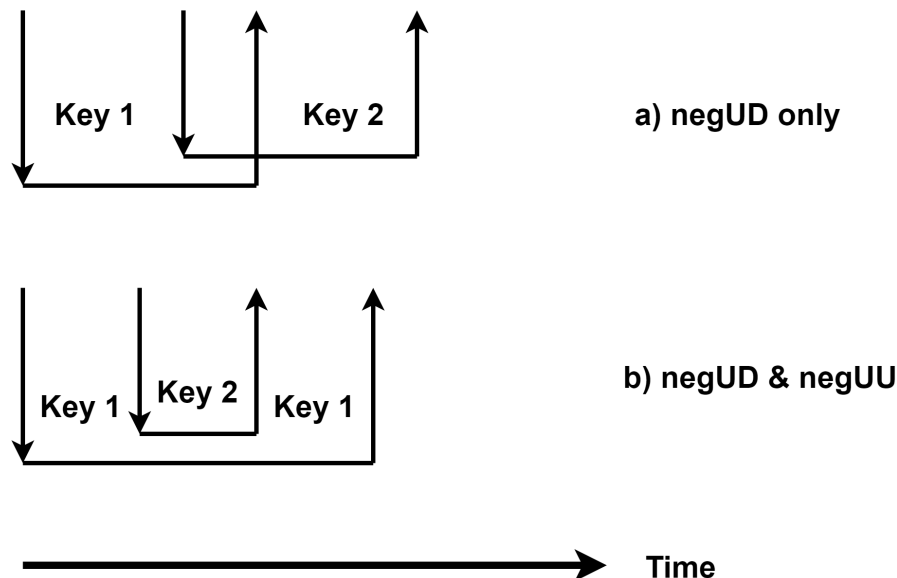


Figura 2.10. Rappresentazione del negUD (figura a) e del negUU (figura b) (fonte: [41]).

Il calcolo del negUD avviene andando a considerare la percentuale di sovrapposizioni che si verificano rapportate al numero di coppie di pulsanti premuti. Un ultimo parametro che sfrutta l'analisi temporale è il negative Up-Up (negUU) che è caratterizzato da un rilascio anticipato di un tasto rispetto a quello precedente (figura 2.10 b). Un negUU si verifica in concomitanza di due negUD consecutivi ed è interessante notare come alcuni utenti, durante la loro sessione di digitazione, non manifestino alcun tipo di negUD (e di conseguenza negUU).

Editing Features

Andiamo a vedere ora, una serie di elementi che possono essere analizzati in base a caratteristiche associabili alle abitudini dell'utente e, quindi per questo motivo, slegate completamente dal fattore temporale. Una prima caratteristica da analizzare può essere il numero di volte che l'utente commette un errore durante la digitazione: questo può essere rilevato a partire dal numero di occorrenze in cui viene premuto il tasto Delete o Canc. L'error rate, definito come il rapporto che intercorre tra il numero di errori compiuti e il numero totale di parole digitale, ci permetterà quindi di avere un'indicazione sulla percentuale degli errori commessi da un utente. Sulla tastiera esistono inoltre pulsanti "speciali" adibiti a funzionalità specifiche che spesso hanno lo stesso fine e la cui unica differenza è quella di trovarsi in posizioni spaziali differenti. Se quindi in alcuni casi la scelta di un pulsante piuttosto che un altro è condizionata dal suo uso (ad esempio i tasti di eliminazione di un carattere visti prima), in altri casi la decisione ricade totalmente sull'utente, permettendo in questo modo di poter definire aspetti psicologici e comportamentali legati unicamente alle sue preferenze. Un esempio riguarda l'uso del tasto Shift o del CapsLock per digitare una lettera maiuscola, o l'uso dello Shift sinistro piuttosto che destro con il quale poter scegliere il carattere alternativo associato ad un determinato pulsante⁶.

2.8.3 Mobile keystroke dynamics

Come visto nella parte introduttiva, i dispositivi mobile stanno prendendo sempre più piede negli ultimi anni, diventando a tutti gli effetti dei sistemi general purpose simili a dei veri e propri computer. L'enorme aumento di dati sensibili presenti nei nostri smartphone richiede quindi meccanismi di protezione adeguati e sufficienti nel garantire sicurezza e protezione all'utente.

La trasposizione del keystroke dynamics su dispositivi mobile è un processo complicato in quanto subentrano fattori di incertezza e di variazione rispetto al metodo tradizionale. Siamo infatti abituati ad utilizzare il nostro computer sempre attraverso situazioni definite, solitamente seduti, mentre l'utilizzo di uno smartphone può essere fatto in svariate posizioni che possono influenzare drasticamente la velocità di scrittura tipica di un utente. Per questo motivo non basterà più creare un solo modello, ma sarà necessario creare un insieme N di modelli tramite il quale descrivere le N situazioni possibili. Per semplicità si può supporre di individuare tre posizioni abituali di scrittura, ovvero seduto, in piedi e durante il cammino e due modalità di scrittura differenti, in orizzontale o in verticale. Lo switch da un modello all'altro avviene in modo diretto tramite l'analisi dei dati prelevati dal giroscopio del dispositivo: questa è una fase critica che determina il fallimento o la riuscita del sistema. Nel momento in cui l'utente preme un tasto sullo schermo, vengono analizzate le rispettive coordinate x , y e z , e viene associato uno dei modelli precedentemente descritti; per le pressioni successive si andrà a calcolare la variazione delle tre coordinate per definire se ci si trova ancora nello stesso modello o meno.

⁶In questo caso è possibile ricavare un'ulteriore informazione tramite l'analisi di come il tasto Shift viene rilasciato rispetto al pulsante concorrente.

In uno studio condotto da Heather Crawford e Ebad Ahmadzadeh [42], su 39 partecipanti in cui sono stati analizzati più di 300000 parole, è emerso in modo evidente come la poca precisione fornita da un modello unico influenzi negativamente i valori del FAR e del FRR, a discapito invece di un sistema multi-modello (tabella 2.11)

	FRR (%)	FAR (%)
No Position	39-41.9	23-35.6
Sit	4.5-10.8	10.5-18.6
Stand	10.3-15.8	20.2-39.8
Walk	9.2-17.7	23.3-31.1

Figura 2.11. Confronto dei valori di FRR e FAR non tenendo conto della posizione e considerando la posizione (seduto, in piedi, camminando).

Esistono però due problemi sostanziali nel sistema appena descritto. Il primo riguarda il numero di modelli preso in considerazione, che non permette di descrivere appieno le modalità di scrittura di un utente: potremmo infatti scrivere più velocemente se abbiamo una camminata lenta e senza ostacoli, o avere una scrittura più lenta e imprecisa nel caso in cui stiamo camminando velocemente e abbiamo una serie di ostacoli che ci interrompono e ci distraggono più volte dal processo di scrittura. Il secondo problema riguarda invece il riconoscimento di scrittura all'interno di un modello piuttosto che un altro, che in caso di identificazione errata creerà casi di falsi negativi, non permettendo all'utente di autenticarsi. Alla luce di tutto questo è evidente come questo meccanismo non possa ancora essere usato come mezzo di autenticazione primario, a causa della sua poca precisione. In ogni caso le potenzialità del keystroke dynamics possono essere usate per rafforzare sistemi di autenticazione ordinari, formando meccanismi più complessi di password hardening [43] o come mezzo di monitoraggio secondario, al fine di verificare l'autenticità dell'utente in modo trasparente durante l'utilizzo del suo dispositivo, andando poi eventualmente a richiedere nuovamente il login nel caso in cui venga rilevata un'anomalia nella modalità di scrittura.

2.8.4 Authentication through an Hidden Markov Model Behavior

Continuiamo l'analisi fatta sui dispositivi mobile, andando ad analizzare una tecnica che studia il comportamento dell'utente durante la sua navigazione e la sua interazione con il dispositivo. HMM [44], in particolare, è un modello che va a rilevare i movimenti orizzontali e verticali che vengono riprodotti su uno schermo touch, andando a rilevare caratteristiche come la pressione, l'area, la durata e la posizione. Dopo la fase di training, in cui viene ricreato un template in base alle caratteristiche registrate dall'utente, avviene lo step di autenticazione: dal momento che un singolo tratto potrebbe dare dei risultati poco precisi, vengono combinati un insieme di parametri prima di fornire un risultato concreto, andando in questo modo ad incrementare la robustezza del sistema. Il sistema di continuous authentication, viene abilitato in tre situazioni differenti:

- La Intra-session rappresenta l'autenticazione a breve termine che viene eseguita subito dopo il login, appena è stata conclusa la fase di training. Per questa tipologia di sessione abbiamo un EER medio pari al 5.49% con il singolo tratto e allo 0.37% per il multi-tratto⁷. Per i sistemi in cui si vuole raggiungere una massima sicurezza (FAR=0), l'usabilità rimane ancora piuttosto alta. Nonostante questo, la Intra-session può presentare delle problematiche di sicurezza nell'istante in cui un potenziale attaccante si impadronisce del dispositivo, subito dopo il login dell'utente: in questo caso infatti il modello terrà conto sia del template del proprietario, costruito in fase di training, sia dei gesti immessi dal malintenzionato, creando la possibilità che venga favorito proprio quest'ultimo (dal momento che la fase di training e quella di testing vengono realizzate nella stessa sessione). In generale questa prima fase non rappresenta la protezione massima, che viene invece delegata nelle due fasi successive.
- Nella Inter-session, l'autenticazione avviene dopo un breve periodo di tempo, che si attesta intorno ai 10-12 minuti successivi all'avvenuto login. Grazie a questo intervallo di tempo, in questo contesto non si presentano le problematiche di sicurezza rilevate nella Intra-session. Per quanto riguarda invece i valori di EER abbiamo un valore pari al 7.8% per il singolo tratto e all'1.7% per il multi-tratto.
- Infine abbiamo la Long-term che rappresenta l'autenticazione più impegnativa, dal momento che viene fatta sui dati che vengono ricavati durante il primo giorno di utilizzo. Infatti in questo caso si rileva un EER maggiore rispetto ai casi precedenti, con valori pari al 9.21% per singolo tratto e al 2.28% per multi-tratto.

HMM non rappresenta l'unico tipo di implementazione per quanto riguarda il modello appena trattato, ma ne esistono altre varianti come mostrato nella tabella 2.1.

<i>Autori</i>	<i>Utenti testati</i>	<i>Classificatori utilizzati</i>	<i>EER (%)</i>
Frank et al. [45]	41	SVM, kNN	0.00 - 4.00
Zhang et al. [46]	50	Sparsity-based classifiers	0.77
Li et al. [47]	75	SVM	3.0
Feng et al. [48]	40	Random forest	7.75
Servadda et al. [49]	138	Differenti	10.50
Zhao et al. [50]	78	Distanza	6.33 - 15.40

Tabella 2.1. Schema delle tecniche basate sul tocco dello schermo.

2.8.5 Authentication by web browsing history

Andiamo ora ad analizzare un sistema biometrico che studia le abitudini di navigazione di un utente attraverso il pattern dei siti web visitati in precedenza. Il primo

⁷In [44] con multi-tratto si intende un'interpolazione dei dati che viene fatta su 11 singoli tratti.

elemento di spicco di questo meccanismo risiede nella totale assenza di un legame con l'hardware: non interessa infatti come l'utente ricerchi i vari siti, se tramite l'ausilio della tastiera o di un touchscreen (cosa che abbiamo visto invece avere importanza nel meccanismo di keystroke dynamics, in cui il cambio di supporto, tramite il quale vengono inseriti gli input, può produrre risultati diversi), ma solo l'ordine con il quale quei siti vengono ricercati. Nel sistema proposto [51] avviene innanzitutto una prima fase di registrazione in cui l'utente definisce l'ordine e la preferenza dei siti ad un Identity Provider; prima di poter navigare, il client deve sempre connettersi all'IdP, che terrà traccia dei siti visitati confrontandoli con il modello definito in fase di registrazione. Ogni qual volta l'utente carica la pagina di un sito, il browser web invia un HTML Request all'IdP su cui è registrato, per scaricare un'immagine in possesso del Provider; a quel punto l'IdP, dopo avere inviato la relativa HTML Response, registrerà il sito visitato dal client. L'autenticazione viene infine approvata o respinta in base ad un rate definito dal sistema (che può essere una percentuale arbitraria di corrispondenze esatte): nel caso in cui non venga soddisfatta quella soglia, verranno chieste conferme ulteriori al fine di verificare l'identità dell'utente.

Oltre a quali siti vengono visitati, si può pensare di fare un'analisi sul come, attraverso lo studio di una serie di parametri comportamentali [52] che si manifestano all'interno di una stessa sessione:

- Pauses, che è il tempo speso da un utente su una pagina web; esso viene calcolato come differenza tra due timestamp successivi.
- Burstiness, una caratteristica intrinseca nel comportamento umano che in questo contesto viene calcolato attraverso le variazioni di pausa tra una visita e l'altra.
- Time between revisits, indica il numero di visite di una stessa pagina all'interno di una stessa sessione.
- Genres, tramite il quale è possibile categorizzare le pagine web dell'utente in base al loro stile, alla loro formattazione e al loro contenuto.

Questi parametri però, applicati singolarmente, non sono in grado di definire caratteristiche comportamentali individuali abbastanza forti da poterne pensare l'implementazione, mentre la loro applicazione sul modello analizzato nella prima parte potrebbe migliorarne l'accuratezza complessiva.

Esiste inoltre un'altra tecnica che permette di realizzare un'autenticazione basata sulla navigazione web, che risulta però essere più accurata ed affidabile rispetto a quelle trattate in precedenza. Essa prende il nome di "Markov Sequence based Identity Authentication" (MSIA) [53] e si basa su un modello Markoviano⁸, in cui ogni anello della catena viene descritto attraverso cinque attributi:

- uid, che permette di identificare l'utente produttore della catena Markoviana.

⁸Processo stocastico in cui la probabilità di transizione da un punto ad un altro, dipende unicamente dallo stato del sistema

- url, denota l'URL della pagina web visitata.
- ref, indica i links associati all'URL.
- tstamp, il timestamp relativo all'accesso alla pagina web.
- lastsec, il tempo di permanenza sul sito espresso in secondi.

Utilizzando quindi il modello prodotto da MSIA, si è in grado di riconoscere l'identità dell'utente che sta navigando su internet: la precisione del meccanismo è direttamente proporzionale al numero di componenti di cui è formato MSIA. È possibile, però, che durante la “partenza a freddo” del sistema venga rilevata un'imprecisione troppo elevata, causata da un modello Markoviano ancora troppo acerbo. In questo caso, si può associare ad MSIA una tecnica di “Sequence and Preference based Identity Authentication” (SPIA) [53], dove vengono analizzati ulteriori aspetti temporali e comportamentali dell'individuo, con il quale è possibile ricreare un modello finale più attendibile. Secondo uno studio condotto su sei studenti, la tecnica MSIA è in grado di arrivare ad un'accuratezza pari all'83.6%, che può salire fino al 90% se usata in coppia con SPIA.

Infine, è bene precisare, come sia possibile andare a realizzare una trasposizione del meccanismo visto in questo paragrafo anche su mobile, andando a studiare il pattern di apertura delle app [54].

2.8.6 Authentication using In-Air Handwriting Style Verification

MoCRA [55] è un sistema di autenticazione che permette di identificare l'individuo attraverso parametri comportamentali e biometrici intrinseci nella sua scrittura. Tutto questo viene fatto attraverso la rilevazione di stringhe di caratteri scritti nell'aria che vengono rilevati da un apposito sensore in grado di riconoscere la profondità del tratto, chiamato “Leap Motion Controller”. Il fatto di essere legati ad un dispositivo hardware è sicuramente uno svantaggio, anche rispetto alle altre tecniche viste precedentemente; d'altra parte però, questo meccanismo è in grado di fornire più precisione con una media del valore di EER che varia intorno all'1.18%, tale da permetterne l'implementazione anche per fornire autenticazione primaria tramite challenge-response. Il modo in cui opera questo sensore è tramite la suddivisione del tratto in “stroke segment”, ovvero un blocco di costruzione base dell'elaborato tale da poterne ricavare un elemento distintivo: questo permette la creazione di un sample legato all'individuo che ne permette l'identificazione in base al come viene scritto e non solo in base al suo contenuto (vedi figura 2.12). Il sistema di registrazione di MoCRA lavora attraverso due fasi:

- Enrollment, dove vengono catturate le prime stringhe disegnate dall'utente. Questa è la fase primaria che permette di definire un sample associato all'utente.
- Testing, in cui viene testato ed affinato il sample creato nella fase successiva

User-1					
	when	comes	out	other	but
User-2					
	as	with	makes	some	even
User-3					
	when	comes	out	other	but

Figura 2.12. In questo esempio le parole scritte dagli utenti 1 e 3 sembrano molto simili, ma in realtà alcune lettere (come la ‘o’ e la ‘t’) sono identificabili attraverso “stroke segment” differenti (fonte: [55]).

La fase di autenticazione avviene tramite la rappresentazione su schermo di stringhe prodotte in modo casuale, che dovranno essere ridisegnate dall’utente. MoCRA controllerà come prima cosa se ciò che viene scritto dall’utente combacia con quello riprodotto sullo schermo mentre, come ultimo, andrà a creare un sample a partire dai dati raccolti, che verrà confrontato con quello definito in fase di registrazione. In termini di sicurezza, il sistema è completamente esente da smudge attack in quanto non avviene un’interazione diretta tra sensore e utente; inoltre abbiamo visto come i sistemi biometrici tradizionali possano riscontrare delle limitazioni in alcuni ambiti particolari (ad esempio avere le dita sporche o bagnate potrebbe non garantire un riconoscimento ottimale delle impronte digitali), cosa che invece non accade per questo sistema di autenticazione. Infine anche gli shoulder surfing attack perdono d’efficacia, a causa dell’alta difficoltà che riscontrerebbe l’attaccante nell’osservare e acquisire il tratto stilistico della vittima.

2.8.7 Context-Aware Security by Hierarchical Multilevel Architectures

CASHMA [56] rappresenta un’architettura concreta tramite il quale è possibile utilizzare un insieme di caratteristiche biometriche dell’individuo al fine di realizzare un sistema di continuous authentication funzionale. È bene precisare che nel meccanismo di cui andremo ora a spiegare i dettagli, non viene specificato il tipo di sistema biometrico che deve essere implementato sul dispositivo, lasciando in questo modo libero arbitrio al progettista. CASHMA è composto dai seguenti elementi (vedi figura 2.13):

- Databases of templates, che contengono le informazioni biometriche degli utenti regolarmente registrati al servizio.

- Una serie di computational server, che serviranno a verificare la validità dei dati presenti nel database con quelli inviati dal client.
- Authentication server, che fa da cerniera tra il client e tutta la parte computazionale di verifica dei dati inviati dall'utente.
- CASHMA authentication service, racchiude le tre componenti viste sopra e rappresenta la struttura principale su cui è fondata CASHMA.
- Web services, ovvero tutti quei servizi e quegli applicativi web che richiedono una fase di autenticazione e che sono registrati ad un CASHMA authentication service per poter essere abilitati ad una continuous authentication.
- Client, ovvero il dispositivo utente che acquisisce i dati biometrici e li trasmette al CASHMA authentication server per poter accedere ad un determinato web service.

Le trasmissioni tra le varie componenti dell'architettura CASHMA devono avvenire attraverso canali sicuri per evitare attacchi Man In The Middle (MITM).

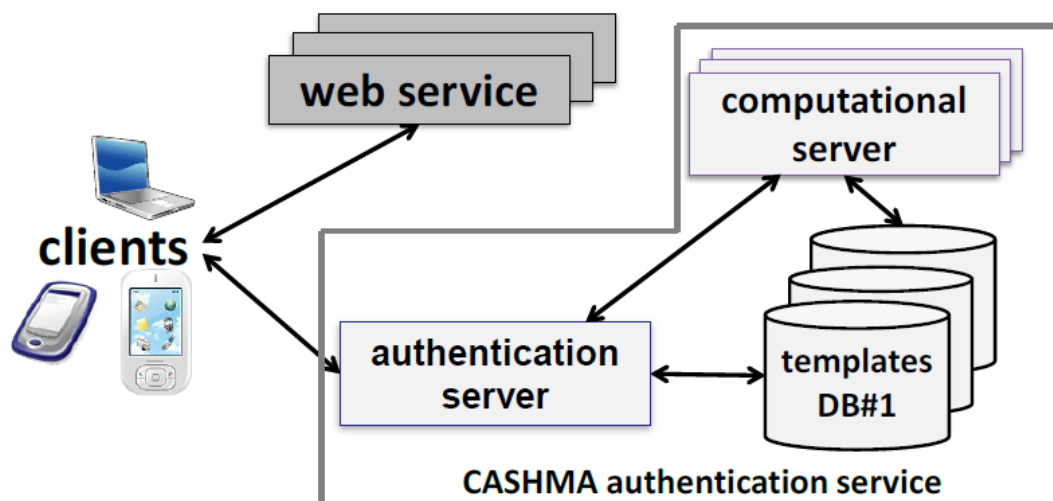


Figura 2.13. Architettura CASHMA (fonte: [56]).

Dopo avere visto l'architettura vediamo ora il funzionamento, proponendo l'esempio di un utente che vuole autenticarsi ad un servizio web. Come prima cosa il client deve autenticarsi, utilizzando i sistemi biometrici presenti sul proprio dispositivo che verranno inviati all'authentication server per la verifica. Se l'utente è registrato al CASHMA authentication service e se i suoi dati combaciano con quelli memorizzati nel database, l'authentication service provvederà a rilasciare un certificato tramite il quale il client potrà provare la propria identità col web service. Il CASHMA certificate è un elemento fondamentale perché permette di aggiornare i valori necessari a mantenere aperta la sessione tra il client e il web service, utilizzando un valore di Timestamp ed un Sequence Number (quest'ultimo necessario per evitare replay attack). L'idea generale del protocollo si basa quindi sul prelevare in

modo saltuario i dati biometrici dell'utente, verificarli col CASHMA authentication service e, in caso di matching, aggiornare il time expired della sessione. Se invece un ipotetico attaccante prenda il posto dell'utente dopo la fase iniziale di autenticazione, CASHMA riconoscerà l'intruso e non invierà al client un nuovo certificato con il quale poter rinfrescare i dati della sessione, ma gli permetterà ugualmente di partecipare a quella sessione fino al termine del suo time expired. Analizzando quest'ultimo aspetto si potrebbe pensare di regolare in modo adeguato il timestamp della sessione cercando di trovare il giusto tradeoff per evitare finestre temporali di aggiornamento troppo brevi, che potrebbero appesantire inutilmente il dispositivo e abbassare di conseguenza l'usabilità dell'utente, o troppo lunghe, in quanto queste ultime consentirebbero l'accesso delle risorse all'attaccante per lunghi periodi di tempo.

2.8.8 Behavioral biometric system: uno sguardo al futuro

Andiamo ora a descrivere brevemente un paio di tecniche di cui non si conoscono ancora bene i dettagli implementativi, ma la cui commercializzazione potrebbe dare spunti positivi in una ipotetica continuous authentication del futuro. Il primo sistema di cui andiamo a parlare è il progetto Qualcomm Fingerprint Sensors [57], presentato dall'azienda nel corso del 2017, che dimostra la possibilità di integrare il sensore di impronte digitali sotto lo schermo del dispositivo. Questo porterebbe prevalentemente ad avere maggiore usabilità per l'utente che non dovrebbe più ricercare il sensore in punti a volte scomodi del terminale, dal momento che avrebbe tutta la superficie dello smartphone per potersi autenticare. Una possibile implementazione in termini di continuous authentication, potrebbe essere fatta andando ad integrare il sensore lungo tutta la superficie dello schermo, permettendo in questo modo di rilevare e riconoscere l'utente mentre svolge le sue attività. Purtroppo ad oggi, questo sistema ha ancora dei limiti implementativi in quanto il pannello su cui può essere montato il sensore deve necessariamente essere un OLED e non superare lo spessore di 1200µm, andando di conseguenza a tagliare fuori una grossa fetta dei dispositivi presenti attualmente sul mercato. Agli inizi del 2018, Vivo e Huawei hanno presentato due smartphone che implementano la tecnologia appena trattata. Togliendo il fatto che l'estensione del riconoscimento non copra tutto lo schermo ma solo un'area ben definita, il risultato globale non è stato soddisfacente, a causa dei lunghi tempi richiesti ed alla difficoltà nel riconoscimento, fattori che sottolineano come l'uso di questa tecnologia risultava ancora acerba. Un anno dopo, nel 2019, quasi tutti i produttori hanno cercato di integrare questa tecnologia all'interno dei propri telefoni top di gamma, con risultati più soddisfacenti rispetto al passato grazie all'abbandono della tecnologia capacitiva in favore a quella ottica o ad ultrasuoni.

Una serie di ricercatori dell'università di Buffalo hanno presentato durante il MobiCom del 2017 un sistema in grado di poter effettuare un'autenticazione continuativa attraverso il rilevamento del battito cardiaco. Cardiac Scan [58], questo il nome del sistema, riuscirebbe a rilevare la conformazione del cuore ed i suoi battiti cardiaci attraverso una serie di sensori pochi costosi e che non richiedono un diretto contatto con l'utente; inoltre il battito cardiaco è un'informazione non clonabile in

alcun modo, in quanto rappresenta una caratteristica unica e non controllabile del corpo umano di un individuo. Attualmente l'EER del Cardiac Scan si piazza su un 4.52%, un valore troppo alto per poterne pensare una commercializzazione oggi, ma che rappresenta un'idea rivoluzionaria che potrebbe davvero cambiare in futuro il mondo dell'autenticazione di un sistema informatico.

Infine, durante il Mobile World Congress 2019, LG ha presentato il primo dispositivo al mondo che è in grado di riconoscere l'utente attraverso la posizione e lo spessore delle vene presenti nella mano. Questa caratteristica viene anche sfruttata per poter interagire con alcune parti del sistema, come l'apertura di determinate applicazioni o l'interazione con la fotocamera. Di certo questo non rappresenta un meccanismo trasparente di continuous authentication, come può essere quello di key-stroke dynamics (paragrafo 2.8.1), ma si tratta comunque di un'analisi biometrica che viene eseguita mentre l'utente sta per completare una specifica attività.

2.9 Sistemi multipli in una continuous authentication

Quelle che abbiamo trattato fino ad ora, rappresentano una serie di tecniche che possono essere implementate al fine di offrire una continuous authentication all'interno di un dispositivo. Ogni sistema però, preso singolarmente, può soffrire di un insieme di problematiche che nel complesso possono andare ad inficiare negativamente sul meccanismo di autenticazione [59]:

- Una raccolta di dati imprecisi causata da fattori esterni o ambientali, come quella derivante da una scarsa illuminazione, potrebbero non riconoscere correttamente l'utente.
- Una singola fonte di informazioni potrebbe non essere in grado di fornire un numero soddisfacente ed esatto di dati. Supponiamo ad esempio che un incidente comprometta temporaneamente alcune funzionalità fisiche dell'utente: in questo caso l'analisi di quegli elementi porterebbe alla realizzazione di modelli di autenticazione errati.
- Spoof attack⁹, potrebbero essere attuati ad esempio attraverso l'uso di fotografie.

Tutti i problemi evidenziati possono essere limitati andando a costruire un modello basato su più sistemi di autenticazione: risulta infatti molto più difficile realizzare spoof attack, nell'istante in cui l'attaccante deve simulare più meccanismi contemporaneamente. Nella tabella sottostante 2.2 è presente un sommario delle tecniche che sfruttano la combinazione incrociata di più fonti di dati per il riconoscimento dell'utente. Spiccano tra queste il modello proposto da Fridman [60], dove l'utilizzo

⁹Attacco in cui un malintenzionato o un programma impersonano in modo illegittimo l'identità di un individuo.

combinato della stilometria, il pattern delle applicazioni usate, il comportamento sul web e l'uso del GPS permettono di avere valori di EER che variano dall'1% al 5%, in base all'ampiezza temporale usata per raccogliere i dati.

<i>Autori</i>	<i>Tecniche utilizzate</i>	<i>Utenti testati</i>	<i>EER (%)</i>
Fridman et al. [60]	Stilometria, uso delle applicazioni, web browsing, GPS	200	1 (30 minuti) - 5 (1 minuto)
Shi et al. [61]	Accelerometro, touch screen, microfono, storia delle posizioni rilevate	7	-
McCool et al. [62]	Volto, voce	152	11.9 (maschi), 13.3 (femmine)
Sitova et al. [63]	Accelerometro, giroscopio, tocchi sullo schermo, keystroke dynamics	100	7.16 (in cammino), 10.05 (fermi)
Saevanee et al. [64]	Analisi linguistica del parlato e della scrittura, keystroke dynamics	30	3.3 - 4.4

Tabella 2.2. Schema delle tecniche basate sulla fusione di più sistemi di riconoscimento.

2.10 Altri sistemi di autenticazione

La continuous authentication rappresenta a tutti gli effetti un modo semplice, veloce e affidabile per autenticare l'utente durante la sua sessione lavorativa. Purtroppo però, come abbiamo visto anche negli esempi precedenti, non esiste un sistema in grado di garantire una sicurezza totale sia per quanto riguarda i Physiological Biometric che i Behavioral Biometric system. Per questo motivo tratteremo in questa parte una serie di sistemi che si focalizzano su aspetti non comportamentali dell'individuo e che allo stesso tempo non influiscano negativamente sull'usabilità dell'utente finale.

2.10.1 Graphical-based password keystroke dynamics

Introduciamo come primo aspetto un sistema di password hardening basato su due dei meccanismi di autenticazione di cui abbiamo già discusso precedentemente: le password grafiche e i meccanismi di keystroke dynamics. Quello che si va ad utilizzare è un modello simile allo schema di Blonder fondato sulla scelta di aree all'interno di un'unica immagine, a cui viene aggiunto una tecnica di riconoscimento dei tempi di pressione delle suddette zone. Il sistema di autenticazione è composto da tre fasi [65]:

- Enrollment phase, che rappresenta la fase di registrazione. In questa parte, data la dimensione degli schermi mobile, avremo una risoluzione dell'immagine ridotta a 50mm x 60mm, suddivisa in 30 quadrati 10mm x 10mm. L'utente dovrà cliccare su una serie di aree che porterà poi alla definizione della sequenza tramite cui ci si autenticerà.
- Classifier building phase, in cui viene creato il sample a partire dai dati acquisiti nella fase precedente.
- Authentication phase, ovvero la fase di autenticazione vera e propria.

Durante il processo di acquisizione dei dati, oltre ai dati relativi alla sequenza delle aree selezionate, verranno calcolati i tempi di pressione secondo 4 aspetti principali [65]:

- Down-Up (DU), il tempo che intercorre tra la pressione e il rilascio di un'area.
- Up-Down (UD), l'intervallo di tempo tra il rilascio di un'area e la pressione di quella successiva.
- Down-Down (DD), l'intervallo di tempo tra la pressione di un'area e la pressione di quella successiva.
- Up-Up (UU), il tempo che intercorre tra il rilascio di un'area e il rilascio di un'area successiva (figura 2.14).

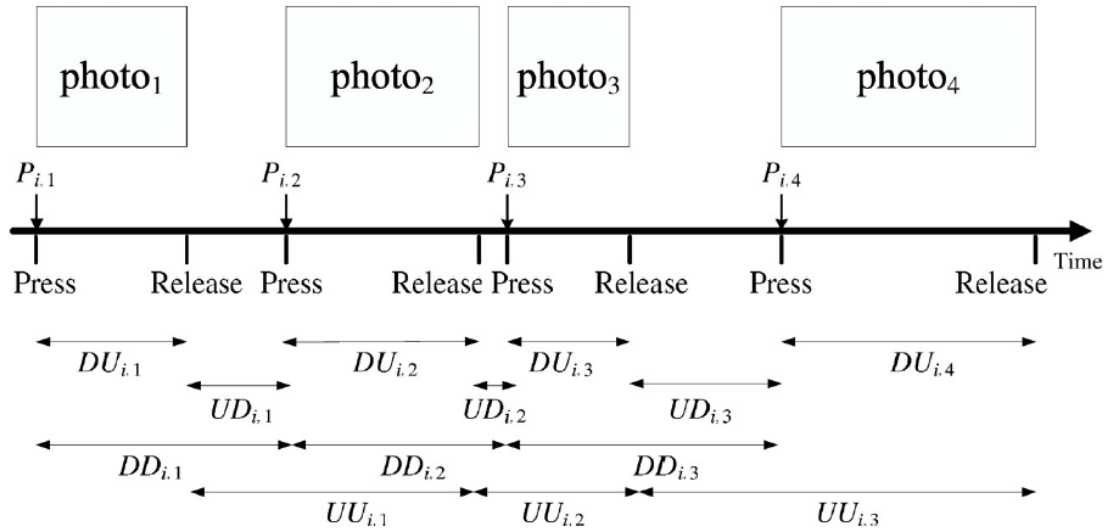


Figura 2.14. Grafico basato sulla tempistica delle pressioni di 4 aree (fonte: [65]).

La peculiarità di questo sistema è quello di avere un duplice monitoraggio in fase di autenticazione, che non riguarderà più soltanto la correttezza della sequenza, ma che esaminerà anche come essa verrà digitata; il sistema infatti non permetterà l'accesso all'utente che avrà digitato la sequenza nel modo giusto ma tramite una tempistica sbagliata.

Tra i vantaggi di questo meccanismo abbiamo sicuramente un controllo maggiore nei confronti degli shoulder surfing attack, la cui efficacia viene annullata dal keystroke dynamics presente all'interno del sistema. Inoltre si potrebbe pensare di ampliare la sua efficacia e la sua precisione grazie agli sviluppi tecnologici presenti negli smartphone, come utilizzare la peculiarità degli schermi moderni di riconoscere l'intensità della pressione, che permetterebbe di avere quindi un elemento di analisi ulteriore. Infatti, secondo gli studi condotti da Ting-Yi Chang [65], il sistema raggiunge un EER del 12.2% nella sua versione classica, ma si riduce fino al 6.9% per la controparte con sensore di pressione.

2.10.2 Systems Out-Of-Band-Based

Un sistema out-of-band-based è un tipo di autenticazione a due fattori che richiede un metodo secondario di verifica, attraverso un canale di comunicazione separato rispetto a quello usato per comunicare le informazioni primarie. Vedremo ora due sistemi di autenticazione che hanno bisogno di questo meccanismo per poter regolarmente funzionare.

OTP

Con OTP ci eravamo lasciati all'inizio di questo capitolo (paragrafo 2.2.2), cercando di capire come il server possa spedire in modo sicuro un elenco di password all'utente. In questo caso infatti diventa una necessità utilizzare un canale out-of-band, dal momento che l'impiego dello stesso mezzo, causerebbe un'elevata esposizione a replay e MITM attack. La comunicazione più efficace che si può intraprendere è sicuramente attraverso un contatto diretto tra colui che vuole fornire le password e l'utente, in quanto essa rappresenta l'unica circostanza esente da qualsiasi tipo di sistema informatico (un esempio concreto di azienda che implementa questa metodologia sono le banche, che utilizzano un canale diretto per fornire i codici d'accesso per i pagamenti online all'utente). Nonostante questo rappresenti il metodo più sicuro, il confronto diretto rimane un sistema poco usabile e spesso non adottato dalle grandi aziende, che preferiscono adottare mezzi differenti come la spedizione di un token elettronico o l'invio di SMS direttamente sul telefono dell'utente.

Fourth-factor authentication

Prima di poter parlare della fourth-factor authentication è bene introdurre con un esempio come funziona il sistema a due fattori, tramite l'uso di un PIN alfanumerico ed un token code:

- Fase di registrazione: in un primo momento si fornisce all'utente il PIN e il token code; viene inoltre creato e memorizzato lato server il template che se confrontato permetterà il riconoscimento del client.
- Fase di autenticazione: L'utente deve fornire al server presso cui vuole mostrare la propria identità, sia il PIN che il token code (in questo caso possiamo considerare ad esempio un TOTP).

Fourth-factor authentication [66] è un sistema in grado di fornire supporto nel caso in cui vengano persi uno o entrambi gli elementi necessari per l'autenticazione a due fattori. Esso si basa sull'utilizzo di un canale out-of-band sul quale una persona fidata, definita in fase di registrazione, fornisce un vouchcode con il quale l'utente può concludere la sua autenticazione. Vediamo nel dettaglio come si compone questa parte (vedi figura 2.15):

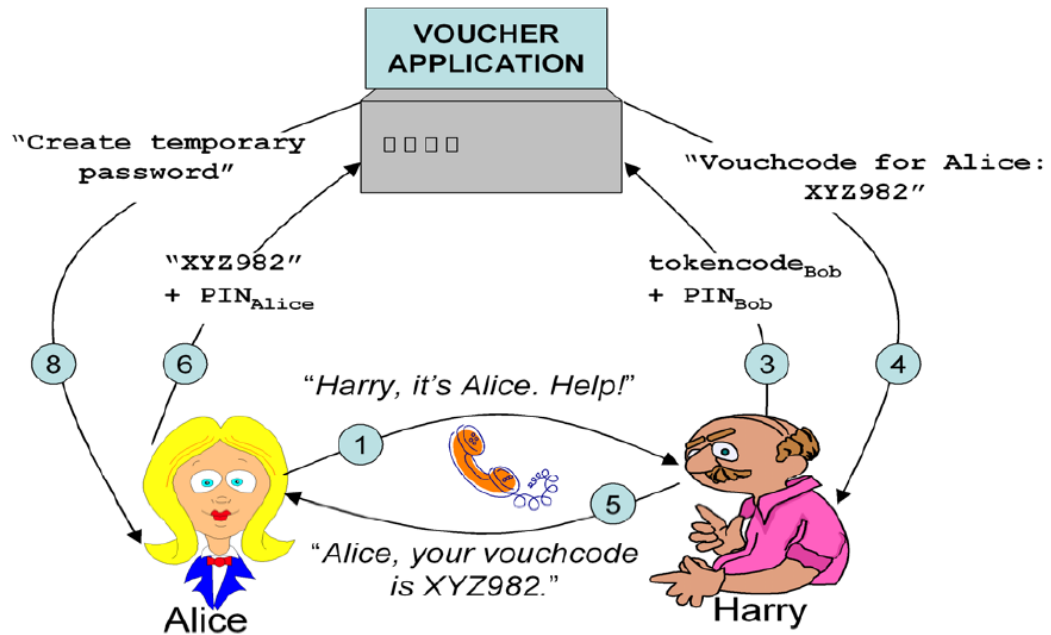


Figura 2.15. Le sei fasi della Fourth-factor authentication (fonte: [66]).

1. In questa prima fase l'utente contatta su un canale secondario l'aiutante definito in fase di registrazione.
2. L'aiutante deve in qualche modo riconoscere l'identità dell'utente. Per fare questo è possibile utilizzare una telefonata o il contatto diretto, mentre è vietato dal protocollo l'utilizzo di mail, in quanto mezzo soggetto a tecniche di phishing e quindi non affidabile.
3. L'aiutante, dopo avere riconosciuto l'identità dell'utente, deve utilizzare il suo dispositivo per autenticarsi al servizio web sul quale vuole autenticarsi anche l'utente.
4. Come risposta di un'autenticazione avvenuta con successo e dopo avere controllato la veridicità del rapporto che intercorre tra utente e aiutante, il server genererà un vouchcode necessario per terminare l'autenticazione dell'utente.
5. In base al canale secondario scelto nella fase primaria, l'aiutante comunicherà il vouchcode all'utente.

6. Dopo avere inviato il vouchcode, il server attiverà una sessione per il richiedente assieme ad un timeout. Il client avrà quindi a disposizione un breve periodo di tempo per potersi autenticare col server tramite il vouchcode fornitogli dall'aiutante.

La robustezza del meccanismo appena visto è fortemente dipendente dal tipo di interazione che utente ed aiutante decidono di intraprendere. Comunicare ad esempio il vouchcode a voce alta in mezzo ad una folla di persone potrebbe introdurre attacchi di tipo social engineering, rendendo di conseguenza tutto il sistema completamente inutile.

2.10.3 Multi-Factor Systems

I sistemi multi-factor rappresentano un ottimo meccanismo per rendere più sicura l'autenticazione ad un servizio: se da una parte l'impiego di una tecnica come la fourth-factor authentication [66] permetta di recuperare parte delle informazioni perse, è anche vero che il suo utilizzo richiede una certa quantità di tempo, che a volte può risultare fatale per la user-experience dell'utente. Andremo ora a vedere due sistemi che trattano in modo differente questa tipologia di struttura.

Authentication with plural Servers in the correct order

Il primo sistema che andremo a vedere ha l'obiettivo di rafforzare il protocollo multi-factor rendendo più difficile per un malintenzionato il compito di impersonare l'utente, basandosi sull'ordine con il quale vengono immesse le informazioni.

Ad esempio, come mostrato nella figura 2.16, supponiamo che l'utente voglia autenticarsi con un Service Provider e che per farlo debba ottenere tre certificati di autenticazione dai corrispettivi Identity Provider. Il sistema proposto da Yuki Kuzumazawa [67] non solo verificherà la correttezza dei certificati, ma anche l'ordine con il quale essi sono stati prodotti, aggiungendo quindi un livello ulteriore di sicurezza a quello standard. L'ordine con il quale devono essere contattati i vari IdP viene definito dall'utente in fase di registrazione, rimanendo un segreto tra lui ed il SP.

Multi-Sign-On Authentication Collector

Nei sistemi multi-factor l'usabilità ricopre un livello critico a causa della moltitudine di meccanismi che l'utente deve imparare a maneggiare per potersi autenticare; inoltre, nel sistema visto prima [67], questa criticità risulta ancora più rilevante a causa dell'informazione aggiuntiva introdotta dall'ordine in cui devono essere disposti i vari IdP. Se tutti i servizi web a cui noi siamo soliti accedere durante una giornata, avessero un'implementazione multi-factor, la facilità e l'immediatezza a cui siamo abituati verrebbe meno rendendo nel complesso molto frustrante la fase di autenticazione per ognuno di essi.

Per questo motivo andiamo a discutere di un sistema [68] basato su OpenID Single Sign-On (di cui verranno discussi i dettagli nel paragrafo 2.11.1) in cui l'utente può scegliere il tipo di IdP da usare per potersi autenticare. In particolare

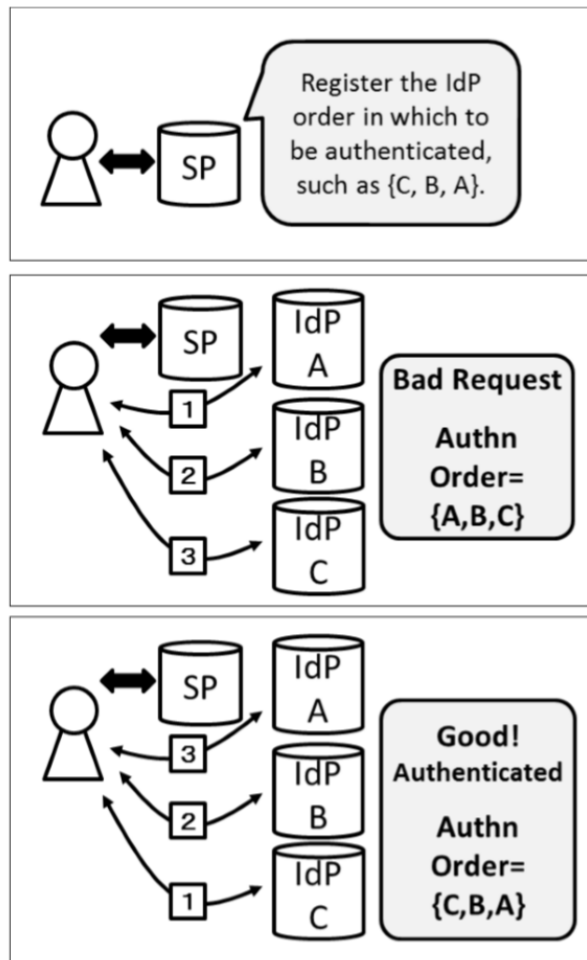


Figura 2.16. Esempio di autenticazione Multi-Factor in cui deve essere specificato l'ordine (fonte: [67]).

l'autenticazione si basa su un sistema a punteggio (vedi figura 2.16), formato da uno score primario associato all'SP e ad uno parziale legato al singolo IdP; l'utente dovrà autenticarsi con una serie di IdP, tali da rendere la somma dei loro punteggi parziali, maggiore o uguale al valore del punteggio di SP (ovviamente tutti gli IdP devono essere registrati presso il relativo SP). Lo score associato ad ogni IdP non è causale, ma è direttamente proporzionale alla sua sicurezza e robustezza: un sistema come la password avrà sicuramente un punteggio inferiore ad un sistema di riconoscimento biometrico, che a sua volta sarà maggiore di un token elettronico.

2.10.4 Zero-Trust Authentication

L'ultimo sistema che andiamo ad analizzare in questa categoria sfrutta l'abilità innata della mente umana di costruire legami semantici su un ristretto gruppo di concetti. L'idea di ZeTA [69] è infatti quella di ricreare un sistema d'interazione binario di tipo challenge-response su una coppia di parole scelte dall'utente in fase di registrazione assieme ad un operatore logico che le lega. Supponiamo ad esempio che vengano scelte le parole "Abete OR Cane" e "Rosso AND Mela". In questi

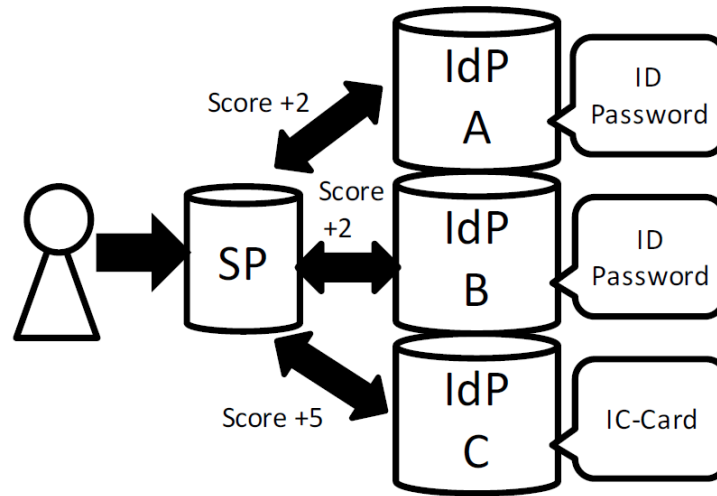


Figura 2.17. Esempio di un'architettura multi-factor basata su un sistema a punteggio (fonte: [68]).

casi alla domanda “È un animale?” riceverà una risposta positiva solo la prima coppia, mentre “È un colore?” avrà risposta negativa in entrambi i casi. Il numero di domande che vengono proposte cambia in base al contesto, così come la loro tipologia; questo viene fatto per evitare di incorrere in guessing attack, dove un malintenzionato potrebbe studiare l'esito delle risposte date per cercare di ricostruire un modello del segreto. Per quanto riguarda il discorso riguardante brute-force attack, esso è fortemente dipendente dal numero di domande che vengono poste all'utente; in ogni caso si può pensare di introdurre un login timer il cui valore aumenta ad ogni tentativo fallito, per sventare in questo modo questa tipologia d'attacco. Infine anche shoulder surfing attack hanno poca efficacia, sia perché l'utente non digita direttamente la coppia di parole tramite tastiera e sia perché la tipologia di domande cambia ad ogni tentativo di autenticazione. Nonostante questo, si tratta pur sempre di un segreto che viene messo nelle mani di un utente, che in quanto tale potrebbe venire soggiogato da tecniche di social engineering.

2.11 Sistemi federati

Dopo avere esaminato alcuni dei meccanismi di autenticazione, andiamo a vedere cosa offre il mercato attuale sulle implementazioni riguardante i sistemi federati.

2.11.1 OpenID Single Sign-On (SSO)

La peculiarità dei meccanismi chiamati “Single Sign-On”, è quella di poter effettuare il login su diversi siti utilizzando solo un “User Identifier”, definito dal sistema in fase di registrazione. Tutta l'autenticazione viene quindi spostata su un provider OpenID, richiedendo quindi all'utente di autenticarsi una sola volta per sessione;

questo permette di rendere il processo di autenticazione più sicuro, in quanto non si dovrà più andare a contattare più server gestiti da diverse entità. Questa scelta implementativa è un esempio di come un miglioramento relativo alla sicurezza provochi anche un miglioramento sull'usabilità dell'utente finale, al quale basterà ricordarsi un'unica combinazione di ID e password, per potersi autenticare su siti differenti. OpenID può anche implementare meccanismi di autenticazione aggiuntivi come OTP o certificati SSL. L'architettura su cui si basa OpenID è formata da (figura 2.18):

- User Agent, il browser web dal quale l'utente può utilizzare l'OpenID identifier per autenticarsi al sito (relying party).
- OpenID Provider, un server che conterrà i vari OpenID identifier degli utenti definiti in fase di registrazione.
- Relying party, il sito web a cui l'utente vuole collegarsi.

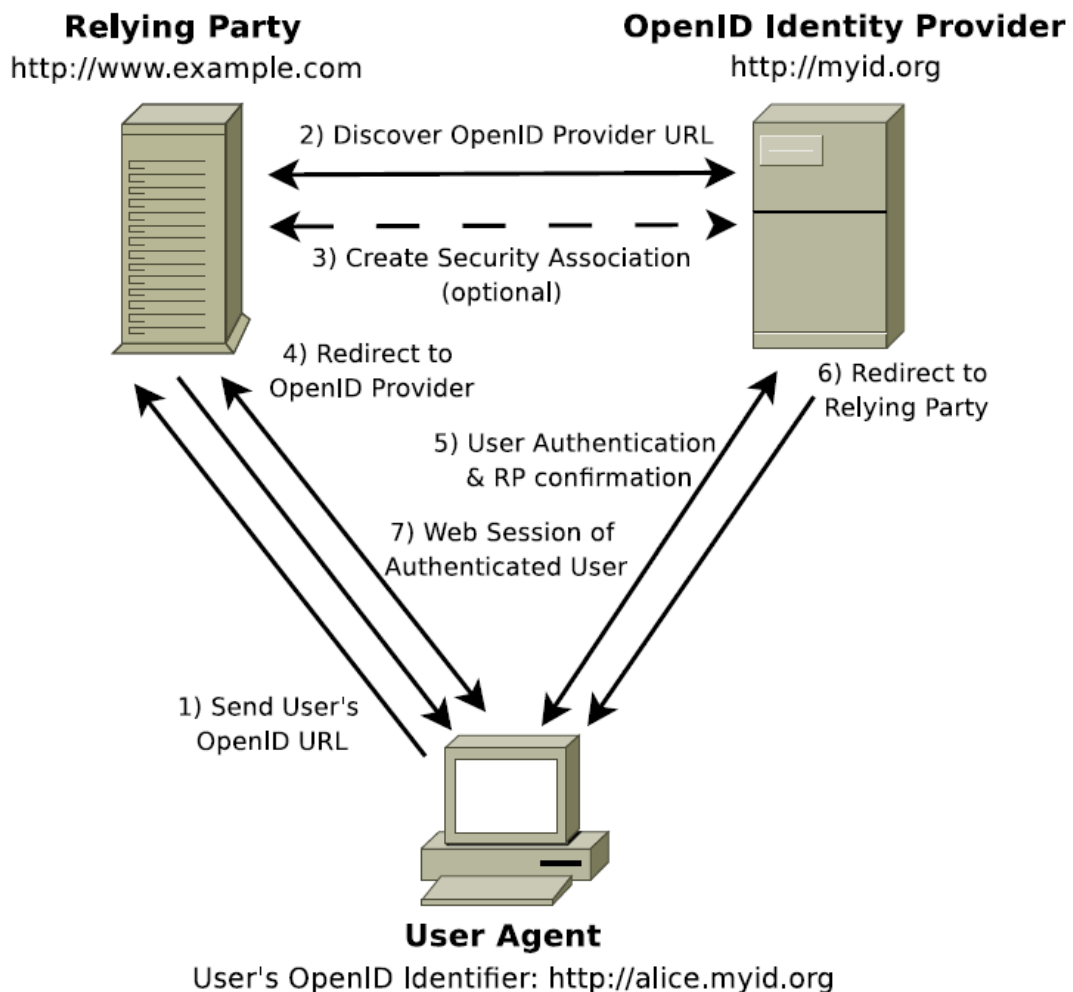


Figura 2.18. Modello di autenticazione tramite OpenID (fonte: [21]).

All'interno di questa architettura gli utenti vengono identificati da un URL chiamato OpenID identifier; questa scelta implementativa è stata fatta per facilitare la traduzione in indirizzo IP da parte del DNS, grazie alla sua natura gerarchica, ma presenta dei grossi problemi a livello di sicurezza dal momento che i dati scambiati tra il relying party e l'openID provider non vengono cifrati, esponendo l'utente a MITM attack. Questo non è un bug, ma rappresenta un errore di progettazione dell'architettura OpenID, che potrebbe essere rivista e migliorata rendendo ad esempio obbligatorio l'utilizzo di HTTPS.

In conclusione, questo sistema di autenticazione presenta, seppur in entità minore, lo stesso problema del meccanismo visto in precedenza, in cui alta usabilità è sinonimo di bassa sicurezza (OpenID Identifier scambiati in chiaro), mentre elevata sicurezza provoca un abbassamento dell'usabilità da parte dell'utente, che dovrà affiancare ad OpenID, altri meccanismi di autenticazione.

2.11.2 Security Assertion Markup Language

SAML [70] è uno standard open source basato su XML¹⁰ e definito da OASIS¹¹, che viene usato per la condivisione di risorse riguardanti l'identità, l'autenticazione e l'autorizzazione tra sistemi differenti. Il meccanismo di SSO può utilizzare proprio il protocollo SAML per richiedere, ricevere e formattare i dati necessari al sistema per autenticare l'utente.

La condivisione avviene attraverso l'assertion, un documento XML trasmesso tra service provider ed identify provider, che può contenere tre tipi di informazioni:

- Il tempo in cui l'utente ha effettuato l'accesso ed il metodo di autenticazione utilizzato.
- Attributi SAML, ovvero specifiche parte di dati che forniscono specifiche informazioni sull'utente.
- Una risposta fornita da un provider, che indica se l'autorizzazione ad una determinata risorsa può essere concessa o è negata.

Il funzionamento di SAML, all'interno di un sistema SSO, avviene nell'istante successivo all'autenticazione, quando l'identify provider diventa in grado di trasmettere le assertion ai service provider, dopo che l'utente tenta di accedere a dei determinati servizi. Dal punto di vista dell'usabilità, SAML rimane un meccanismo interno al sistema federato, risultando quindi completamente trasparente agli occhi dell'utente.

¹⁰Extensible Markup Language [71].

¹¹Organizzazione indipendente che sviluppa gli standard XML nel mondo web.

2.11.3 Open AuTHentication

L'obiettivo di OATH [72] è quello di realizzare un sistema di autenticazione aperto e senza diritti d'autore utilizzando, dove possibile, standard già esistenti, attraverso una serie di componenti hardware alla portata di tutti gli utenti. Gli scenari d'utilizzo di un sistema OATH sono molto vasti e spaziano dai siti online delle banche, a dispositivi desktop, fino ad arrivare alle VPN. Inoltre il sistema permette la coesione di framework diversi, impedendo in questo modo di essere vincolati ad un solo venditore. Andiamo a questo punto a vedere l'architettura di OATH (vedi figura 2.19):

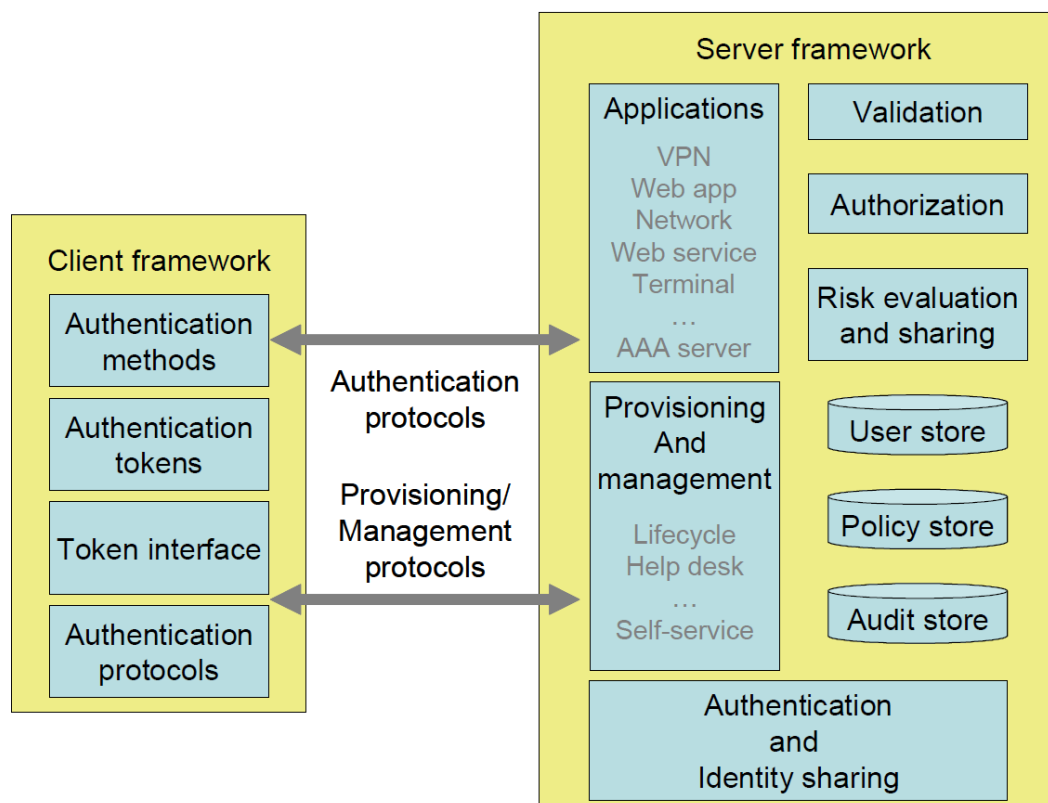


Figura 2.19. Modello di autenticazione tramite OpenID (fonte: [72]).

- Il client framework, rappresenta l'insieme di tutti i metodi, i tokens, le interfacce ed i protocolli implementabili su un'architettura OATH.
- Il provisioning e management framework, si occupa di gestire il ciclo di vita del software e delle credenziali di sicurezza per un sistema di autenticazione.
- Il validation framework, permette di verificare la validità delle credenziali in base al tipo di sistema implementato.
- L'applications, fa da intermediario tra il client che vuole autenticarsi e il protocollo che viene implementato.

- L'authorization, decide l'insieme delle risorse a cui il client può accedere dopo essersi autenticato (il concetto di autorizzazione verrà descritto nel capitolo 3).
- L'user store, è un DB all'interno del quale sono contenute le informazioni inerenti al client.
- Il policy store, è un unico DB all'interno del quale sono memorizzate le politiche di tutti i componenti dell'architettura (esiste anche un'architettura in cui viene definita una policy store dedicata per i singoli componenti).
- L'audit store, è un DB che contiene un repository con i controlli per i vari componenti (utilizzando strutture simili a quelle del policy store).
- L'authentication and identity sharing è la componente che implementa le tecnologie e i modelli necessari ad abilitare la condivisione del sistema di autenticazione.
- Risk evaluation and sharing, permette di valutare il rischio associato ad una particolare transazione.

A partire da questa architettura, analizzeremo ora nel dettaglio la parte riguardante gli authentication method, ovvero quella componente che attraverso i protocolli (che rendono possibile la comunicazione tra client e server), permette di descrivere come l'utente finale deve autenticarsi attraverso OATH. Tra i sistemi proposti sono presenti le password (nonostante esse siano nel complesso non raccomandate da OATH per i motivi di cui abbiamo ampiamente discusso all'inizio di questo capitolo), le OTP, i meccanismi challenge-response, le Transaction Signing¹², i certificati supportati da una PKI ed i sistemi biometrici. Pertanto andremo a vedere più da vicino quali sono le tipologie di RFC che seguono le politiche open source e free-royalty di OATH, e che in quanto tale possono essere implementate in questo modello di autenticazione.

Dynamic Symmetric Key Provisioning Protocol

Il protocollo di "Dynamic Symmetric Key Provisioning Protocol" (DSKPP) definito all'interno dell'RFC 6063 [73], viene utilizzato per l'inizializzazione e la configurazione di chiavi simmetriche in ambito locale o remoto. Il processo di autenticazione avviene attraverso una serie di messaggi scambiati tra client e server, il cui obiettivo è quello di associare la stessa chiave simmetrica alle due parti; per fare questo è prima necessario che il server verifichi l'autenticità del client (è bene precisare che la stessa verifica può essere fatta da parte dell'utente, attuando in questo modo una mutua autenticazione). Esistono due varianti di DSKPP: il protocollo two-pass, che supporta il trasporto di una chiave esistente verso un modulo crittografico, ed il four-pass in cui la generazione della chiave avviene dinamicamente in base ad una

¹²Un metodo avanzato di C-R in cui l'utente deve fornire oltre alla risposta una serie di dettagli relativa alla transazione in atto.

serie di variabili, attraverso una key agreement. Andiamo a valutare la sicurezza di questo meccanismo, analizzando una serie di attacchi informatici che è possibile attuare su questo tipo di comunicazione:

- Cancellazione dei messaggi, il cui unico risultato è quello di causare un denial of service sul servizio, non interrompendo però il lavoro del server.
- Inserimento di pacchetti, la cui pericolosità dipende dalla protezione che viene data ad un particolare campo “AuthenticationData” nel messaggio di inizializzazione di DSKPP: se infatti un’attaccante riuscisse a recuperare quel pacchetto potrebbe costringere l’utente a richiedere una nuova inizializzazione della chiave.
- I messaggi replay e di reordering possono essere rilevati e neutralizzati.
- Se la comunicazione non avviene attraverso un canale sicuro come SSL o TLS, un MITM attack sarebbe in grado di rubare la chiave simmetrica del client.

HMAC-based One-Time Password Algorithm

“HMAC-based One-Time Password” (HOTP) è un algoritmo OTP event-based, descritto nell’RFC 4226 [74], che rispetto al meccanismo tradizionale, sfrutta la funzione di hash HMAC per garantire integrità e autenticità ai messaggi scambiati tra client e server. Il funzionamento di HOTP si basa sull’incremento di un contatore e sull’utilizzo di una chiave simmetrica conosciuta unicamente dal token e dal server, che avrà il compito di validarla in fase di autenticazione. La chiave fornita dal token non viene inviata in chiaro ma attraverso una funzione HMAC-SHA-1, come definito nell’RFC 2104 [75]. A livello di sicurezza, l’insieme degli attacchi riproducibili su DSKPP non sono realizzabili, rendendo unicamente possibili brute-force attack. È da sottolineare che HOTP non è in grado di cifrare la trasmissione dei dati e che quindi bisognerà utilizzare dei meccanismi esterni per proteggere la riservatezza e la privacy dei pacchetti. Infine in termini di usabilità vale lo stesso discorso che è stato affrontato in precedenza per OTP.

OATH challenge-response Algorithm

“OATH challenge-response” (OCRA) [76] è un sistema aperto che sfrutta l’algoritmo HOTP al fine di realizzare una one-way o una mutua autenticazione. Per fare questo, OCRA amplia il concetto di One-Time Password attraverso l’utilizzo di una funzione crittografica (di default viene utilizzata HOTP-SHA1-6¹³) che riceve in input il segreto condiviso tra il client e il server, assieme ad una concatenazione di dati utili alla realizzazione della comunicazione. La sicurezza di questo sistema è direttamente influenzata dal tipo di funzione crittografica utilizzata: in questo caso avendo usato HOTP, l’unica tipologia di attacchi possibili sono quelli a forza bruta, sempre supponendo che la comunicazione tra le due parti avvenga tramite un canale sicuro come SSL, TLS o IPsec.

¹³L’ultima cifra della sigla indica il punto in cui viene troncato il risultato di hashing.

Portable Symmetric Key Container

“Portable Symmetric Key Container” (PSKC) [77] rappresenta un formato XML-based per il trasporto di chiavi simmetriche ed i relativi metadati, necessario per fornire supporto all’importazione e all’esportazione dei pacchetti tra client e server. La crittografia dei dati segreti, necessaria per mantenere l’integrità delle informazioni, può avvenire tramite l’utilizzo di DES, Triple-DES, chiavi asimmetriche, chiavi passphrase o firma digitale. Per quanto riguarda invece l’autenticità dei dati trasmessi, essa può essere garantita attraverso la cifratura del payload del pacchetto o, in alternativa, utilizzando uno dei protocolli già esistenti, con il quale viene realizzato un canale sicuro di trasmissione.

2.12 Framework usati per il management dei sistemi di autenticazione

Dopo avere visto una serie di implementazioni open source, vediamo invece alcuni modelli di autenticazione che richiedono una spesa per poter essere utilizzati dall’utente finale.

2.12.1 Vault

Vault [78] è un progetto di HashiCorp [79] che viene usato per gestire i “segreti” all’interno di un sistema di sicurezza informatico. Con il termine “segreti” si intendono tutte quei meccanismi di autenticazione utilizzabili da un utente per accedere ad uno o più servizi, come ad esempio nome utente e password, token API o certificati TLS. Una volta che il servizio Vault viene avviato, esso si trova in uno stato sigillato, che non permette di effettuare alcun tipo di operazione: in questa fase, tutti i dati sensibili vengono cifrati attraverso una chiave, generata in precedenza dal sistema. La chiave viene a sua volta protetta da una Master key che, attraverso l’algoritmo di condivisione del segreto di Shamir [80], viene suddivisa in 5 Key Share (vedi figura 2.20). Di default, il sistema richiede 3 Key Share per poter ricostruire una Master Key, ma è possibile variare a proprio piacimento il numero di parti in cui può essere suddivisa, fino ad ottenere una singola Key Share che coincide esattamente con la Master Key.

Dopo che l’utente si autentica, indipendentemente dal meccanismo che decide di utilizzare, il sistema Vault crea un token¹⁴ che viene associato all’ID di quell’utente: tutti gli accessi futuri a quella risorsa verranno fatti automaticamente attraverso CLI¹⁵, andando ad includere il token nella richiesta:

```
$ vault login -method=github token=<token>
```

¹⁴Il cui funzionamento è simile a quello delle sessioni su internet.

¹⁵Command-Line Interface.

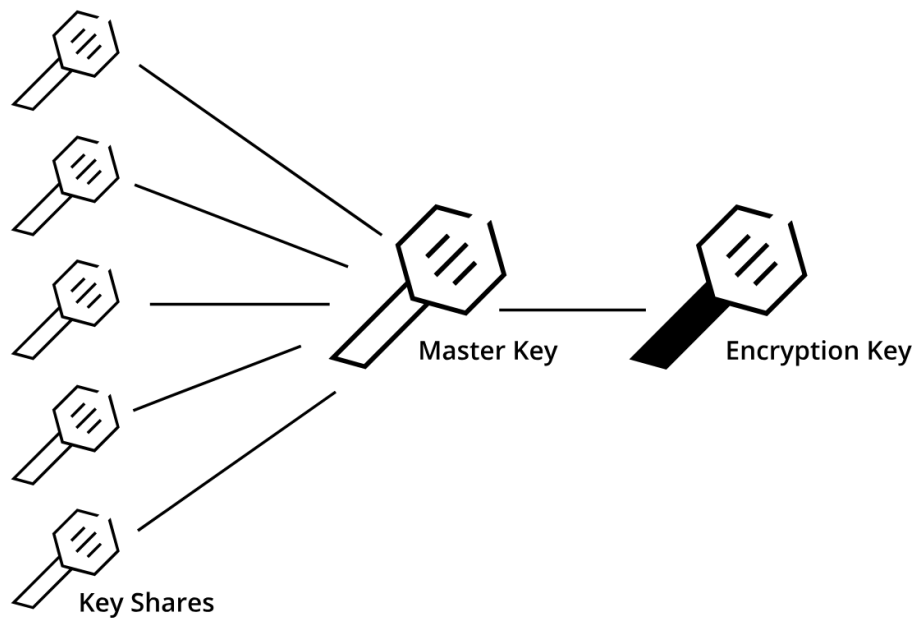


Figura 2.20. Schema delle chiavi utilizzate dal sistema Vault (fonte: [78]).

Se invece il metodo a cui si vuole accedere non supporta il CLI, la procedura deve essere fatta manualmente attraverso specifiche API.

2.12.2 Entrust IdentityGuard Adaptive Authentication

“Entrust IdentityGuard Adaptive Authentication” (EIGA) [81] è un sistema di autenticazione che sfrutta gli elementi e le caratteristiche presenti sul dispositivo dell’utente, per fornire un livello di protezione maggiore durante la fase di login, in modo del tutto trasparente. Per fare questo, EIGA richiede l’installazione di un adaptive authentication SDK kit, installabile sia su desktop che su mobile, necessario per validare, in fase di autenticazione, le politiche tramite cui verranno poi definite le granularità e il peso che verrà attribuito ad ogni accesso. Nella figura 2.21 sono presenti solo alcuni dei parametri che possono essere sfruttati da EIGA per definire l’autenticazione dell’utente. Possono infatti essere sfruttati elementi come: il certificato del dispositivo, la geolocalizzazione dell’IP tramite cui vedere l’appartenenza o meno di un blacklist o capire se è plausibile che l’utente si trovi proprio in quel posto, la velocità della connessione, il tempo e la durata delle sessioni o più in generale i parametri comportamentali dell’utente. In termini di usabilità EIGA è flessibile e totalmente trasparente, non influenzando quindi né negativamente, né positivamente rispetto al metodo di autenticazione implementato dall’utente.

2.12.3 IBM Security Access Manager

IBM Security Access Manager [82] è un software utilizzabile dalle aziende per controllare e semplificare il processo di autenticazione degli utenti che usufruiscono dei loro servizi. Esso si basa sull’utilizzo di una serie di politiche definite dall’azienda

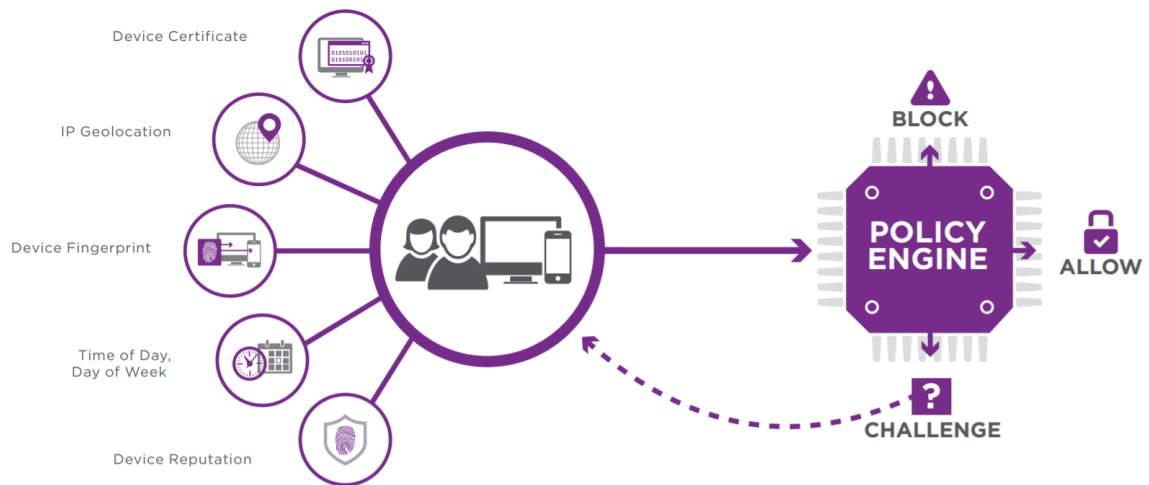


Figura 2.21. Schema della policy di EIGA (fonte: [81]).

che vengono applicate nell'istante in cui l'utente intraprende determinate azioni. Supponiamo ad esempio che il software venga utilizzato da una banca per gestire gli accessi dei suoi clienti in base al tipo di contratto stipulato: l'azienda potrebbe decidere di impostare un vincolo su una somma di denaro oltre il quale ogni tipo di transazione verrà rifiutata, oppure definire il tipo di autenticazione che l'utente dovrà affrontare alla richiesta di un particolare servizio. Tutto il meccanismo è gestito in modo totalmente automatico da parte dell'utente, il quale in base alle scelte da lui intraprese otterrà un rifiuto, un'accettazione o la richiesta di un'ulteriore conferma. Un altro prodotto interessante della menzionata azienda statunitense si chiama IBM Security Access Manager for Single Sign-On e rappresenta un programma, questa volta fruibile dall'utente, che permette di usufruire di un meccanismo SSO simile a quello visto nell'implementazione open source di OpenID.

2.13 Conclusione

Abbiamo analizzato in questo secondo capitolo lo stato dell'arte per quanto riguarda i sistemi di autenticazione. Nel capitolo successivo andremo a completare l'analisi tecnica focalizzandoci sul processo di autorizzazione, per poi vedere, nel quarto capitolo, quale sia l'usabilità percepita dagli utenti rispetto alle varie metodologie trattate.

Capitolo 3

Autorizzazione

L'Autorizzazione [83] viene definita come un sistema tramite cui salvaguardare:

- La riservatezza, garantendo che le informazioni siano accessibili solo a coloro che sono autorizzati ad avere accesso.
- L'integrità, tutelando l'accuratezza e la completezza delle informazioni e dei metodi di elaborazione.
- La disponibilità delle risorse, assicurandosi in questo modo che gli utenti autorizzati abbiano accesso alle informazioni e alle risorse associate quando richiesto.

Questo tipo di meccanismo è necessario all'interno di un sistema di sicurezza informatico, perché permette di creare una sorta di gerarchia, tramite la quale definire il ruolo e il range delle azioni che ogni singolo utente avrà nei confronti di una determinata risorsa.

3.1 Politiche e controlli sugli accessi

Un sistema di autorizzazione deve come prima cosa definire una politica di sicurezza, ovvero un insieme di proprietà di alto livello, che descrive quali devono essere gli accessi autorizzati per un determinato sistema informatico. Per fare ciò è necessario utilizzare un modello, come ad esempio:

- Modello Bell-La Padula [84], usato per garantire la segretezza delle risorse attraverso:
 - Classi di accesso, che definiscono una classificazione gerarchica degli elementi.
 - Modi di accesso che definiscono i diritti (read, append, execute e write) con il quale poter interagire su di essi.

- Modello Biba [85], che integra il modello precedente fornendo l'integrità dei dati.

Oltre ad una politica di sicurezza, è necessario però fornire anche un meccanismo di basso livello, sia software che hardware, che sia in grado di implementarle, come ad esempio un Access Control List tramite il quale è possibile rilevare chi ha diritto ad accedere ad una determinata risorsa e con quali privilegi. Esistono svariati modelli per implementare il controllo degli accessi, come ad esempio:

- Discretionary Access Control (DAC) [86], in cui il proprietario della risorsa decide chi può accedere ad essa e con quali privilegi. La vulnerabilità principale di questo sistema risiede però nell'alta libertà che viene data all'utente finale, il quale potrebbe fornire privilegi ad un possibile attaccante.
- Mandatory Access Control (MAC) [87], in cui è la politica definita in precedenza a stabilire chi può accedere ad una risorsa e con quali modalità, realizzando quindi una gestione degli accessi centralizzato e definito a priori. Questo meccanismo limita le possibilità dell'utente, ma permette di avere un controllo più sicuro sulle risorse.
- User-Based Access Control (UBAC), dove l'ACL contiene la lista degli utenti autorizzati ad accedere ad una determinata risorsa. Poco flessibile nei sistemi composti da molti utenti.
- Role-Based Access Control (RBAC) [88], in cui il controllo degli accessi ed i privilegi sulle risorse, sono definiti in base ai ruoli che i singoli utenti assumono all'interno di una determinata organizzazione.

I modelli visti finora rappresentano una visione tradizionale, e in alcuni casi obsoleta, per implementare una politica di controllo degli accessi. Quello che andiamo ad analizzare ora, invece, riguarda schemi differenti che cercano di migliorare alcune criticità dei suddetti modelli.

3.1.1 Access Control tramite Path Condition

Nel paragrafo precedente, i primi due modelli presentavano dei grossi limiti a causa del forte vincolo sull'individuo e sui suoi diritti d'accesso; questo è stato tuttavia risolto con l'introduzione del modello RBAC che, sostituendo l'accesso della persona con quello del ruolo che esso ricopre, ha introdotto maggiore flessibilità e scalabilità, due fattori fondamentali in un sistema di autorizzazione. Nonostante RBAC sia uno dei meccanismi più utilizzati esso presenta comunque dei limiti, derivanti dal mascheramento di informazioni riguardanti le relazioni che sussistono tra i singoli individui e le rispettive risorse. Quello di cui andremo a parlare in questo paragrafo, riguarda l'utilizzo di un approccio che combina la scalabilità del modello RBAC con la bassa granularità degli altri modelli.

Il meccanismo di access control tramite path condition proposto da Crampton [89], utilizza un sistema a grafo in cui i nodi rappresentano le entità, che possono

essere concrete e rappresentare quindi l'individuo o una risorsa, o logiche e riferirsi quindi ad un gruppo di entità, esattamente come accade nel modello RBAC. Le relazioni tra le varie entità vengono invece definite attraverso una "path condition", ovvero un insieme di policy formate da delle azioni, che indicano se l'entità è autorizzata o meno, e da delle regole che se verificate scatenano le azioni relative. Se vengono verificate più regole, si può decidere di applicare tutte le relative azioni, attraverso un principio di priorità, o solo le azioni relative alla prima. D'altra parte, se non viene verificata alcuna regola, è possibile adottare un'azione di default. Da un punto di vista pratico, questo modello permette di relazionare tre tipologie diverse di entità, rappresentabili attraverso un user-object, un user-group e un group-object; questo consente di garantire un'elevata flessibilità a livello implementativo, con la possibilità di poter riprodurre lo stesso modello RBAC. Tuttavia un aspetto da considerare risiede nella complessità intrinseca nella struttura dati a grafo: prendiamo ad esempio in analisi la fase di path matching dove, dato un soggetto, la risorsa a cui si vuole accedere ed una path condition, viene ritornato un valore booleano che permette di verificare se i requisiti sono stati soddisfatti oppure no. Questo tipo di operazione, associata alla struttura dati utilizzata, appesantisce inevitabilmente l'algoritmo generale del sistema.

3.1.2 Context-Based Access Control in sistemi mobile

Spostiamo ora l'attenzione su uno scenario differente, quello del mobile. Negli ultimi anni la tecnologia che viene utilizzata per costruire gli smartphone è sempre più avanzata, rendendoli dei veri e propri computer piuttosto che dei semplici terminali adibiti alle chiamate. Oltre ad aumentare la capacità di calcolo, è aumentato però anche il numero di sensori montati su di essi, incrementando di conseguenza il rischio che dati sensibili, ricavati da applicativi di terze parti, vengano utilizzati per usi impropri senza l'esplicita autorizzazione dell'utente. Prima di entrare però nel dettaglio del sistema di autorizzazione proposto in questo paragrafo, è bene precisare come avviene il controllo degli accessi all'interno di uno smartphone. L'OS che prenderemo in analisi è Android, un sistema operativo che si basa sul kernel linux e che, grazie ad un'architettura multi-layered, permette agli sviluppatori di migliorare il supporto per la sicurezza e la privacy [90]. Il meccanismo che veniva utilizzato nelle versioni di Android meno recenti, prevedeva che prima dell'installazione di un applicativo venisse dichiarata una lista di permessi, definiti all'interno del file "AndroidManifest.xml", che rispettavano il modello Ask-On-Install (AOI): affinché l'app venisse installata correttamente sul dispositivo, l'utente doveva accettare categoricamente tutti i permessi, non avendo la possibilità di decidere su quali elementi della lista dare il consenso o meno. Se l'utente non approvava l'intera lista di permessi, l'applicazione non veniva installata, limitando indirettamente la libertà di scelta dell'utente. Tuttavia, dalla versione Marshmallow di Android, è stata introdotto un nuovo metodo di gestione dei permessi chiamato Ask-Of-First-Use (AOFU), in cui i permessi ad una risorsa vengono richiesti, dopo l'installazione, solo

nell'istante in cui l'applicazione la sta usando per la prima volta¹. In questo caso viene adottato un modello di tipo Identity Based Access Control, dove i singoli permessi vengono specificati individualmente per ogni applicativo. Purtroppo Android soffre di una forte frammentazione del suo sistema operativo, ed i dati del 5 febbraio 2018 rilevano che quasi il 50% degli utenti Android utilizzi ancora una versione del sistema operativo antecedente a Marshmallow [91]. Per quanto riguarda l'usabilità del meccanismo AOFU, rimando alla lettura del paragrafo 4.5.3.

Per quanto riguarda entrambi i modelli analizzati, AOI e AOFU, è chiaro come essi costituiscano una limitazioni nella libertà di scelta dell'utente, soprattutto se si considera che ipoteticamente un'applicazione può usufruire di quei permessi anche quando non viene esplicitamente avviata dall'utente. Un modello che può essere utile in questo caso è il CBAC [92], con il quale il controllo degli accessi su un applicativo viene calcolato in base al contesto in cui l'utente si trova. Il contesto può essere definito attraverso l'analisi di due elementi:

- Il tempo, andando a definire il range temporale in cui la politica può essere applicata.
- Il luogo, che specifica dove la politica può essere applicata. Esso rappresenta un elemento fondamentale che ben si lega con il concetto di mobilità dello smartphone.

In particolare la posizione dell'individuo può essere rilevata o tramite il sistema di triangolazione del GPS e della rete mobile, che sono in grado di definire una buona approssimazione in ambienti esterni, oppure attraverso l'utilizzo del wi-fi che, grazie alla precisione più elevata, è in grado di rilevare l'utente in tutti i contesti indoor. Alcuni esperimenti condotti da Shebaro [92] hanno evidenziato come questi tre meccanismi utilizzati per il monitoraggio costante dell'individuo², provochino un consumo della batteria del 5% in più, ogni ora, rispetto a quando il sistema CBAC non viene abilitato. Questo rappresenta sicuramente uno scoglio sull'autonomia generale del dispositivo, che potrebbe però essere parzialmente risolto andando a rilevare la posizione dell'utente in intervalli di tempo più ampi.

Per quanto riguarda l'architettura, CBAC è formato dai seguenti elementi (vedi figura 3.1):

- Il Context Provider (CP), che attraverso un database interno mantiene ed aggiorna i dati riguardante la posizione dell'individuo.
- L'Access Controller (AC), che assegna o toglie l'autorizzazione di una risorsa ad un determinato applicativo.
- Il Policy Manager (PM), che fornisce un'interfaccia tramite cui poter realizzare delle policy collegabili ad un contesto specifico.

¹I singoli permessi possono essere negati, ma questo potrebbe portare ad un non funzionamento parziale o totale dell'applicazione.

²In particolare il sistema rileva la posizione dell'utente ogni 30 secondi.

- Il Policy Executor (PE), che esegue le policy generate tramite il processo di PM.

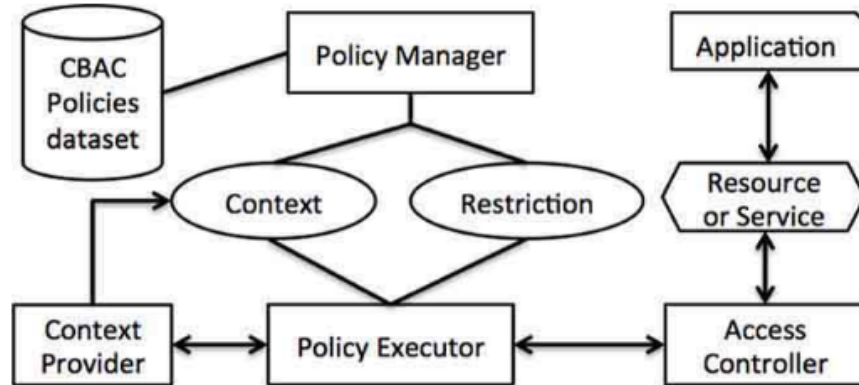


Figura 3.1. Architettura CBAC (fonte: [92]).

Nel momento in cui un applicativo vuole fare accesso ad una risorsa, l'AC verifica se essa è autorizzata facendo una richiesta al PE; se la richiesta è autorizzata allora il PE controlla se ci siano policy associate all'applicativo. Se esiste almeno una policy, il PE fa richiesta al CP per rilevare il contesto posizionale e temporale rispetto al quale quella politica è stata definita: solo se quel contesto combacia con quello presente nella policy, l'AC potrà applicare le restrizioni contenute nella policy stessa, mentre in caso contrario essa verrà ignorata. Questo meccanismo permette quindi di poter definire delle limitazioni sul numero di risorse a cui un app può accedere.

In termini di sicurezza, il sistema CBAC è in grado di risolvere inoltre un problema riguardante l'UID, un user ID univoco che il sistema operativo assegna ad un applicativo per riferirsi ad esso; nel caso in cui due applicazioni siano state sviluppate dalla stessa società, ad entrambe verrà assegnato lo stesso UID. Il problema nasce dal fatto che il sistema operativo Android non applica le policy sull'applicazione ma sull'UID: questo permetterebbe a due applicazioni diverse (e potenzialmente anche con privilegi differenti) ma con lo stesso UID, di poter condividere le risorse, scatenando un *privilege escalation attack*³ [93]. Tramite CBAC invece, la policy viene adottata sul nome del package legato all'applicativo e non più sull'UID [92].

3.1.3 Attribute-Based Access Control

In un modello ABAC, il controllo degli accessi ad una risorsa viene verificato attraverso una combinazione di attributi che, rendendo valida la policy, permettono all'utente di accedere alla risorsa. La definizione di un attributo non ha vincoli particolari, ma deve appartenere ad una delle seguenti categorie:

³Attacco in cui viene sfruttato un bug o un difetto di progettazione del sistema per ottenere accesso a delle risorse protette da un utente o un applicativo.

- Attributi relativi al soggetto, ad esempio riguardanti il ruolo che ricopre all'interno di un'azienda.
- Attributi del soggetto in base all'azione che vuole compiere su una determinata risorsa.
- Attributi relativi all'oggetto e che descrivono quindi la risorsa.
- Attributi relativi allo scenario in cui si sta svolgendo l'azione.

In aggiunta i metadata permettono di definire alcune informazioni su come deve essere utilizzato un attributo all'interno di un determinato scenario. In particolare, i metadata che descrivono gli attributi si dividono in tre categorie [94]:

- Accuracy, in cui viene descritta la semantica e la sintassi corretta da utilizzare per usufruire correttamente di un determinato attributo.
- Integrity, dove vengono descritti i protocolli da utilizzare per permettere una trasmissione sicura, tramite cui mantenere l'integrità e la confidenzialità degli attributi trasmessi.
- Availability, necessari per garantire il caricamento e il recupero degli attributi da parte del relying party, ovvero il sistema che fornisce l'accesso data una determinata richiesta.

ABAC risulta nel complesso un modello molto utilizzato, grazie alla sua elevata flessibilità e alla possibilità di migliorare sistemi già esistenti. RBAC infatti, nonostante sia un modello popolare per la gestione degli accessi controllati, ha una serie di limitazioni derivate dalla bassa usabilità nel setup iniziale e nel controllo, oltre a non fornire un supporto adeguato alla definizione di attributi dinamici in un sistema multi-dominio. Tramite l'uso del sistema appena descritto, è invece possibile inserire gli attributi nel modello, rendendolo complessivamente più semplice ed efficace da utilizzare [95]. D'altro canto, se in un sistema RBAC è facile intuire quali siano le risorse a cui l'utente può accedere, la stessa cosa non può essere detta per un modello ABAC dove, per determinare l'esito di un'autorizzazione, è necessario eseguire un set di regole più o meno ampio. Infine, entrambi i modelli soffrono di una struttura fortemente centralizzata che oltre ad inficiare sulla robustezza (trattandosi infatti a tutti gli effetti di un single point of failure), creano anche problemi sul carico che il server deve affrontare in caso di richieste multiple [96].

Analizziamo ora XACML, uno standard usato per la definizione di policy che viene applicato proprio nel modello Attribute Based Access Control appena visto.

3.2 eXtensible Access Control Markup Language

XACML [97] è un linguaggio definito da OASIS basato su XML, che permette di descrivere le politiche di autorizzazione tra un soggetto ed una risorsa. Di per sé, XACML definisce il formato dati che deve essere utilizzato per imporre una richiesta

e per generare la relativa risposta, ma non impone alcun vincolo sul protocollo che deve essere utilizzato per trasferire i suddetti messaggi.

Ipotizziamo ora uno scenario piuttosto classico di utilizzo, composto da un utente che vuole accedere ad una determinata risorsa (vedi figura 3.2). Supponiamo inoltre che un'eventuale fase di autenticazione sia stata già completata in precedenza, e che le policy relative alla risorsa siano state definite da uno o più utenti. La richiesta dell'utente viene smaltita da un Enterprise Service Bus, un'infrastruttura software che è in grado di gestire richieste multiple proveniente da diverse fonti su una o più risorse; essa comunica direttamente con il Policy Enforcement Point. Il PEP contatterà il Policy Decision Point, l'unità che, dopo avere raccolto una serie di parametri, deciderà se concedere o meno l'accesso a quella risorsa. Dal momento che solitamente il PEP è un Firewall, e in quanto tale non è in grado di comunicare in XACML, la richiesta viene prima intercettata da un Context Handler, che farà da tramite durante tutte le comunicazioni. Il PDP, per prendere la decisione finale, può attingere a due tipologie di informazioni:

1. Al Policy Information Point. Questo passaggio è opzionale, ma permette di raccogliere attributi aggiuntivi sulla richiesta di accesso che il PDP sta valutando. L'interazione tra PDP e PIP passa dal CH, mentre il PIP può attingere al database dell'utente attraverso diversi sistemi, come RDBMS⁴, directory o flat structured file⁵.
2. Al Policy Administration Point. Questo tipo di richiesta è obbligatoria in quanto permette al PDP di recuperare le policy associate alla risorsa.

Il PDP a questo punto invierà la XACML response al CH, che indicherà al PEP se la richiesta di accesso ha avuto un esito positivo o meno. Il PEP, prima di fornire la risposta definitiva all'utente, dovrà contattare l'Obligations Service, che andrà a controllare la validità e la correttezza degli attributi relativi alla policy.

3.2.1 XACML: tecniche avanzate

Andremo ora a trattare una serie di tecniche che, pur basandosi sempre su XACML, introducono alcuni concetti che sono in grado di migliorare il modello tradizionale sotto diversi aspetti.

Set-based Algebra-XACML

Con l'incremento dei Web service e la loro composizione in soluzioni web sempre più complesse ed articolate, è cresciuta di conseguenza anche la complessità e la dimensione di una policy XACML. Dal momento che il processo decisionale attuato dal

⁴Relational DataBase Management System.

⁵Database in cui i file sono memorizzati in plain text.

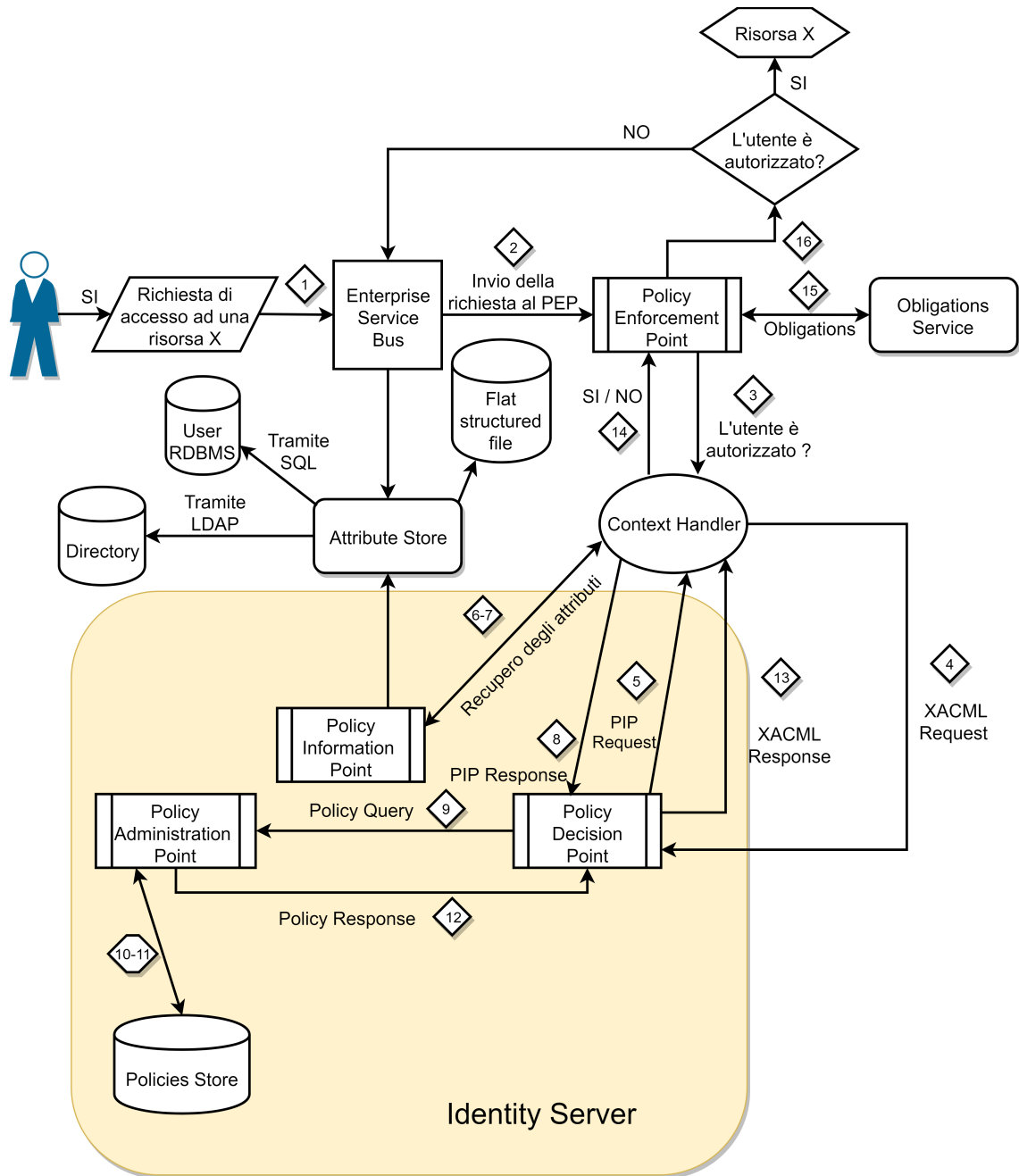


Figura 3.2. Architettura di un sistema di autorizzazione ad una risorsa con valutazione delle rispettive policy.

PDP deve essere fatto in tempo reale per ogni richiesta ricevuta da un PEP, è naturale che policy più complesse richiedano più tempo per essere gestite, provocando un vero e proprio collo di bottiglia nel processo di autorizzazione globale.

SBA-XACML [98] si propone come un linguaggio che, pur mantenendo la stessa struttura di XACML, riesce a rendere più efficiente il processo di valutazione di una policy, attraverso l'introduzione di un'algebra formale e una logica deduttiva migliorata. L'architettura di SBA-XACML sfrutta un compilatore in grado di convertire la sintassi e la struttura del linguaggio in XACML: questo può essere fatto in

modo del tutto indipendente durante la fase di valutazione della policy, attraverso una “Policy Evaluation Module”. In particolare questo modulo, dato come valore d’ingresso una richiesta SBA-XACML e un set di policy, è in grado di rilevare in tempo reale se le regole sono soddisfatte o meno, fornendo la risposta associata.

Tutto questo viene fatto con un’efficienza di valutazione migliore rispetto al modello classico, che sfrutta un livello intermedio in cui le policy vengono automaticamente convertite e che permettono quindi di elaborare richieste dalle 2.4 alle 15 volte più velocemente, in base alla dimensione della policy.

XACML with an Ontology Web Language

Discutiamo in questo paragrafo di un’implementazione di XACML che avviene attraverso OWL, un linguaggio web semantico progettato per rappresentare elementi complessi e le loro relative relazioni. Rispetto all’architettura classica di cui abbiamo discusso in precedenza, il linguaggio che andiamo ad introdurre in questo paragrafo utilizza il concetto di Description Layer Reasoner, un componente che si interfaccia direttamente con il PDP e che, al fine di ricavare la valutazione di una policy, permette di generare tre differenti ontologie [99]:

- Una Policy Terminological BOX (TBOX), che contiene le policy attive, ricavate attraverso il risultato dell’operazione algoritmica attuata dal PDP.
- Un Domain TBOX, che rappresenta un ritratto significativo del dominio dell’applicazione contenente l’intera collezione degli oggetti e delle relazioni coinvolte.
- Un Domain Assertional BOX (ABOX), contenente le diverse descrizioni degli individui e delle risorse coinvolte.

Sia il Domain TBOX che l’ABOX sono memorizzati e gestiti dal PIP.

Per quanto riguarda il processo di valutazione, esso segue le seguenti fasi:

- Il PDP dopo aver ricevuto una XACML request da parte di un context handler, recupera la policy TBOX dal PAP, e la Domain TBOX e l’ABOX dal PIP.
- Successivamente vengono interrogate le ontologie appena recuperate definendo in questo modo se il soggetto ha diritto o meno di accedere ad una determinata risorsa.
- Il PDP genera infine un messaggio di XACML response che verrà inoltrato al context handler.

In termini di prestazioni, sebbene la conversione da una policy alle relative strutture ontologiche non comporti l’utilizzo di grosse risorse, la stessa cosa non può essere detta per il processo di validazione: in questo caso infatti, come viene anche evidenziato nell’articolo “Using OWL Reasoning for Evaluating XACML Policies” [99], abbiamo tempistiche di processamento fino a due ordini di grandezza inferiori rispetto a SBA-XACML [98]. Questa problematica potrebbe tuttavia essere risolta realizzando un codice più ottimizzato, accompagnato da un hardware progettato per reagire in modo più reattivo agli stimoli provenienti dai vari PEP.

Privacy-Preserving XACML

Nel modello tradizionale di XACML, una volta che la risorsa richiesta viene correttamente autorizzata dal PDP, essa viene direttamente trasferita all'utente finale. XACML non prevede alcun meccanismo in grado di cifrare i dati durante il trasferimento e, per questo motivo, questa fase potrebbe presentare alcune criticità. Al fine di risolvere queste problematiche, si potrebbe ampliare il modello originale andando ad introdurre la cifratura delle risorse, accompagnate dalla relativa firma digitale, che permetterebbero di garantire in questo modo la riservatezza e l'autenticità dei dati trasmessi. Il modello proposto da Seol [100], realizza questo meccanismo all'interno di un contesto medico, dove i pazienti vogliono accedere alle proprie analisi. In questo caso i file memorizzati nel database dell'ospedale sono scritti in XML, e vengono usate le proprietà di XML encryption e XML digital signature, per applicare rispettivamente la cifratura e la firma digitale (figura 3.3).

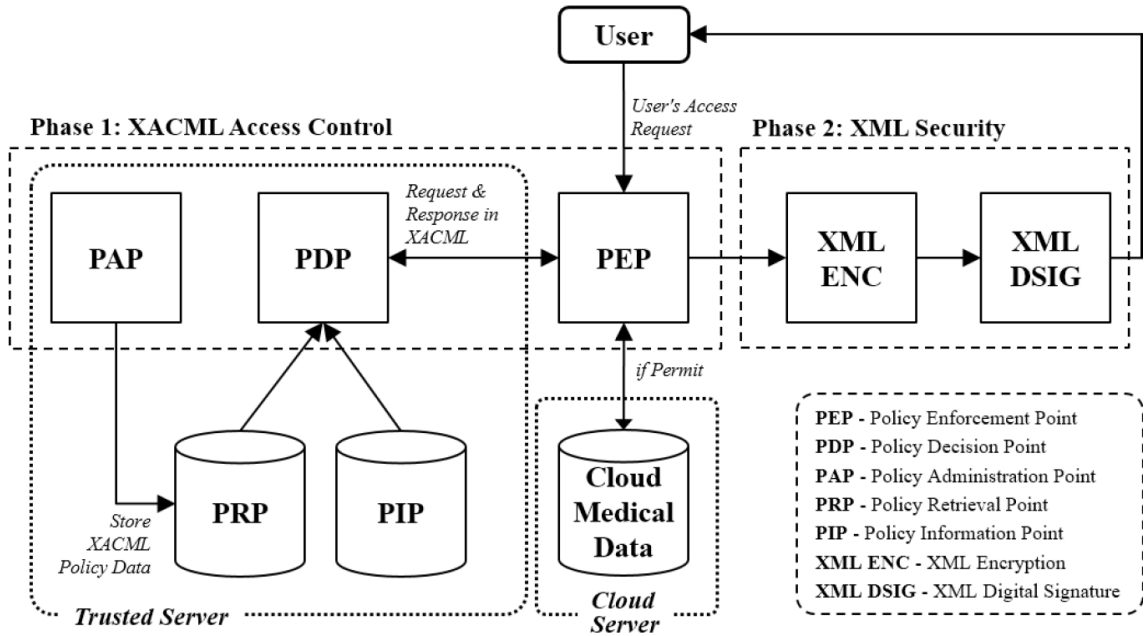


Figura 3.3. Modello proposto in ambito medico da [100]).

Fine-Grained Access Control using XACML policy

Il modello XACML definito da OASIS definisce l'insieme delle specifiche tramite cui poter implementare un profilo RBAC [101]. Questo specifico meccanismo risulta sicuramente funzionale, ma non è in grado di fornire un differente livello di accesso alla risorsa, in base al tipo di granularità richiesta. Per ovviare a questa problematica, Shibli [102] propone un profilo XACML basato su un modello Fine-Grained Access Control (FGAC). A livello pratico non viene intaccato il modello tradizionale, ma bensì estesa la parte del Policy Administration Point (figura 3.4), che ora raggrupperà le policy su più livelli, in base alla granularità richiesta.

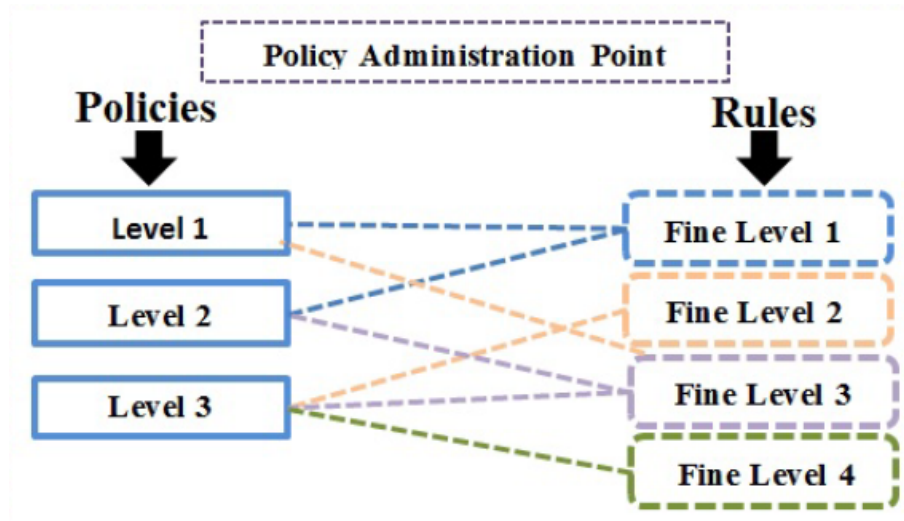


Figura 3.4. Policy Administration Point with Fine-Grained Access Control of policy (fonte: [102]).

Access Control Policy with Satisfiability Modulo Theories

La sintassi espressiva proposta da XACML per la definizione di policy può risultare poco intuitiva e, per questo motivo, portare l'autore a commettere degli errori durante la loro scrittura. Risulta quindi necessaria l'introduzione di un software che rilevi eventuali problematiche, attraverso dei meccanismi automatici che siano in grado di verificare che una certa proprietà venga applicata nel modo corretto alla policy.

X2S [103] è un framework di verifica delle policy XACML che, basandosi sull'uso di un Satisfiability Modulo Theories, fornisce un miglioramento significativo sulle prestazioni, oltre che garantire un'analisi più approfondita rispetto ai sistemi tradizionali. L'architettura di X2S si basa su due elementi (figura 3.5):

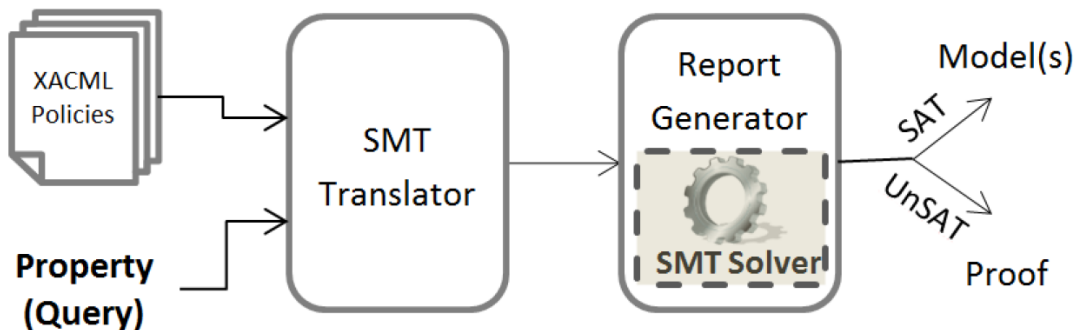


Figura 3.5. Architettura SMT (fonte: [103]).

- Il SMT Translator, che viene implementato direttamente all'interno del Policy Decision Point [104]: data una policy in ingresso, il suo obiettivo è quello

di estrapolare le informazioni importanti per la valutazione della stessa e di fornire in output le formule che codificano l'applicabilità di ogni regola.

- Il Report Generator, che fornendo i dati raccolti dall'SMT Translator al SMT Solver, è in grado di mostrare i risultati in un formato più user-friendly.

XACML with Graph Databases

All'inizio di questo paragrafo su XACML, abbiamo visto come il Policy Decision Point rappresenti un elemento fondamentale all'interno dell'intero processo decisionale di autorizzazione. Esso, tuttavia, può essere causa di un possibile collo di bottiglia nel momento in cui devono essere processate un numero ingente di policy contemporaneamente, dal momento che, richieste multiple al policy store e all'attribute store (vedi figura 3.2) rallenterebbero l'intero sistema. Una delle possibili soluzioni con il quale è possibile velocizzare l'accesso alle risorse, è quella di ridistribuire il modo in cui esse vengono memorizzate, ad esempio andando ad optare per un database a grafo⁶ come Neo4j, uno dei sistemi più performante presenti attualmente sul mercato [105]. Nell'articolo "Modeling XACML Security Policies Using Graph Databases" [106], viene presentata una possibile rappresentazione all'interno del quale gli archi descrivono l'interazione tra due elementi, mentre i nodi possono rappresentare una policy, una singola regola o un set dei due elementi.

U-XACML: XACML with Usage CONtrol

La modalità con il quale vengono valutate le policy, all'interno di un meccanismo di access control statico, non prevede la loro rivalutazione se viene rilevato un cambiamento del contesto in cui esse vengono applicate. Il modello di Usage CONtrol [107, 108] è in grado di estendere il sistema tradizionale attraverso un monitoraggio completo sull'uso delle risorse che permette, nel caso in cui le condizioni per un utente non risultino più soddisfatte, di interromperne immediatamente l'accesso. U-XACML [109] è un'estensione del linguaggio tradizionale, che viene definito appositamente per eseguire Usage CONtrol sulle policy.

History-Based U-XACML

È possibile introdurre un miglioramento sul modello U-XACML appena descritto, attraverso un controllo sulle policy in cui l'esecuzione di una determinata azione non dipende solo dall'azione stessa, ma anche da tutte quelle che sono state precedentemente eseguite dal sistema [110]. La tecnica è stata arricchita anche da "POLPA", un linguaggio algebrico, con il quale è possibile definire con più semplicità le policy in fase di scrittura, oltre che a fornire un tracciamento completo sulle azioni eseguite.

⁶Database che memorizzano i dati in base alla teoria dei grafi, usando nodi, proprietà e archi per rappresentare i dati.

MCACM: Access control in multi-clouds environment based on XACML

Parliamo ora di MCACM [111], un'implementazione basata su eXACML⁷ tramite il quale è possibile gestire il controllo degli accessi in un ambiente multi-server. Vediamone il funzionamento andando a considerare un utente che da un cloud A vuole accedere ad una risorsa contenuta in un cloud B (figura 3.6):

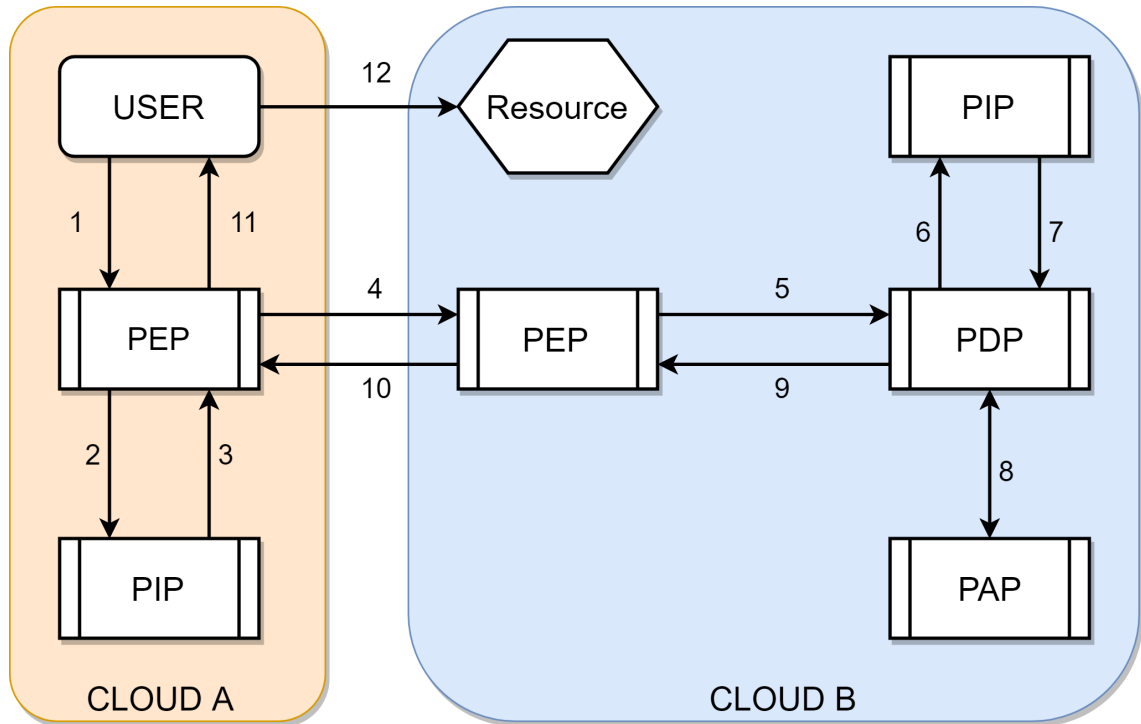


Figura 3.6. Architettura di una richiesta di autorizzazione tramite eXACML.

1. L'utente invia la richiesta al PEP locale, che a sua volta inoltrerà la query al PIP.
2. Il PEP locale, dopo avere raccolto le informazioni dal PIP, invia la query ed i relativi attributi al PEP del cloud B.
3. Il PDP del cloud B elabora la richiesta attraverso le informazioni ricavate da entrambi i cloud e fornisce una risposta al PEP del cloud A.
4. Il PEP inoltra la risposta all'utente.

⁷Estensione del linguaggio XACML, applicato in contesti multi-server.

Business Processing-XACML

OASIS propose per la prima volta nel 2005 [112] (revisionato poi nel 2014 [113]) un profilo XACML basato su RBAC che fosse in grado di elaborare policy di tipo role-based access control; il modello iniziale, tuttavia, non supportava la SoD⁸, permettendo in questo modo ad un singolo utente di poter danneggiare (intenzionalmente o accidentalmente) un determinato sistema. BP-XACML [114] supera questo limite, attraverso l'introduzione di nuovi elementi come il “performer list”, un nuovo policy set e un nuovo attributo tramite cui riconoscere le istanze di un attività. L'architettura di BP-XACML è stata progettata per essere compatibile con quella di RBAC-XACML (come mostrato in figura 3.7): questo significa che le policy RBAC possono essere gestite indipendentemente dal flusso con il quale si svolge invece il processo di autorizzazione.

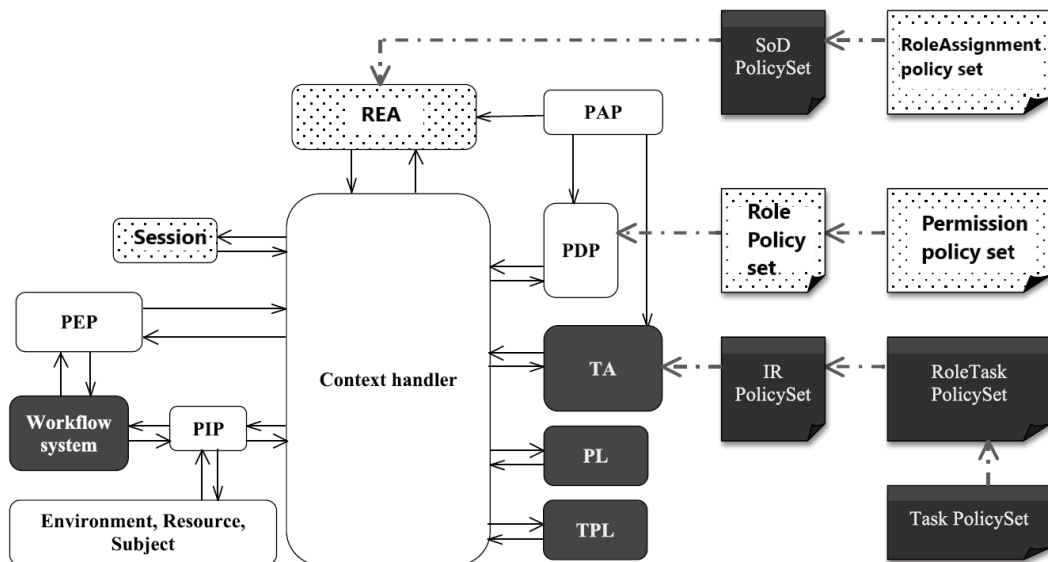


Figura 3.7. Architettura BP-XACML: gli elementi con sfondo bianco si riferiscono all'architettura classica di XACML, quelli con sfondo puntinato riguardano gli elementi aggiunti con RBAC-XACML ed infine i restanti vengono introdotti dal modello BP-XACML (fonte: [114]).

3.2.2 SAFAX

Parliamo infine, in quest'ultimo paragrafo, di un'implementazione di XACML molto usata nei servizi di clouding. Quest'ultimi infatti, seppur molto adoperati, possono presentare delle criticità nell'istante in cui più utenti vogliano condividere tra loro dei dati presenti su servizi differenti, dal momento che ognuno di essi implementa un meccanismo di controllo degli accessi proprietario, che ne rende in generale complicata la sincronizzazione.

⁸Separation Of Duties, caratteristica di un sistema in cui nessun singolo individuo ha la capacità di eseguire da solo una determinata attività.

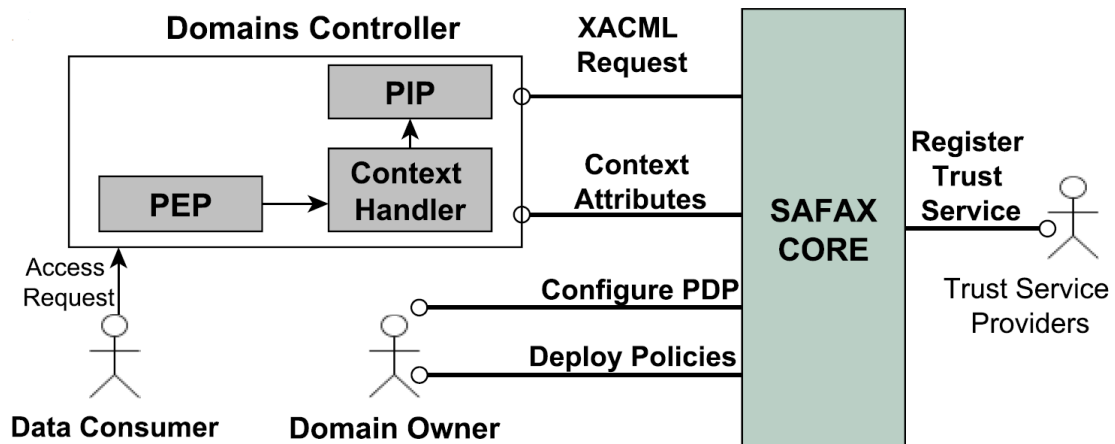


Figura 3.8. Architettura di SAFAX (fonte: [115]).

SAFAX [115] partendo da un'architettura XACML, si pone come obiettivo quello di garantire un'interazione tra servizi di clouding differenti, attraverso un meccanismo estremamente scalabile. La sua architettura (figura 3.8) risulta molto simile a quella di XACML, a parte il “deploy policy”, che permette all'utente proprietario del dominio di sviluppare le proprie politiche ed il “register service”, che è in grado di raccogliere le richieste di registrazione a SAFAX. Ma la grande differenza rispetto a XACML risiede sicuramente nel core (figura 3.9), che è composto da:

- Il Router, che si preoccupa di fare il forwarding delle varie richieste.
- Il Service Repository, che fornisce a tutti i meccanismi di clouding regolarmente iscritti, di interagire con altri servizi.
- SAFAX assegna un Policy Decision Point per ogni utente che viene identificato attraverso un URL univoco, detto PDP-URL. Il PDP gestirà a quel punto tutte le richieste di accesso ai dati dell'utente
- Il Policy Administrator Point, che facilita invece la gestione e il controllo delle policy definite dall'utente. Inoltre, quando l'utente aggiorna una policy, il PDA forza il PDP a rielaborare la nuova policy.
- SAFAX fornisce supporto per diverse implementazioni di XACML e, di conseguenza, di differenti PDP. Il PDP Configuration permette all'utente di selezionare il PDP più consono per il suo caso d'uso.

Vediamo a questo punto un esempio di come interagisce il sistema, nell'istante in cui viene inviata una richiesta di accesso ad una determinata risorsa. In particolare, supponiamo che un certo utente A voglia inviare una richiesta per accedere ai dati di un utente B, appartenente ad uno specifico DC; questa richiesta viene intercettata dal PEP del DC che la inoltrerà al context handler, il quale elaborerà una valida XACML request. A sua volta il CH, tramite l'ausilio del router, invierà la richiesta al PDP che, basandosi sul PDP-URL univoco, inizierà la fase di valutazione della

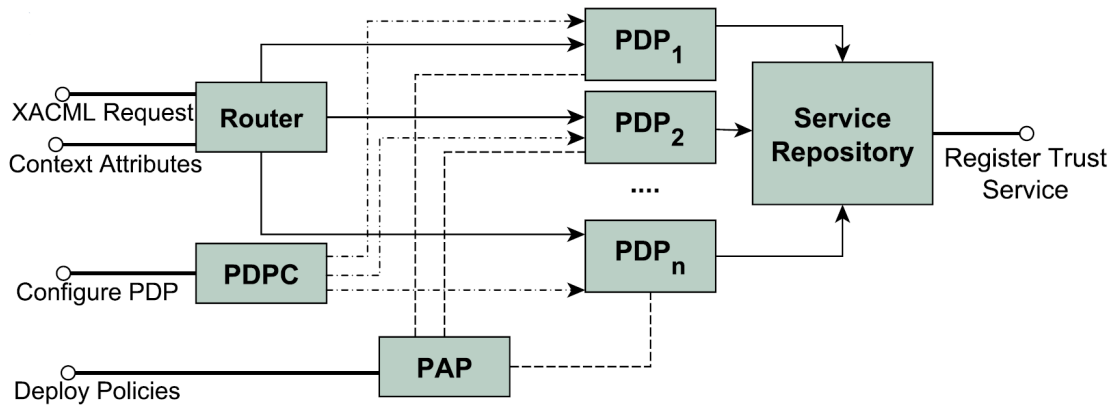


Figura 3.9. Architettura del core di SAFAX (fonte: [115]).

policy. Durante questa fase decisionale, il PDP potrebbe avere bisogno di consultare uno o più trust services definiti all'interno della policy valutata. Successivamente, il PDP elabora una risposta che viene inoltrata al context handler il quale avrà il compito di definire la XACML response finale. Essa viene prima inviata al PEP, ed infine propagata verso l'utente richiedente.

In conclusione, SAFAX realizza un'implementazione più efficiente dei singoli componenti XACML, migliorando in questo modo sia l'usabilità per gli utenti che devono condividere i propri dati e sia per i provider, i quali non dovranno realizzare più metodi ad hoc per il controllo degli accessi.

3.3 Open Authorization

OAuth 1.0 [116] è un sistema di autorizzazione basato sull'uso di token che ha il compito di monitorare l'accesso a risorse e dati sensibili. L'architettura di OAuth è composta da tre parti (figura 3.10):

- Il client, che richiede le informazioni (associabile all'applicazione dell'utente).
- Il server, che possiede la risorsa (ad esempio Facebook).
- L'utente dell'applicazione, iscritto al servizio fornito dal server.

In una prima fase è necessario che l'utente registri l'applicazione al server, il quale fornirà al client una coppia di chiavi segrete (consumer key e consumer secret), necessarie durante la fase di autorizzazione. Nell'istante in cui l'utente vuole accedere ad una risorsa del server, il client lo reindirizza ad un URL fornitogli dal server stesso, il cui codice sorgente contiene le chiavi segrete definite in fase di registrazione e un URL secondario. OAuth non impone nessun vincolo su come implementare il sistema di autenticazione, la cui scelta rimane vincolata al proprietario della risorsa. Dopo che l'utente si è autenticato e che il server ha verificato e approvato la sua identità, verrà fatta una redirect tramite l'URL fornito precedentemente al client;

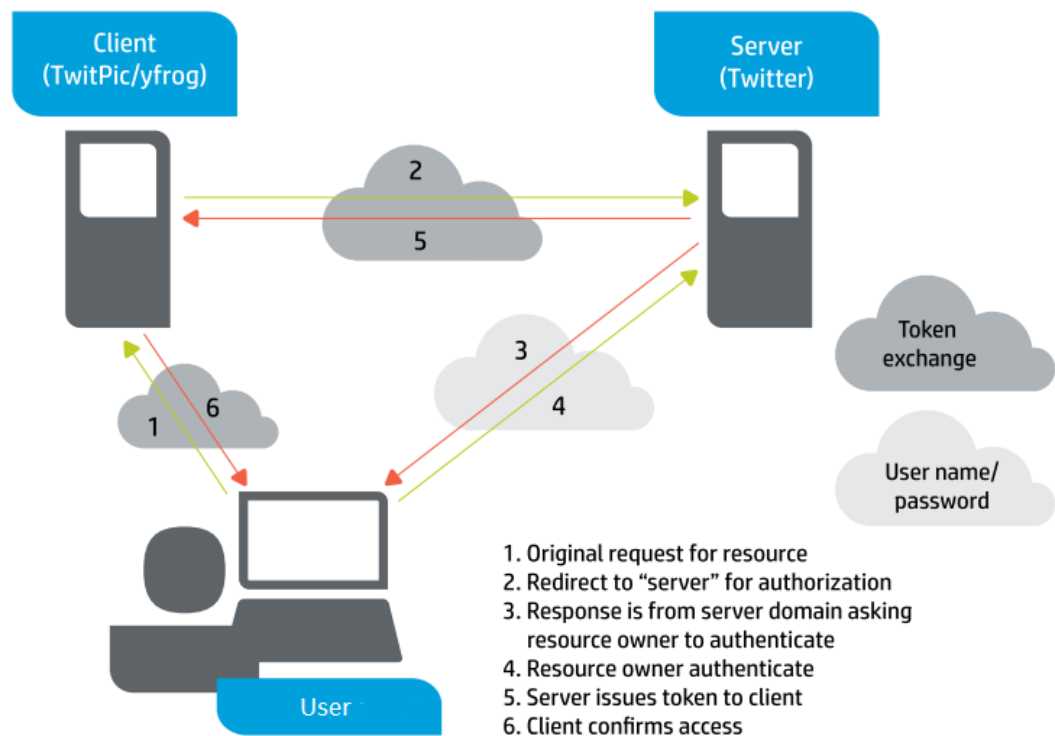


Figura 3.10. Modello di Autorizzazione tramite OAuth (fonte: [Hewlett Packard Enterprise](#)).

l'URL contiene al suo interno una stringa fondamentale, che rappresenta il token tramite il quale il client potrà inoltrare successivamente le richieste dell'utente al server.

La progettazione del client determina l'usabilità del sistema, che raggiungerà il vertice massimo nell'istante in cui le varie redirect che intercorrono tra il server e il client rimarranno totalmente trasparenti all'utente finale. OAuth non è però esente da problemi di sicurezza causati in parte da falle presenti a livello di progettazione, e in parte derivanti da un uso superficiale del protocollo [117]. Il problema principale riguarda la memorizzazione e la trasmissione delle chiavi segrete consumer key, consumer secret e del token. Questi segreti vengono memorizzati all'interno del server e devono essere formati da stringhe ragionevolmente complesse, in modo tale da evitare brute-force attack. Inoltre è necessario implementare un meccanismo di sicurezza, come ad esempio la cifratura, per non permettere a nessuno di conoscere le credenziali dell'utente in caso di un attacco fisico diretto ai danni del database del server. Invece per quanto riguarda la trasmissione è consigliabile utilizzare un canale TLS/SSL, che permette di evitare intercettazioni e attacchi MITM durante lo scambio dei dati, mentre i server possono limitare replay attack tramite l'utilizzo degli attributi nonce e timestamp forniti dal protocollo, il cui compito è rispettivamente quello di firmare la richiesta del client e definire il tempo di conservazione del nonce. Infine il protocollo non supporta la mutua autenticazione, rendendo in questo modo il client esposto ad attacchi di phishing, in cui l'utente può essere ridiretto verso server malevoli.

Nell'ottobre del 2012 è stata presentata una seconda versione del protocollo [118], in cui viene aggiunto un ruolo di mediatore che intercorre tra il client ed il server. Nonostante questo, il funzionamento generale e di conseguenza le problematiche presenti in OAuth 2.0, sono le medesime di quelle della versione precedente. I due protocolli sono inoltre incompatibili fra di loro.

OAuth, nonostante i suoi limiti e i suoi difetti, rappresenta tuttora il sistema di autorizzazione più utilizzato all'interno della rete, ed implementato da aziende come Google, Facebook e Yahoo!.

3.4 Gestione e creazioni delle policy in Vault

Continuiamo il discorso iniziato su Vault nel secondo capitolo (vedi paragrafo 2.12.1), andando a vedere come questo sistema gestisce il meccanismo di autorizzazione [119].

In Vault viene usato il principio RBAC, tramite il quale è possibile definire i privilegi che gli utenti hanno su una determinata risorsa, identificata in questo caso da un path. Di default la policy è di tipo deny⁹, mentre la sintassi che viene utilizzata per descriverla, può essere JSON o seguire un formato proprietario di Hashicorp [79], chiamato HCL [120]. Nell'esempio:

```
path "secret/foo" {  
  capabilities = ["read"]  
}
```

viene associato il diritto di lettura ad una risorsa identificata dal path “secret/-foo”.

Esistono due tipi di policy:

- La default policy, che viene collegata automaticamente alla creazione di un token e che descrive alcune funzionalità base necessarie per il corretto funzionamento del sistema. Essa non può essere eliminata direttamente dall'utente, ma può essere modificata ed adattata ai diversi usi.
- La root policy, che non può essere eliminata o modificata. Tutti gli utenti associati ad una root policy diventano utenti root e, in quanto tale, possono esercitare qualsiasi diritto all'interno di Vault.

3.5 Conclusione

L'analisi fatta in questo capitolo riguarda prevalentemente la gestione del controllo sugli accessi, dal momento che essa rappresenta un forte punto d'interazione tra

⁹Questo significa che se viene creata una policy vuota o una in cui è presente un conflitto, essa esercita sempre un divieto e mai un permesso.

sistema ed utente. Tutti gli altri aspetti sono stati solo presentati e non approfonditi nel dettaglio (come accade invece per i sistemi di autenticazione), proprio perché essi appartengono a dei meccanismi più interni, difficilmente manipolabili dall'utente e, quindi, valutabili in termini di usabilità.

Capitolo 4

Usabilità

Dopo avere analizzato nei due capitoli precedenti i sistemi di autenticazione e autorizzazione da un punto di vista tecnico, andremo ora a visionare qual è il loro rapporto rispetto all'usabilità percepita dall'utente. Per quanto riguarda i sistemi di autenticazione, lo studio viene fatto prima sulla categoria del sistema e poi sui singoli meccanismi che la compongono. Per i sistemi di autorizzazione vedremo invece come l'utente utilizzi le policy e le access control list per proteggere dei dati sensibili.

4.1 Usabilità nelle Textual Password

Abbiamo visto nel capitolo precedente come le password ripetibili, seppur rappresentino il mezzo di autenticazione più usato dalla massa, abbiano dei grossi problemi a livello di usabilità, oltre che di sicurezza. Nemmeno l'utilizzo di password complesse, così come descritto nel documento “800-63, Appendix A” (edizione originale di uno standard tutt'ora in evoluzione) rilasciato nel 2003 dal National Institute of Standards and Technology, ha avuto il risultato sperato e a dirlo è lo stesso creatore della guida, Bill Burr. Le linee fornite dall'articolo (implementate ampiamente nel panorama moderno) obbligavano l'inserimento di almeno un simbolo e un numero all'interno della password, portando così l'utente a creare stringhe che erano sicuramente più complesse rispetto a quelle tradizionali, ma anche più difficili da ricordare e quindi meno usabili (vedi figura 4.1). Utilizzare delle politiche di restrizione, tuttavia, non significa necessariamente limitare l'usabilità: ad esempio costringere l'utente ad inserire come password un insieme di parole di senso compiuto (passphrase), significa sia aumentare il livello di sicurezza, rendendo in questo modo molto più difficile che si verifichino brute-force attack, che facilitarne l'apprendimento.

La trasposizione di questo meccanismo su mobile non ha avuto un esito positivo, con tempi di inserimento fino al 20% più lunghi ed una probabilità doppia di commettere errori [121], rispetto al sistema tradizionale. Neanche il fatto di poter utilizzare tecniche di inserimento alternative come lo Swype, la compilazione automatica delle parole e la visibilità parziale dell'ultimo carattere inserito, hanno generato un miglioramento sull'usabilità, che risulta essere sempre bassa ed associata ad una user experience deludente e frustrante.

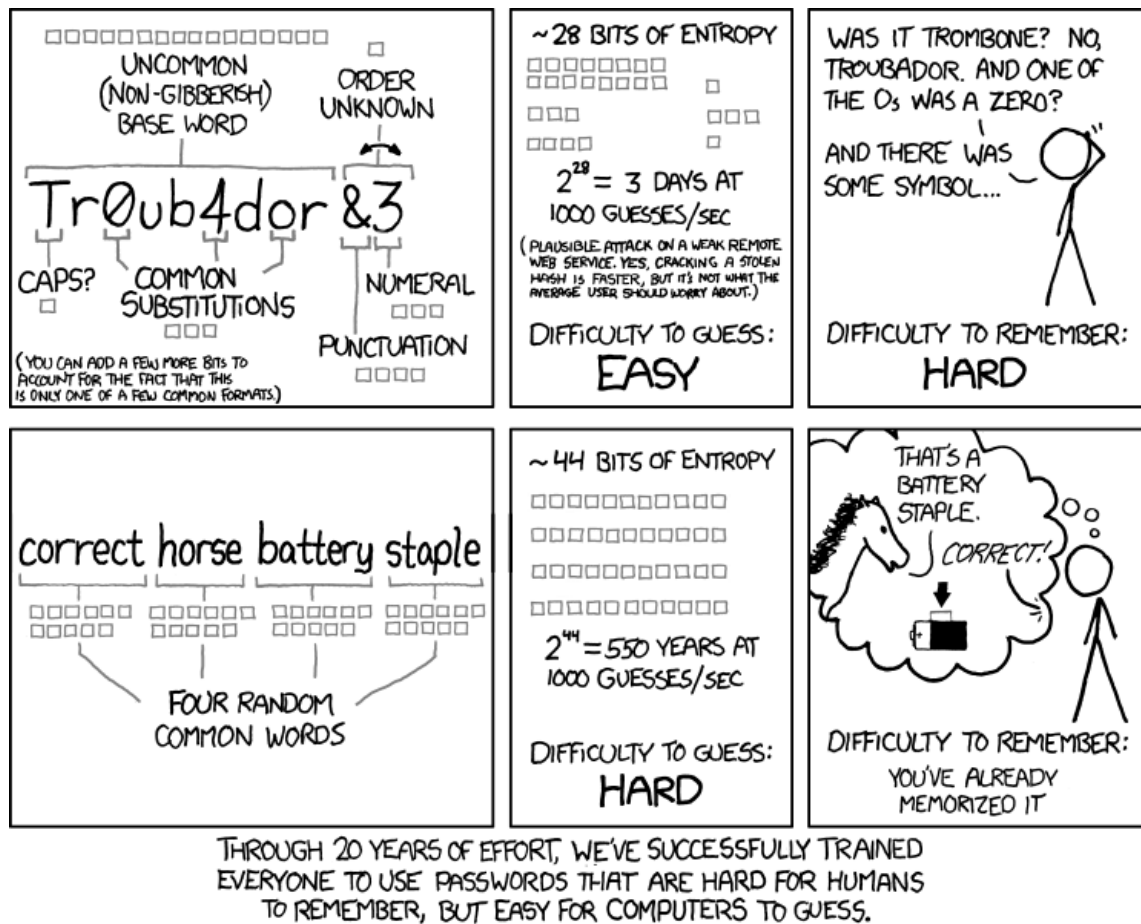


Figura 4.1. di Randall Munroe per il suo blog XKCD (fonte: [xkcd](http://xkcd.com)).

Andremo ora ad analizzare nel dettaglio l'impatto che i sistemi password-based (visti nel capitolo precedente) hanno sull'usabilità dell'utente finale.

Analisi delle tecniche

Abbiamo visto nel secondo capitolo (paragrafo 2.2.2) come OTP permetta di risolvere una serie di rischi legati alla sicurezza. D'altra parte, in termini di usabilità, si vanno a perdere una serie di vantaggi che invece sono presenti nelle password ripetibili. Come prima cosa non si ha più l'immediatezza legata al principio di "knowledge" presente nel meccanismo classico, dal momento che l'utente dovrà avere a sua disposizione un elenco di password non personali; è inoltre impensabile che l'utente provi a memorizzarle tutte in quanto spesso le stringhe vengono generate dal server in modo pseudo-casuale (oltre al fatto che la lunghezza dell'elenco può contenere anche decine di password). Le OTP, infine, possono risultare frustranti nell'istante in cui, per essere utilizzate, portino l'utente a dover interrompere il proprio flusso operativo; questo è necessario in quanto il server ha bisogno di un meccanismo OOB per trasmettere le chiavi al client.

Supponiamo ad esempio che un utente stia per concludere una transazione con la propria carta di credito, ma che prima di autorizzare il pagamento, la banca voglia

come ulteriore prova l’inserimento da parte del cliente di una OTP inviata tramite SMS. Supponendo che tutti i passaggi debbano necessariamente essere effettuati attraverso lo stesso dispositivo, l’utente dovrà:

1. Uscire temporaneamente dal sito di acquisti.
2. Andare sull’app di messaggistica.
3. Ricordarsi la stringa di caratteri (che potrebbe non essere memorizzata in un solo colpo dall’utente, richiedendo in questo caso la riesecuzione di tutti i passaggi).
4. Ritornare sul sito di acquisti (sperando che la transazione sia stata salvata in background sulla RAM).
5. Sottomettere la OTP.

Risulta evidente come questa serie di passaggi possa risultare poco usabile per l’utente, soprattutto per quelli meno pratici. Tuttavia negli ultimi anni sono state introdotte delle tecniche al fine di migliorare l’esperienza d’uso del meccanismo OTP, come ad esempio la possibilità di utilizzare il copia/incolla, evitando che in questo modo l’utente debba memorizzarsi una stringa, oppure tramite la presenza di finestre pop-up che appaiono durante la ricezione degli SMS, non richiedendo così uno swipe tra diversi applicativi. Esistono infine altre tecniche come quella proposta da Ravi Surana [122], in cui il dispositivo riconosce automaticamente l’arrivo di una OTP e, con un semplice tocco, la inserisce direttamente nel form opportuno.

Il sistema TOTP ha le stesse caratteristiche di usabilità di OTP, non presentando però la problematica appena discussa, grazie al fatto che la password viene comunicata su un token esterno al dispositivo con il quale ci si autentica. Un fattore che potrebbe però determinare dei problemi di usabilità, oltre che di sicurezza, è la finestra temporale all’interno del quale una password è considerata valida: intervalli troppo lunghi potrebbe favorire la presenza di attacchi mentre, periodi troppi brevi, aumentano la possibilità di errori, che possono scaturire durante la digitazione o dal fatto che la stringa è stata inserita al di fuori della validità temporale. In generale l’RFC 6238 [23], consiglia di utilizzare un intervallo temporale pari a 30 secondi, in modo tale da rendere proporzionale il rapporto che sussiste tra usabilità e sicurezza.

Infine, l’usabilità di S/KEY può essere comparata a quella delle password ripetibili, dal momento che l’utente usa un segreto condiviso con il server, per generare di volta in volta i relativi codici.

4.2 Usabilità nelle Graphical Password

I sistemi basati su password testuali hanno nel complesso una bassa usabilità se usati per fornire un alto livello di sicurezza nel sistema di autenticazione. Il motivo principale per cui accade tutto questo è legato alla password che nel complesso richiede un elevato sforzo cognitivo e flessibilità nel memorizzare stringhe alfanumeriche sempre

nuove. D'altra parte la "Picture Superiority Effect Theory" [123] indica proprio come l'essere umano sia molto più incline a memorizzare contenuti visivi, come ad esempio le immagini e i video piuttosto che stringhe formate da numeri e caratteri; basti pensare che la maggior parte delle tecniche di memorizzazione si basano proprio sull'associare parole e numeri a delle figure, piuttosto che a dei caratteri. In termini generali è quindi possibile sostenere che l'usabilità all'interno di sistemi di questo tipo sia elevata, anche se, come andremo a vedere nel dettaglio ora, per ogni tecnica possono subentrare fattori che vanno ad inficiare negativamente sulla valutazione finale.

Per capire meglio quale possa essere la percezione e l'usabilità di un sistema che fa uso di queste tecniche, andiamo a vedere come prima cosa un caso pratico, per poi concentrarci sull'analisi delle tecniche grafiche viste nel secondo capitolo.

Caso 1

Nell'esperimento condotto da Mohd Anwar e Ashiq Imran [124], l'obiettivo è quello di verificare quali fossero le esperienze d'uso di 33 utenti nel dover utilizzare, come sistema di autenticazione sul proprio dispositivo, un meccanismo basato su PIN piuttosto che uno basato sul pattern lock¹; la prova è stata fatta sia attraverso l'uso di uno smartphone che di un tablet, per vedere come la dimensione dello schermo possa influire sull'usabilità dell'utente. Vediamo ora i vari risultati che sono emersi da questa prova:

- Il tempo di login è un fattore fondamentale per l'analisi dell'usabilità finale: tempi di accesso troppo lunghi potrebbero infatti portare l'utente a disabilitare il meccanismo all'interno del proprio device. Durante la comparazione, non è stata rilevata una differenza di accesso così elevata da compromettere l'esperienza utente. Il sistema che richiede meno tempo è il pattern lock su smartphone, mentre quello che ne richiede di più risulta essere il PIN su tablet, dove, a causa dell'ingrandimento dei tasti sulla superficie dello schermo, l'utente è costretto a coprire con il proprio dito distanze maggiori.
- La differenza tra i tassi di login positivi effettuati tra i due sistemi è simile, ed è formato dall'88% del PIN e dall'83% del pattern lock.
- In termini di facilità di memorizzazione, il PIN viene ricordato solo dal 39% dei partecipanti, mentre la password grafica ottiene un 56%, confermando in questo modo la teoria di Curran [123].
- Un fattore che invece ha determinato una grossa differenza sull'usabilità finale è la grandezza dello schermo: I partecipanti non hanno infatti gradito la difficoltà riscontrata nell'inserimento del PIN sul tablet rispetto che sullo smartphone. Al contrario, invece, non sono state riscontrate particolari problematiche nell'introduzione del pattern lock, rilevando al contrario una preferenza sul dispositivo con schermo più ampio.

¹Sistema di autenticazione grafico, molto usato sui dispositivi android, in cui l'utente accede tracciando una sequenza di segmenti all'interno di uno schema puntato.

- Infine in termini di vulnerabilità ad attacchi, essa dipende molto dal tipo di sicurezza implementata dall'utente. In generale il 75% dei partecipanti ha deciso di utilizzare un pattern lock di complessità e lunghezza adeguata, e l'83% un PIN formato da sequenze non prevedibili. Rispetto al modo in cui gli utenti si autenticavano, il 43% era potenzialmente soggetto a shoulder surfing attack durante l'inserimento del PIN, mentre il 50% da smudge attack per il pattern lock (questo tipo di autenticazione è basato sul trascinamento del dito sullo schermo, e di conseguenza è molto più sensibili a questo tipo di attacchi rispetto al PIN). Per quanto riguarda i brute-force attack, esistono 10000 combinazioni differenti ricreabili tramite l'inserimento del PIN e 389112 per il pattern lock (se viene considerato l'inserimento di una sequenza complessa formata da 9 punti), rendendo quindi, da questo punto di vista, più sicuro il secondo metodo. Come ultima analisi, si può infine notare come gli utenti preferiscano autenticarsi utilizzando sequenze più complesse, preferendo quindi avere sistemi più sicuri ma meno usabili.

Analisi delle tecniche

Il sistema grafico più semplice, analizzato nel capitolo precedente, è sicuramente la tecnica basata sul Blonder Scheme [25]. L'usabilità di questo meccanismo è variabile poiché dipende fortemente dal grado di precisione che viene utilizzato per il riconoscimento del punto selezionato: se l'area di precisione è elevata l'utente avrà meno difficoltà nel digitare la sequenza corretta ma sarà più semplice effettuare dictionary o brute-force attack, mentre nel caso contrario avremo una sicurezza maggiore a discapito di una bassa usabilità per l'utente finale che potrà riscontrare delle difficoltà nell'inserire la giusta sequenza di input. Questo tipo di problematica non è presente solo in questo sistema, ma anche in tutti quelli in cui, per autenticarsi, è necessario selezionare un'area precisa dello schermo [26]. Nella tecnica di Persuasive Cued-Click Points, inoltre, il fatto di persuadere l'utente a selezionare aree che per lui potrebbero non essere semplici da memorizzare, hanno un esito potenzialmente negativo sulla sua usabilità.

I sistemi che non evidenziano problematiche di shoulder surfing, invece, superano questo tipo di criticità, grazie all'abbandono di sistemi di autenticazione che si basano sul tocco di punti predefiniti. Nonostante questo, l'introduzione di nuovi metodi di riconoscimento ha portato inevitabilmente ad altre problematiche che, in alcuni casi, possono compromettere l'usabilità degli utenti. Prendiamo ad esempio in analisi PassMatrix: secondo lo studio condotto da Sun [27], su 30 utenti che non hanno familiarità con l'uso di schemi grafici, il tempo medio necessario per registrarsi a questo sistema è di 90.5 secondi. Per quanto riguarda il tempo di autenticazione, si parte da una media di 47 secondi per la prima, fino ad arrivare a 16.5 secondi dopo che l'utente ha fatto un po' di pratica con il meccanismo. In generale queste rimangono comunque delle stime, in quanto il tempo è fortemente condizionato dalla praticità che l'utente ha nel gestire dispositivi che fanno uso del touchscreen. In ogni caso oltre tre quarti dei partecipanti all'esperimento hanno ritenuto il tempo richiesto per autenticarsi accettabile, rendendo, in definitiva, il meccanismo in questione abbastanza usabile dalla massa.

Per il sistema di autenticazione presentato da Danish [28], non esistono dei dati concreti che siano in grado di fornire un riscontro positivo o negativo riguardo la sua usabilità.

Infine, per quanto riguarda IBRAS [29], si possono rilevare due situazioni contrastanti. Da una parte abbiamo un sistema fortemente usabile, grazie all'uso delle immagini come password, mentre dall'altra parte abbiamo un sistema poco utilizzabile a causa dell'immagine stessa, che sarà sempre legata al dispositivo utilizzato e mai all'utente che lo utilizza, come invece accade con la password alfanumerica. Quest'ultimo aspetto infatti può generare delle problematiche: se il sistema di autenticazione viene usufruito da più dispositivi personali legati all'utente, basterà infatti trasferire le immagini su di essi, ma nel caso in cui l'utente voglia autenticarsi tramite dispositivi a lui esterni, non potrà farlo tramite autenticazione basata su IBRAS, perché non potrà disporre dell'immagine necessaria ad autenticarsi. Questo rappresenta un limite in termini di usabilità, in quanto in quei casi l'utente sarà costretto a portarsi l'immagine sempre con sé o ad utilizzare meccanismi di archiviazione di cloud computing.

4.3 Usabilità in una continuous authentication

A causa dell'elevato numero di fattori presente in questo tipo di analisi, può risultare complicato esaminarne l'usabilità. Per quanto riguarda la precisione con il quale vengono esaminati i dati raccolti, in un contesto simile, è consigliabile favorire una False Rejection Rate anziché una False Acceptance Rate. Infatti, se da una parte alzare i tassi di FAR renderebbe il sistema meno sicuro, bisogna considerare anche che valori troppo alti di FRR creerebbero frustrazione nell'utente legittimo. Inoltre, dal momento che la continuous authentication non rappresenta il sistema di autenticazione primaria, ma solo un mezzo con il quale è possibile monitorare costantemente l'operato dell'utente, risulta nel complesso più sensato garantire una maggiore usabilità a discapito della sicurezza, evitando in questo modo che l'utente venga interrotto troppe volte durante il suo processo produttivo. Tutto ciò deve quindi trasformarsi nella realizzazione di un sistema di sicurezza che sia facile da usare e semplice da comprendere per gli utenti, indipendentemente dalle loro capacità e dai loro background [125]. Per capire quale sia la reazione degli utenti in merito a questa tipologia di autenticazione, andiamo ad analizzare alcuni casi pratici.

Caso 1

Nel primo caso di studio [126], condotto su 37 utenti, viene valutata prima l'usabilità da un punto di vista empirico per poi cercare, in un secondo momento, di comprendere quale possa essere la percezione degli utenti nei confronti di questa tecnologia, anche in relazione al meccanismo di autenticazione classico. In termini assolutistici, l'usabilità percepita nei sistemi di CA (che in questo caso viene rilevata in base allo studio dei tocchi riprodotti dall'utente sullo schermo) è ottima, con il 91% dei partecipanti che lo considera persino migliore del meccanismo tradizionale.

Tuttavia il 35% di essi trova frustrante i falsi rifiuti che possono scaturire a causa delle incertezze di cui soffre il sistema (come evidenziato nel secondo capitolo). Per quanto riguarda invece la sicurezza percepita dagli utenti, l'82% risulta soddisfatto della CA preferendola, in alcuni casi (30%), al meccanismo di autenticazione primario.

In conclusione si può affermare che la maggior parte dei partecipanti sia rimasto soddisfatto della CA e questo rappresenta un'ulteriore conferma su come questo sistema possa rappresentare un ragionevole compromesso tra sicurezza ed usabilità all'interno del mondo dei sistemi di autenticazione.

Caso 2

Abbiamo visto come la richiesta continua di autenticazione da parte del sistema può risultare frustrante nell'esperienza d'uso finale; essa tuttavia rappresenta una pratica necessaria, soprattutto nella fase iniziale, per addestrare il dispositivo a riconoscere il proprio utilizzatore. Questo principio viene sfruttato nel modello di Alert proposto da Nathan Clarke [127], in cui vengono presentati 6 metodi incrementali di sicurezza per monitorare la CA:

1. Usando i dati recenti presenti in cache.
2. Usando i dati rimanenti in cache.
3. Usando i dati successivi immessi dall'utente.
4. Introducendo un metodo di autenticazione intrusivo, ma altamente affidabile.
5. Richiedendo un ulteriore metodo di autenticazione intrusivo.
6. L'errore di autenticazione successivo richiama un blocco di sistema.

Come si può evincere dallo schema, esso fornisce tre possibilità non invasive, in modo tale da diminuire il più possibile la presenza di falsi negativi e i relativi disagi per l'utente. Se invece per tre volte consecutive il sistema non riconosce l'utente, allora verranno utilizzati due metodi di autenticazione primaria che, se falliscono, porteranno al blocco definitivo del dispositivo. Se da una parte l'estensione della finestra degli errori genera un meccanismo più usabile, dall'altra porta ad un indebolimento della sicurezza globale, in quanto un malintenzionato dovrebbe commettere 5 errori consecutivi prima di venire definitivamente bloccato, permettendogli, in quell'intervallo di tempo, di poter intaccare come vuole il sistema. Questo rappresenta un compromesso tuttavia accettabile dal momento che esiste a priori un sistema di autenticazione primaria in grado di ridurre questa problematica.

Questa tecnica è stata introdotta all'interno di un caso d'uso che ha coinvolto 27 partecipanti: come prima cosa essi hanno dovuto registrare il proprio volto, la voce e le caratteristiche comportamentali di scrittura, in modo tale che potessero essere riconosciuti nel meccanismo secondario di CA. Il risultato finale è ottimo, in quanto il 92% degli utenti si sono ritenuti soddisfatti del sistema proposto, considerandolo persino più sicuro del sistema di autenticazione primaria. Anche il sistema di Alert

a scala è stato valutato positivamente dall'utenza. Per quanto riguarda invece le tecniche utilizzate per la CA, c'è stata una preferenza per la voce e keystroke dynamics rispetto al volto, a causa delle incertezze di valutazione che possono scaturire nel momento in cui l'utente non si trova perfettamente in linea con il sensore fotografico del dispositivo.

Caso 3

L'ultimo studio che prenderemo in considerazione [128], verte su uno schema di continuous authentication basato, come mostrato in figura 4.2, su tre livelli, "None", "Some" e "All", differenziati in base alla conoscenza che il dispositivo ha del proprio utilizzatore. Questo permette di fare un'analisi non solo sull'usabilità e l'esperienza dell'utente, ma anche sulla qualità del sistema di riconoscere delle possibili intrusioni. L'esperimento viene fatto su tre scenari, ognuno con una difficoltà ed una serie di obiettivi differenti: i 30 partecipanti, assegnati in modo casuale ad una delle tre possibilità, dovranno cercare di risolvere i tasks relativi.

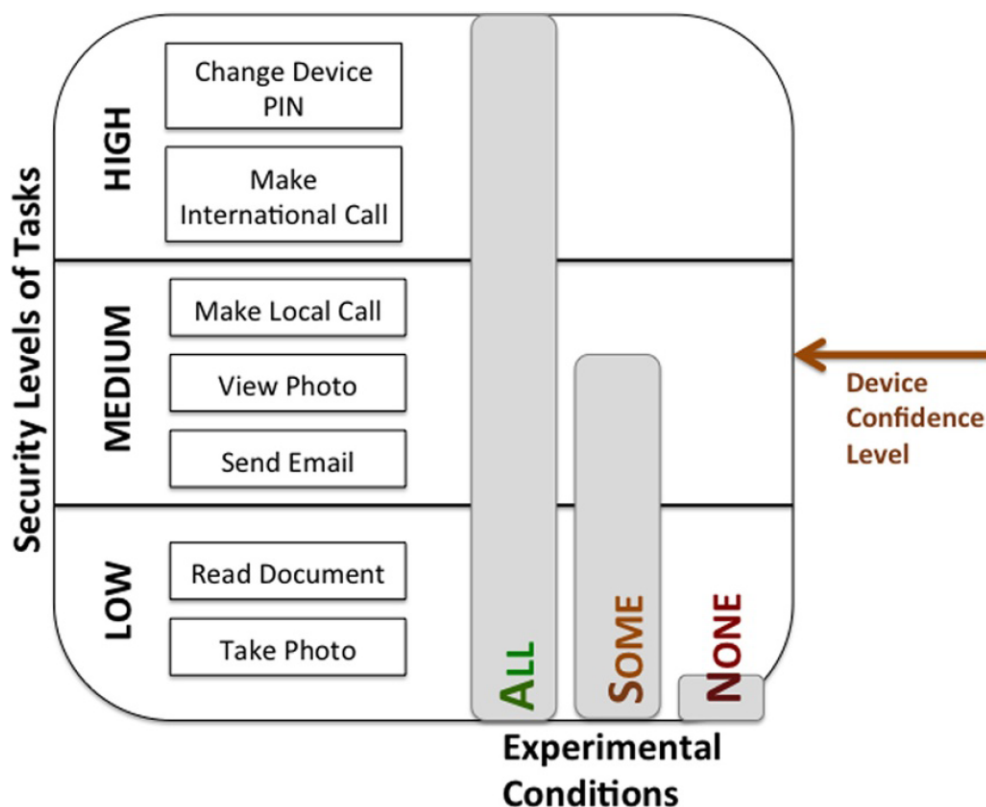


Figura 4.2. Interazione tra tasks, i relativi livelli di sicurezza, il livello di confidenza del device e il livello di confidenza del meccanismo di autenticazione (fonte: [128]).

Dal questionario posto prima dell'esperienza viene rilevato che il 73% degli utenti percepiscono la CA come un sistema di autenticazione più sicuro rispetto ai metodi tradizionali, seppure la maggioranza di essi hanno manifestato l'interesse di voler testare il sistema prima di poter dare una risposta definitiva. Questo è un comportamento comune nella psiche umana, associata alla paura dello sconosciuto [129],

che in questo caso si riflette su un meccanismo di cui l'utente non ha un controllo ed un'interazione diretta, come invece accade con i meccanismi di autenticazione tradizionali. Nonostante questo, dopo che i partecipanti hanno potuto provare direttamente il meccanismo di CA, il 90% di loro lo ha ritenuto valido, sostenendo di volerlo introdurre all'interno dei propri dispositivi personali.

Analisi delle tecniche

Analizziamo ora l'usabilità delle tecniche trattate nel capitolo precedente, utilizzabili in un meccanismo di continuous authentication.

Parametri fisici

Per quanto riguarda le tecniche che si basano su elementi fisici dell'utente, sistemi incentrati sull'iride e sul volto presentano una bassa usabilità, derivata dal fatto che il rilevamento delle loro caratteristiche avviene su coordinate spaziali e non attraverso un contatto fisico col sensore. Questo si traduce in frustrazione nell'utente che deve ogni volta posizionare il dispositivo in un'angolazione specifica affinché la rivelazione avvenga nel modo corretto. Un altro fattore che influenza negativamente queste due tecniche riguarda l'ambiente circostante, il quale se troppo buio o luminoso, potrebbe non garantire un esatto rilevamento.

L'utilizzo dello sguardo come sistema biometrico presenta di per sé un'alta usabilità [35], nonostante i tempi medi di autenticazione risultino ancora troppo elevati [36]: questo porta a sconsigliare l'implementazione di questo meccanismo in un sistema di autenticazione primaria, mentre il suo utilizzo in una CA potrebbe non creare problemi per l'esperienza d'uso dell'utente, anche se rimangono presenti le difficoltà evidenziate nei due sistemi precedenti.

La voce, invece, è una caratteristica che pur non richiedendo un contatto fisico tra utente e sensore, non necessita che il dispositivo sia posto in determinate angolazioni, grazie alla capacità panoramica dei microfoni presenti su quest'ultimi. La luminosità bassa o alta dell'ambiente non crea problemi, ma spazi rumorosi potrebbero rendere difficile il rilevamento della voce così come situazioni in cui siamo impossibilitati a parlare, renderebbero inutilizzabile il meccanismo.

L'impronta digitale rimane la soluzione biometrica più implementata in assoluto sugli smartphone. Il fatto che debba esserci un contatto diretto tra sensore e dito e che l'analisi venga fatta attraverso un modello bidimensionale, rende il meccanismo reattivo ed intuitivo da utilizzare [128, 130]. L'unica limitazione può scaturire nell'istante in cui il dito per il riconoscimento sia per qualche motivo inutilizzabile (perché magari momentaneamente impegnato o presenta dei tagli temporanei), ma in quel caso basta riconfigurare un'altra impronta, se non è già stato fatto in fase di registrazione. Il numero di utenti che per motivi fisici non potrà mai utilizzare questo sistema di autenticazione è basso [33] ed è anche questo il motivo per il quale viene implementato in un numero di dispositivi sempre crescente.

Parametri Comportamentali

Come abbiamo visto nei casi d'uso descritti in precedenza, il *keystroke dynamics* rappresenta la tecnica più utilizzata per testare la sicurezza e l'usabilità della CA. Questa tecnica risulta estremamente usabile ed apprezzata dagli utenti [126, 127], a patto che vengano analizzati un numero di parametri sufficienti da rendere affidabile e preciso il sistema [41]. D'altra parte però, lo stesso discorso non può essere fatto all'interno del panorama mobile, dove il numero elevato di elementi in gioco può compromettere la stabilità del meccanismo [42], con una possibilità più elevata di FRR ed un conseguente abbassamento dell'usabilità.

Il sistema HMM, basato sui tocchi riprodotti sul touchscreen, si attesta su ottimi risultati: in particolare, se impostato un livello molto alto di sicurezza, FAR = 0, l'usabilità del modello raggiunge picchi piuttosto alti, con valori del FRR inferiori al 2% [44].

Anche per quanto riguarda l'autenticazione basata sul pattern con il quale vengono visitati siti web e app [51, 52, 53, 54], il livello di usabilità rimane ottimo, grazie alla completa trasparenza del sistema agli occhi dell'utente. L'unico motivo di frustrazione potrebbe derivare da un modello poco preciso, che abbiamo visto essere superabile grazie a tecniche come MSIA e SPIA [53].

Infine, il sistema presentato da MoCRA [55] presenta un'usabilità variabile, determinata dalla lunghezza del sample che l'utente deve ricomporre in aria per autenticarsi: stringhe troppo lunghe da ricreare potrebbero aumentare l'affidabilità del meccanismo, ma allo stesso tempo ridurre l'usabilità dell'utente, a causa di un tempo di autenticazione troppo lungo. Inoltre, scrivere parole in aria non rappresenta un'attività comune per l'utente e questo si trasforma nella perdita della trasparenza tipica della *continuous authentication*.

4.4 Usabilità negli altri sistemi di autenticazione

In questa categoria non ci sono purtroppo delle caratteristiche comuni tra i vari sistemi, in quanto essi adottano spesso tecniche molto differenti tra loro per verificare l'autenticità dell'utente. Andremo quindi a vedere direttamente quale sia l'usabilità percepita dai vari utenti, per ognuna delle singole tecniche analizzate nel capitolo precedente.

Analisi delle tecniche

Partiamo con l'unione di due meccanismi di cui abbiamo già discusso, il *keystroke dynamics* e le *Graphical Password* [65]. In termini di precisione, l'introduzione di un parametro biometrico introduce inevitabilmente un errore, che potrebbe influire negativamente sull'usabilità del meccanismo finale. D'altra parte il fatto di dare all'utente il libero arbitrio sulla scelta delle aree da selezionare, può essere positivo in termini di usabilità, se usato in modo appropriato.

Per quanto riguarda l'autenticazione *fourth-factor* [66], essa risulta nel complesso poco usabile a causa dei limiti che vengono posti, ma che sono necessari per

fornire un’elevata sicurezza nel meccanismo di recupero password. Questo sistema rappresenta il classico esempio in cui aumentare la sicurezza significa compromettere l’esperienza d’uso dell’utente (anche se in questo caso sono da apprezzare scelte implementative come quella del “vouchcode”, che sono in grado di rendere, nel loro piccolo, più usabile il sistema).

Nel sistema “uthentication with plural servers in the correct order” [67] è necessario che venga fornito anche un ordine sui certificati prodotti oltre che alla loro validità. In termini di usabilità questo si traduce in un peggioramento rispetto al sistema classico, a causa dell’informazione aggiuntiva che l’utente deve necessariamente memorizzare.

Discorso contrario per il meccanismo Multi-Sign-On [68] dove, grazie al sistema a punteggio, viene fornita più flessibilità all’utente finale; esso potrà infatti decidere quali e quanti meccanismi usare per potersi autenticare, mantenendo allo stesso tempo inalterata la struttura di base di un sistema multi-factor.

Infine, in ZeTA, abbiamo un’usabilità che dipende fortemente dal numero di errori commessi dagli utenti. Secondo lo studio fatto da Gutmann [69] su 20 partecipanti, la percentuale di risposte esatte raggiunge il 93.125%, con una distribuzione degli errori data da: AND=1.6%, OR=3.3%, AND NOT=11%, OR NOT=12.5%. Risulta evidente come il tasso di errori si concentri prevalentemente sulle congiunzioni NOT, la cui assenza potrebbe ridurre le imprecisioni effettuati dagli utenti.

4.5 Usabilità nei sistemi di autorizzazione

Discuteremo in questa parte, qual è l’approccio che può avere un utente, nell’istante in cui si trova a dover definire delle policy per proteggere dei dati sensibili. Un primo caso di studio potrebbe essere quello delle mail, uno dei metodi preferiti dagli utenti per condividere file [131], dove tramite le funzionalità “to”, “cc” e “ccn” si può identificare l’insieme degli indirizzi a cui una email può essere spedita; questo rappresenta un metodo semplice ma non efficace per definire una ACL completa, in quanto il mittente non può definire alcun permesso specifico sui propri destinatari e sull’uso che essi possono fare sulla risorsa condivisa². Per venire incontro a questa problematica, è possibile utilizzare Facemail [132], un software che tramite una lista di volti (figura 4.3) è in grado di fornire un’idea su quante e quali persone potrebbero ricevere i dati condivisi.

La mail rappresenta quindi un classico esempio in cui un’elevata semplificazione di un meccanismo porta sia ad una forte usabilità che ad una limitazione nelle funzionalità di cui l’utente può usufruire, oltre che ad una condivisione non voluta dei propri file. Analizzeremo però ora quale potrebbe essere il comportamento dell’utente nell’istante in cui abbia la possibilità di adottare meccanismi di creazione delle policy più complesse, esaminando, in un secondo momento, alcuni criteri da considerare durante la fase di design.

²È anche vero infatti che i destinatari potrebbero potenzialmente fare ciò che vogliono dei file ricevuti, condividendolo ad esempio con la propria lista dei contatti.

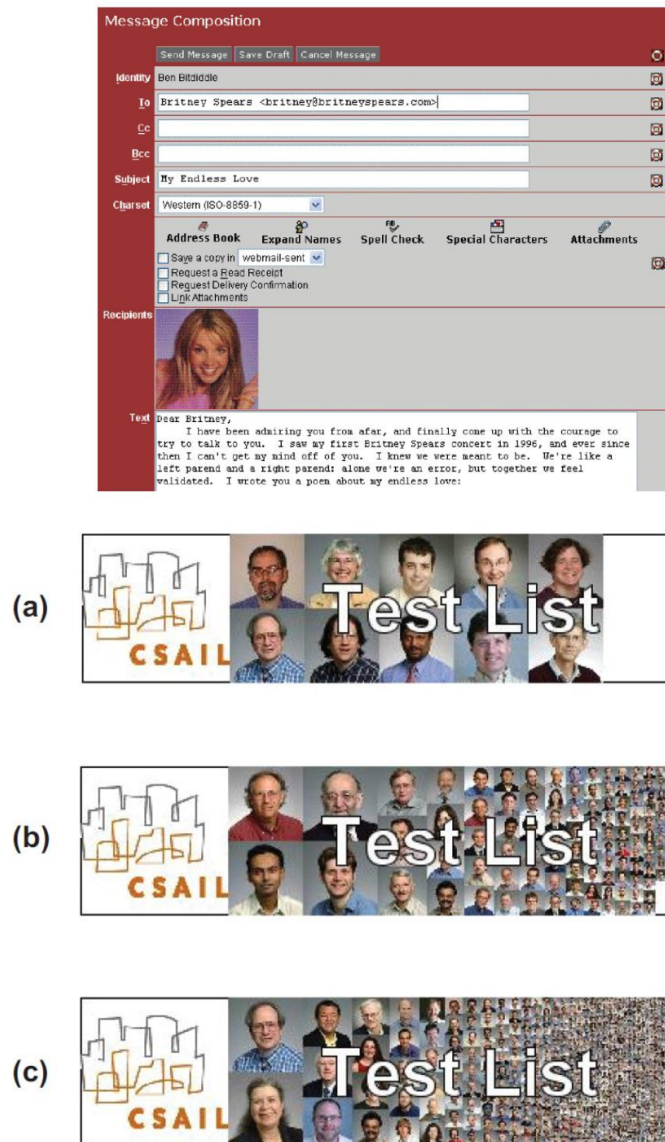


Figura 4.3. Uso di Facemail con 3 liste di potenziali destinatari: a) 10 facce, b) 100 facce, c) 1000 facce (fonte: [132]).

4.5.1 Definizione di una policy

Partiamo dal presupposto che non tutti gli utenti hanno le capacità e l'esperienza per poter definire in modo efficace una policy e che, il software utilizzato per determinarla, deve come prima cosa fornire la possibilità di determinare la granularità con il quale il proprietario vuole proteggere le sue risorse [133]. La user-interface rappresenta quindi un elemento cardine, dal momento che una bassa usabilità del sistema potrebbe portare l'utente a commettere errori durante la costruzione dell'ACL [134], compromettendo a priori il sistema di autorizzazione. Un esempio di software che può essere preso in analisi è quello proposto da Zurko [135], la cui bassa

usabilità ed il numero di errori che l'utente può accidentalmente commettere, viene analizzata nell'articolo di Smetters [136].

Prima di esaminare nel dettaglio i risultati su come gli utenti definiscono una ACL, è però bene precisare che la maggior parte di essi tendono a non farlo proprio, andando ad includere i propri dati sensibili in dispositivi personali, che non vengono condivisi con altri individui³ [137]. Tutti coloro che invece si cimentano nella costruzione di una policy, possono definire i permessi secondo una definizione individuale o attraverso gruppi di utenti [88]. Quest'ultimo permette di velocizzare il processo di costruzione di una policy dal momento che i privilegi non vengono assegnati in modo individuale, ma vengono condivisi tra i membri del gruppo. Allo stesso tempo però questa condivisione può determinare una mancanza di trasparenza, dal momento che un membro del gruppo potrebbe dare accidentalmente accesso alla risorsa in modo non intenzionale ad altri utenti.

Vediamo a questo punto l'analisi condotta da Smetters [136] in cui viene analizzato il comportamento di 16 utenti sulla gestione di 49672 oggetti, di cui 3375 ad accesso individuale ed i restanti condivisi tra gruppi. Il software che hanno utilizzato per gestire i privilegi è DocuShare, tramite il quale è stata tenuta traccia di tutte le modifiche fatte nell'arco di 9 anni. Durante questo periodo solo il 5.2% degli oggetti ha subito un cambiamento esplicito sugli accessi, rilevando quindi che gli utenti raramente cambiano i permessi su una cartella o un file, preferendo invece aggiungere l'oggetto in una cartella in cui l'ACL è già stata definita. Bisogna inoltre considerare che la maggior parte delle modifiche effettuate potrebbe essere stata causata da un errore [134] e non da una diretta volontà degli utenti. Infatti, nell'istante in cui viene modificata l'ACL di una cartella tramite l'interfaccia del programma, vi è la possibilità di applicare tale modifica in modo ricorsivo su tutti i file contenuti in quella cartella (azione di default per DocuShare). Questo viene confermato dal fatto che la maggior parte degli oggetti figli, in cui c'era stato un cambiamento nei privilegi, possiede gli stessi permessi della cartella padre che lo ospita. Infine, come ultimo aspetto è stato anche analizzato il tipo di ACL definito dagli utenti, rilevando che nella maggior parte dei casi si trattava di policy complesse. Anche qua, il motivo è da attribuirsi nuovamente all'errore umano, dal momento che la maggior parte di queste ACL venivano copiate da policy già esistenti, generando in questo modo accessi potenzialmente indesiderati.

4.5.2 Conflitti

Abbiamo visto nella parte precedente come l'errore umano possa rappresentare un forte ostacolo durante la definizione di una policy. Un'altra possibile problematica che può verificarsi, è però quella dei conflitti: supponiamo ad esempio che una regola A permetta ad un utente X di poter leggere e scrivere su un file Y, mentre una regola B vieti quei permessi ad un gruppo Z, in cui appartiene anche l'utente X. Se le regole A e B vengono applicate insieme, si verificherà un conflitto sui privilegi che l'utente X potrà esercitare sul file Y. Il problema principale non riguarda tanto il conflitto di

³Ritenendo questa pratica un sistema di autorizzazione adeguato per proteggere i propri dati.

regole, che è una situazione che può accadere anche agli autori di policy più esperti, ma le difficoltà che l'utente comune riscontra nel ricercarli e correggerli [138].

Analizziamo quindi, due metodi che possono essere utilizzati per definire una precedenza nell'istante in cui si verifica un conflitto tra due regole:

- Specificity precedence, in cui una regola applicata su un'entità specifica ha precedenza rispetto ad una generica. Con entità si può indicare l'elemento che detiene i permessi (utente o gruppo), oppure l'oggetto che viene protetto (file o cartella).
- Deny precedence, dove la negazione ha la priorità rispetto al permesso.

L'uso di questi approcci non è però in grado di fornire protezione totale dai conflitti ed ancora una volta sarà l'usabilità del programma a determinare la riuscita o meno del sistema. Per capire meglio, prendiamo in analisi lo studio condotto da Reeder, [139] in cui viene utilizzato un modello di conflict-resolution sul file system NTFS di Microsoft, che integra al suo interno entrambi i modelli di cui discusso in precedenza. L'obiettivo dell'esperimento proposto è di vedere quali sono le decisioni che gli utenti prendono per cercare di correggere un eventuale conflitto.

Supponiamo un primo esempio in cui bisogna dare ad un determinato utente i diritti di scrittura e lettura su un file. L'utente appartiene però a due gruppi differenti, uno in cui l'accesso a quel file viene negato ed uno in cui invece viene permesso, realizzando quindi un potenziale conflitto. Adottando il metodo deny precedence, quello che ci aspettiamo è che venga data priorità al primo gruppo, negando quindi i permessi all'utente. Il problema nasce però dal fatto che l'utente può non essere a conoscenza dell'esistenza della regola deny e, credendo di applicare unicamente quella di tipo allow, penserà erroneamente di avere garantito i diritti a quell'utente. Infatti, secondo l'analisi condotta da Reeder [139], su 54 partecipanti, il motivo principale per il quale gli utenti commettono questo tipo di errori è causato da un'interfaccia poco chiara di Windows, che non fornisce alcun feedback concreto sui reali privilegi impostati e sulla presenza, quindi, di eventuali conflitti.

Prendiamo infine in analisi un caso di due regole in cui la prima nega all'utente A l'accesso ad una cartella admin, mentre la seconda consente ad un gruppo, di cui fa parte l'utente A, di accedere ad un file contenuto nella medesima cartella. Non essendoci una regola più specifica dell'altra, si crea un potenziale conflitto, che viene risolto dal modello dando precedenza alla risorsa piuttosto che alla dimensione, fornendo quindi priorità alla seconda regola. Anche in questa situazione, rispetto allo studio fatto in precedenza, viene rilevato che la maggior parte degli utenti applica le due regole pensando che la precedenza venga attribuita alla dimensione, realizzando nel complesso una policy il cui risultato è differente da quello sperato.

4.5.3 Design

Analizzati quindi gli aspetti critici che intercorrono durante la costruzione di una policy, proviamo ad evidenziare, in accordo con [133, 140], una serie di elementi che bisognerebbe prendere in considerazione durante la progettazione del software:

- La maggior parte degli utenti, soprattutto quelli meno esperti nel campo dell'informatica, non hanno bisogno di definire policy complesse; per questo motivo, uno dei primi aspetti che dovrebbe mettere in evidenza il software in uso, è la granularità con il quale si vogliono definire le regole.
- Capita spesso che due conoscenti possano condividere dei dispositivi personali e questo può essere un problema se consideriamo che l'utente è incline a nascondere in esso dei dati sensibili [137]. Una soluzione adottabile potrebbe essere quella di avere un'interfaccia guest da abilitare nel momento antecedente al prestito, che vieta l'accesso a quelle risorse. La gestione di questa interfaccia dovrebbe essere fatta in modo discreto, senza andare a porre cambiamenti grafici rispetto a quella tradizionale, in modo da non compromettere la fiducia dell'utente [141].
- L'interfaccia deve essere semplice e funzionale al fine di ridurre la possibilità di introdurre degli errori. Inoltre è necessaria anche una user-experience appagante evitando in questo modo che l'utente, durante la composizione delle policy, possa sentirsi frustrato.
- Può essere utile introdurre un log dettaglio consultabile dall'utente, che tiene traccia di tutti gli accessi e delle eventuali modifiche effettuate sulle proprie risorse.
- Il programma dovrebbe permettere la modifica dinamica delle policy già definite in precedenza.
- Il software dovrebbe rendere più visibile il controllo degli accessi, incorporando le politiche ad il contesto di utilizzo.
- Un modo efficace per risolvere le problematiche inerenti ai conflitti durante la definizione di una policy, può essere fatto non solo applicando dei metodi di conflict-resolution efficaci, ma garantendo anche che venga fornito all'utente il giusto feedback visivo.

Vediamo ora una serie di implementazioni che prendono spunto dagli elementi appena trattati.

Sharing Palette

La sharing palette [142] è un'interfaccia grafica che, non facendo uso di una access control list, è in grado di semplificare la gestione dei dati condivisi dall'utente. Il sistema mostrato nella figura 4.4.a si divide in tre parti: quella in alto a sinistra indica i file condivisi con tutta la rete, quella in basso a sinistra i file condivisi unicamente con uno o più individui, la cui definizione viene fatta nella terza parte, quella a destra. I file che non vengono condivisi, non sono presenti nello schema, in modo tale da non arricchire l'ambiente di informazioni non utili. La feature più interessante di questo meccanismo, rimane sicuramente la capacità di fornire in ogni istante all'utente la consapevolezza del gruppo o degli individui a cui sta condividendo una risorsa, semplicemente scorrendo il mouse sull'elemento, come mostrato nella figura 4.4.b.

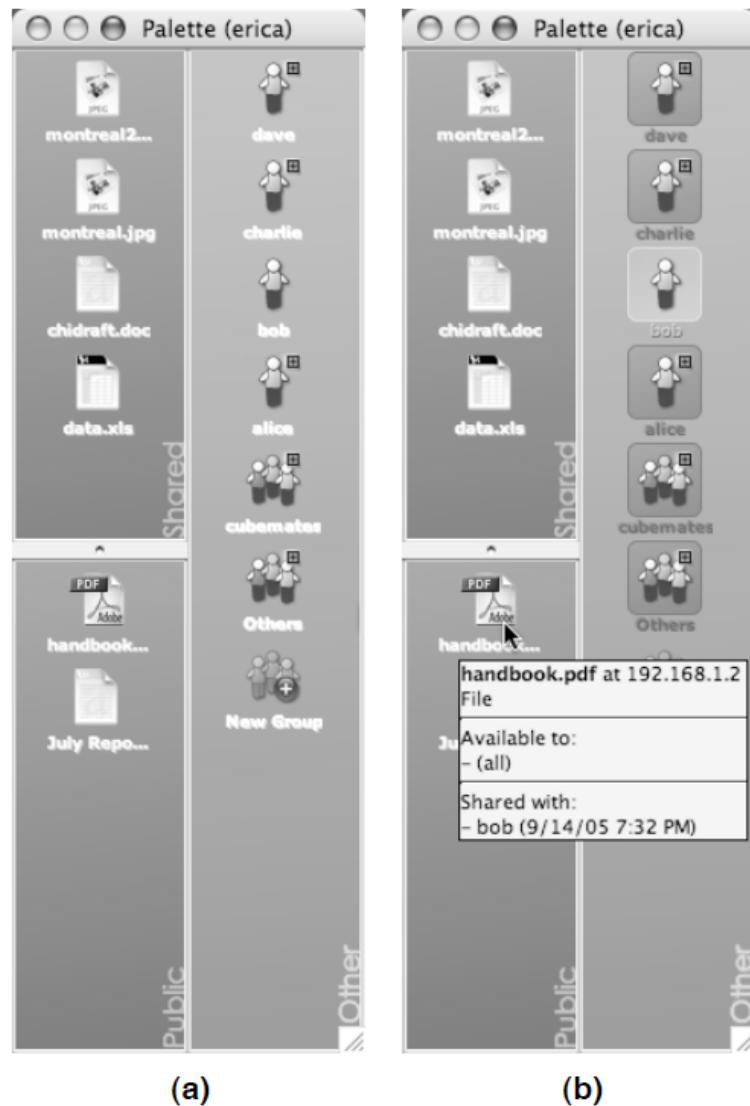


Figura 4.4. Sharing Palette (fonte: [142]).

Proximity Display

Solitamente l'interfaccia utente degli applicativi più utilizzati, non mette in primo piano informazioni riguardante il controllo degli accessi, perché tipicamente l'utente non è interessato a questo tipo di feature. Bisogna però sottolineare che alcuni tipi di piattaforma (come ad esempio Facebook) preferisce volontariamente omettere questo tipo di informazione, con l'obiettivo di far focalizzare l'attenzione del consumatore sui propri contenuti e sulle pubblicità [143]. Le conseguenze di tutto ciò è rilevabile direttamente da uno studio di Jennifer King [144], secondo cui solo un utente su 516 sa esattamente chi può vedere i contenuti che carica sulla piattaforma.

Affinché queste problematiche vengano corrette, è necessario quindi che tutti i maggiori programmi di file sharing (e tra questi anche i vari social network), implementino dei meccanismi che siano trasparenti, usabili e allo stesso tempo funzionali.

Un esempio potrebbe essere la tecnica di Proximity Display [145], che se applicata ad esempio ad una galleria di immagini, non apporta alcuna modifica estetica sostanziale, se non nell'istante in cui viene mosso il cursore sopra una risorsa: in quel caso infatti apparirà un elenco di gruppi con una serie di icone, a identificare il tipo di permesso applicato. Esso, inoltre, utilizza delle icone anziché delle tradizionali scritte, al fine di rendere più veloce ed immediata la comprensione da parte dell'utente.

Expandable Grid

Expandable Grid [134] rappresenta un meccanismo in grado di visualizzare su schermo in modo semplice ed efficace, i diritti che ogni singolo utente o gruppo possono avere su una risorsa. L'interfaccia è composta essenzialmente da una matrice (figura 4.5) nelle cui celle sono presenti le informazioni di access-control rappresentate tramite quadrati: ognuno di essi rappresenta un permesso, mentre il colore indica lo stato deny, allow o partial. La differenza sostanziale rispetto ai meccanismi classici, risiede però, dato un set di policy applicate, nella segnalazione dinamica dei conflitti che permette di limitare la probabilità che l'utente possa commettere un errore.

Turtle Guard

Abbiamo visto nel paragrafo 3.1.2, come la gestione dei permessi sulle versioni android più recenti possa essere completata attraverso un meccanismo di tipo AOFU. Questo si realizza tramite il fatto che, la decisione dell'utente, viene richiesta solo nell'istante in cui l'applicazione vuole fare un effettivo uso di una determinata risorsa. Se in un primo momento questo approccio può apparire ottimale perché consente una scelta consapevole da parte dell'utilizzatore, nella pratica può risultare poco usabile, dal momento che l'utente potrebbe essere sopraffatto da un numero elevato di richieste [146]. Una soluzione a questa problematica può essere fornita dal machine learning, con la quale si potrebbe realizzare un sistema decisionale dei processi automatico, basato su un set di parametri forniti dall'utente [147]. TurtleGuard [148], utilizza proprio questa tecnica con l'obiettivo di settare automaticamente sull'applicativo un insieme di permessi, semplificando in questo modo l'operato dell'utente. Per evitare che l'utente rimanga ignaro dei reali permessi di cui può disporre l'applicazione, TurtleGuard utilizza un insieme di popup per notificargli tutte le azioni che sono state effettuate dal sistema; in più, tutti i permessi, possono essere resettati manualmente, secondo le proprie esigenze.

L'interfaccia classica di Android, in entrambi i modelli AOI e AOFU, presenta alcune criticità a livello di usabilità e user-experience: risulta, ad esempio, spesso complicato per l'utente andare a visionare quali siano i permessi legati ad una specifica applicazione, in quanto questo tipo di informazione viene “nascosta” all'interno delle impostazioni generali; non vengono, inoltre, mai forniti dei dettagli sulla validità dei permessi rilasciati dall'utente. Su TurtleGuard è stato invece fatto un ottimo lavoro durante la fase di design, permettendo di realizzare un'applicazione separata dal sistema operativo che, lavorando in background, risulta trasparente ed

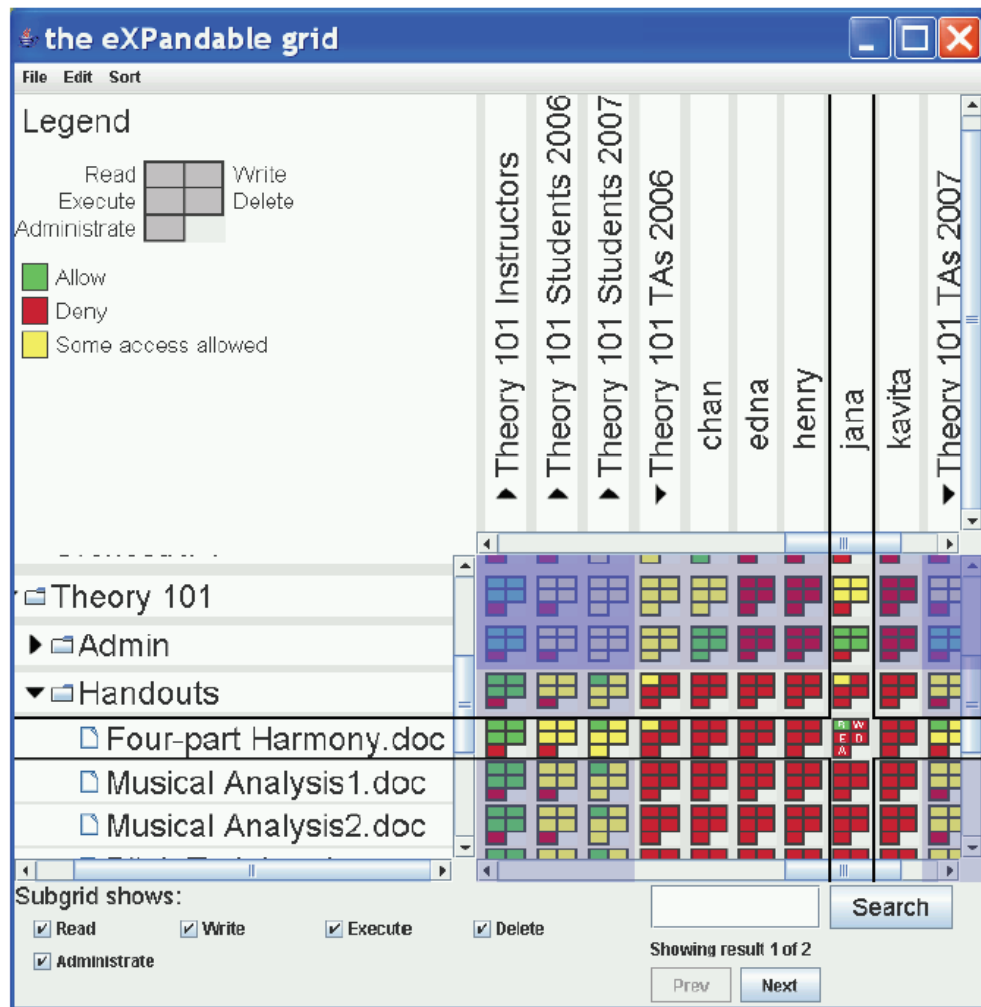


Figura 4.5. Expandable Grid (fonte: [134]).

allo stesso tempo accessibile in ogni istante. Andando infatti a cliccare sulla relativa icona, l'utente può facilmente gestire i permessi assegnati alla propria lista di applicazioni, differenziando quelli rilasciati quando l'app è in esecuzione, da quelli che hanno una validità continua.

4.6 Valutazione comparativa sull'usabilità

Arrivati a questo punto è bene riassumere i risultati evidenziati nel quarto capitolo, attraverso uno schema riassuntivo (vedi figura 4.6) che permette di comparare le differenze di usabilità che intercorrono tra i vari sistemi di autenticazione.

Tra i parametri di valutazione esistono quello che si basano su delle caratteristiche intrinseche al singolo sistema:

- False Rejection Rate, ovvero la possibilità che il sistema rilevi falsi negativi. Il livello di FRR dipende dalla precisione di rilevamento del sistema utilizzato.

- L'interfaccia del sistema e la relativa user-experience.
- La velocità con cui un utente, che non ha mai visto l'interfaccia del sistema in precedenza, è in grado di eseguire una specifica attività.
- La soddisfazione personale di un utente nell'usufruire di un determinato sistema.

Questi elementi, tuttavia, non verranno analizzati in questo paragrafo in quanto dipendenti all'implementazione e, per questo motivo, non forniscono un metodo di valutazione valido sull'intera categoria.

Vediamo ora i parametri su cui si baserà invece la valutazione:

- Tempo Funzionamento (TF): tempo richiesto dall'utente per imparare il funzionamento del sistema.
- Tempo Login (TL): tempo richiesto dall'utente per effettuare il login.
- Errori Utente (EU): possibilità che l'utente commetta degli errori durante il login.
- Errori Sistema (ES): possibilità che il sistema commetta degli errori durante il login (ovvero l'introduzione di un FRR).
- Affinità (A): affinità che intercorre tra utente e sistema.
- Limiti nell'applicabilità (LA): indica gli scenari all'interno del quale non è possibile utilizzare il sistema.
- Sforzo cognitivo (SC): complessità che riscontra l'utente nell'utilizzare il sistema.
- Trasparenza (T): impatto che il sistema ha sull'utente e sulle sue attività.

Dal momento che non esiste una scala comune su cui è possibile valutare questi parametri, ne è stata definita una relativa: se ad esempio il sistema in analisi ha un'affinità alta con l'utente ed ha pochi limiti di applicabilità, tutti e due i parametri avranno una valutazione elevata, dal momento che l'usabilità derivante è in entrambi casi alta.

La scala di valutazione può quindi assumere quattro valori differenti:

- 1 = per nulla usabile
- 2 = poco usabile
- 3 = abbastanza usabile
- 4 = molto usabile

È inoltre da segnalare che in alcune valutazioni è possibile assegnare un range di valori ('÷') o un insieme puntuale di essi (','); questo può succedere nel momento in cui il valore esatto dipende dalla specifica implementazione utilizzata e non dall'intera categoria.

Sistema	TF	TL	EU	ES	A	LA	SC	T
Password testuali ripetibili	4	2÷4	3÷4	4	1	4	4	1
OTP	4	3÷4	3	4	2	3	4	1
TOTP	4	4	2÷3	4	2	4	4	1
S/KEY	4	2÷4	3÷4	4	1	4	4	1
Blonder Scheme	3	4	3÷4	4	4	4	4	1
Cued-click points	3	3	3÷4	4	4	4	3	1
Persuasive CCP	2	3	3÷4	4	3	4	3	1
PassMatrix	1	1	2	4	3	4	1	1
Alignment based graphical password	1	1	2	4	3	4	1	1
IBRAS	1	2÷3	4	4	4	1	3	1
Volto	4	4	4	1÷4	4	2	4	1,4
Iride	4	4	4	1÷4	4	2	4	1,4
Impronta digitale	4	4	4	1÷4	4	3	4	1,4
Sguardo	4	2÷3	4	1÷4	4	2	4	1,4
Voce	4	3÷4	4	1÷4	4	2	4	1,4
Keystroke dynamics	4	1÷2	4	1÷4	4	3	4	4
Mobile keystroke dynamics	4	1÷2	4	1÷4	4	4	4	4
Input su schermo	4	1÷2	4	1÷4	4	3	4	4
Web & app history	2	1÷2	4	1÷4	4	3	4	4
In-air handwriting	3	2	4	1÷4	3	2	3	1
Keystroke dynamics in graphical password	3	3÷4	3÷4	1÷4	2÷4	3	4	4
Fourth factor	1	1	2÷3	4	1	2	1	1
Plural server in the correct order	2	1	1÷4	4	1	3	2	1
Multi-Sign-On Authentication Collector	2	1	1÷4	4	1	3	2	1
ZeTA	2	2	1÷4	4	3	3	3	1

Figura 4.6. Tabella comparativa sull'usabilità.

4.7 Conclusione

Conclusa anche l'analisi sull'usabilità dei sistemi di autenticazione e autorizzazione, non ci resta che andare a capire in che contesti essi possano essere implementati. Questa parte, fondamentale per comprendere lo scenario in cui verrà inserito poi il prototipo finale, verrà approfondita nel capitolo successivo.

Capitolo 5

Scenari e casi d'uso

Il caso d'uso viene definito come [149]:

“la descrizione di una serie di interazioni, finalizzata al raggiungimento di un obiettivo, tra uno o più fattori esterni ed il sistema in considerazione”.

È possibile definire un insieme di elementi, al fine di caratterizzare al meglio un caso d'uso [150]:

- Attore: utente o gruppo di individui che interagiscono con un sistema al fine di raggiungere un determinato obiettivo. Tra questi è possibile individuare l'attore primario (obbligatorio) e gli attori secondari (facoltativi), che possono collaborare con il primo per il raggiungimento dell'obiettivo finale.
- Obiettivo: indica il traguardo finale che deve essere raggiunto dall'attore.
- Sistema: in questa tipologia verranno racchiuse tutte le possibili tecniche che l'attore può usare per raggiungere l'obiettivo.
- Main Success Scenario (facoltativo e sostituibile con un riassunto descrittivo): elenco numerato delle interazioni che intercorrono tra sistema e attore (ed eventualmente attori secondari).
- Pre-condizioni: condizioni che devono essere verificate al fine di poter svolgere un determinato caso d'uso.
- Trigger (facoltativo): evento che viene scatenato al verificarsi di una determinata condizione.
- Estensioni: descrizione delle eventuali problematiche che possono scaturire durante l'esecuzione del caso d'uso.
- Miglioramenti: serie di elementi che possono essere presi in considerazione al fine di migliorare l'usabilità e la sicurezza del caso d'uso analizzato.
- Diagramma (facoltativo): rappresentazione grafica del caso d'uso che considera l'interazione (totale o parziale) degli aspetti visti in precedenza.

Tutti questi aspetti verranno analizzati nel dettaglio negli scenari che seguono.

5.1 Scenario #1: acquisto online

Il primo scenario è composto da un attore che vuole completare una transazione commerciale presso un sito online.

Attore

L'utente o l'azienda (che è possibile rappresentare come un gruppo di utenti) che vogliono effettuare un acquisto su internet.

Obiettivo

Completare con successo la transazione commerciale (punto di vista dell'attore) o impedirne l'avvenimento in caso di un uso illecito (punto di vista del sistema e del commerciante, oltre che dell'eventuale vittima del furto digitale).

Sistema

Il sistema è duplice ed è composto dal sito di vendita online, da cui l'attore principale effettua l'acquisto e da un Payment Gateway, il circuito finanziario con il quale il sito ha stretto un accordo.

Pre-condizioni

Innanzitutto si presuppone che l'utente si sia già autenticato (ove richiesto) sul sito online, attraverso uno scenario simile a quello che verrà presentato nel caso d'uso successivo. Il sito non dispone di un servizio che permette di memorizzare in modo sicuro i dati della carta di credito: questo significa che l'utente deve per ogni volta inserire le informazioni bancarie. Viene inoltre assodato che sia stata effettuata una mutua autenticazione sul sito o che comunque si disponga di un certificato valido che sia in grado di confermare la veridicità di quest'ultimo. L'attore principale deve possedere dei dati bancari autentici e validi, oltre che ad esserne il proprietario diretto, e deve avere un credito sufficiente per coprire l'acquisto che vorrebbe effettuare. Deve infine essere presente una connessione a internet.

Main Success Scenario

L'interazione tra attore e sistema si compone delle seguenti fasi (figura [5.1](#)):

- 1 L'attore decide di effettuare un acquisto presso un sito online. Il sistema, dopo avere verificato la disponibilità dell'oggetto, invia formalmente il prezzo al compratore.
- 2 L'utente, se è d'accordo con il prezzo, accetta l'offerta. A questo punto il sito propone al compratore un form all'interno del quale poter inserire i dati relativi alla carta di credito, con il quale vuole effettuare l'acquisto.

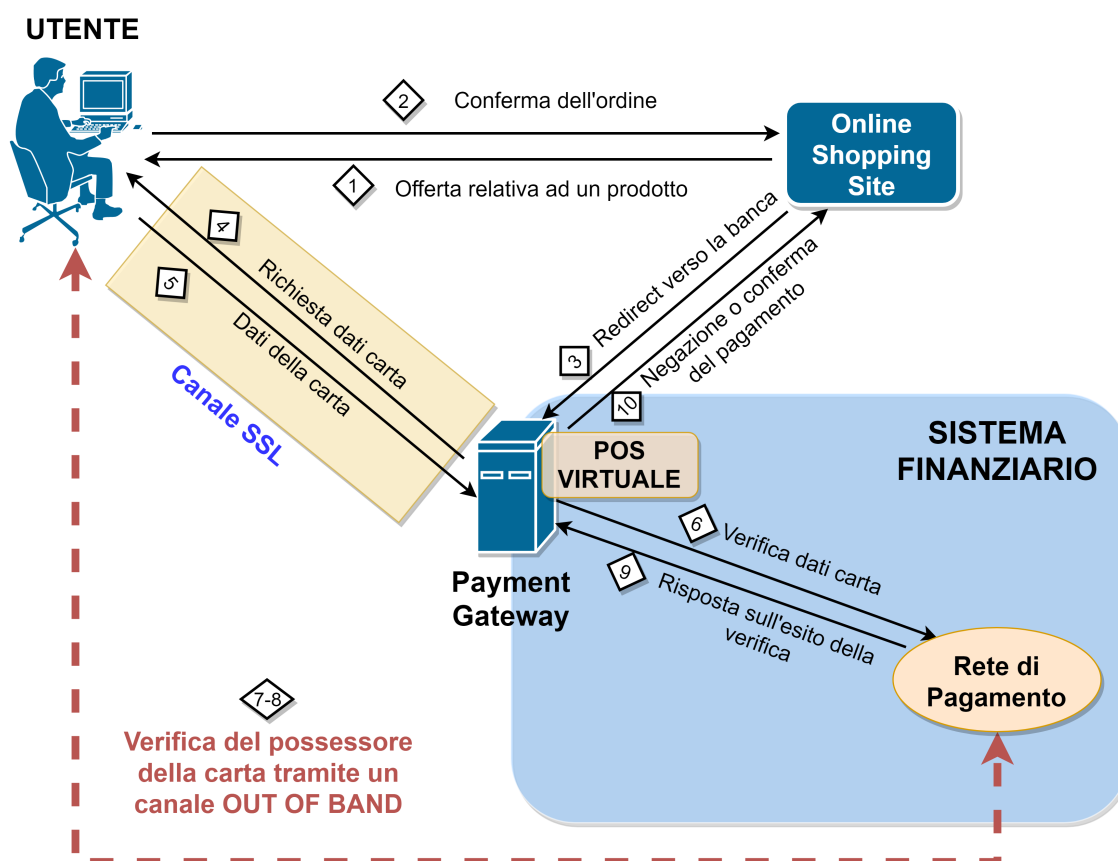


Figura 5.1. Diagramma dello scenario #1.

- 3 Solitamente il sito non gestisce direttamente i dati della carta di credito ma viene utilizzato l'uso di un **trigger**: quando viene rilevata una transazione commerciale, il sito effettua una redirect verso il proprio Payment Gateway, che si occuperà direttamente della pratica.
- 4 Il PG apre un canale SSL con l'attore; solo a quel punto l'utente può inserire i dati bancari.
- 5-6 Il PG, ricevuti i dati immessi dall'utente, utilizza un software chiamato Point Of Sale per trasformare quei dati in informazioni adatte al mondo finanziario.
- 7-8 Se i dati inseriti sono corretti, la banca può richiedere (**trigger**) un controllo ulteriore per verificare che l'attore sia effettivamente il proprietario della carta di credito di cui ha inserito gli estremi. Questa verifica dovrebbe essere fatta all'interno di un canale Out Of Band, differente da quello utilizzato per effettuare la transazione principale.
- 9-10 Se l'identità della persona viene confermata ed i dati inseriti sono corretti, la transazione viene approvata dal sistema bancario e viene inviata la relativa conferma al sito web e al compratore.

Estensioni

Di seguito l’elenco delle problematiche che si potrebbero presentare nell’MSS appena descritto:

- Nel punto 2 l’inserimento di dati relativi alla carta di credito dell’utente in un form rappresenta un grosso pericolo, dal momento che il commerciante potrebbe appropriarsi di quelle informazioni per commettere delle frodi ai danni dell’attore. In questo caso risulta fondamentale conoscere il sito con il quale si sta interagendo (vedi pre-condizioni).
- Nel punto 4, il trasferimento di dati nella rete è un processo molto pericoloso in quanto sensibile ad una serie di attacchi: in questo caso l’utilizzo di un canale sicuro è obbligatorio.
- Nei punti 7-8, il tipo di verifica OOB effettuata dalla banca, potrebbe risultare meno sicuro rispetto ad altri. Ad esempio il protocollo Signaling System Seven (SS7) presenta ormai da parecchi anni alcune criticità [151], che permetterebbero di effettuare in modo abbastanza semplice un attacco MITM, col fine di rubare il token OTP. Segnalo inoltre la possibilità di clonare la sim della vittima, tramite l’ausilio di tecniche di social engineering, avendo in questo modo un accesso diretto ai codici OTP inviata dalla banca.

Miglioramenti

Basandoci sul punto 7-8 del diagramma 5.1, vediamo i sistemi OTP implementati da due banche, Unicredit e FinecoBank:

- Unicredit [152] fornisce tre differenti meccanismi per la gestione dei codici OTP. Il primo, caratterizzato da un token elettronico, rappresenta il metodo più sicuro, dal momento che usufruisce di un canale di comunicazione separato da quello utilizzato per comunicare i dati bancari. Un sistema molto simile è quello chiamato “Password Card”, una tesserina contenente 40 codici prestampati: qui l’unica criticità risiede nel fatto che un’eventuale foto del supporto permetterebbe di conoscere tutte le password OTP utilizzabili dall’utente. L’ultimo sistema è quello del Mobile Token, un generatore virtuale di token OTP, implementato direttamente nell’app Unicredit. In questo caso il livello di sicurezza si abbassa in quanto potrebbero rientrare problematiche di social engineering o Malware che sarebbero potenzialmente in grado di intaccare il corretto funzionamento del sistema. In ogni caso, il sistema messo in atto da Unicredit presenta un alto livello di flessibilità, usabilità e sicurezza, non introducendo alcun sistema basato sul protocollo SS7.
- FinecoBank [153], utilizza unicamente l’SMS come metodo per fornire le password OTP ai propri clienti. Esso, nonostante sia caratterizzato da un semplice utilizzo, si basa sul protocollo SS7, risultando di conseguenza facilmente vulnerabile da un MITM. Inoltre il canale di comunicazione utilizzato dall’SMS,

potrebbe risiedere sullo stesso dispositivo con il quale si è avviata la transazione commerciale, violando in questo modo i requisiti dell’OOB. FinecoBank non fornisce sistemi OTP alternativi all’SMS.

In conclusione, risulta evidente come da un punto di vista della sicurezza bisognerebbe evitare l’utilizzo del protocollo SS7, adottando l’uso di tecniche alternative come fatto da Unicredit.

Da un punto di vista dell’usabilità, si potrebbe intervenire andando a implementare delle API android [\[154\]](#) che siano in grado di riconoscere ed inserire automaticamente nel form l’SMS inviato dalla banca (come discusso nel paragrafo [4.1](#)).

5.2 Scenario #2: dispositivo personale

Il secondo scenario è composto da un attore che vuole accedere ad un proprio dispositivo personale.

Attore

L’utente che vuole accedere ad un determinato dispositivo personale.

Obiettivo

Completare con successo l’autenticazione (punto di vista dell’attore) o impedirne l’avvenimento in caso di un uso illecito (punto di vista del sistema e del proprietario dell’account).

Sistema

I sistemi in gioco sono il dispositivo locale e l’authentication server.

Pre-condizioni

Nel caso l’autenticazione necessiti della rete, il dispositivo deve avere una connessione ad internet. L’attore deve inoltre possedere (fisicamente o intrinsecamente) l’input corretto con il quale si vuole autenticare. L’autenticazione può richiedere l’uso di un authentication server, attraverso tecniche come:

- Repeatable Textual Password ([2.2.1](#))
- One-Time Password ([2.2.2](#))
- Time-based One-Time Password ([2.2.3](#))

- S/KEY ([2.2.4](#))
- Fourth-factor authentication ([2.10.2](#))
- Authentication with plural Servers in the correct order ([2.10.3](#))
- Multi-Sign-On Authentication Collector ([2.10.3](#))
- Zero-Trust Authentication ([2.10.4](#))

Oppure, può risultare sufficiente la memoria interna del dispositivo con il quale si sta effettuando l'autenticazione:

- Physiological Biometric Systems ([2.7](#))
- Keystroke dynamics ([2.8.1](#))
- Authentication through an Hidden Markov Model Behavior ([2.8.4](#))
- Authentication by web browsing history ([2.8.5](#))
- Authentication using In-Air Handwriting Style Verification ([2.8.6](#))

Mentre esistono sistemi che si possono adattare ad entrambi i meccanismi, come ad esempio la categoria delle graphical password ([2.3](#)).

Si ipotizza poi che il dispositivo utilizzato per immettere l'input, non presenti alcun software dannoso¹ che possa in qualche modo compromettere la corretta riuscita dell'operazione. Viene inoltre assodato che sia stata effettuata una mutua autenticazione con il server adibito all'autenticazione o che comunque si disponga di un certificato valido che sia in grado di confermare la veridicità di quest'ultimo.

Main Success Scenario

L'interazione tra attore e sistema si compone delle seguenti fasi (figura [5.3](#)):

CASO A

- 1 L'utente inserisce un input all'interno del dispositivo con l'obiettivo di autenticarsi. L'input può essere immesso attraverso l'uso di credenziali o tramite sistemi biometrici.
- 2-3 Nel caso in cui l'utente abbia immesso un input i cui dati di confronto risiedono nella memoria locale, la comparazione viene fatta direttamente con il sample creato in fase di registrazione, all'interno del dispositivo stesso.

CASO B

¹Come ad esempio un keylogger, strumento in grado di effettuare lo sniffing dei dati inseriti, direttamente dal dispositivo dell'utente.

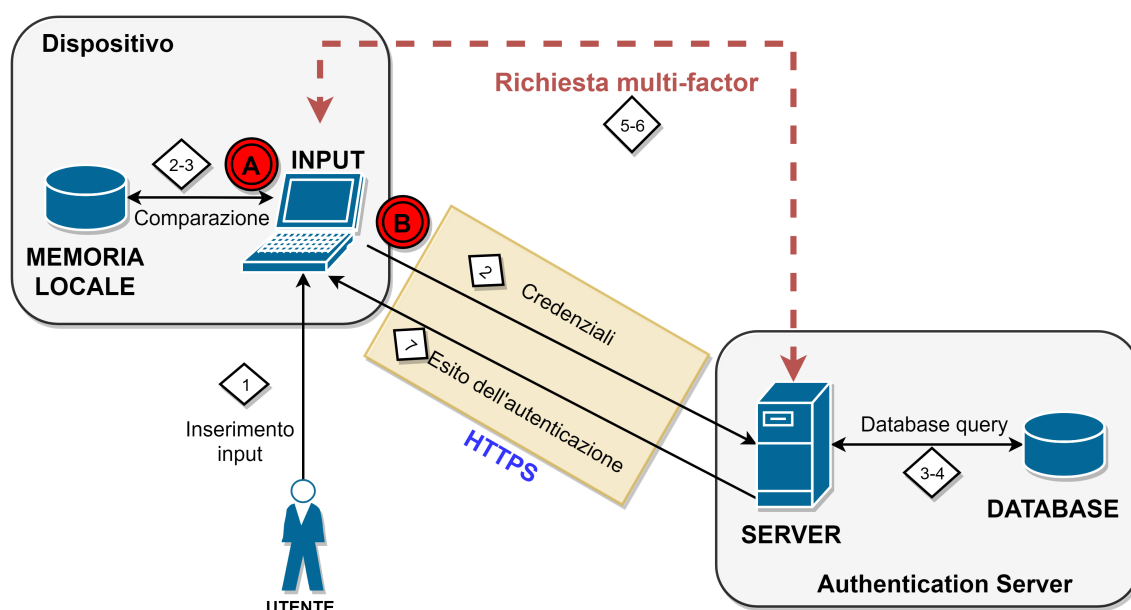


Figura 5.2. Diagramma dello scenario #2.

- 1 L'utente inserisce un input all'interno del dispositivo con l'obiettivo di autenticarsi. L'input può essere immesso attraverso l'uso di credenziali (es. username e password) o tramite sistemi biometrici (es. impronta digitale).
- 2 Le credenziali vengono spedite tramite la rete, verso il relativo authentication server.
- 3-4 L'authentication server fa una query al proprio database, per vedere se è presente una corrispondenza: se non viene rilevato alcun match, l'autenticazione è considerata fallita.
- 5-6 In base al sistema ed al tipo di registrazione fatta, l'authentication server può richiedere un'ulteriore prova dell'identità dell'utente, tramite meccanismo multi-factor (**trigger**).
- 7 L'authentication server invia l'esito dell'autenticazione all'utente, insieme all'eventuale risorsa.

Estensioni

Di seguito l'elenco delle problematiche che si potrebbero presentare nell'MSS appena descritto:

- Nel primo punto, l'introduzione da parte dell'utente di una password ripetibile o di una sequenza grafica, potrebbe essere rilevata da un malintenzionato attraverso tecniche di shoulder surfing e smudge attack. Al contrario l'utilizzo di sistemi biometrici, sia fisici che comportamentali, non introducono in questa fase nessuna problematica.

- Nel caso A, se l’input immesso è di natura biometrica, l’utente potrebbe non essere riconosciuto correttamente dal dispositivo, a causa dell’incertezza di misurazione presente in questo tipo di tecnologia (vedi paragrafo 2.7).
- Nel punto 2 del caso B, il trasferimento di dati nella rete è un processo molto pericoloso in quanto sensibile ad una serie di attacchi: l’utilizzo di un canale sicuro risulta quindi obbligatorio.
- Nei punti 3-4, un eventuale attacco al database dell’authentication server, potrebbe compromettere le credenziali di accesso di tutti i rispettivi utenti.
- Per quanto riguarda il sistema multi-factor, punti 5 e 6, possono presentarsi le problematiche evidenziate nello scenario precedente (5.1).

Miglioramenti

Il sistema analizzato presenta diverse possibilità implementative, dando quindi la possibilità al progettista di realizzare la soluzione che sia in grado di coniugare al meglio usabilità e sicurezza.

Se il meccanismo multi-factor utilizzato si basa su OTP, valgono gli elementi discussi nello scenario precedente (5.1). Se invece utilizza altre tecniche, bisogna cercare di non appesantire troppo il carico di lavoro dell’utente, per evitare di ottenere un sistema complessivamente poco usabile.

Infine, il database dell’authentication server non dovrebbe in alcun modo memorizzare le credenziali degli utenti in chiaro, ma utilizzare tecniche di secret management con il quale mettere al sicuro i dati.

5.3 Scenario #3: account

Il terzo scenario è composto da un attore che vuole accedere ad un determinato account.

Attore

L’utente che vuole accedere ad un determinato account.

Obiettivo

Completare con successo l’autenticazione (punto di vista dell’attore) o impedirne l’avvenimento in caso di un uso illecito (punto di vista del sistema e del proprietario dell’account).

Sistema

I sistemi in gioco sono il dispositivo locale e l’authentication server.

Pre-condizioni

Il dispositivo deve avere una connessione ad internet. L'attore deve inoltre possedere (fisicamente o intrinsecamente) l'input corretto con il quale si vuole autenticare, ad esempio attraverso una delle tecniche viste in precedenza:

- Repeatable Textual Password ([2.2.1](#))
- One-Time Password ([2.2.2](#))
- Time-based One-Time Password ([2.2.3](#))
- S/KEY ([2.2.4](#))
- Fourth-factor authentication ([2.10.2](#))
- Authentication with plural Servers in the correct order ([2.10.3](#))
- Multi-Sign-On Authentication Collector ([2.10.3](#))
- Zero-Trust Authentication ([2.10.4](#))
- Vault ([2.12.1](#))

Si ipotizza che il dispositivo utilizzato per immettere l'input, non presenti alcun software dannoso che possa in qualche modo compromettere la corretta riuscita dell'operazione. Viene inoltre assodato che sia stata effettuata una mutua autenticazione con il server adibito all'autenticazione o che comunque si disponga di un certificato valido che sia in grado di confermare la veridicità di quest'ultimo.

Main Success Scenario

L'interazione tra attore e sistema si compone delle seguenti fasi (figura [5.3](#)):

- 1 L'utente inserisce un input all'interno del dispositivo con l'obiettivo di autenticarsi.
- 2 Le credenziali vengono spedite tramite la rete, verso il relativo authentication server.
- 3-4 L'authentication server fa una query al proprio database, per vedere se è presente una corrispondenza: se non viene rilevato alcun match, l'autenticazione è considerazione fallita.
- 5-6 In base al sistema ed al tipo di registrazione fatta, l'authentication server può richiedere un ulteriore prova dell'identità dell'utente, tramite meccanismo multi-factor (**trigger**).
- 7 L'authentication server invia l'esito dell'autenticazione all'utente, assieme all'eventuale risorsa.

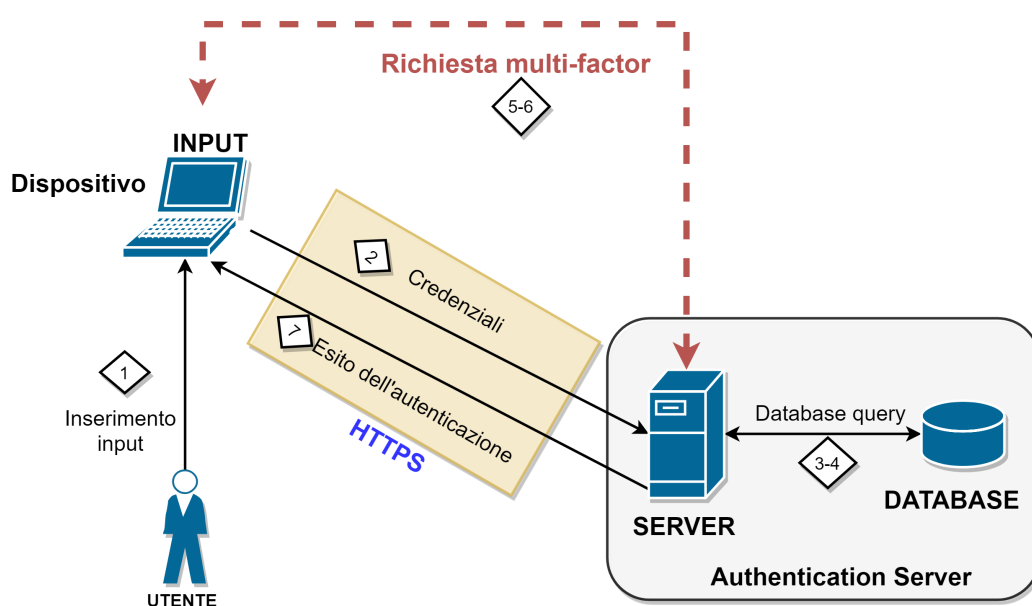


Figura 5.3. Diagramma dello scenario #3.

Estensioni

Di seguito l'elenco delle problematiche che si potrebbero presentare nell'MSS appena descritto:

- Nel primo punto, l'introduzione da parte dell'utente di una password ripetibile o di una sequenza grafica, potrebbe essere rilevata da un malintenzionato attraverso tecniche di shoulder surfing e smudge attack. Al contrario l'utilizzo di sistemi biometrici, sia fisici che comportamentali, non introducono in questa fase nessuna problematica.
- Nel punto 2, il trasferimento di dati nella rete è un processo molto pericoloso in quanto sensibile ad una serie di attacchi: l'utilizzo di un canale sicuro risulta quindi obbligatorio.
- Nei punti 3-4, un eventuale attacco al database dell'authentication server, potrebbe compromettere le credenziali di accesso di tutti i rispettivi utenti.
- Per quanto riguarda il sistema multi-factor, punti 5 e 6, possono presentarsi le problematiche evidenziate nello scenario precedente (5.1).

Miglioramenti

Il sistema analizzato presenta diverse possibilità implementative, dando quindi la possibilità al progettista di realizzare la soluzione che sia in grado di coniugare al meglio usabilità e sicurezza.

Se il meccanismo multi-factor utilizzato si basa su OTP, valgono gli elementi discussi nello scenario precedente (5.1). Se invece utilizza altre tecniche, bisogna cercare di non appesantire troppo il carico di lavoro dell'utente, per evitare di ottenere un sistema complessivamente poco usabile.

Infine, il database dell'authentication server non dovrebbe in alcun modo memorizzare le credenziali degli utenti in chiaro, ma utilizzare tecniche di secret management con il quale mettere al sicuro i dati.

5.4 Scenario #4: condivisione di risorse presenti su cloud, attraverso l'utilizzo di Temporary Constraint

Il quarto scenario è composto da un attore principale che vuole condividere una risorsa, memorizzata in cloud, con degli attori secondari, tramite link temporanei.

Attore

L'attore principale è l'utente possessore di una risorsa, contenuta in un cloud, che deve essere condivisa con l'utente A e B, i due attori secondari.

Obiettivo

L'obiettivo dell'attore principale è quello di ricavare un ticket, relativo alla risorsa contenuta in cloud, che conceda per un periodo limitato l'accesso agli attori secondari.

Sistema

Il sistema è composto da un servizio di cloud che contiene la risorsa da condividere e da un dispositivo A che appartiene all'utente A. Sul dispositivo è presente un file-system user-level, associato all'account che l'utente A ha sul servizio di clouding.

Pre-condizioni

L'attore principale e l'utente A devono essersi già autenticati tramite il proprio cloud service, attraverso uno scenario simile a quello presentato nel caso d'uso precedente (paragrafo 5.3). Il servizio di clouding utilizzato dall'utente A deve essere lo stesso di quello usato dall'attore principale. L'utente B non ha nessun particolare tipo di vincolo.

Si ipotizza infine che tutti gli attori abbiano una connessione internet e che i dispositivi da loro utilizzati non presentino alcun software dannoso che possa in qualche modo compromettere la corretta riuscita delle operazioni effettuate.

Main Success Scenario

L'interazione tra attore e sistema si compone delle seguenti fasi (figura 5.4):

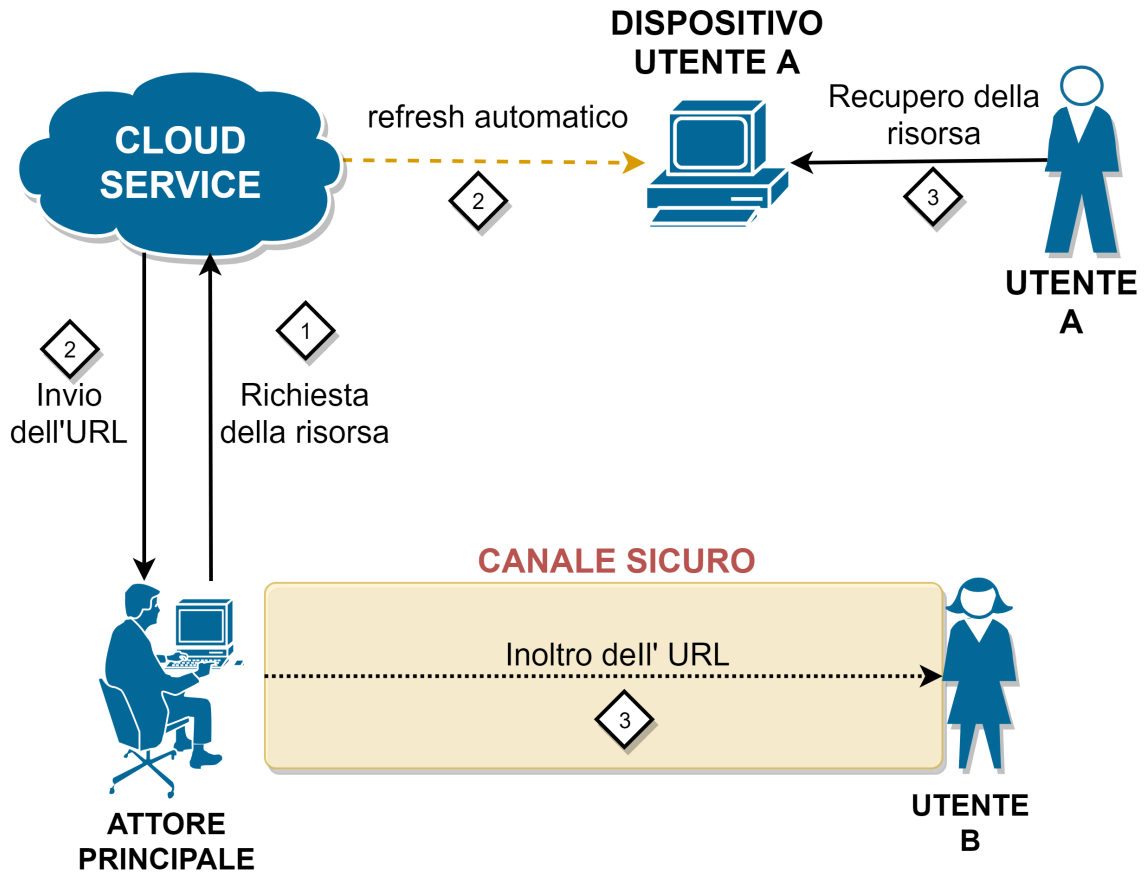


Figura 5.4. Diagramma dello scenario #4.

UTENTE A

- 1 L'attore principale richiede, al proprio servizio di clouding, il ticket temporaneo con il quale poter condividere la risorsa.
- 2 Dopo aver autorizzato l'accesso alla risorsa, il cloud service aggiorna automaticamente il file system dell'utente.
- 3 L'utente troverà la risorsa direttamente all'interno del proprio file system, sul suo dispositivo personale, per un periodo di tempo che è stato definito in precedenza dall'attore principale.

UTENTE B

- 1 L'attore principale richiede, al proprio servizio di clouding, il ticket temporaneo con il quale poter condividere la risorsa.

- 2 Dopo aver autorizzato l'accesso alla risorsa, il cloud service crea un URL tramite cui è possibile accedervi. L'URL, che ha una validità temporale limitata, viene inviato all'attore principale.
- 3 L'utente A inoltra l'URL all'utente B, tramite la rete o attraverso l'utilizzo di un canale OOB.

Estensioni

Di seguito l'elenco delle problematiche che si potrebbero presentare nell'MSS appena descritto:

- Il punto 3 dell'utente B rappresenta un punto di criticità, dal momento che qualunque MITM che riuscisse a rubare quel link, sarebbe poi in grado di accedere alla risorsa.
- Il punto 2 degli utenti A e B, potrebbe far scaturire la stessa problematica presente nel punto precedente. Tuttavia, trattandosi in questo caso di un processo interno al sistema di cloud service, la probabilità che si manifesti risulta di gran lunga inferiore rispetto al caso precedente².
- Un problema intrinseco del meccanismo di temporary constraints, è che l'attore principale non ha nessun controllo sulla risorsa che viene condivisa con gli utenti A e B.

La natura dell'URL, nonostante dipenda dal tipo di cloud service utilizzato, solitamente è composto dalle seguenti informazioni [155]:

- Il periodo di tempo all'interno del quale è garantito l'accesso alla risorsa
- Il percorso tramite cui è possibile accedere alla risorsa
- I permessi
- Una firma, associata dal cloud service (diversa da una firma digitale)

È chiaro come il link non impone alcun tipo di ACL sulla risorsa, permettendone quindi una possibile condivisione con altri contatti.

- Infine un attaccante potrebbe cambiare la parte dell'URL relativa ai permessi, per ampliare l'uso che può avere su quella risorsa.

Miglioramenti

Generalmente il meccanismo di temporary constraints, seppure presenti un'elevata usabilità, ha dei forti limiti a livello di sicurezza, dal momento che viene a mancare

²In questo caso l'unico fattore che entra in gioco è la sicurezza che è stata applicata al sistema di clouding.

un contatto diretto con la risorsa condivisa. Inoltre la firma applicata sull’URL dovrebbe assumere le caratteristiche della firma digitale, in modo tale da salvaguardare l’integrità del link ed evitare manipolazioni dirette o guessing attack.

Per quanto riguarda la spedizione del link, bisognerebbe cercare di utilizzare canali alternativi a quelli della rete IP, anche se approcci di questo tipo potrebbero abbassare l’usabilità del sistema. Servizi di clouding come MEGA [156] impongono l’utilizzo di chiavi crittografiche necessarie per la decifratura della risorsa condivisa; esse dovrebbero essere trasmesse in canali OOB e sicuri, così come segnalato dal sito stesso (vedi figura 5.5):

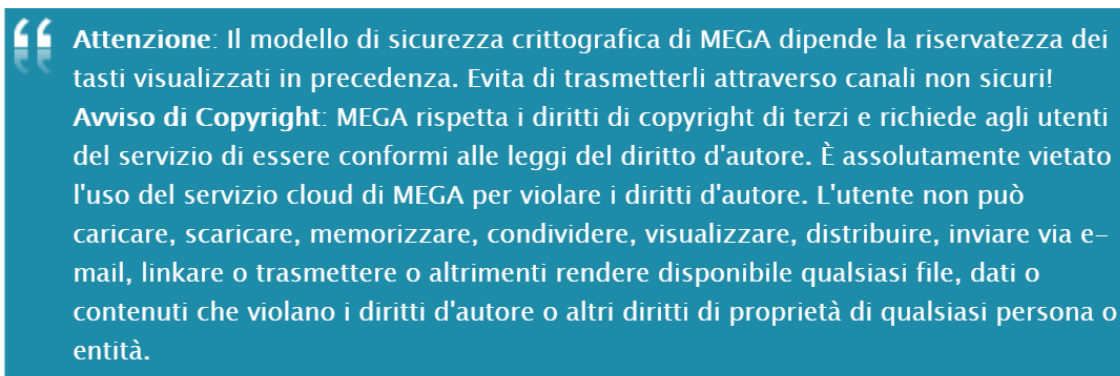


Figura 5.5. Segnalazione fornita MEGA alla creazione del link di condivisione della risorsa (fonte: [156]).

5.5 Scenario #5: condivisione di risorse presenti su cloud, attraverso l’utilizzo di Setting Per User Permission

Il quinto scenario è composto da un attore principale che vuole condividere una risorsa online con degli attori secondari, tramite condivisione di bucket.

5.5.1 Nozioni teoriche

In tutti i servizi di storage, le risorse possono essere divise in “object” e “bucket”, dove quest’ultimo rappresenta la root directory all’interno del quale risiedono i vari oggetti (che possono a loro volta essere distinti in file o cartelle). La condivisione di un bucket deve avvenire assieme alla costruzione di un Access Control List che contiene gli ID degli utenti che potranno accedere a quella risorsa, oltre che ai relativi permessi. Inoltre l’identificativo con il quale viene richiamato un bucket deve essere univoco sia all’interno del cloud che tra cloud service differenti.

Attore

L'attore principale è l'utente possessore di una risorsa, contenuta in un cloud, che deve essere condivisa con l'utente A e B, i due attori secondari.

Obiettivo

L'obiettivo dell'attore principale è di condividere un bucket su cui è stata definita un'ACL (contenente gli ID dei due attori secondari).

Sistema

Il sistema è composto dallo storage cloud dell'attore principale, al cui interno è presente la risorsa da condividere.

Pre-condizioni

Gli attori secondari devono possedere un account che sia riconoscibile dal servizio di cloud utilizzato dall'attore principale. Si suppone inoltre che essi siano già autenticati al proprio cloud service (secondo uno dei meccanismi visti nello scenario 5.3), che non deve essere necessariamente lo stesso di quello utilizzato dall'attore principale.

Si ipotizza che tutti gli attori abbiano una connessione internet e che i dispositivi da loro utilizzati non presentino alcun software dannoso che possa in qualche modo compromettere la corretta riuscita delle operazioni effettuate. Devono infine essere rispettate le seguenti regole:

- **Regola A:** i privilegi definiti dall'ACL devono riguardare le operazioni di creazione, eliminazione, lettura, scrittura e comparazione di tutti gli oggetti ricorsivamente contenuti nel bucket.
- **Regola B:** solo il possessore e i beneficiari del bucket possono operare su di esso.
- **Regola C:** i beneficiari non possono concedere alcun tipo di permesso ad altri utenti, su uno o più oggetti contenuti nel bucket.

Main Success Scenario

L'interazione tra attore e sistema si compone delle seguenti fasi (figura 5.6):

1. Gli utenti A e B inviano i propri ID all'attore principale.
2. L'attore principale fa richiesta al cloud storage del bucket su cui verrà applicata una ACL rispetto agli ID precedentemente inviati.

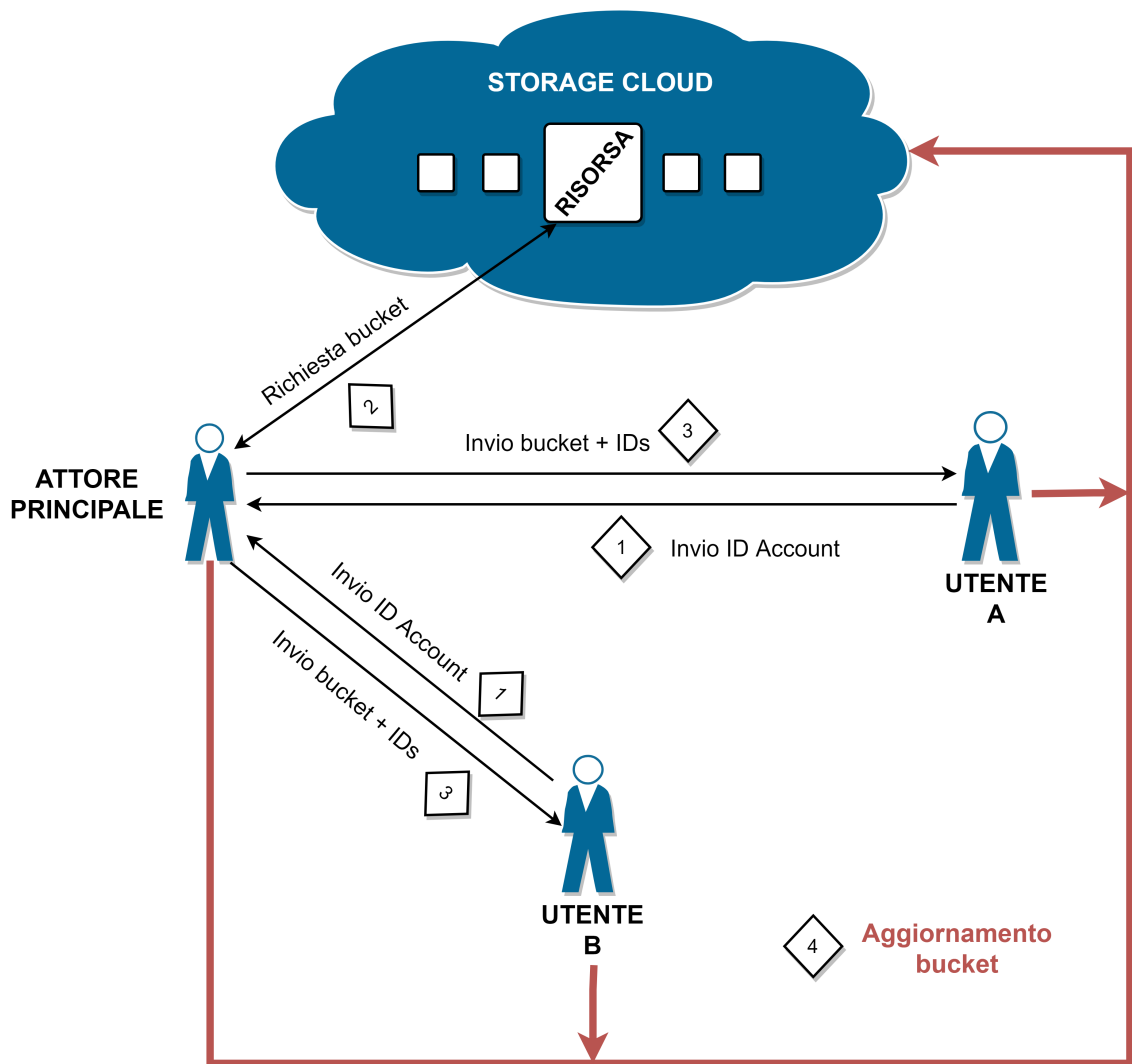


Figura 5.6. Diagramma dello scenario #5.

3. L'attore principale, ricevuto il bucket, lo inoltra agli utenti A e B.
4. Quando uno degli attori carica un nuovo oggetto nel bucket, esso deve necessariamente associargli una nuova ACL tramite il quale fornire l'elenco degli ID che possono esercitare il diritto di lettura su di esso.

Estensioni

Di seguito l'elenco delle problematiche che si potrebbero presentare nell'MSS appena descritto:

- Nel punto 1, un attaccante potrebbe rubare l'ID dell'utente A o B; questo, tuttavia, rappresenta un problema solo marginale dal momento che il malintenzionato non sarebbe comunque in grado di accedere al relativo account di clouding.

- Un discorso analogo può essere applicato al punto 3. È però da segnalare che, in questo caso, tutti i ricevitori del bucket vengono a conoscenza degli ID di cui è composta l'ACL, andando di conseguenza ad intaccare la privacy di tutti i beneficiari della risorsa.
- Nel punto 4 se l'utente A o B carica un nuovo oggetto, egli può garantire l'accesso di lettura ad altri utenti senza che l'attore principale lo sappia, o persino escludere altri beneficiari del bucket. Il primo caso viola la regola C, mentre il secondo caso contraddice la regola A. La tipologia di permessi che il proprietario dell'oggetto può imporre su di esso riguarda solo la lettura, non coinvolgendo in alcun modo la creazione, l'eliminazione, la scrittura e la comparazione della risorsa.

Miglioramenti

Nonostante le informazioni trasportate nel punto 1 e 3 non rappresentino da sole una strategia con il quale l'attaccante potrebbe accedere al bucket, è comunque consigliabile utilizzare un canale di trasmissione sicuro.

Per evitare problematiche di privacy, in cui ogni utente viene a conoscenza dell'ID degli altri beneficiari della risorsa, l'attore principale potrebbe creare tanti bucket quanti sono gli ID. Ovviamente questo tipo di approccio è applicabile solo nel caso in cui il numero dei beneficiari è ridotto, dal momento che, in caso contrario, la gestione dei bucket risulterebbe critica, oltre che diminuire l'usabilità dell'attore principale.

Per quanto riguarda il punto 4, la problematica è causata da un problema intrinseco del caso d'uso appena trattato. L'unica soluzione adottabile dall'attore principale (senza violare l'algoritmo) sarebbe quella di escludere tutti i partecipanti che violano le regole relative al bucket. Se invece fosse possibile agire al di sopra dello scenario descritto, l'eliminazione del punto 4 permetterebbe di rispettare tutte le pre-condizioni, a discapito però di un sistema che risulterebbe più chiuso nelle sue funzionalità.

5.6 Scenario #6: continuous authentication

Il sesto scenario è composto da un attore principale che, durante l'esecuzione di un'attività, viene monitorato da un algoritmo di continuous authentication.

Attore

Lo scenario è composto da un attore principale, l'utente fruitore del sistema, e da un potenziale attaccante che vuole fare un uso illecito del dispositivo.

Sistema

Il sistema è caratterizzato da un dispositivo su cui è montato sopra un meccanismo di continuous authentication (presentati nel paragrafo 2.8), in grado di rilevare automaticamente la presenza dell'attore principale o di un malintenzionato.

Obiettivo

Il corretto funzionamento del sistema.

Pre-condizioni

Si ipotizza che il sistema abbia integrato al suo interno un algoritmo in grado di monitorare l'attività dell'utente anche in una fase successiva all'autenticazione primaria (ad esempio tramite i meccanismi trattati nel paragrafo 2.8). L'attore principale deve avere a sua disposizione delle credenziali d'accesso valide tramite il quale potersi autenticare sul dispositivo, attraverso uno scenario simile a quello visto nel paragrafo 5.2. Al contrario l'attaccante non è in possesso delle informazioni necessarie per accedere al sistema, né è in grado di ricavarle in un secondo momento.

Main Success Scenario

Analizziamo lo scenario attraverso due punti di vista differenti, quello del proprietario del dispositivo e quello dell'attaccante, attraverso lo schema 5.7:

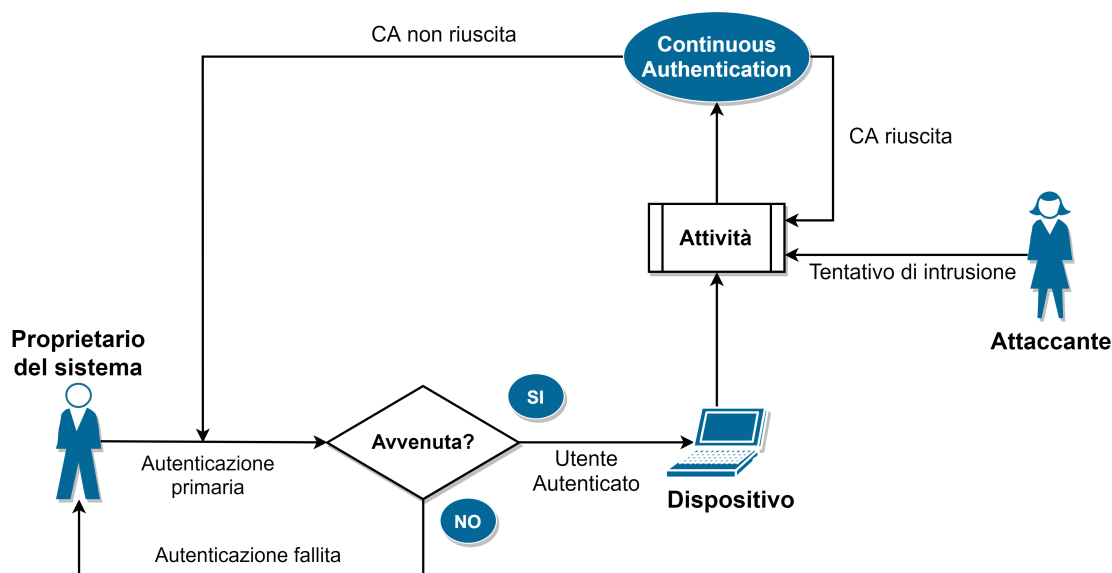


Figura 5.7. Diagramma dello scenario #6.

Utente

- L'utente, dopo essersi autenticato al dispositivo, inizierà a svolgere su di esso alcune attività.

- Attraverso un'analisi dei pattern utilizzati dall'utente per interagire con il dispositivo, viene verificata la sua attendibilità.
- Se durante l'analisi di CA non vengono rilevati dei problemi, l'utente può continuare ad interagire col sistema.
- In caso contrario un **trigger** blocca l'esecuzione dell'attività, richiedendo nuovamente l'autenticazione primaria all'utente che, dopo essersi nuovamente autenticato, potrà riprendere l'attività interrotta.

Malintenzionato

- L'attaccante, dopo avere rubato il dispositivo (già sbloccato) all'utente, inizierà a svolgere alcune attività con l'obiettivo di rubare dati sensibili alla vittima.
- Dopo un certo periodo, il sistema rileverà un'anomalia nei confronti dell'utilizzatore del dispositivo e lo bloccherà (**trigger**), richiedendo all'utente di autenticarsi.
- L'attaccante, non avendo a disposizione i dati necessari per autenticarsi, non potrà più accedere al dispositivo.

Estensioni

Di seguito l'elenco delle problematiche che si potrebbero presentare nell'MSS appena descritto:

- Il punto cardine della CA è la precisione con il quale vengono rilevate le anomalie. Un'accuratezza alta sarebbe in grado di rilevare con più facilità accessi illeciti, ma causerebbe anche un abbassamento dell'usabilità dell'utente che, non venendo riconosciuto dal sistema, sarebbe costretto ad autenticarsi nuovamente.
- Il tempo che il sistema necessita per rilevare la presenza di un malintenzionato, potrebbe non essere sufficiente ad impedire il verificarsi di azioni illecite.
- la CA non deve essere applicata nel caso in cui l'utilizzatore sia un individuo a cui il proprietario ha momentaneamente prestato il dispositivo.

Miglioramenti

Affinché la CA venga integrata opportunamente, è necessario implementare algoritmi che monitorano aspetti differenti dell'utilizzatore. Per rendere il sistema più preciso è inoltre possibile integrare tecniche di machine learning, in modo tale da avere un identikit del proprietario più attendibile.

Infine per risolvere la problematica del prestito, si potrebbe implementare l'idea dell'interfaccia guest [137], già discussa nel paragrafo 4.5.3.

5.7 Implementazione prototipale

Rispetto agli scenari evidenziati in questo capitolo, lo sviluppo del prototipo si baserà sul secondo di essi (paragrafo 5.2), ovvero l'utilizzo di un'autenticazione primaria con il fine di accedere a delle determinate risorse. Essa infatti permette di introdurre una maggiore flessibilità rispetto agli altri scenari, dando quindi la possibilità di intervenire in modo più mirato al miglioramento dell'usabilità generale del sistema. Questo modello verrà inoltre ampliato attraverso un'analisi di continuous authentication, simile a quello discusso nel paragrafo 5.6, con il quale fornire un ulteriore livello di sicurezza, senza però andare ad intaccare l'usabilità finale dell'utente.

Non verranno invece implementate tecniche sul controllo degli accessi e la condivisione delle risorse, come quelle evidenziate nei paragrafi 5.4 e 5.5, in quanto richiederebbero l'esistenza di un sistema di autenticazione già maturo ed integrato all'interno di un file system.

L'ambito di applicazione sarà di tipo Desktop e si baserà sul sistema operativo Windows 10.

Capitolo 6

Metodologie di valutazione del prototipo

Rispetto ai parametri di valutazione discussi nel paragrafo 4.6, andremo a vedere in questo capitolo quali sono le metodologie che verranno adottate per misurare l'usabilità del prototipo finale, di cui i dettagli saranno discussi nel capitolo successivo.

6.1 Struttura

Per poter valutare l'usabilità del sistema, è necessario analizzare tre aspetti differenti:

1. L'efficacia, che consiste nell'accuratezza con il quale gli utenti superano determinati task.
2. L'efficienza, che è il rapporto che intercorre tra l'efficacia ed il grado con il quale si è completato un determinato obiettivo, in relazione al tempo ed alle risorse utilizzate dall'utente per superarlo.
3. La soddisfazione percepita dall'utente nell'utilizzare il sistema.

Mentre i primi due parametri sono valutabili tramite tecniche analitiche, la soddisfazione è un fattore fortemente intrinseco nell'utente e, per questo motivo, misurabile unicamente attraverso metodologie soggettive.

Nello studio presentato da Frøkjær, durante il CHI Proceedings¹ [157], se l'analisi dell'usabilità non prende in considerazione l'insieme dei parametri che comprende efficacia, efficienza e soddisfazione dell'utente, si rischia di avere un indicatore finale non affidabile. Per questo motivo, la valutazione che verrà fatta sul prototipo

¹Conference on Human Factors in Computing Systems

cercherà di comprendere tutte e tre le proprietà, comprese eventuali analogie che le accomunano.

Di conseguenza, andremo a suddividere l'analisi in due fasi:

- Uno studio oggettivo, che viene analizzato direttamente tramite i dati raccolti durante l'esecuzione del tool e con il quale valuteremo l'efficacia e l'efficienza del sistema.
- Uno studio soggettivo, che viene analizzato tramite un questionario proposto al termine del test e con il quale verrà valutata la soddisfazione dell'utente.

6.2 Analisi oggettiva

Vediamo quali sono le metodologie di valutazione adottabili all'interno di un'analisi strettamente oggettiva. Alcuni dei risultati raccolti saranno inoltre in grado di fornire un'idea su alcuni parametri soggettivi che verranno poi approfonditi nell'analisi successiva.

6.2.1 Processo di registrazione

Un primo parametro fondamentale è il tempo richiesto dall'utente per completare la fase di registrazione, rilevabile tramite l'utilizzo di un timer. Esso, oltre a fornire un risultato concreto sul tempo medio richiesto da un utente per effettuare la registrazione, permette di avere un'idea su alcuni fattori soggettivi, come ad esempio l'affinità tra sistema e utente, e la complessità che esso potrebbe riscontrare nel capirne il funzionamento. Inoltre, per rendersi conto della familiarità percepita dell'utente per completare la fase di registrazione, si può andare a valutare la somma complessiva dei click richiesta per il completamento dell'attività.

6.2.2 Processo di autenticazione

In modo analogo alla misurazione precedente, viene utilizzato un timer per valutare il tempo richiesto dall'utente per effettuare il login. La presenza di eventuali fallimenti durante il processo di login non porta alla terminazione della valutazione, ma costituisce una parte integrante di essa. Una volta raccolti tutti i risultati sarà possibile capire qual è il tempo richiesto da un utente per autenticarsi, oltre ad avere un'idea soggettiva sullo sforzo cognitivo che viene richiesto all'utente per completare l'attività.

6.2.3 Completamento di specifici task

Un ulteriore parametro di valutazione, riguarda il completamento di operazioni secondarie, effettuabili dopo il superamento del sistema di autenticazione primaria.

Esso riguarda task come il cambio della password, dell'username, la simulazione di scrittura di un testo, la cancellazione del proprio account o il logout. Anche in questo caso, viene valutato il numero di click e degli eventuali fallimenti.

6.2.4 Completamento della simulazione continuous authentication

Tra i task proposti nel paragrafo precedente, la simulazione di scrittura di un testo (effettuato per il riconoscimento di keystroke dynamics), viene introdotta come attività obbligatoria all'interno dei test. Esso, oltre a verificare se l'utente viene correttamente identificato, permette di rilevare altre caratteristiche, come il tempo richiesto per il completamento dell'attività ed il tempo di digitazione, da cui è possibile ricavare il tempo che l'utente impiega per capire il funzionamento dell'interfaccia.

6.2.5 Numero di fallimenti

Durante il completamento di tutte le operazioni discusse in precedenza, viene valutato l'eventuale numero di fallimenti commesso dagli utenti. È inoltre da precisare che si può incorrere in un fallimento se un evento² interrompe l'esecuzione di un'attività in corso.

6.2.6 Log dettagliato

Tutte le operazioni effettuate dall'utente durante il test, vengono registrate all'interno di un log dettagliato. Esso permetterà di avere un'idea più chiara sui valori numerici registrati, oltre che comprendere meglio quale è stato l'orientamento dell'utente all'interno dell'interfaccia.

6.2.7 Logout

Il logout, oltre a chiudere la sessione corrente, permette di inviare tutti i dati raccolti in precedenza ad uno specifico indirizzo mail, in modo tale da poter essere consultati in un secondo momento. In quanto tale, la fase di logout è obbligatoria per tutti i partecipanti al test.

6.2.8 Interpretazione dei risultati

Al fine di capire meglio i valori ricavati al termine della simulazione, andrò a spiegare in questa sezione come vengono valutate le singole attività:

²Dove con evento si intende l'inizio della stessa attività o di una differente.

- **Registrazione:** il timer viene avviato alla pressione del tasto “SignUp”, presente nella prima schermata. Nessun tasto permette di annullare o stoppare il timer, se non il termine della procedura. Il numero di click che viene conteggiato, riguarda unicamente quelli effettuati nella schermata di registrazione. Si incorre in un fallimento se la password o l’username inseriti non soddisfano i requisiti minimi di accettazione. L’inserimento di un username già esistente non viene valutato come errore, in quanto l’utente non può sapere a priori qual è la lista degli username già memorizzati.
- **Login:** il timer viene avviato alla pressione del tasto “Login”, presente nella prima schermata. Nessun tasto permette di annullare o stoppare il timer, se non il termine della procedura. Il numero di click che viene conteggiato, riguarda unicamente quelli effettuati nella schermata di login. Si incorre in un fallimento se si inserisce una password o un username errato.
- **Cambio dell’username:** il timer viene avviato alla pressione del tasto “Change Username”, presente nella schermata Home. Nessun tasto permette di annullare o stoppare il timer, se non il termine della procedura. Il numero di click che viene conteggiato, riguarda tutti quelli che vengono effettuati nella schermata dell’attività e nella Home. Si incorre in un fallimento se si esce (intenzionalmente o non³) dalla schermata dell’attività prima che la medesima sia stata conclusa. Se l’utente inserisce un username che è già presente nel DB, esso viene segnalato, ma non provoca l’incremento del numero dei fallimenti relativi all’attività.
- **Eliminazione dell’account:** il timer viene avviato alla pressione del tasto “Delete Username”, presente nella schermata Home. Nessun tasto permette di annullare o stoppare il timer, se non il termine della procedura. Il numero di click che viene conteggiato, riguarda tutti quelli che vengono effettuati nella schermata dell’attività e nella Home. Si incorre in un fallimento se si esce (intenzionalmente o non⁴) dalla schermata dell’attività prima che la medesima sia stata conclusa. Quando l’attività viene completata, il sistema effettua automaticamente il logout (vedi [A.10.5](#)) e, di conseguenza, tutte le azioni legate ad esso.
- **Cambio della password:** il timer viene avviato alla pressione del tasto “Change Password”, presente nella schermata Home. Nessun tasto permette di annullare o stoppare il timer, se non il termine della procedura. Il numero di click che viene conteggiato, riguarda tutti quelli che vengono effettuati nella schermata dell’attività e nella Home. Si incorre in un fallimento se si esce (intenzionalmente o non³) dalla schermata dell’attività prima che la medesima sia stata conclusa o se la nuova password non soddisfa i requisiti minimi di accettazione.

³La pressione del tasto ‘CANCEL’ all’interno della schermata dell’attività, incrementa di 1 il numero di fallimenti.

⁴La pressione del tasto ‘NO’ all’interno della schermata dell’attività, incrementa di 1 il numero di fallimenti.

- **Simulazione di KD:** il timer viene avviato alla pressione del tasto “SIMULATION”, presente nella schermata Home. Nessun tasto permette di annullare o stoppare il timer, se non il termine della procedura. Il numero di click che viene conteggiato, riguarda unicamente quelli effettuati nella schermata di simulazione. Si incorre in un fallimento se non viene verificato un match tra il testo inserito durante la simulazione e quello creato in fase di registrazione.

La struttura utilizzata per raccogliere i dati è un oggetto JSON mentre, la chiave che viene utilizzata per raccogliere il singolo dato, è composta dalla seguente sintassi:

`<nomeAttività>_<parametroMisurato>_<counter>`

L'introduzione di un counter permette di poter rilevare anche la stessa attività, ripetuta più volte, all'interno della medesima simulazione.

6.3 Analisi soggettiva

Rispetto alle metodologie oggettive evidenziate in precedenza, l'applicazione di una valutazione soggettiva dell'usabilità risulta molto difficile da applicare, in quanto:

- La misurazione non può avvenire in modo puntuale, come accaduto in precedenza.
- All'interno di un questionario, gli utenti tendono a non applicare un processo decisionale basato sul modello di Bayesian [158], in cui tutte le decisioni vengono prese con l'obiettivo di dare una soluzione ottimale, ma utilizzano un modello più semplicistico, con il quale si cerca di fornire, nel minor tempo possibile, una soluzione convincente.
- Jacob Nielsen consiglia di focalizzare maggiormente l'attenzione su quello che gli utenti fanno, piuttosto che su quello che dicono [159], in quanto:
 - Tendono a parlare bene del sistema che stanno valutando, piuttosto che criticarlo
 - Spesso non ricordano quello che hanno fatto nella fase precedente al test
 - Sono inclini a razionalizzare il proprio comportamento

Per questo motivo, bisogna accettare che nessun metodo soggettivo sarà mai in grado di fornire una valutazione esatta, dal momento che è impossibile rimuovere in esso il fattore umano. Detto ciò, è possibile utilizzare alcune tecniche ed accortezze sulle modalità con il quale vengono condotte le analisi, con l'obiettivo di ottenere, complessivamente, una valutazione più accurata.

Vedremo quindi ora quale sarà la metodologia che verrà utilizzata all'interno dei test soggettivi.

6.3.1 Parte 1: Spiegazione

Prima che l'utente possa iniziare a provare il prototipo, è necessario accertarsi che abbia appreso appieno il funzionamento del sistema di autenticazione. Questo viene fatto attraverso una breve introduzione (vocale o scritta) in cui vengono spiegate le varie attività di cui sarà composto il test. In questa fase bisogna però stare attenti a non divulgare informazioni sul “come” le azioni debbano essere completate, per evitare che i risultati finali possano essere condizionati.

Al termine della spiegazione verrà poi chiesto se tutte le informazioni fornite siano state chiare e se ci fossero dei dubbi prima di iniziare; dopo quest'ultima domanda non verrà fornita più alcun'altra chiarificazione durante lo svolgimento del test⁵.

6.3.2 Parte 2: Pre-test

Dopo che l'utente ha chiaro in mente come è strutturata la simulazione, è necessario inquadrare quali siano le conoscenze di quest'ultimo sui sistemi di autenticazione e, più in generale, sul mondo informatico. Questo tipo di analisi verrà condotta tramite un breve test che permetterà di posizionare l'utente all'interno di tre categorie differenti:

- **Gruppo A**, che racchiude tutti coloro che lavorano o studiano nel campo dell'informatica, e che quindi conoscono bene l'ambiente dei sistemi di autenticazione.
- **Gruppo B**, che racchiude chi se la cava con l'uso del computer e possiede una conoscenza informatica elementare.
- **Gruppo C**, che racchiude tutti coloro che hanno una bassa esperienza con il mondo informatico e che potrebbero non conoscere il significato di “sistema di autenticazione”.

Questo tipo di categorizzazione è fondamentale per capire appieno i risultati dei test: ci aspetteremo infatti che gli utenti del gruppo A ottengano risultati migliori a quelli del gruppo C, non a causa del sistema di autenticazione in sé, ma da una differente esperienza avuta in questo settore.

Le domande che verranno proposte in questo breve pre-test saranno:

1. Prima dell'inizio di questo test, conoscevi il significato di “sistema di autenticazione” e “sistema di autorizzazione”? [SI/NO]

In caso affermativo, che sistemi conosci?

⁵Esclusi chiarimenti di natura tecnica, come ad esempio un malfunzionamento del programma durante la fase di simulazione.

2. Utilizzi o hai mai utilizzato, in tutta la sua completezza, un sistema operativo basato su UNIX, differente da quello di Apple? [SI/NO]

In caso affermativo, che distribuzioni hai utilizzato?

3. Conosci e sai utilizzare almeno un linguaggio di programmazione? [SI/NO]

In caso affermativo, che linguaggi conosci?

4. Utilizzi o hai mai utilizzato almeno 3 programmi della suite Office di Microsoft? [SI/NO]

In caso affermativo, che programmi utilizzi/hai utilizzato?

Considerando il fatto che per ogni risposta affermativa viene aggiunto un punto, si possono ottenere i seguenti risultati:

- 4 punti → Gruppo A
- 2-3 punti → Gruppo B
- 0-1 punto → Gruppo C

6.3.3 Parte 3: Questionario finale

Infine, per analizzare il grado di soddisfazione degli utenti, verrà proposto un questionario, al termine del test pratico, composto da una serie di affermazioni. Il metodo di valutazione scelto per valutare i quesiti è la scala di Likert, dal momento che rappresenta un giusto compromesso tra affidabilità, usabilità e velocità di completamento del test [160]. Essa, inoltre, permette di avere risultati più precisi rispetto ad uno slider classico, dove invece viene riscontrata una maggiore difficoltà nel selezionare un valore puntuale [161]. La scala è composta dai seguenti parametri di valutazione:

1. Per niente d'accordo
2. Disaccordo
3. Indeciso
4. D'accordo
5. Molto d'accordo

Il modello di Likert, seppur presenti un'interfaccia e un'usabilità elevata, può soffrire di alcune problematiche: ad esempio, come viene discusso anche nell'articolo "Effects of Acquiescent Response Set on Patient Satisfaction Ratings" [162], se le domande vengono presentate sotto forma di batteria, la tendenza dell'utente sarà quella di fornire sempre la stessa risposta, indipendentemente dal contenuto del quesito. Questo fenomeno è chiamato "response set" ed un modo per evitarlo è quello di ridurre il numero di domande che vengono proposte all'interno dello stesso gruppo.

Vediamo ora come sarà composto il questionario di valutazione:

1. **Prima batteria** - Ho trovato accettabile il tempo richiesto per:
 - a. imparare il funzionamento del sistema
 - b. completare la fase di registrazione
 - c. completare la fase di login
 - d. completare la simulazione di keystroke dynamics
 - e. completare le attività secondarie⁶
2. **Seconda batteria** - Ho trovato soddisfacente utilizzare:
 - a. il sistema primario di autenticazione
 - b. il meccanismo di keystroke dynamics
3. **Terza batteria** - Il grado di sicurezza percepita
 - a. dal sistema di autenticazione è elevato
 - b. dal sistema di riconoscimento di keystroke dynamics è elevato
 - c. durante lo svolgimento delle attività secondarie è elevato
4. **Quarta batteria** - Il grado di facilità d'utilizzo
 - a. del sistema di autenticazione è elevato
 - b. del sistema di riconoscimento di keystroke dynamics è elevato
 - c. delle attività secondarie è elevato
5. **Quinta batteria** - Ho trovato familiare l'ambiente utilizzato per:
 - a. registrarmi
 - b. effettuare il login
 - c. effettuare l'attività di keystroke dynamics
 - d. svolgere le attività secondarie

Oltre a queste batterie di domande a risposta chiusa, ne proporrò una serie facoltative a risposta aperta, che permetterà di raccogliere qualche dettaglio in più sull'esperienza vissuta dall'utente durante il test:

1. Hai incontrato particolari difficoltà nell'utilizzare il sistema? [SI/NO]
Se sì, in quali frangenti?
2. Utilizzeresti questo meccanismo per proteggere i tuoi dispositivi o i tuoi dati?
Motiva la risposta.

⁶Cambio password, cambio username, eliminazione dell'account.

3. Il fatto che il sistema sia in grado di riconoscerti in base al modo in cui digiti, lo ritieni un fattore:

- a. sicuro. [SI/NO]
- b. usabile. [SI/NO]

Motiva le tue risposte.

4. Ulteriori commenti liberi.

Capitolo 7

Prototipo

La prima parte di questo capitolo verterà sull'analisi dei tool da cui è iniziato lo sviluppo del sistema di autenticazione. Verranno poi presentati, nell'ultima parte, tutti i miglioramenti e le implementazioni che hanno permesso di incrementare la sicurezza e l'usabilità di quello che è poi diventato a tutti gli effetti il prototipo finale. Per un'analisi più approfondita sulla struttura e sulle tecniche utilizzate per la realizzazione del sistema, rimando alla lettura del manuale del programmatore (vedi appendice [A](#)).

7.1 Analisi dei Tool

Rispetto agli scenari #2 (paragrafo [5.2](#)) e #6 (paragrafo [5.6](#)) del capitolo precedente, il prototipo sarà diviso in due parti:

- Mutating Graphical Password System, un sistema grafico con funzionalità anti-shoulder surfing, che verrà utilizzato per l'autenticazione primaria.
- Keystroke dynamics with typing DNA recorder, un sistema di continuous authentication che verrà impiegato nella fase successiva all'autenticazione primaria.

Entrambi i tool sono scritti in Java e utilizzano le librerie grafiche AWT¹, permettendo una perfetta interoperabilità tra le due parti.

7.2 Tool 1: Mutating Graphical Password System (graphical password)

MGPS [[163](#)] è un sistema di autenticazione che prende spunto dal pattern lock, e che si basa sull'inserimento di segmenti all'interno di uno schema puntato di dimensioni 8x8. Abbiamo visto nel paragrafo [2.4](#), come questa categoria di sistemi sia

¹Abstract Window Toolkit, una libreria di Java composta da un insieme di classi e interfacce con il quale è possibile realizzare applicazioni GUI.

fortemente incline a problematiche di shoulder surfing e smudge attack, in quanto un semplice sguardo durante l’inserimento (o anche nell’istante successivo per i dispositivi touch) potrebbe compromettere completamente l’efficacia del sistema. Per correggere queste mancanze, vengono inserite delle nuove funzionalità all’interno della schermata di login del tool, con il quale è possibile combattere le criticità appena descritte:

- Durante l’autenticazione, l’utente non deve immettere la sequenza tramite l’apposita griglia (come fatto invece durante il processo di registrazione), ma inserendo le coordinate relative agli estremi dei segmenti che dovrebbero essere tracciati. Ad esempio, nella griglia 7.1, per inserire il tratto in rosso, l’utente dovrà immettere le coordinate relative: T2 - G5 - H9. Non essendoci un’interazione diretta tra utente e griglia, vengono come prima cosa sventati eventuali smudge attack.
- Se le coordinate presenti nelle celle della griglia fossero statiche (ad esempio come mostrato nella figura 7.2), un malintenzionato potrebbe risalire al pattern corretto attraverso una fotografia, andando poi successivamente a vedere gli input immessi dall’utente. Per eliminare questa minaccia, dopo ogni inserimento, il contenuto delle celle cambia, rendendo del tutto inutili le combinazioni mostrate in precedenza (7.1). Le coordinate inserite dall’utente saranno quindi relative alla specifica griglia, mostrata in un determinato istante; in questo modo per l’attaccante sarà molto più difficile avviare manovre di shoulder surfing.

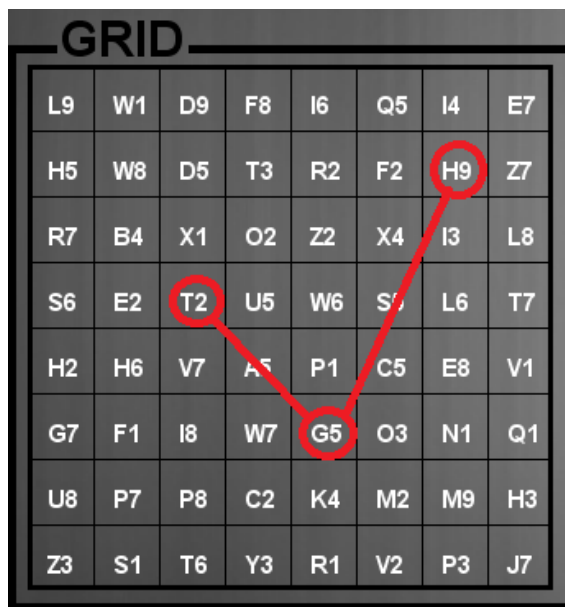


Figura 7.1. Esempio di inserimento delle coordinate statico.



Figura 7.2. Esempio di inserimento delle coordinate dinamico.

7.2.1 Creazione della password

Una volta che l'utente ha immesso la sequenza, se essa avrà una dimensione inferiore a 3 punti, il sistema la rifiuterà chiedendo di inserirla nuovamente, ma non andando a rimuovere le coordinate precedentemente inserite. Questo rappresenta l'unico vincolo che il sistema richiede durante l'inserimento della password.

Una volta che la sequenza viene accettata, prima di essere inserita all'interno del DB, viene prelevata la prima coordinata e ricavato il relativo hash; questo procedimento viene iterato fino al raggiungimento della sequenza di hash finale che verrà poi confrontata in fase di login. A questo punto l'informazione viene memorizzata all'interno del DB che, in questo caso, viene realizzato con un file locale con estensione txt, in cui i dati contenuti vengono opportunamente cifrati (per i dettagli implementativi vedi il paragrafo 7.4).

7.2.2 Verifica della password

L'interfaccia di login è composta dalla griglia principale e da due form in cui devono inseriti l'username e la password con cui si è registrati; sul fondo sono inoltre presenti due pulsanti con il quale è possibile confermare o cancellare ciò che è stato inserito. Come accennato in precedenza, l'utente non deve toccare i punti sulla griglia, ma inserire le coordinate che mutano dinamicamente sullo schermo. Se l'utente non digita nulla per un tempo pari a 5 secondi, la griglia cambierà i suoi valori anche se non è stata immessa alcuna coordinata. Dopo che sono stati immessi tutti i valori, viene ricavato l'hash finale che sarà confrontato con il sample costruito in fase di registrazione. Se il matching ha esito positivo l'utente viene autenticato e reindirizzato verso la sua home, altrimenti gli viene richiesto di inserire nuovamente la sequenza.

7.2.3 Home

Una volta che l'utente si è registrato, esso entrerà nella sua home, tramite il quale potrà:

- Cambiare username
- Cambiare la password
- Eliminare l'account
- Fare il logout
- Ritornare alla schermata home

7.3 Tool 2: typingDNA (keystroke dynamics)

Il tool di keystroke dynamics è formato quasi unicamente dall'algoritmo nudo e crudo, senza una struttura di contorno solida, come accade invece nel caso precedente. Il sito [164] fornisce una demo interattiva con il quale poter provare le funzionalità dell'algoritmo mentre, il repository di github [165], mette a disposizione le relative API. Esse sono compatibili sia con javascript, per un utilizzo web service oriented, che con java, con il quale è possibile realizzare delle implementazioni locali. Rispetto a quest'ultimo caso, le API permettono un adattamento con JavaFX o con AWT di JavaSE.

7.3.1 API

Tra i metodi java disponibili abbiamo²:

- **TypingDNARecorder.getTypingPattern(optionsObject)**, è la funzione principale; ritorna una stringa associato al pattern misurato durante la digitazione e richiede come input i seguenti parametri:
 - *type* (int) → Il parametro permette di indicare il tipo di analisi che deve essere condotta: 0 per testi casuali di 120-180 caratteri, 1 per testi brevi specifici e 2 per introdurre una migliore accuratezza su quest'ultimi.
 - *length* (int) → Parametro opzionale che permette di indicare la lunghezza dei caratteri su cui si vuole applicare l'algoritmo; 0 per ignorare.
 - *text* (String) → Solo se type=1, permette di specificare il testo di confronto.
 - *textId* (int) → Opzionale e solo per type=1, permette di definire un ID personalizzato per il testo digitato. Se vale 0, il parametro viene ignorato.
 - *caseSensitive* (boolean) → Opzionale e solo per type=1, permette di indicare se l'analisi su testi specifici deve essere case sensitive o meno.

²Le funzioni sono tutte statiche e vengono quindi richiamate direttamente dalla classe TypingDNARecorder.

- **TypingDNARecorder.reset()**, permette di cancellare tutte le misurazioni rilevate fino a quell'istante, dal relativo stack.
- **TypingDNARecorder.start()**, utilizzato per avviare la misurazione. Esso viene richiamato automaticamente in fase di inizializzazione, ma può essere impiegato anche manualmente per riavviare la misurazione dopo uno stop().
- **TypingDNARecorder.stop()**, utilizzato per terminare la misurazione.

È inoltre da segnalare che per effettuare il confronto tra due pattern viene eseguita una chiamata POST all'indirizzo "https://api.typingdna.com/match", tramite un webservice che viene messo a disposizione dagli stessi autori del tool [166]; la risposta, fornita in formato JSON, permetterà di capire se esiste o meno una corrispondenza tra i due pattern. Il meccanismo utilizzato dall'API non è pubblico, ma questo si tratta di un aspetto di secondaria importanza dal momento che, l'obiettivo della tesi, rimane quello di valutare l'usabilità dei fruitori del sistema.

7.3.2 Analisi del pattern inserito

Il tipo di input immesso dall'utente può riguardare la scrittura di un testo specifico o di uno qualsiasi; in base al tipo di analisi scelta deve essere costruito un diverso approccio di testing.

Testo uguale

In questo caso l'utente viene autenticato in base allo stesso testo utilizzato in fase di registrazione. La valutazione viene fatta sia sulle caratteristiche di keystroke dynamics, che sulla corretta corrispondenza delle informazioni inserite. È da notare che il sistema può accettare testi compresi da 4 a centinaia di caratteri, ma che l'accuratezza è fortemente dipendente dalla lunghezza del testo inserito: input troppo brevi potrebbero quindi innalzare il valore di FAR e FRR.

Vediamo ora alcuni casi d'uso in cui è possibile integrare questo tipo di approccio:

- Email e password, composto in media da 30 caratteri (usato per il login).
- Username e password, composto in media da un minimo di 15 caratteri (usato per il login).
- Email, composto in media da un minimo di 15 caratteri (usato per il recupero di password).
- Dati della carta di credito e nome del proprietario, composto tipicamente da almeno 30 caratteri (usato per acquisti online).
- Analisi focalizzata unicamente sul KD, tramite la scrittura di un testo breve fissato dal sistema³.

³Per un corretto riconoscimento, l'input finale immesso dall'utente deve essere necessariamente uguale a quello proposto dal sistema.

Testo casuale

Un altro modo per valutare l'input dell'utente, può essere fatto tramite l'inserimento di testi casuali. In questo caso la correttezza del testo inserito rispetto a quello proposto dal sistema, non pregiudica la riuscita dell'autenticazione, in quanto viene valutato unicamente la parte relativa al KD.

Possono infine essere applicati una serie di accorgimenti, al fine di rendere la rilevazione più precisa:

- Evitare di usare lo stesso testo sia per la fase di registrazione che per quella di login. Un individuo infatti, sapendo già qual è l'input che deve inserire, potrebbe involontariamente velocizzare il suo modo di scrivere e non essere riconosciuto dal sistema.
- È buona norma richiedere in fase di registrazione la scrittura di due testi, in modo tale che il sample di confronto finale risulti più attendibile.
- Per i testi complessi composti da 15/20 caratteri differenti, se la loro frequenza di apparizione si avvicina quanto più possibile ad essere equiprobabile, la precisione del profilo utente potrebbe migliorare.
- Nella creazione del testo, l'utilizzo di caratteri appartenenti alla riga inferiore della tastiera, potrebbe riprodurre un sample di valutazione più preciso [164].
- Come già detto prima, l'utilizzo di testi più lunghi permette una rilevazione più precisa. In particolare è consigliato utilizzare dai 160 ai 200 caratteri per la fase di registrazione e dai 120 ai 140 per quella di login.
- È meglio focalizzare l'analisi sulla digitazione di caratteri minuscoli, anziché numeri o simboli.
- È consigliabile utilizzare dei testi con una media di 15 caratteri diversi, anche se la lunghezza complessiva risulta più lunga di quella raccomandata.

7.4 Costruzione del prototipo

Rispetto ai due tool appena analizzati, sono state definite alcune scelte implementative:

- Per quanto riguarda i dati username/pattern, riferiti al sistema 7.3, la memorizzazione della sola sequenza di hash non è sufficiente. Durante la fase di simulazione, infatti, sarà necessario estrarre la stringa originale, al fine di poterla comparare con quella ricavata durante la fase finale del test. Per questo motivo è stata creata una classe "MyCipher.java" (all'interno del package "kd") che si occupa di effettuare la cifratura e la decifratura dei dati.

L'algoritmo utilizzato è AES-128-CBC [167].

- Anche per la memorizzazione delle credenziali ho deciso di uniformarmi al meccanismo di cui ho parlato sopra.
- Vengono utilizzati in tutto 3 file per registrare le credenziali degli utenti:
 - 1 file principale per username e relativa sequenza
 - 1 file principale per username e pattern KD
 - 1 file temporaneo per trasmettere informazioni relative all'algoritmo di KD tra il thread principale e l'Event Dispatching Thread (EWT) di AWT

7.5 Miglioramenti applicati

A partire dal prototipo grezzo, vediamo quindi quali sono le migliorie presenti nel programma finale:

- Il tempo di generazione delle coordinate all'interno della griglia era inizialmente di 5 secondi, anche se non viene effettuata nessuna azione dall'utente. Questo tempo può risultare troppo basso e generare, in alcuni casi, frustrazione nell'utilizzatore. Per questo motivo ho deciso di prolungare il tempo da 5 a 15 secondi, in modo tale da fornire una finestra temporale più ampia in cui l'utente può applicare la sua scelta. Oltre questo, la griglia cambia i valori delle coordinate se:
 - Viene cancellata uno o più delle coordinate che sono state precedentemente inserite.
 - Viene inserito una coordinata che non è presente tra quelle nella griglia.
- La griglia e la dimensione dei caratteri, con il quale vengono rappresentate le coordinate, hanno ora una dimensione più grande. I caratteri presentano inoltre un font più snello, al fine di migliorarne la visibilità durante la lettura.
- Ridotto il numero della password minima richiesta durante la fase di registrazione da sei a tre. Sono di conseguenza variati anche i messaggi di segnalazione rispetto alla lunghezza della password:
 - 0-2, weak: la password inserita risulta troppo breve e, per questo motivo, non viene considerata valida.
 - 3-5, good: la lunghezza della password soddisfa i requisiti minimi per essere accettata dal sistema.
 - 6+, strong: la password, oltre a rispettare i requisiti minimi, ha una lunghezza tale da venire considerata "robusta".

Riguardo quest'ultimo aspetto, il termine "robusta" indica l'alta resistenza che questo meccanismo ha nei confronti dei brute force attack. Se andiamo infatti a considerare una password di lunghezza pari a 3 (requisito minimo di accettazione), il numero di combinazioni assume un valore estremamente

elevato, in quanto ogni singola coordinata viene generata a partire da una lettera compresa tra [A-Z] e una cifra [1-9]:

$$C_1 = 26 \cdot 10 = 260 \quad (7.1)$$

Se poi applichiamo questo risultato all'esempio precedente, otteniamo che il numero di combinazioni totale risulta essere pari a:

$$C_3 = 26 \cdot 26 \cdot 26 = 17576 \quad (7.2)$$

In merito a tutto questo, una riduzione del requisito minimo di validità della password, non reca nessun danno in termini di sicurezza, ma permette di aumentare l'usabilità dell'utente durante l'inserimento in fase di login.

- Durante la fase di login, l'utente può confermare i dati inseriti anche tramite la pressione del tasto “enter”⁴.
- Durante l'attività di cambio dell'username, l'utente può confermare i dati inseriti anche tramite la pressione del tasto “enter”⁴.
- Durante la fase di registrazione, l'utente può confermare i dati inseriti anche tramite la pressione del tasto “enter”⁴.
- Ora, al termine della fase di simulazione di keystroke dynamics, apparirà nella home un apposito pulsante tramite il quale visionare il risultato del test.
- L'inserimento di due coordinate durante la fase di login, saranno intervallate da una stringa separatrice, al fine di migliorare la visibilità dell'input inserito.
- Sono state ingrandite le finestre relative al processo di KD.
- Realizzata una miglior interfaccia grafica per il pannello di KD, sia per la parte di registrazione che per quella di simulazione.
- Inserita la possibilità di scegliere tra italiano e inglese per la parte di simulazione di KD.
- Risolto un problema presente nel tool originale riguardante il sistema di autenticazione primaria. L'operazione di logout, se confermata, riportava alla schermata iniziale di welcome, ma se da essa si ritentava il login, il processo si bloccava. Per risolvere questa problematica ho imposto che l'operazione di logout chiudesse definitivamente tutti i processi, evitando in questo modo che si potessero verificare problemi di sincronizzazione.
- Risolto un problema presente nel tool originale riguardante il sistema di autenticazione primaria. Al termine dell'operazione di cancellazione dell'utente,

⁴Inizialmente solo l'apposito pulsante “OK” permetteva la convalida dei dati.

all'interno della schermata Home, tutte le finestre vengono chiuse ma il processo principale rimane ancora attivo (senza però esserci la possibilità di effettuare alcuna operazione). Ho corretto questa problematica andando a forzare la chiusura del processo.

- Risolto un problema presente nel tool originale riguardante il conteggio dei punti inseriti all'interno della griglia. Nel tool originale, il contatore adibito a questa funzione non veniva mai azzerato e, di conseguenza, qualsiasi ripetizione dell'attività di registrazione e cambio della password, mostrava un valore errato.
- Il tool originale del sistema di autenticazione primario, presenta un grave bug: ogni volta che dalla home si tenta di cambiare la password o di cancellare il profilo utente, vengono automaticamente rimossi i dati in locale di tutti gli utenti. Per risolvere questa problematica ho dovuto ricostruire completamente il meccanismo interno di memorizzazione dei dati, andando in particolare ad intervenire su:
 - sistema di salvataggio dei dati in fase di registrazione
 - sistema di comparazione dei dati in fase di login
 - cambio della password
 - eliminazione dei dati associati all'utente, assieme ai dati relativi al pattern di digitazione
 - cambio dell'username sia sul file contenente le credenziali, che su quello contenente i pattern di KD

7.6 Conclusione

L'insieme delle migliorie appena trattate ha permesso di realizzare un prototipo finale più concreto ed usabile, che è stato testato tramite il modello visto in precedenza e di cui i risultati verranno discussi nel capitolo successivo. A partire da esso è stato poi creato un prototipo di autenticazione avanzato, che contiene al suo interno una simulazione sempre basata sul riconoscimento di keystroke dynamics, applicato però ad uno scenario di utilizzo reale. Questo sistema verrà presentato successivamente nell'appendice C, ma non subirà un processo di valutazione come è accaduto invece per il prototipo originale.

Capitolo 8

Risultati

Dopo circa un mese di simulazioni, sono stati raccolti un totale di 22 campioni, suddivisi in:

- 9 utenti per il gruppo A.
- 5 utenti per il gruppo B.
- 8 utenti per il gruppo C.

A tutti i partecipanti è stato proposto il modello discusso nel capitolo [6](#), composto da un sistema di tracciamento passivo per rilevare le statistiche oggettive e un questionario finale con il quale è stata valutata la percezione soggettiva. Quello che vedremo in questo capitolo riguarda unicamente i dati aggregati mentre, per quanto riguarda i dati grezzi dei singoli individui, rimando alla lettura dell'appendice [D](#).

8.1 Analisi dei dati oggettivi

L'analisi dei dati oggettivi verrà eseguita su ognuna delle singole attività completate dall'utente:

- Registrazione
- Login
- Attività secondarie, composte da:
 - Cambio username
 - Cambio password
- Simulazione di keystroke dynamics

8.1.1 Registrazione

L'attività di registrazione è caratterizzata da due fattori, ognuno dei quali può influire negativamente o positivamente sui risultati finali:

- La prima parte è composta dalla scelta dell'username e della password grafica, dove una sequenza più corta richiederà meno tempo ad essere scritta e memorizzata rispetto ad una più lunga.
- La seconda parte è la scrittura del testo per il riconoscimento di KD, dove il tempo di completamento dipende quasi unicamente dall'abilità che ha l'utente nella scrittura con tastiera.

A partire da questa premessa, nei grafici 8.1 e 8.2 vengono presentati i tempi ed il numero di click richiesti da ogni singolo utente per completare la fase di registrazione.

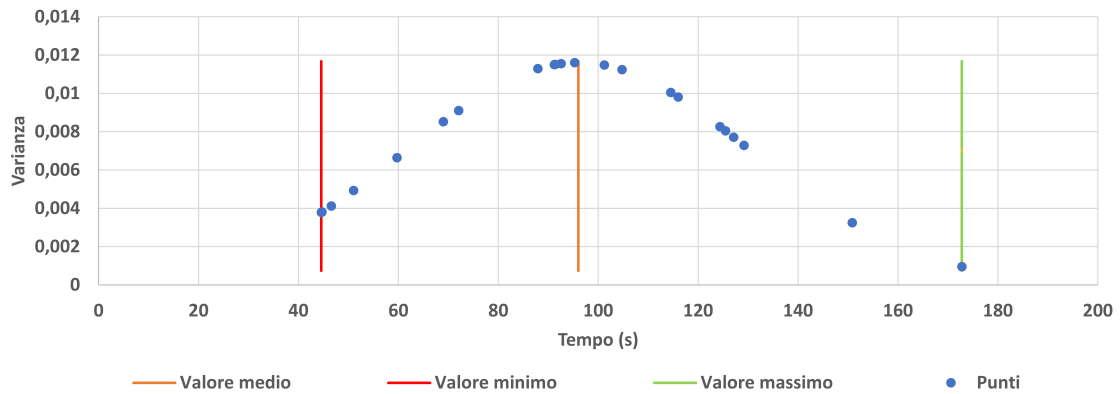


Figura 8.1. Grafico rappresentante le tempistiche richieste da ogni singolo utente per completare la fase di registrazione.

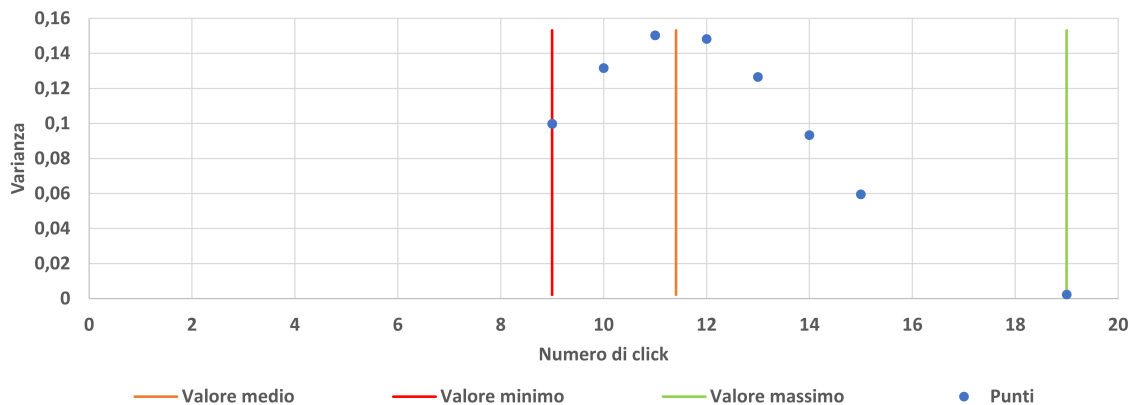


Figura 8.2. Grafico rappresentante il numero di click richiesti dai singoli utenti per completare la fase di registrazione.

Un primo riscontro interessante riguarda la relazione tra il risultato ottenuto ed il gruppo di appartenenza dell'utente che ha completato il test. Come mostrato nel

grafico 8.3, gli appartenenti al gruppo A hanno terminato la fase di registrazione in tempi mediamente lunghi, nonostante siano stati riconosciuti come utenti esperti (in particolare, cinque di loro hanno impiegato più di 92 secondi per completare l'attività).

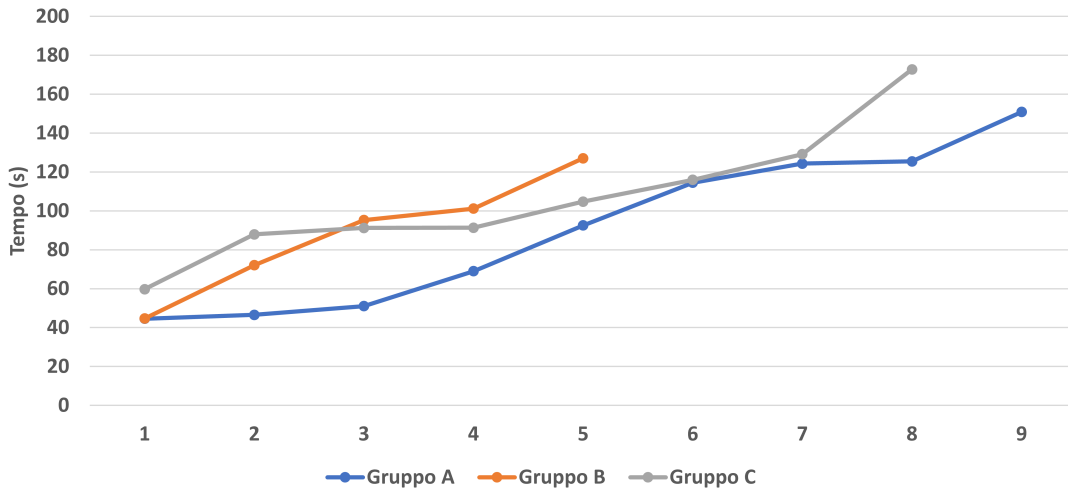


Figura 8.3. Comparazione tra il tempo richiesto per completare la fase di registrazione rispetto al gruppo di appartenenza del partecipante.

Il motivo per il quale alcuni utenti esperti hanno ricevuto delle tempistiche negative è da attribuire al differente approccio che hanno avuto nel completare la simulazione: tre di essi infatti hanno scelto di utilizzare una password più robusta (composta da 5 o 6 punti), che ha richiesto un tempo di scelta, digitazione e memorizzazione maggiore rispetto ad una sequenza di lunghezza minima. Alcuni dei partecipanti, inoltre, ha deciso di utilizzare l'opzione “show password”, per rivedere la sequenza appena immessa e quindi sfruttare appieno le funzionalità messe a disposizione dall'interfaccia di registrazione. Rispetto a questi risultati, il numero di click effettuati dai partecipanti dagli utenti durante il test rimane in linea con le scelte da loro effettuate, non mostrando alcun risultato incoerente.

Per quanto riguarda gli utenti del gruppo B e C, i risultati negativi derivano prevalentemente dai lunghi tempi di scrittura relativa alla parte di KD.

Infine, secondo i vincoli definiti nel paragrafo 6.2.8, è stato rilevato un unico fallimento tra i 22 partecipanti, causato da un utente che si è dimenticato di inserire l'username all'interno dell'apposito campo.

8.1.2 Login

All'interno della fase di autenticazione sono stati riscontrati un totale di quattro fallimenti, causati da un inserimento errato della password. Questo risultato è determinato da una iniziale difficoltà che alcuni partecipanti hanno constatato nel comprendere il meccanismo di inserimento della sequenza¹.

¹Vai all'analisi soggettiva, paragrafo 8.2, per maggiori dettagli.

Si può invece notare una certa coerenza sul periodo di completamento del processo di autenticazione, per tutti coloro che non hanno commesso errori. Al contrario, i 4 utenti che hanno fallito l'inserimento al primo tentativo, hanno totalizzato le tempistiche peggiori, come mostrato nel grafico 8.4.

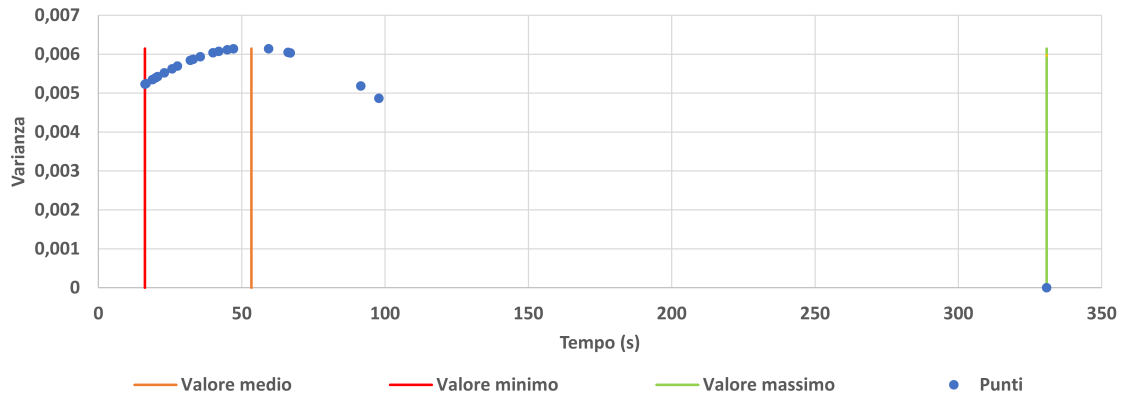


Figura 8.4. Grafico rappresentante le tempistiche richieste da ogni singolo utente per completare la fase di autenticazione.

Anche per quanto riguarda il numero di click effettuati (vedi grafico 8.5), gli utenti che hanno fallito il primo inserimento hanno ottenuto i valori più elevati.

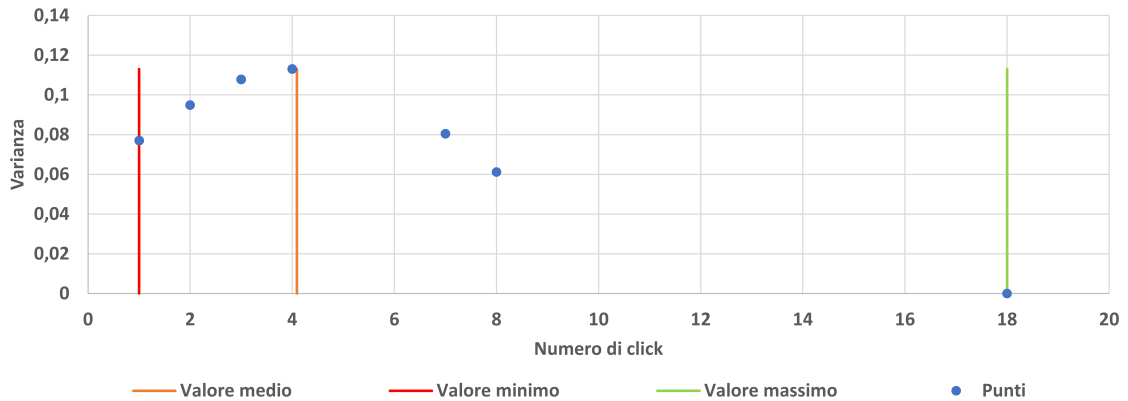


Figura 8.5. Grafico rappresentante il numero di click richiesti dai singoli utenti per completare la fase di autenticazione.

Se si va infine a comparare i tempi di registrazione con quelli di autenticazione (vedi grafico 8.6), si evince come l'andamento delle due linee segua la stessa tendenza². Questo permette di dimostrare come le due fasi siano fortemente legate tra di loro dal momento che, chi ha scelto una password più lunga, ha mediamente impiegato più tempo a completare sia la registrazione che l'autenticazione.

²Andando ad escludere le possibili incertezze derivanti da coloro che hanno fallito questa parte di test.

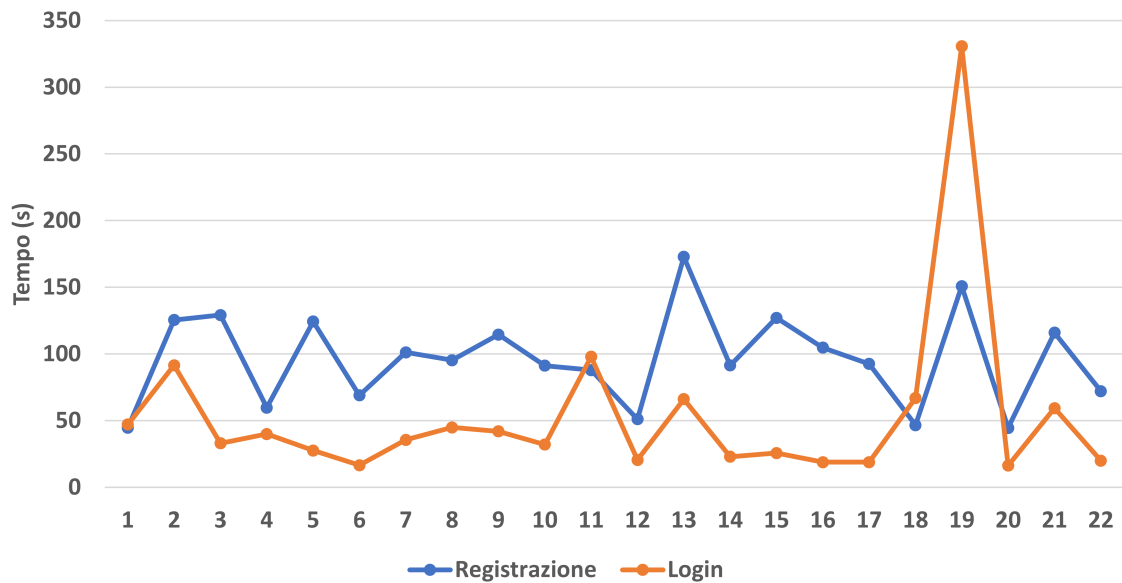


Figura 8.6. Comparazione tra i tempi richiesti per completare la fase di registrazione e quella di autenticazione.

8.1.3 Attività secondarie

Per quanto riguarda il completamento delle attività secondarie, si può notare come non vi sia nessun punto d'intersezione tra le due linee presenti nel grafico 8.7. Risulta quindi evidente come l'attività di cambio della password richieda più tempo per essere completata rispetto a quella di modifica dell'username. Inoltre, il tempo che intercorre tra le due linee assume in media lo stesso valore, mostrando quindi una sorta di coerenza nella difficoltà riscontrata per terminare le due attività rispetto allo svolgimento delle singole simulazioni.

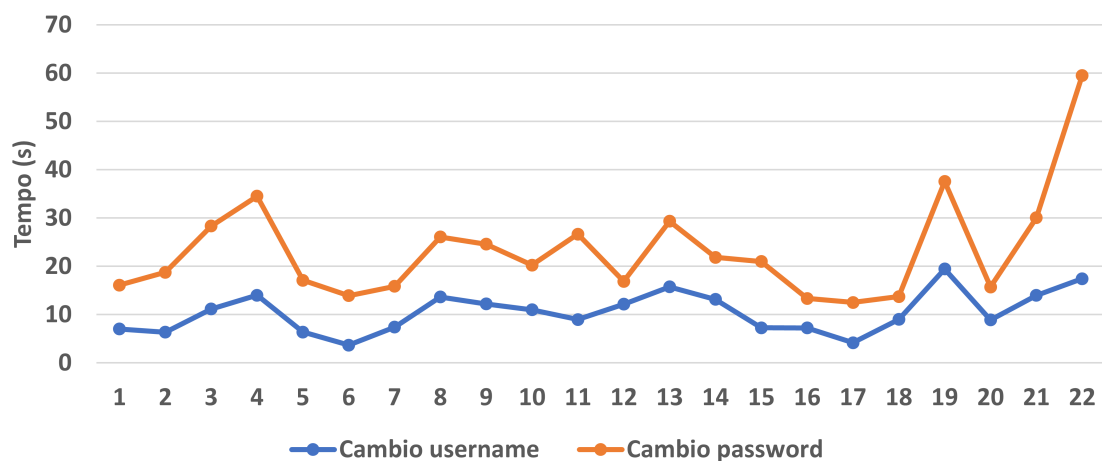


Figura 8.7. Grafico rappresentante le tempistiche richieste da ogni singolo utente per completare l'attività di cambio username e cambio password.

Rispetto al grafico 8.8, quasi tutti gli utenti sono riusciti a completare il cambio dell'username con soli due click. Al contrario il cambio della password ha richiesto

un utilizzo di click direttamente proporzionale alla lunghezza della nuova sequenza scelta.

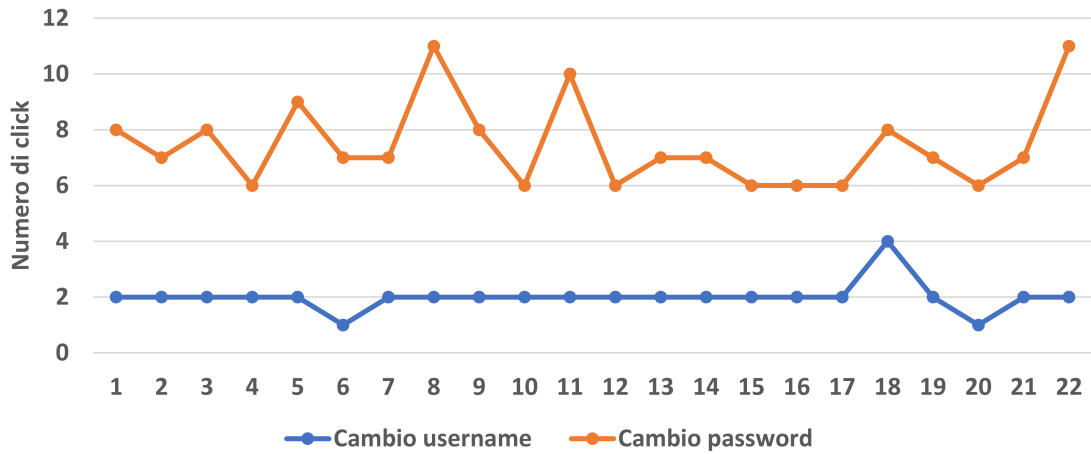


Figura 8.8. Grafico rappresentante il numero di click richiesti dai singoli utenti per completare l'attività di cambio username e cambio password.

Nessun utente ha infine fallito il completamento delle attività secondarie³.

8.1.4 Simulazione di keystroke dynamics

Tutti gli esiti delle simulazioni hanno ricevuto un feedback positivo, dimostrando quindi l'elevata affidabilità del meccanismo di KD. I tempi necessari per il completamento della prova, sono in linea con quelli calcolati durante la registrazione e dipendono unicamente dalla velocità di scrittura dell'utente (vedi grafico 8.9).

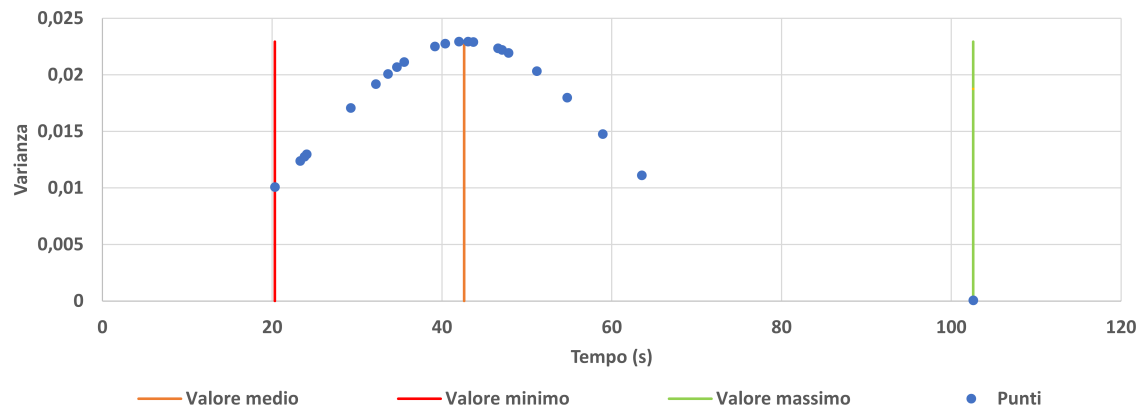


Figura 8.9. Grafico rappresentante le tempistiche richieste da ogni singolo utente per completare la simulazione di keystroke dynamics.

³Viene registrato un solo caso di falso positivo all'interno dell'attività di cambio della password.

8.2 Analisi dei dati soggettivi

Trattiamo ora, in questa seconda parte, le risposte che i 22 utenti hanno fornito durante la compilazione del questionario di valutazione del prototipo, i cui dettagli sono stati già spiegati all'interno del paragrafo 6.3.3.

Rispetto alle prime due batterie di domande (vedi grafici 8.10 e 8.11) l'attività che ha riscontrato maggiori valutazioni negative è quella di autenticazione, in linea con quanto emerso dall'analisi oggettiva. Il riscontro negativo rappresenta, tuttavia, solo una piccola parte dei risultati (due utenti), dal momento che la maggior parte dei partecipanti ha comunque trovato soddisfacente utilizzare questo tipo di meccanismo. Per quanto riguarda le altre attività, sono invece emersi solo giudizi neutri o favorevoli, tra cui spicca la simulazione di KD che ha ricevuto unicamente riscontri positivi.

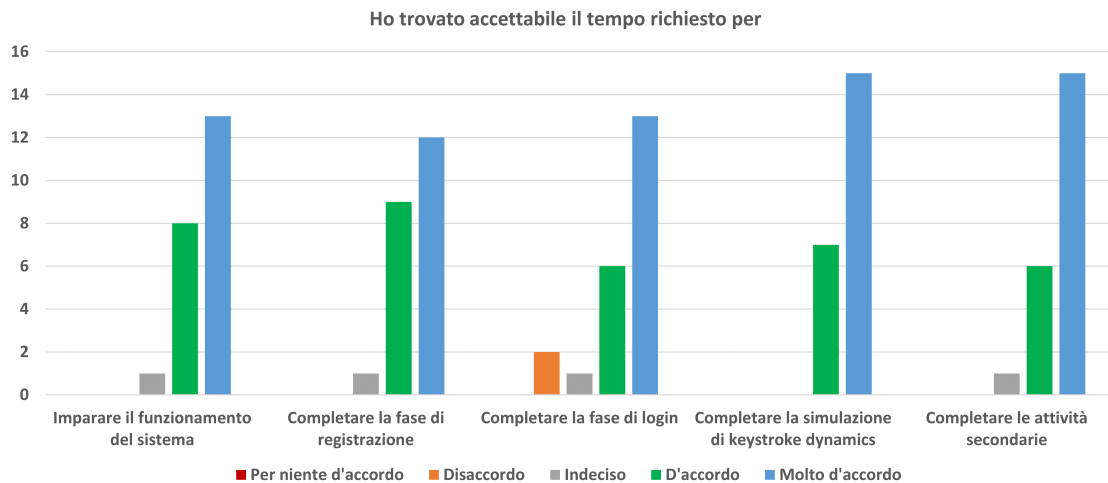


Figura 8.10. Risultati della prima batteria di domande del questionario.

Un altro dato da non trascurare riguarda la terza batteria di domande (vedi grafico 8.12) dove sono state manifestate alcune perplessità sulla sicurezza fornita dal sistema di keystroke dynamics. Nessuna problematica, invece, per le altre attività.

In termini di facilità d'utilizzo (grafico 8.13), si ripropongono gli stessi risultati riscontrati nelle prime due batterie di domande.

L'interfaccia grafica del sistema di autenticazione, seppure non rappresenti l'elemento di valutazione principale del prototipo, è stato in grado di fornire una risposta sulla user experience percepita dai singoli partecipanti. Come mostrato nel grafico 8.14, l'ambiente del sistema di autenticazione ha ricevuto una valutazione generalmente positiva; l'impatto negativo maggiore è stato riscontrato ancora una volta nel processo di autenticazione, probabilmente da tutti coloro che hanno avuto qualche difficoltà nel completare quella fase.

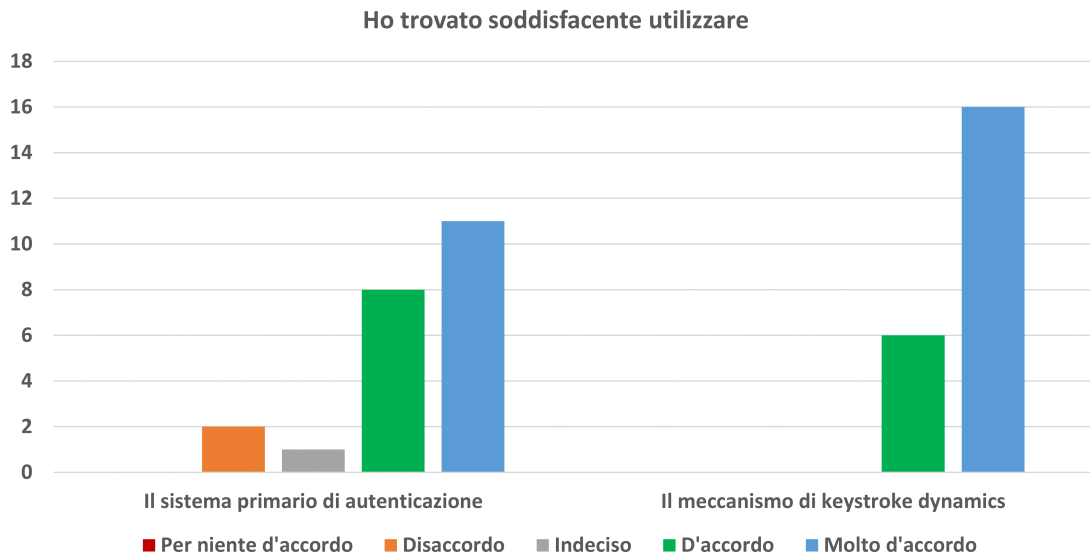


Figura 8.11. Risultati della seconda batteria di domande del questionario.

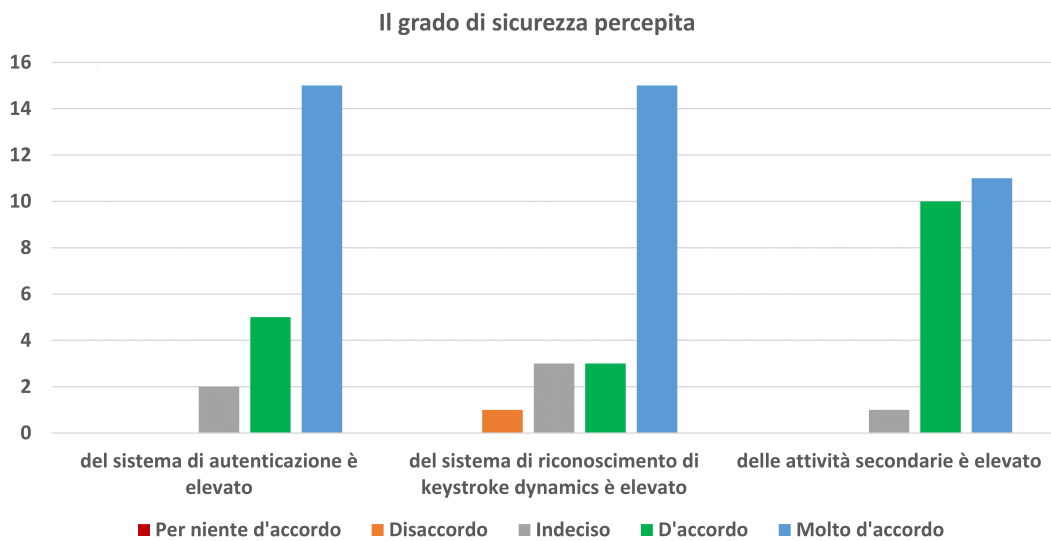


Figura 8.12. Risultati della terza batteria di domande del questionario.

8.3 Conclusione

Proviamo quindi a fornire ora un giudizio generale in base alle valutazioni viste in precedenza⁴:

- Il sistema di autenticazione primario ha ricevuto grossi apprezzamenti per

⁴Anche grazie ai commenti liberi presenti nel questionario finale, che mi hanno permesso di avere un'idea più chiara su alcune valutazioni date.

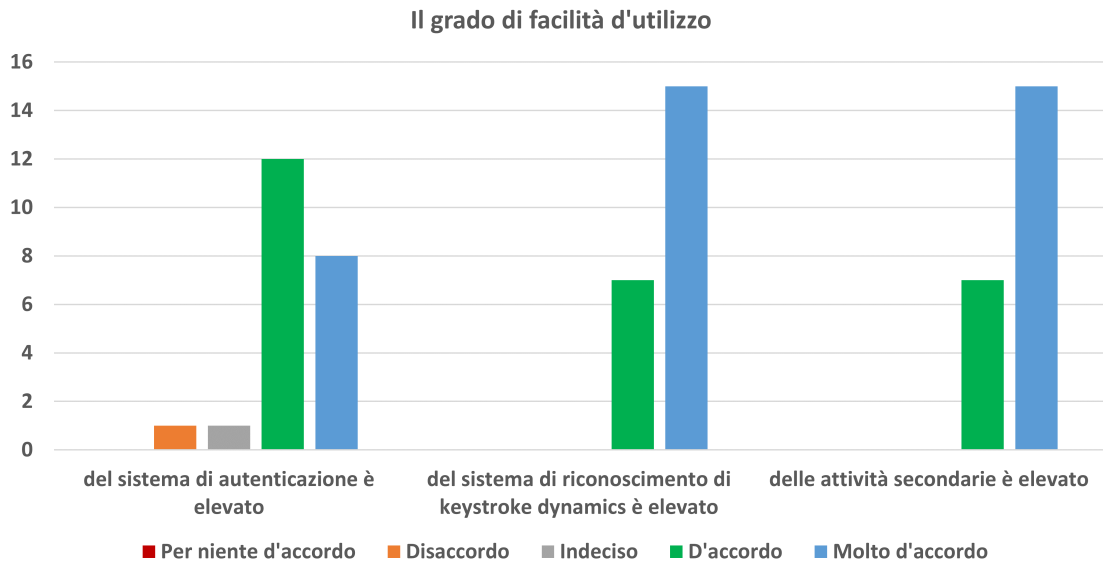


Figura 8.13. Risultati della quarta batteria di domande del questionario.

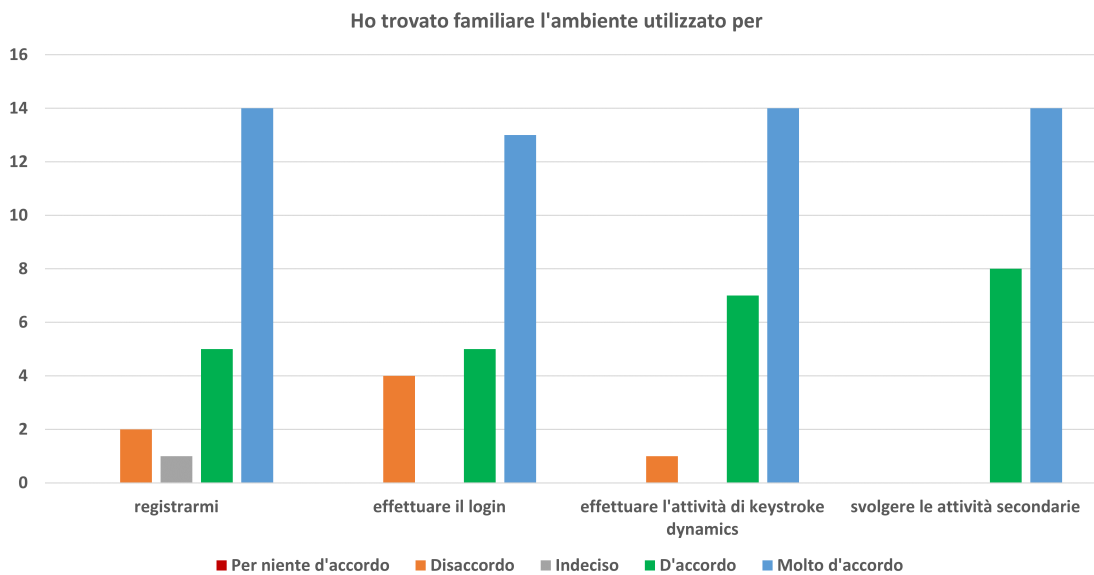


Figura 8.14. Risultati della quinta batteria di domande del questionario.

quanto riguarda il livello di sicurezza percepita, grazie al particolare meccanismo di inserimento delle coordinate in fase di login, con il quale vengono eliminate le problematiche di shoulder surfing e smudge attack. Purtroppo questo aspetto rappresenta anche la critica maggiore, dal momento che alcuni utenti hanno avuto alcune difficoltà a completare il processo di autenticazione⁵. In particolare, alcuni di essi, hanno trovato complicato posizionare i punti della propria sequenza all'interno della griglia di autenticazione, a causa della

⁵Ritengo, in ogni caso, che questo tipo di criticità sia legata prevalentemente al primo login e che vada a sparire mano a mano che si acquisisce familiarità col sistema.

manca delle coordinate esterne (appositamente non presenti al fine di non fornire alcun feedback visivo ad un potenziale attaccante). Un'altra critica mi è stata mossa sul fatto che, questo tipo di mancanza, condiziona indirettamente l'utente a selezionare punti che risiedono vicino al bordo, riducendo di conseguenza il numero di combinazioni totali. Per risolvere questa problematica si potrebbe creare all'interno della griglia alcuni riferimenti grafici passivi, come una differente colorazione delle celle centrali rispetto a quelle esterne, incrementando quindi il numero di punti visivi a cui l'utente può fare riferimento durante la scelta della password⁶. Nessuna osservazione è stata invece avanzata sulla memorizzazione della password grafica, a conferma della "Picture Superiority Effect Theory" [123] discussa nel paragrafo 4.2.

- Il meccanismo di *keystroke dynamics* è sicuramente l'attività che ha generato più stupore e perplessità tra gli utenti. A livello di usabilità è stato fornito all'unanimità un giudizio positivo, grazie all'elevata trasparenza del sistema mentre, per quanto riguarda la sicurezza, alcuni partecipanti hanno avuto qualche dubbio sull'effettiva capacità di riuscire a riconoscere correttamente il proprietario del sistema (nonostante tutte le simulazioni effettuate abbiano ricevuto un riscontro positivo). Quest'ultima valutazione è da attribuire, principalmente, a due motivazioni:
 - In primo luogo, potrebbe essere stato frainteso l'uso per cui è stato pensato il sistema di KD: esso, infatti, non potrà mai sostituire completamente il sistema di autenticazione primario, proprio a causa della possibilità che si possa verificare un falso positivo⁷. La filosofia che è presente dietro la *continuous authentication* è quindi quella di avere un riconoscimento costante, in tutte quelle situazioni in cui il sistema di autenticazione primario non è in grado di agire.
 - Gli utenti potrebbero aver percepito il sistema poco sicuro a causa dell'elevata usabilità, un po' come è accaduto nel caso di "Apple Pay" [4].

⁶Una tecnica simile al *Persuasive Cued-Click Points* [26], vista nel paragrafo 2.4.3.

⁷La presenza di un falso negativo andrebbe invece a compromettere l'usabilità, lasciando intatta la sicurezza dell'utente.

Capitolo 9

Conclusione finale

A partire dall'analisi che è stata condotta sullo stato dell'arte dei sistemi di autenticazione, risulta abbastanza evidente come esistano diverse soluzioni implementative: se una parte di essa rimane ancora vincolata al concetto di “password testuali”, altre cercano di spaziare tra modelli differenti, offrendo interessanti spunti implementativi. Ed è proprio da queste ultime tecniche che ho cercato di costruire il mio prototipo di autenticazione, andando a toccare concetti diversi da quelli che siamo abituati a vedere, per capire quale potesse essere la reazione del pubblico nel provare qualcosa che si allontana fortemente dalle tecniche comunemente utilizzate.

L'ultima domanda presente all'interno del questionario di valutazione del prototipo, chiedeva:

“Utilizzeresti questo meccanismo per proteggere i tuoi dati?”

Penso che cercare di convincere un utente a cambiare le proprie abitudini non sia semplice, soprattutto per tutte quelle persone che non hanno un particolare interesse in questo campo e che vedono l'autenticazione unicamente come un obbligo necessario per poter accedere ad una determinata risorsa. Nonostante queste premesse, tra i 22 utenti che hanno partecipato al test ben 19 hanno risposto positivamente alla domanda¹, riponendo quindi fiducia in un sistema nuovo, che richiederà sicuramente uno sforzo cognitivo maggiore durante i primi utilizzi. Questo dimostra però, allo stesso tempo, come il modello di username/password, su cui si basano la maggior parte dei sistemi con cui interagiamo ogni giorno, abbia ormai stancato ogni tipologia di utente.

Il lavoro a cui sono giunto con questa tesi non può considerarsi concluso, ma presenta vari spunti interessanti che potrebbero essere approfonditi in futuro:

- Un primo aspetto riguarda il miglioramento del prototipo attuale, che sia in grado di risolvere alcune delle lacune evidenziate nel capitolo precedente; in questo modo si avrebbe modo di bilanciare meglio il gap che intercorre attualmente tra la forte usabilità del meccanismo di KD e l'elevata sicurezza del sistema di autenticazione primario.

¹Tra di esse, sette risposte provengono da persone che non conoscevo di persona.

- Vi è poi tutto lo studio legato alla continuous authentication, che permette di approfondire l'utilizzo delle tecniche viste nel paragrafo 2.8 all'interno di un contesto più realistico e meno prototipale. L'analisi potrebbe essere condotta tramite l'implementazione del sistema di autenticazione all'interno di scenari più complessi, che siano in grado di descrivere in modo più accurato le attività condotte quotidianamente dagli utenti². Un ottimo punto di partenza, in questo caso, può sicuramente essere il prototipo di autenticazione avanzato, in cui viene presentata una simulazione di scrittura di una mail applicata al riconoscimento di KD e di cui ho descritto il funzionamento e l'implementazione all'interno dell'appendice C.
- Infine, approfonditi i due elementi precedenti, si potrebbe andare a trattare tutta la parte relativa ai sistemi di autorizzazione, all'interno di un meccanismo di autenticazione più concreto e già integrato all'interno di un file system esistente, vedendo se esiste la possibilità di estendere la “continuous authentication” all'interno di una “continuous authorization”.

In conclusione, mi ritengo soddisfatto del lavoro e dei risultati che sono emersi in questi due anni di ricerca, con la speranza di essere riuscito, anche solo in piccolissima parte, a diminuire la distanza che intercorre tra due mondi diametralmente opposti, ma i cui contatti stanno diventando sempre più frequenti. L'uomo e la macchina.

²In particolare, sarebbe interessante capire come varia l'affidabilità di una tecnica di continuous authentication in questa tipologia di scenario e se, l'utilizzo di meccanismi multipli, sia in grado di fornire un'accuratezza migliore.

Appendice A

Manuale del programmatore

L'obiettivo di questa prima appendice è quello di spiegare, nel dettaglio, quale sia l'architettura del prototipo di valutazione. L'analisi che verrà condotta sarà prettamente tecnica ed è indirizzata a tutti coloro che vogliono capire meglio il funzionamento intrinseco al sistema di autenticazione. Per quanto riguarda invece le azioni che devono essere compiute dall'utente al fine di utilizzare il sistema sul proprio dispositivo, rimando alla lettura dell'appendice [B](#).

Questa sezione viene suddivisa in due parti:

- Ambiente di sviluppo e struttura generale del prototipo.
- Classi.

A.1 Ambiente di sviluppo e struttura generale del prototipo

Il programma è scritto completamente in Java, tramite il SE development kit 8 ed il JDK 1.8, mentre l'intero sviluppo è stato fatto all'interno dell'IDE Eclipse, nella versione "Photon" (giugno 2018). Per quanto riguarda la parte relativa all'interfaccia grafica, sono state utilizzate le librerie di AWT e Swing.

Vedremo ora nel dettaglio come è strutturato il prototipo (vedi figura [A.1](#)), tralasciando la cartella "src" che verrà poi approfondita a partire dal paragrafo [A.2](#).

A.1.1 External library

In questa cartella sono contenuti tutti i file jar necessari per il corretto funzionamento del sistema che, in quanto tali, sono stati tutti aggiunti al build path del progetto Java. Il motivo per il quale vengono inseriti all'interno di questo programma verrà spiegato più avanti ([A.2](#)), ma in generale il loro uso è fondamentale al fine di:

1. Implementare il formato JSON.

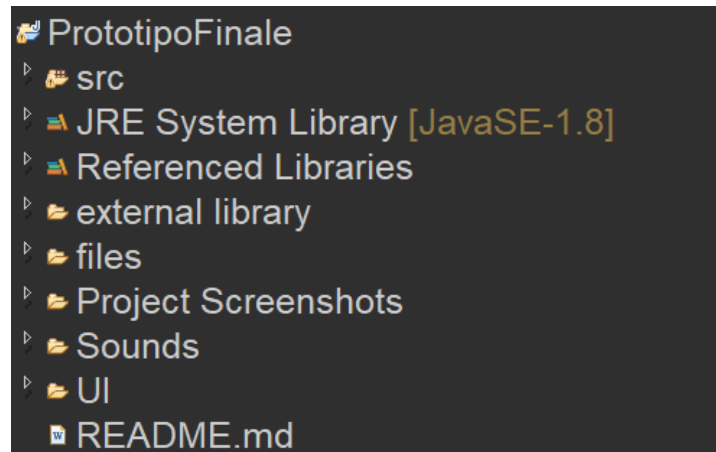


Figura A.1. Struttura del “PrototipoFinale”.

2. Inviare richieste HTTP.
3. Inviare mail.

Presento ora l’elenco di tutte le dipendenze inserite all’interno del prototipo:

- commons-codec-1.9.jar
- commons-logging-1.2.jar
- forms-1.3.0.jar
- httpclient-4.4.jar
- httpcore-4.4.jar
- javaee.jar
- json-simple-1.1.1.jar
- mail.jar

Come descriverò successivamente nel manuale utente, non è necessario aggiungere queste librerie durante la simulazione del prototipo di valutazione, dal momento che viene già fatto a priori durante la creazione dell’eseguibile.

A.1.2 Files

Il sistema di autenticazione necessita, in alcuni punti, di alcuni file di supporto, che vengono racchiusi all’interno della cartella “files”:

- userCredentials.txt, che contiene le credenziali degli utenti nel formato:

- username
- password
- UserPattern.txt, che contiene informazioni riguardanti il meccanismo di key-stroke dynamics nel formato:
 - username
 - lingua utilizzata per la simulazione
 - pattern di KD
- TmpPattern.txt, che contiene l’username dell’utente e che viene condiviso tra processi differenti all’interno del programma.

A.1.3 Project Screenshots e README.md

La piattaforma di base deriva da un progetto opensource [163] di cui ho già discusso nel paragrafo 7.2. Nel file “README.md” è contenuta una breve descrizione del tool originario mentre, nella cartella “Project Screenshots”, sono presenti alcuni diagrammi e screen dell’interfaccia grafica¹.

A.1.4 Sounds

La cartella Sounds contiene un insieme di musiche e suoni che possono essere integrati all’interno del sistema di autenticazione, al fine di fornire un feedback uditivo durante la pressione di un pulsante².

A.1.5 UI

La cartella UI, contiene un insieme di immagini che è possibile applicare come sfondo alle finestre del programma.

A.2 Classi

Analizziamo ora il contenuto della cartella “src”, contenente le classi utilizzate dal prototipo. La loro spiegazione verrà fatta attraverso un ordine logico, seguendo il flusso di operazioni eseguite dall’utente (partendo dal processo di registrazione, fino ad arrivare al logout).

¹Elementi che erano già presenti nel tool originario

²È stato scelto di non utilizzare alcuna musica da riprodurre in background.

A.3 HomePage package

Il package HomePage contiene il codice riguardante la prima finestra che appare una volta lanciato il programma (vedi figura A.2). In particolare, all'interno dell'interfaccia è possibile effettuare una duplice scelta:



Figura A.2. Finestra HomePage del prototipo di autenticazione.

1. Sign Up, che permette all'utente di registrarsi all'interno del sistema.
2. Login, con il quale l'utente potrà accedere alla propria pagina personale, una volta che avrà completato la procedura di autenticazione.

Il package HomePage è composto dalle classi Home.java e HomePn.java.

A.3.1 Home.java

La classe Home.java contiene il metodo

```
public static void main(String[] args);
```

che è il punto di partenza dal quale viene lanciato l'intero programma. Al suo interno viene creata ed inizializzata una variabile "Home h", attraverso gli altri metodi presenti nella medesima classe. La finestra Home ha una dimensione non regolabile e relativa (tramite i parametri scrnsize.width e scrnsize.height) all'ampiezza dello schermo sul quale viene riprodotta.

Vediamo quindi quali sono i metodi presenti all'interno della classe.

public Home()

Questa funzione rappresenta il metodo costruttore della classe Home che, in quanto tale, viene richiamato nel momento in cui viene creata un'istanza dell'oggetto. Esso inizializza i seguenti elementi:

- Gli aspetti estetici e funzionali della finestra, come il titolo
`setTitle("Mutating Graphical Password System");`
o il comportamento che deve assumere una volta che viene chiusa
`setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
- Il pulsante tramite il quale viene avviata la fase di autenticazione
`CustButton btLogin = new CustButton("Login");`

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista (utilizzata per monitorare le azioni dell'utente durante il test).
- L'avvio del timer, utilizzato per monitorare il tempo di login dell'utente. Esso viene inizializzato solo se non è già stato avviato in precedenza, evitando in questo modo sovrascritture che potrebbero inficiare sui risultati finali.
- L'incremento del counter relativo al numero di click effettuati durante la fase di login.
- L'esecuzione del metodo

`loginClick();`

- Il pulsante tramite il quale viene avviata la fase di registrazione
`CustButton btSignUp = new CustButton("Sign Up");`

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista (utilizzata per monitorare le azioni dell'utente durante il test).
- L'avvio del timer, utilizzato per monitorare il tempo di registrazione dell'utente. Esso viene inizializzato solo se non è già stato avviato in precedenza, evitando in questo modo sovrascritture che potrebbero inficiare sui risultati finali.
- L'incremento del counter relativo al numero di click effettuati durante la fase di registrazione.
- L'esecuzione del metodo

```
signUpClick();
```

- L’inizializzazione della variabile “m” relativa alla classe SignUp

```
static SignUp m = new SignUp();
```

- L’inizializzazione della variabile “l” relativa alla classe Login

```
static Login l = new Login();
```

public void signUpClick()

Metodo che disabilita la schermata Home (vedi metodo disableHome()) ed avvia l’interfaccia e le funzionalità relative alla fase di registrazione.

public void loginClick()

Metodo che disabilita la schermata Home (vedi metodo disableHome()) ed avvia l’interfaccia e le funzionalità relative alla fase di autenticazione.

public void enableHome()

Metodo che viene richiamato dalle altre funzioni per riabilitare l’interfaccia grafica relativa alla schermata Home e disabilitare le eventuali finestre attive tramite i metodi disableSignUp() e disableLogin(). Questa funzione viene chiamata nel momento in cui dalla schermata di autenticazione o di registrazione si vuole annullare l’azione corrente, per ritornare alla homepage.

public void disableLogin()

Metodo che viene richiamato dalle altre funzioni per disabilitare l’interfaccia grafica relativa alla schermata di autenticazione.

public void disableSignUp()

Metodo che viene richiamato dalle altre funzioni per disabilitare l’interfaccia grafica relativa alla schermata di registrazione.

public void disableHome()

Metodo che viene richiamato dalle altre funzioni per disabilitare l’interfaccia grafica relativa alla schermata Home.

A.3.2 HomePn.java

All'interno di questa classe troviamo una serie di funzioni che vanno ad arricchire l'interfaccia della schermata home.

Tutti gli elementi vengono disegnati tramite l'uso della variabile `g2d` di tipo `Graphics2D`:

```
public abstract class Graphics2D extends Graphics;
```

Essa è un'estensione della classe `Graphics` che permette di avere una maggior libertà sul controllo della geometria, la gestione dei colori ed il layout del testo.

Vediamo quindi quali sono i metodi principali contenuti in questa classe.

public HomePn()

Il costruttore della classe `HomePn`, si occupa unicamente di caricare all'interno della variabile `"BufferedImage back"`, lo sfondo che verrà poi applicato all'interno dell'interfaccia. Dal momento che l'operazione richiede l'apertura di un file, è necessario gestire l'eventuale `IOException`:

```
try {
    back = ImageIO.read(new File(".\\UI\\back3.jpg"));
}
catch (IOException e) {
    e.printStackTrace();
}
```

public void paintx(Graphics g)

Metodo che si occupa di applicare una serie di elementi grafici all'interno dell'interfaccia:

- Aggiunta dello sfondo caricato in precedenza nella variabile `"back"`.
- Settaggio della scritta `"MGPS"` con relativo colore e font.
- Applicazione della cornice ad una distanza interna di 10 punti rispetto alle dimensione della schermata.

A.4 SignUp package

Il package `SignUp` contiene il codice riguardante l'interfaccia e gli algoritmi utilizzati per la prima parte del processo di autenticazione, ovvero la password grafica e la scelta del linguaggio. L'acquisizione del pattern di KD verrà discusso nel paragrafo [A.5](#).

L'interfaccia (vedi figura A.3) è composta da un primo form “USERNAME” che dovrà essere compilato manualmente dall'utente, mentre il secondo viene completato in automatico in base alle scelte effettuate all'interno della “PASSWORD GRID”. Sono infine presenti 4 pulsanti:

- Show password, che mostra l'ordine temporale con il quale è stata inserita la password.
- Clear, che cancella la password inserita.
- Cancel, che permette di tornare alla schermata precedente.
- Ok, con il quale confermare i dati inseriti.

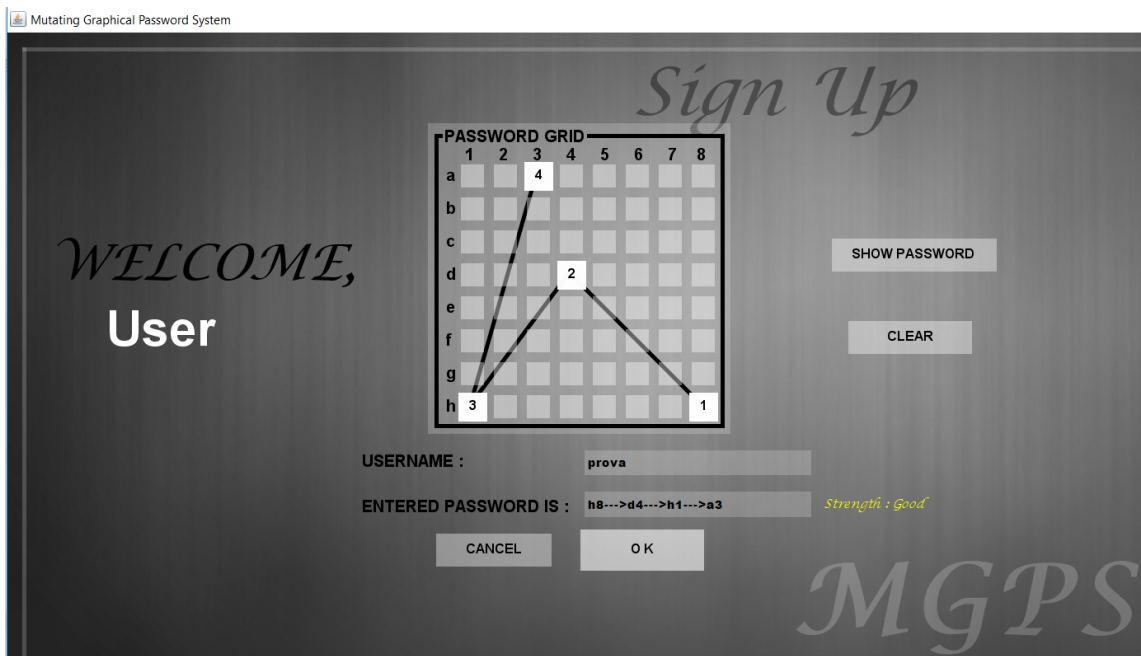


Figura A.3. Finestra di registrazione del prototipo di autenticazione.

Vediamo quindi quali sono le classi di cui è composto il package SignUp.

A.4.1 SignUpPanel.java

La classe SignUpPanel, oltre a rappresentare il punto d'ingresso dell'interfaccia di registrazione, contiene una serie di variabili globali fondamentali per il corretto funzionamento del prototipo:

- Le variabili contenente i nomi dei file sui quali verranno memorizzati i dati degli utenti, secondo lo schema discusso nel paragrafo A.1.2.

```
private static final String FILENAME_PATTERN =  
    ".\\files\\UserPattern.txt";  
private static final String FILENAME_CREDENTIALS =  
    ".\\files\\UserCredentials.txt";
```

- La chiave ed il vettore di inizializzazione usato per cifrare i dati degli utenti, secondo l'algoritmo AES-128-CBC.

```
private final String key = "Bar11f45cara23p5";  
// 128 bit key  
private final String initVector = "vbmc45pl12plxm23";  
// 16 bytes IV
```

- Le variabili utilizzati per valutare i parametri oggettivi relativi alla fase di registrazione

```
private static long startTime = 0;  
private static int clickCounter = 0;  
private static int failures = 0;
```

Vediamo ora quali sono i metodi fondamentali utilizzati all'interno della classe `SignUpPanel`³.

void SignUpPanel()

Questa funzione rappresenta il metodo costruttore della classe `SignUpPanel` che, in quanto tale, viene richiamato nel momento in cui viene creata un'istanza dell'oggetto. Esso inizializza i seguenti elementi:

- La griglia contenente le coordinate da selezionare (vedi [A.4.4](#))

```
BtPanel btPan = new BtPanel();  
btPan.setBorder(null);  
btPan.setBackground(new Color(102, 102, 204));  
btPan.setBounds(510, 100, 382, 407);  
btPan.setOpaque(false);  
add(btPan);  
btPan.setLayout(null);
```

- Gli aspetti estetici e funzionali della finestra come lo sfondo e le caselle in cui inserire l'username. Tra le variabili che vengono inizializzate, vi è "AnimPanel"

```
AnimPanel animPan = new AnimPanel(btPan);
```

³Verrà omessa la spiegazione di tutti i metodi elementari, il cui unico scopo è quello di incrementare o settare il valore di alcune variabili.

essenziale per mostrare la sequenza temporale della password inserita dall'utente (vedi [A.4.5](#) per maggiori dettagli) alla pressione del tasto "SHOW PASSWORD".

Al form username, viene aggiunto un KeyListener, che si avvia ad ogni input inserito da tastiera e che richiama la funzione `procPassUser()`.

```
txtUusr.addKeyListener(new KeyAdapter() {  
    public void keyReleased(KeyEvent e) {  
        procPassUser(e);  
    }  
});
```

- Il pulsante tramite il quale viene terminato il processo di registrazione (con relativo ritorno alla homepage principale)

```
CustButton btnCancel = new CustButton("CANCEL");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la fase di registrazione.
- L'esecuzione del metodo

```
Home.enableHome();
```

- L'inizializzazione di tutti gli elementi legati alla griglia, tramite la funzione

```
clearButtons(false);
```

- Il pulsante tramite il quale viene confermato il processo di registrazione, relativo ai soli dati username e password

```
CustButton btnOK = new CustButton("OK");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la fase di registrazione.
- Se il metodo `procPass()` ritorna true, viene chiamata la funzione `procOK()`

```
if(procPass()){  
    try {  
        procOK();  
    }  
}
```

- Il pulsante tramite il quale viene cancellata la sequenza immessa dall'utente prima di quel momento

```
CustButton btClr = new CustButton("CLEAR");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la fase di registrazione.
- L'esecuzione del metodo

```
public void clearButtons (boolean clrUserName);
```

- Il pulsante tramite il quale viene mostrata la sequenza temporale, relativa alle coordinate della password precedentemente immesse

```
CustButton btShowPass = new CustButton("SHOW PASSWORD");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la fase di registrazione.
- L'esecuzione del metodo

```
public synchronized void startAnim();
```

- Un `MouseListener`

```
btClick();
```

che viene utilizzato per rilevare tutti i click del mouse che vengono eseguiti all'interno della griglia.

```
public void paint(Graphics g);
```

Questo metodo, richiamato in fase di inizializzazione, permette di applicare gli effetti grafici (font, colore, dimensione e posizionamento) a tutti gli elementi contenuti nella schermata di registrazione.

```
public void btClick();
```

Al fine di rilevare il corretto input all'interno della griglia, sono stati costruiti 64 pulsanti associati ai relativi `MouseListener`, uno per ogni coordinata.

```
CustButton[] Btarr=new CustButton[64];

for(i=0;i<64;i++){
    Btarr[i].addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent e) {
            ...
        }
    });
}
```

Nel momento in cui viene cliccato uno dei 64 elementi nella griglia, vengono scatenati i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la fase di registrazione.
- L'esecuzione del metodo

```
clickProc(x);
```

public void clearButtons (boolean clrUsrName);

Il metodo clearButtons permette di pulire e resettare tutti gli elementi grafici inseriti nella griglia, oltre che i campi password e username⁴

```
Btarr[i].clear();
Btarr[i].setEnabled(true);
Btarr[i].removeMouseMotionListener(Btarr[i]);

txtPass.setText("");
txtUsr.setText("");
```

Dopo l'eliminazione di tutti i dati inseriti dall'utente, il processo rimane attivo sulla schermata di registrazione.

public void clickProc(int x);

In questo metodo, il punto che è stato selezionato all'interno della griglia viene aggiunto alle altre coordinate già acquisite in precedenza

```
Password=Password+Btarr[x].getName();
```

Nel form relativo alla password, viene inserito una freccia tra una coordinata inserita e l'altra, al fine di rendere il risultato globale più visibile

⁴Quest'ultima solo se è stato inserito in precedenza (e quindi il valore di clrUsrName è true)

```
if(!txtPass.getText().equals(""))
    txtPass.setText(txtPass.getText() + "--->" +
        Btarr[x].getName());
```

Se l'elemento che si sta registrando non è il primo ad essere stato inserito, viene utilizzata la classe `AnimPanel.java` ([A.4.5](#)) per disegnare sulla griglia la riga che intercorre tra l'ultima coordinata inserita e quella precedente. Viene infine incrementata la variabile "count", che tiene traccia del numero totale di punti inseriti.

public boolean procPass();

In questo metodo vengono verificati che i campi username e password soddisfino i requisiti minimi per poter essere accettati:

- `if(txtUsr.getText().length()==0)`

Se non è stato inserito alcun username, viene ritornato un valore "false".

- `else if(usernameAlreadyExists(txtUsr.getText()))`

Se l'username inserito è stato già registrato in precedenza, viene ritornato un valore "false".

- `else if(Password.length()<6)`

Se la lunghezza della password inserita è inferiore a 6 punti, viene ritornato un valore "false"⁵.

- Se non viene riscontrato nessuno dei casi precedenti, la funzione ritorna un valore "true" e le credenziali vengono salvate all'interno dei relativi file, tramite i metodi

```
saveCredentials(usr,Password);
savePattern(usr);
```

Tutte le casistiche evidenziate vengono monitorate andando a registrare la relativa azione all'intero della lista; ove necessario, viene inoltre incrementato il numero totale di fallimenti.

public synchronized void startAnim();

Metodo `synchronized` che lancia un thread in parallelo (vedi la classe `AnimPanel.java` [A.4.5](#) per maggiori dettagli)

```
Thread th=new Thread(animPan);
th.start();
```

⁵Nel codice originario, vengono suddivise le casistiche in cui `Password.length` è uguale a zero o minore di sei. Nella spiegazione queste informazioni vengono invece omesse, per rendere la lettura più scorrevole e meno prolissa.

public void procOK();

Con il metodo procOK si conclude la prima parte del processo di registrazione

```
Home.disableSignUp();
```

Vengono poi resettati tutti i valori grafici associati alla griglia ed infine avviato il processo di selezione della lingua, attraverso il metodo startLanguages() che si appoggia alla classe Languages.java (vedi paragrafo [A.4.3](#)).

```
Languages l = null;  
l = new Languages();  
l.start(null);
```

private void procPassUser(KeyEvent e);

Il metodo procPassUser verifica, come prima cosa, se sia stato premuto il tasto enter

```
if(e.getKeyChar()==KeyEvent.VK_ENTER)
```

In caso affermativo, viene eseguita la stessa procedura che viene eseguita alla pressione del tasto “OK”.

private boolean usernameAlreadyExists(String user);

Metodo che si occupa di verificare che l’username scritto dall’utente in fase di registrazione, non sia già stato inserito da altri in precedenza. Questo viene fatto tramite l’apertura del file “UserCredentials.txt”

```
fr = new FileReader(FILENAME_CREDENTIALS);
```

Una volta aperto il file e gestite le eventuali eccezioni, vengono comparate tutte le righe dispari⁶

```
if(user.compareTo(MyCipher.decrypt(key, initVector,  
    sCurrentLine))==0){  
    match = true;  
    break;  
}
```

Se viene rilevato un match, viene fermato il processo di ricerca e ritornato il valore “true”, tramite il quale verrà segnalata la relativa eccezione.

⁶Vengono analizzate solo le righe dispari in quanto quelle pari contengono le informazioni relative alla password.

```
private void savePattern(String usr);
```

Metodo che registra la prima parte delle informazioni relativa al processo di KD. Questo viene fatto aprendo il file “UserPattern.txt” e scrivendo al suo interno la variabile “usr” cifrata

```
out.append(MyCipher.encrypt(key, initVector, usr));
```

```
private void saveCredentials(String usr, String password);
```

In modo analogo al metodo precedente, vengono salvate le informazioni di username e password all'interno del file “UserCredentials.txt”, dopo essere state cifrate

```
out.append(MyCipher.encrypt(key, initVector, usr));  
out.append(MyCipher.encrypt(key, initVector, password));
```

A.4.2 SignUp.java

La classe SignUp si occupa di definire alcune caratteristiche presenti nella schermata di registrazione, come:

- Il font ed il colore di alcuni elementi.
- Il comportamento che deve avere il sistema alla chiusura della finestra.

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

- Il fatto che la finestra non possa essere ridimensionata.

```
setResizable(false);
```

A.4.3 Languages.java

Abbiamo visto in precedenza che, dopo avere eseguito il metodo procOK(), viene inizializzata una variabile di tipo “Languages”, che si occuperà prevalentemente di due aspetti:

1. L'inizializzazione dell'interfaccia di selezione della lingua (vedi figura [A.4](#)), attraverso il metodo initialize(), richiamato direttamente dal costruttore

```
public Languages() {  
    initialize();  
}
```

2. La creazione di un processo runnable che si occuperà di gestire gli eventi legati all'interfaccia



Figura A.4. Finestra di selezione della lingua del prototipo di autenticazione.

```
public Runnable NewrunnableProcess = new Runnable(){
    ...
}

public void start(String[] args) throws
    InvocationTargetException,
    InterruptedException {
    EventQueue.invokeLater(NewrunnableProcess);
}
```

Vedremo ora nel dettaglio come viene realizzata la schermata di selezione.

private void initialize();

Viene come prima cosa creata una variabile di tipo JFrame

```
private JFrame frame = new JFrame();
```

su di cui, vengono istanziati una serie di elementi⁷:

- La variabile di tipo Choice

```
Choice c=new Choice();
```

ovvero un menù a scorrimento tramite il quale è possibile effettuare una scelta tra italiano ed inglese. Su di esso è associato un MouseListener che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la fase di registrazione.

⁷Di cui riporto solo gli aspetti funzionali e non quelli grafici.

- Il pulsante tramite il quale confermare la scelta effettuata

```
CustButton btnOK = new CustButton("OK");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell’azione appena eseguita all’interno di una lista.
- L’incremento del counter relativo al numero di click effettuati durante la fase di registrazione.
- L’inizializzazione della variabile “language” con l’elemento selezionato dall’utente

```
String choice = c.getItem(c.getSelectedIndex());  
language = choice;
```

- La chiusura del frame corrente

```
frame.dispose();
```

- L’esecuzione del metodo

```
public static void startKD();
```

public static void startKD();

Questo metodo completa la seconda parte del processo di registrazione e ne avvia l’ultima fase, attraverso l’inizializzazione della variabile “kd”, legata al package che verrà successivamente approfondito nel paragrafo [A.5](#).

```
AWT kd = null;  
AWT.language=language;  
kd = new AWT();  
kd.start(null);
```

A.4.4 BtPanel.java

Classe che si preoccupa di disegnare la griglia, all’interno del quale verrà poi inserita la password. `BtPanel` offre anche supporto alla classe `AnimPanel` ([A.4.5](#)) tramite i seguenti metodi:

```
public void setlines(int l[][],int c){  
    lines=l;  
    count=c;  
    repaint();  
}
```

```
public static boolean getAnim(){
    return animOn;
}

public static void setAnim(boolean a){
    animOn=a;
}

public void anim(){
    AnimPanel.setlines(lines, count);
}
```

A.4.5 AnimPanel.java

La classe AnimPanel viene richiamata nel momento in cui viene selezionata l'opzione "SHOW PASSWORD" nella schermata di registrazione. L'intero processo viene istanziato da un thread dedicato che:

1. Come prima cosa ricostruisce la struttura della griglia, attraverso uno schema molto simile a quello adottato da BtPanel
2. Utilizza i metodi visti in precedenza ([A.4.4](#)) per ricavare informazioni fondamentali per la costruzione dell'animazione finale

```
public static void setlines(int l[][],int c){
    lines=l;
    count=c;
}
```

3. Esegue il metodo

```
public void drawIt();
```

che lancia l'animazione finale, andando ad introdurre un delay di 1 secondo tra un frame e l'altro.

```
Thread.sleep(1000);
```

Infine, al termine dell'animazione e prima che il thread corrente venga distrutto, viene ristabilito il layout originale della schermata di registrazione

```
SignUp.contentEnable();
```

A.5 kd package

All'interno del package kd troviamo le classi utilizzate per:

- Ricreare la schermata finale di registrazione (AWT.java)
- Cifrare i dati in scrittura su file (MyCipher.java)
- Decifrare i dati in lettura da file (MyCipher.java)

Le ultime due funzionalità vengono condivise, e quindi utilizzate, anche dal resto del prototipo.

A.5.1 AWT.java

AWT rappresenta la classe principale, il cui compito è quello di fornire l'interfaccia per l'ultima parte di registrazione (vedi figura [A.5](#))



Figura A.5. Finestra di registrazione KD del prototipo di autenticazione.

Esattamente come accade per la classe “Languages.java”, il funzionamento del meccanismo si basa su un processo runnable AWT che si occupa sia dell’aspetto grafico che del funzionamento dei singoli elementi. Una volta creata l’istanza AWT, viene inizializzata la schermata tramite il metodo `initialize()` contenuto nel costruttore della classe

```
public AWT() {  
    new TypingDNARecorder();  
    initialize();  
}
```

Il lancio del processo AWT avviene invece tramite il metodo `start()`, che introduce il `NewRunnableProcess` all’interno della lista degli eventi

```
public void start(String[] args) throws InvocationTargetException,  
InterruptedException {  
    EventQueue.invokeLater(NewRunnableProcess);  
}
```

Prima di andare a descrivere il metodo principale `initialize()`, analizziamo le variabili globali presenti nella classe:

- La variabile che contiene il nome del file su cui verrà memorizzato il pattern di KD

```
protected static final String FILENAME_TMP =  
    ".\\files\\UserPattern.txt";
```

- Le stringhe in italiano e inglese, che verranno adottate in base al valore della variabile “language”

```
private String introductionItalian = "...";  
private String introductionEnglish = "...";  
private String mainItalian = "...";  
private String mainEnglish = "...";  
private String buttItalian1 = "INVIO";  
private String buttItalian2 = "CANCELLA";  
private String buttEnglish1 = "ENTER";  
private String buttEnglish2 = "CANCEL";
```

- Le variabili utilizzate per cifrare e decifrare le stringhe da file (condivise anche con il resto del programma)

```
private final String key = "Bar11f45cara23p5";  
private final String initVector = "vbmc45pl12plx23";
```

private void initialize();

Tra gli aspetti grafici inizializzati dal metodo, vi sono anche quelli legati alle `JLabel` che dovranno essere istanziati in base al valore assunto dalla variabile “languages”

```
if(language.compareTo("ITALIANO")==0){  
    lblNewLabel = new JLabel(introductionItalian);  
}  
else{  
    lblNewLabel = new JLabel(introductionEnglish);  
}
```

Questo aspetto viene applicato a tutte le `JLabel` presenti, tramite le variabili globali di cui discusso in precedenza. Viene inoltre configurata una variabile di tipo `JTextArea` che verrà utilizzata dall’utente per inserire il testo, da cui verrà poi ricavato il pattern di KD.

```
final JTextArea textMain = new JTextArea();
```

Quest'ultima presenta un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la fase di registrazione.

Vengono inizializzati infine due pulsanti, entrambi associati a due differenti `ActionListener`, che vengono abilitati al click del mouse. Il primo pulsante è quello che permette di cancellare le informazioni ricavate fino a quel momento, resettando il processo di KD: per fare ciò viene utilizzato il metodo `reset()` della classe `TypingDNARecorder` (vedi paragrafo [A.6](#)). Il secondo pulsante, invece, è quello con il quale viene confermato l'intero processo di acquisizione, che è composto dai seguenti passaggi:

1. Viene estratta la sequenza che permetterà poi di identificare l'utente

```
String typingPattern = TypingDNARecorder.getTypingPattern  
(type, length, text, textId, extended);
```

2. Vengono salvate su file i dati cifrati riguardanti il pattern e la lingua scelta dall'utente. Quest'ultima servirà per ricostruire l'interfaccia che verrà poi ricreata durante la simulazione finale

```
out.append(MyCipher.encrypt(key, initVector, language));  
out.append("\n");  
out.append(MyCipher.encrypt(key, initVector, typingPattern));
```

3. Viene distrutto il frame corrente

```
frame.dispose();
```

4. Viene stoppato il timer e inseriti tutti i dati all'interno di una struttura JSON, gestita dalla classe `WriteTest.java` (paragrafo [A.12.1](#))

```
Long duration = System.nanoTime()-SignUpPanel.getStartTime();  
Float durationF = Float.valueOf(duration);  
...  
WriteTest.writeOnFile("Registration",durationF,  
                      clicks,SignUpPanel.getFailures());
```

5. Vengono azzerati il timer, il counter ed il numero di fallimenti legati al processo di registrazione, rendendo in questo modo possibile registrare una nuova istanza.

6. Viene abilitata la home principale

```
Home.enableHome();
```

Inoltre, alla pressione di entrambi i pulsanti, vengono registrate le azioni ed i counter relativi all'analisi oggettiva.

A.5.2 MyCipher.java

Come spiegato in precedenza, tutti i dati che vengono scritti su uno dei tre file devono essere cifrati. Allo stesso modo, è necessario introdurre un meccanismo di decifratura nell'istante in cui quei dati devono essere letti ed elaborati dal programma. La classe MyCipher si occupa proprio di questo, ossia realizzare la cifratura e la decifratura di dati tramite i metodi

```
public static String encrypt(String key, String initVector,  
                             String value);  
public static String decrypt(String key, String initVector,  
                             String encrypted);
```

L'algoritmo che viene utilizzato è AES-128-CBC con PKCS#5 [167, 168], mentre gli argomenti presenti nelle funzioni sono:

- String key, la chiave di 128 bit utilizzata per cifrare o decifrare la stringa.
- String initVector, la stringa di 128 bit usata come vettore di inizializzazione.
- String value, la stringa che deve essere cifrata dal metodo encrypt().
- String encrypted, la stringa che deve essere decifrata dal metodo decrypt().

Per quanto riguarda le variabili “key” e “initVector” sono state utilizzate due stringhe casuali contenente 16 caratteri. Dal momento che ognuno di essi assume un peso di 1 byte, si avrà che

$$TOT_{bit} = 8 \cdot 16 = 128bit \quad (A.1)$$

A.6 kdAlgorithm package

Questo package contiene tutti gli algoritmi necessari per estrarre il pattern di KD, a partire dall'input immesso dall'utente. Dal momento che la classe non è stata modificata, ma deriva direttamente dal secondo tool (capitolo 7.3), rimando alla documentazione ufficiale [165] per la spiegazione dettagliata.

A.7 Error package

Il package Error si occupa di segnalare, all'interno dell'interfaccia del prototipo, gli errori che l'utente può commettere durante la sua simulazione. L'unica classe presente è ErrorPn.java.

A.7.1 ErrorPn.java

La classe ErrorPn si presenta con una serie di variabili globali, che identificano parametri come:

- La posizione in cui deve essere mostrato il messaggio di segnalazione⁸

```
int usrPosx,usrPosy,passPosx,passPosy;
```

- Il tipo di messaggio che deve essere scritto

```
String errUsr=new String();  
String errPass=new String();
```

- Il valore che deve essere comparato

```
int length=-1
```

L’inizializzazione delle variabili verrà fatta direttamente all’interno dei metodi di cui ora andremo ad analizzare i dettagli.

```
public ErrorPn(int x1, int y1, int x2, int y2);
```

Metodo costruttore il cui compito principale è quello di inizializzare i punti sul quale verranno mostrati i messaggi di errore.

```
public void paintx(Graphics g);
```

Il metodo paintx viene utilizzato per fornire un output sulla robustezza della password inserita dall’utente durante la fase di registrazione, tramite i parametri discussi nel paragrafo 7.5. Nella prima parte del metodo vengono inizializzate tutti gli elementi grafici del messaggio, come font, colore e dimensione. Il tipo di testo che deve essere scritto viene stabilito da un costrutto if-else, in base al valore assunto dalla variabile length:

- ```
if(length>0 && length<=2){
 g2d.setColor(Color.RED);
 g2d.drawString("Strength : Weak", passPosx, passPosy);
}
```
- ```
else if(length>=3 && length<=5){  
    g2d.setColor(Color.YELLOW);  
    g2d.drawString("Strength : Good", passPosx, passPosy);  
}
```

⁸Con il prefisso -usr si intendono le coordinate del messaggio relativo al campo “User”. Analogamente, il prefisso -pass identifica il campo “Password”.

```
• else if(length>=6){  
    g2d.setColor(Color.GREEN);  
    g2d.drawString("Strength : Strong", passPosx, passPosy);  
}
```

```
public void setLength(int l);
```

Metodo che setta il valore della variabile length.

```
public void setUsrErr(String str);
```

Metodo tramite il quale viene settato l'errore associato all'username.

```
    errUsr=str.toString();
```

```
public void setUsrErr(String str);
```

Metodo tramite il quale viene settato l'errore associato alla password.

```
    errPass=str.toString();
```

A.8 Login package

Il package Login contiene il codice riguardante l'interfaccia e gli algoritmi utilizzati per il processo di login, avviato nel momento in cui viene premuto il relativo pulsante nella schermata home (vedi [A.3](#) per maggiori dettagli). La schermata (figura [A.6](#)) è composta dai seguenti elementi:

- Una griglia composta da una serie di coordinate casuali.
- Un form per l'inserimento dell'username.
- Un form per l'inserimento della password.
- Un pulsante per confermare i dati inseriti.
- Un pulsante per annullare la fase di login.

Vediamo quindi quali sono i metodi presenti nel package Login.

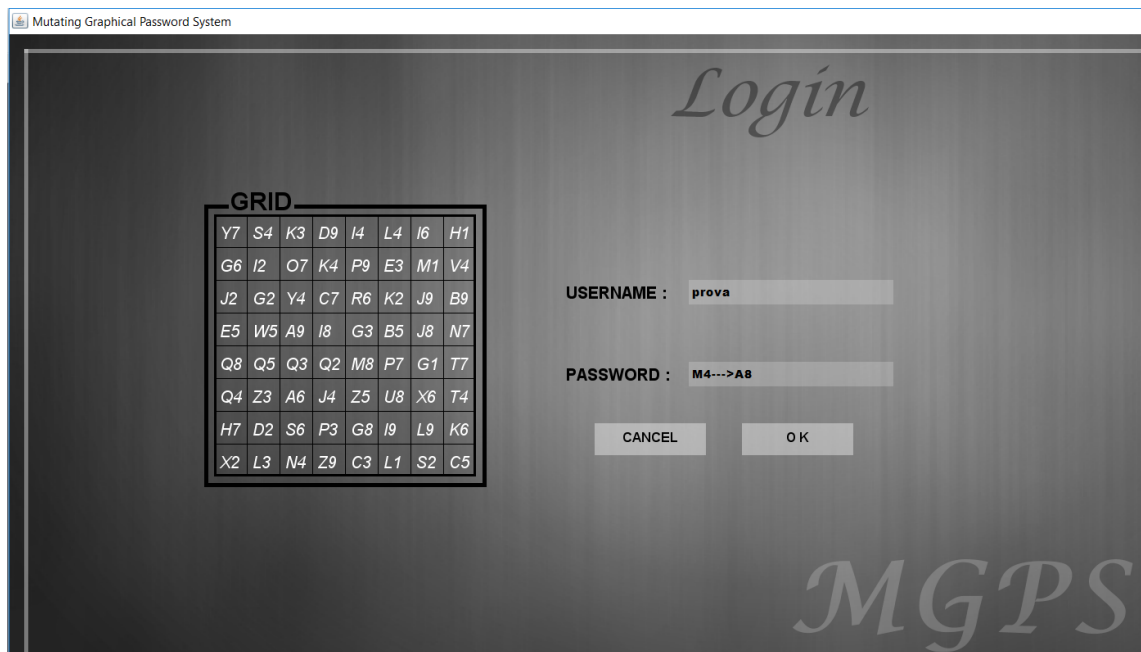


Figura A.6. Finestra di login del prototipo di autenticazione.

A.8.1 LoginPanel.java

La classe LoginPanel, oltre a rappresentare il punto d'ingresso dell'interfaccia di login, contiene una serie di variabili globali fondamentali per il corretto funzionamento del prototipo:

- La variabile contenente il nome del file usato per memorizzare le credenziali degli utenti (secondo lo schema discusso nel paragrafo [A.1.2](#))

```
private static final String FILENAME_CREDENTIALS =
    ".\\files\\UserCredentials.txt";
```

- La chiave ed il vettore di inizializzazione usato per decifrare i dati degli utenti, secondo l'algoritmo AES-128-CBC

```
private final String key = "Bar11f45cara23p5";
// 128 bit key
private final String initVector = "vbmc45pl12plxm23";
// 16 bytes IV
```

- Le variabili utilizzati per valutare i parametri oggettivi relativi alla fase di login

```
private static long startTime = 0;
private static int clickCounter = 0;
private static int failures = 0;
```

Vediamo ora quali sono i metodi fondamentali utilizzati all'interno della classe `LoginPanel`⁹.

public LoginPanel();

Il metodo costruttore viene utilizzato per inizializzare tutti gli elementi presenti nella schermata di login. Durante la costruzione del form per la password, viene aggiunto un `MouseListener` che si attiva al click del mouse e che scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la fase di login.

Oltre al `MouseListener`, viene aggiunto anche un `KeyListener`, che si avvia ad ogni input inserito da tastiera e che richiama la funzione `procPass`

```
txtPass.addKeyListener(new KeyAdapter() {  
    public void keyReleased(KeyEvent e) {  
        procPass(e);  
    }  
});
```

Un'operazione analoga viene fatta anche per il form contenente l'username.

```
txtUsr.addKeyListener(new KeyAdapter() {  
    public void keyTyped(KeyEvent e) {  
        errpn.setUsrErr("");  
    }  
  
    public void keyReleased(KeyEvent e) {  
        procPassUser(e);  
    }  
});  
  
...  
  
txtUsr.addMouseListener(new MouseAdapter() {  
    public void mouseClicked(MouseEvent e) {  
        ...  
    }  
});
```

Per quanto riguarda la griglia, invece, vengono richiamati i metodi contenuti all'interno della classe `Grid.java` (vedi paragrafo [A.8.3](#))

⁹Verrà omessa la spiegazione di tutti i metodi elementari, il cui unico scopo è quello di incrementare o settare il valore di alcune variabili.

```
gridPn = new Grid();  
gridPn.setBounds(250, 220, 320, 320);  
gridPn.setDone(50);
```

Vengono infine implementati due CustButton:

- Il pulsante “CANCEL”, che permette di terminare il processo di login e ritornare alla home principale

```
CustButton btnCancel = new CustButton("CANCEL");
```

Ad esso è associato un MouseListener che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell’azione appena eseguita all’interno di una lista.
- L’incremento del counter relativo al numero di click effettuati durante la fase di login.
- L’esecuzione del metodo

```
public void procCancel() throws  
InvocationTargetException, InterruptedException;
```

- Il pulsante “OK”, con il quale l’utente conferma le credenziali appena inserite

```
CustButton btnOK = new CustButton("O K");
```

Ad esso è associato un MouseListener che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell’azione appena eseguita all’interno di una lista.
- L’incremento del counter relativo al numero di click effettuati durante la fase di login.
- L’esecuzione del metodo

```
public boolean procOK() throws  
InvocationTargetException, InterruptedException
```

public void procPass(KeyEvent e);

Il metodo procPass si occupa di agevolare la scrittura e la cancellazione delle coordinate durante la digitazione della password. Esso richiede come argomento un KeyEvent, che rappresenta il carattere immesso dall’utente in un determinato istante. La funzione può essere innanzitutto descritta come un primo if-else, che separa tre casi:

1. La cancellazione di un punto

```
if(e.getKeyChar()==KeyEvent.VK_BACK_SPACE){  
    ...  
}
```

2. La convalida dei dati, tramite la pressione del tasto enter

```
else if(e.getKeyChar()==KeyEvent.VK_ENTER){  
    ...  
}
```

3. L'inserimento di un carattere

```
else{  
    ...  
}
```

Nel caso in cui sia stato premuto il tasto “backspace”, vi sono ulteriori quattro sotto-casi, tutti dipendenti dalla lunghezza della password già inserita:

1. Se la dimensione è 0, non viene applicata nessuna modifica

```
if(password.length()==0)  
    return;
```

2. Se la dimensione è pari ad 1 o a 2, significa che in precedenza è stato inserito un solo punto, completo o parziale¹⁰. L'eliminazione di un carattere equivale quindi all'eliminazione dell'intero punto.

```
else if(password.length()==1 || password.length()==2 ||  
password.length()==(2+separator.length())){  
    password="";  
    txtPass.setText("");  
    length=0;  
    index=0;  
}
```

3. Se la dimensione è rappresentata da un numero pari diverso da 0 e 2 deve essere eliminato, oltre che al punto, anche il separatore.

```
else if(password.length()%2==0){  
password=txtPass.getText().substring  
(0, txtPass.getText().length()-1 -separator.length());  
txtPass.setText(password);  
length--;  
index--;  
length-=separator.length();  
}
```

¹⁰Ogni punto deve essere composto da due coordinate.

4. Se la dimensione è rappresentata da un numero dispari, non viene invece cancellato il separatore.

Se invece l'utente preme il tasto enter, viene avviata la procedura di conferma e convalida dei dati:

```
try {
    if(!procOK())
        return;
} catch (InvocationTargetException | InterruptedException e_) {
    e_.printStackTrace();
}
```

Infine, se il KeyEvent riguarda un inserimento vengono analizzati due sotto-casi, in base alla lunghezza della password:

1. Se la password ha una dimensione rappresentata da un numero pari, significa che l'utente sta per inserire la prima coordinata del punto che, per definizione, deve necessariamente essere un carattere alfabetico. Inoltre, nel caso in cui il carattere inserito sia minuscolo, esso viene automaticamente convertito in maiuscolo.

```
char c=e.getKeyChar();

if(Character.isLetter(c)){
    if(Character.isLowerCase(c))
        c=Character.toUpperCase(c);
    password=password + c;
    errpn.setPassErr("");
}
```

Se il carattere inserito non è una lettera, ne viene impedito l'inserimento.

2. Se la password ha una dimensione rappresentata da un numero dispari, significa che l'utente sta per inserire la seconda coordinata del punto che, per definizione, deve necessariamente essere un numero.

```
if(Character.isDigit(e.getKeyChar())){
    ...
}
```

Se il carattere inserito non è un numero ne viene impedito l'inserimento mentre, in caso contrario, viene verificato se il punto inserito dall'utente appartenga ad una delle caselle presenti nella griglia. Per fare ciò viene richiamato il metodo `isThere()` della classe `gridPn` (vedi paragrafo [A.8.3](#)).

```
String passcode=gridPn.isThere(password.substring
(password.length()-1, password.length()) + e.getKeyChar());
```

In tutti i casi, indipendentemente dal tasto che è stato inserito, l'attività eseguita viene monitorata ed aggiunta ad una lista, che verrà poi utilizzata per analizzare i comportamenti dell'utente durante la simulazione.

```
public void procPassUser(KeyEvent e);
```

Se l'utente, mentre si trova nel form username, preme il pulsante enter, viene avviata il processo di convalida e conferma dei dati appena inseriti:

```
try {
    if(!procOK())
        return;
} catch (InvocationTargetException | InterruptedException e_) {
    e_.printStackTrace();
}
```

```
public boolean procOK();
```

Il metodo `procOK()` si occupa di convalidare i dati inseriti dall'utente, fornendo un esito positivo o negativo¹¹; quest'ultimo è accompagnato da un contestuale incremento del numero di fallimenti relativo alla fase di login. Inoltre, indipendentemente dal risultato, tutte le azioni vengono registrate all'interno della lista che verrà poi utilizzata per valutare i test.

Come primo controllo, se i campi username e password sono vuoti viene ritornato un valore `false`.

```
if(user.equals("")){
    ...
    return false;
}
if(txtPass.getText().equals("")){
    ...
    return false;
}
```

Dopodichè, viene verificato se esiste una combinazione valida di username/password all'interno del file "UserCredentials.txt"

```
int res=compareCredentials(user,password);
```

In base al valore ritornato dalla funzione, si possono avere tre casi differenti:

1. La password è sbagliata

```
if(res== -1){
    errpn.setUsrErr("");
    errpn.setPassErr("Wrong Password");
    ...
    return false;
}
```

¹¹La funzione ritorna un valore di tipo boolean, che assume un valore `true` in caso di esito positivo e un valore `false` nel caso opposto.

2. L'username non esiste

```
if(res==-2){
    errpn.setPassErr("");
    errpn.setUsrErr("No such Username exists");
    ...
    return false;
}
```

3. Le credenziali inserite sono corrette.

In quest'ultimo caso, vengono stoppati ed inseriti in una struttura JSON tutti i test raccolti all'interno della fase di login (vedi paragrafo [A.12.1](#)). Viene infine dismesso il frame corrente ed abilitato quello relativo alla pagina di welcome (vedi paragrafo [A.9](#)).

```
Home.disableHome();
Home.disableLogin();
Home.disableSignUp();
wel.setVisible(true);
wel.setName(txtUsr.getText(),2);
return true;
```

private int compareCredentials(String user, String password);

Il metodo `compareCredentials()` si occupa di verificare se esiste una corrispondenza delle variabili “user” e “password”, all'interno del file “UserCredentials.txt”. Il parametro di paragone utilizzato è l'utente: se non esiste un matching con nessuno degli username presenti nel file, viene ritornato il valore -2; in caso contrario si passa alla fase successiva in cui viene confrontata la password. A questo punto, se la password presente nel file è uguale a quella inserita dall'utente, l'autenticazione viene completata con successo (return 0), altrimenti viene ritornato il valore -1. Dal momento che tutte le informazioni contenute nel file sono cifrate, sarà necessario utilizzare il metodo della classe `MyCipher` (vedi paragrafo [A.5.2](#)) per leggere i dati in chiaro

```
public static String decrypt(String key, String initVector,
                             String encrypted);
```

A.8.2 Login.java

La classe `Login` si occupa di definire alcune caratteristiche della schermata di login come:

- Il font ed il colore di alcuni elementi.
- Il comportamento che deve avere il sistema alla chiusura della finestra

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

- Il fatto che la finestra non possa essere ridimensionata

```
setResizable(false);
```

A.8.3 Grid.java

La classe Grid.java si occupa di costruire sia la struttura della griglia che il relativo contenuto. Le principali variabili globali utilizzate sono:

- Un vettore che conterrà tutti i 64 punti, definiti ognuno da coordinate casuali

```
String[] temp=new String[64];
```

- Una matrice temporanea in cui viene rappresentato il vettore

```
String[] [] randtemp=new String[8][8];
```

- Una matrice definitiva che rappresenta la griglia attuale

```
String[] [] rand=new String[8][8];
```

```
public Grid();
```

Il metodo costruttore inizializza gli elementi casuali che verranno di volta in volta inseriti all'interno della griglia. Questo viene fatto tramite l'ausilio della classe RandomNum.java ([A.8.4](#)).

```
temp=r.generateRandom();
```

Il vettore “temp” viene poi disposto all'interno della matrice randtemp, tramite il metodo setStr().

```
int k=0;
for(int i=0;i<8;i++){
    for(int j=0;j<8;j++){
        randtemp[i][j]=temp[k++];
    }
}
```

Infine, attraverso un metodo simile a quello appena discusso, vengono trasferiti i valori dalla matrice “randtemp” alla matrice definitiva “rand”.

public void paintx(Graphics g);

A livello grafico, la costruzione della griglia viene fatta tramite il metodo `paintx()`

```
for(int i=1;i<8;i++){
    g2d.drawLine(1, cellSize*i, getWidth()-1, cellSize*i);
    g2d.drawLine(cellSize*i, 1, cellSize*i, getHeight()-1);
}
```

Dopo che è stata definito lo scheletro, vengono inseriti all'interno delle celle gli elementi casuali di cui discusso in precedenza.

```
for(int i=0;i<8;i++){
    for(int j=0;j<8;j++){
        g2d.drawString("" + rand[i][j],
            x-(8*rand[i][j].length()/2)-5, y+10);
        x+=40;
    }
    x=cellSize/2;
    y+=40;
}
```

public void run();

Il metodo `run` gestisce il ciclo di creazione e distruzione dei punti presenti all'interno della griglia (vedi paragrafo 7.5 per maggiori dettagli sul funzionamento). Il punto cardine del sistema è l'interruzione del thread adibito alla creazione e distruzione dei punti casuali

```
Thread.sleep(300);
```

Intervenendo su quel valore si può quindi ridurre o aumentare il tempo che l'utente ha per inserire un intero punto. La creazione di una nuova griglia è accompagnata dal metodo `effectDraw()`, che introduce un'animazione di dissolvenza con il quale vengono eliminati i vecchi valori ed inseriti quelli nuovi.

public String isThere(String s);

Durante la fase di login, le coordinate che l'utente inserisce per definire un punto, difficilmente corrispondono a quelle reali¹²; per questo motivo è necessario un metodo che sia in grado di tradurre le coordinate "fittizie" con quelle reali (dal momento che la comparazione finale della password viene fatta proprio su queste ultime).

In base alle coordinate inserite, il metodo `isThere()` può ritornare due valori differenti:

¹²Potrebbe infatti accadere che durante la generazione casuale degli elementi, siano individuate proprio le coordinate reali.

1. Se viene rilevata una corrispondenza all'interno della matrice "rand", il metodo ritorna le coordinate reali

```
if(rand[i][j].equals(s))
    return "" + c + (j+1);
```

2. Se non viene rilevata nessuna corrispondenza, il metodo ritorna un valore nullo

```
return "";
```

A.8.4 RandomNum.java

Questa classe si occupa di generare i punti casuali da inserire poi all'interno della griglia. Essa è composta principalmente da tre variabili globali che rappresentano:

- L'insieme delle lettere che è possibile scegliere come prima coordinata
- L'insieme delle cifre che è possibile scegliere come seconda coordinata
- Il valore di ritorno del metodo generateRandom(), contenente le 64 combinazioni dei punti casuali

```
String alpha="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

```
String numeric="123456789";
```

```
String[] alphaNumeric=new String[64];
```

```
public String[] generateRandom();
```

Come prima cosa viene creato il primo punto casuale.

```
alphaNumeric[i]=Character.toString(alpha.charAt(r.nextInt(
    alpha.length())));
alphaNumeric[i]=alphaNumeric[i].concat(Character.toString(
    numeric.charAt(r.nextInt(numeric.length()))));
```

I successivi 63 punti vengono gestiti all'interno di un ciclo while, dal momento che nessuno dei nuovi punti deve essere uguale a quelli precedentemente generati. Per verificare ciò, il punto viene inserito all'interno della variabile "temp"

```
temp=Character.toString(alpha.charAt(r.nextInt(
    alpha.length())));
temp=temp.concat(Character.toString(numeric.charAt(
    r.nextInt(numeric.length())));
```

Essa, prima di essere inserita nella variabile finale, viene confrontata con tutti i punti precedenti: se vi è una corrispondenza, il punto viene scartato e si ripete il processo di selezione mentre, in caso contrario, avviene il salvataggio del nuovo valore

```
while(i!=63){  
    flag=1;  
  
    for(j=0;j<=i;j++){  
        if(alphaNumeric[j].compareTo(temp)==0){  
            flag=0;  
            break;  
        }  
    }  
  
    if(flag==1)  
        alphanumeric[++i]=temp;  
}
```

A.9 Welcome package

Il package Welcome si occupa di gestire la parte grafica e logica della schermata di benvenuto, che intercorre subito dopo l'avvenuta autenticazione dell'utente (vedi paragrafo [A.7](#)).

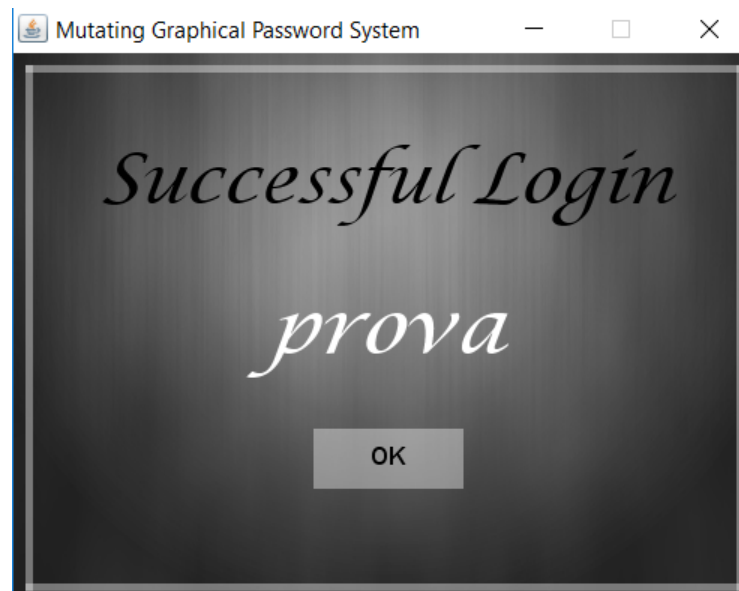


Figura A.7. Finestra di welcome del prototipo di autenticazione.

A.9.1 WelcomePanel.java

la classe si compone da una serie di variabili globali che verranno inizializzate tramite il costruttore ed i metodi, di cui discuteremo ora i dettagli.

public WelcomePanel(String a);

Il costruttore, dopo avere caricato lo sfondo del frame, inizializza il pulsante “OK”, che viene utilizzato per chiudere il frame corrente ed aprire quello relativo alla pagina personale dell’utente. Ad esso è associato inoltre un `MouseListener` che, una volta abilitato tramite click del mouse, esegue il metodo

```
public void procLogout();
```

Tutte le azioni che l’utente compie all’interno di questa interfaccia, non vengono monitorati all’interno dei test oggettivi, in quanto questa rappresenta solo una fase di transizione.

public void paintx(Graphics g);

Metodo che si occupa di definire una serie di aspetti grafici relativi alla schermata welcome:

- Passare l’username dell’utente al nuovo processo

```
name=new String(user);
```

- Indicare il processo dal quale si proviene, attraverso il settaggio di un flag numerico

```
flag=x;
```

Il prototipo originale, infatti, prevedeva il login automatico al termine della fase di registrazione: in questo caso era necessario sapere quale fosse il processo da cui si proveniva, per poter deallocare le giuste risorse. Nella mia implementazione questa scelta è stata rimossa, rendendo il login l’unico meccanismo tramite il quale poter accedere alla propria pagina personale.

public void procLogout();

In base al valore del flag, definito tramite il metodo precedente, si vanno a distruggere le relative risorse

```
if(flag==1)
    SignUpPanel.disableWel();
else if(flag==2)
    LoginPanel.disableWel();
```

Viene infine ricreato il frame relativo alla pagina personale dell’utente, tramite il passaggio del relativo username contenuto nella variabile “name”

```
usr=new UserFrame(name);
```

```
public static void disableUsr();
```

Metodo che chiude il programma corrente, tramite l'esecuzione del relativo comando

```
System.exit(0);
```

A.9.2 Welcome.java

La classe Welcome si occupa di definire alcune caratteristiche della schermata di login come:

- Il font ed il colore di alcuni elementi.
- Il comportamento che deve avere il sistema alla chiusura della finestra

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

- Il fatto che la finestra non possa essere ridimensionata

```
setResizable(false);
```

A.10 User package

Il package User contiene il codice riguardante l'interfaccia e gli algoritmi utilizzati all'interno della pagina personale dell'utente, che viene avviata dopo avere completato la fase di autenticazione. L'interfaccia (figura [A.8](#)) è composta da una serie di pulsanti posti sulla sinistra dello schermo, ognuno dei quali avvia una specifica attività che verrà poi riprodotta sulla parte destra dell'interfaccia.

Vediamo quindi quali sono i metodi presenti nel package User.

A.10.1 UserPn.java

La classe UserPn, oltre a rappresentare il punto d'ingresso dell'interfaccia, contiene una serie di variabili globali fondamentali per il corretto funzionamento del prototipo:

- La variabile contenente il nome del file, che viene usato per passare l'username al processo di KD

```
private static final String FILENAME_CREDENTIALS =  
    "\\files\\TmpPattern.txt";
```

- La chiave ed il vettore di inizializzazione usato per decifrare i dati degli utenti, secondo l'algoritmo AES-128-CBC



Figura A.8. Schermata utente del prototipo di autenticazione.

```
private final String key = "Bar11f45cara23p5";
// 128 bit key
private final String initVector = "vbm45pl12plxm23";
// 16 bytes IV
```

- Le variabili utilizzati per valutare i parametri oggettivi relativi a:
 - cambio password
 - cambio username
 - cancellazione dell'account
 - simulazione di KD

```
private static int counterChangePassword=0;
private static int counterChangeUsername=0;
private static int counterDeleteUsername=0;
private static int counterKD=0;
private static long startTimeChangePassword=0;
private static long startTimeChangeUsername=0;
private static long startTimeDeleteUsername=0;
private static long startTimeKD=0;
private static int failuresChangePassword=0;
private static int failuresChangeUsername=0;
private static int failuresDeleteUsername=0;
private static int failuresKD=0;
```

Vediamo ora quali sono i metodi fondamentali utilizzati all'interno della classe `UserPn`¹³.

`public UserPn(String usr);`

Il metodo costruttore richiede un argomento obbligatorio, tramite il quale viene costruita l'interfaccia utente.

```
username=usr;
```

Vengono poi inizializzati i seguenti tasti:

- Il pulsante tramite il quale è possibile ritornare alla schermata iniziale della pagina personale

```
CustButton home=new CustButton("Home");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter “failuresChangeUsername”, se l'attività del cambio dell'username è già stato avviato

```
if(getStartTimeChangeUsername()!=0)
    failuresChangeUsername++;
```

- L'incremento del counter “failuresDeleteUsername”, se l'attività di eliminazione dell'account è già stata avviata

```
if(getStartTimeDeleteUsername()!=0)
    failuresDeleteUsername++;
```

- L'incremento del counter “failuresChangePassword”, se l'attività del cambio della password è già stata avviata

```
if(getStartTimeChangePassword()!=0)
    failuresChangePassword++;
```

- L'esecuzione del metodo

```
public static void showHome();
```

- Il pulsante tramite il quale è possibile avviare l'attività di cambio dell'username

```
CustButton chgname=new CustButton("Change Username");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

¹³Verrà omessa la spiegazione di tutti i metodi elementari, il cui unico scopo è quello di incrementare o settare il valore di alcune variabili.

- La scrittura dell'azione appena eseguita all'interno di una lista.
- Se non ancora fatto, viene avviato il timer relativo all'attività

```
if(startTimeChangeUsername==0)
    startTimeChangeUsername();
```

In caso contrario viene incrementato il numero di fallimenti ed il counter relativi all'attività

```
else{
    failuresChangeUsername++;
    increaseCounterChangeUsername();
}
```

- L'incremento del counter “failuresDeleteUsername”, se l'attività di eliminazione dell'account è già stata avviata.
- L'incremento del counter “failuresChangePassword”, se l'attività del cambio della password è già stata avviata.
- L'esecuzione del metodo

```
public void showChgName();
```

- Il pulsante tramite il quale è possibile avviare l'attività di cambio della password

```
CustButton chgpass=new CustButton("Change Password");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- Se non ancora fatto, viene avviato il timer relativo all'attività

```
if(getStartTimeChangePassword()==0)
    startTimeChangePassword();
```

In caso contrario viene incrementato il numero di fallimenti ed il counter relativi all'attività

```
else {
    failuresChangePassword++;
    increaseCounterChangePassword();
}
```

- L'incremento del counter “failuresDeleteUsername”, se l'attività di eliminazione dell'account è già stata avviata.
- L'incremento del counter “failuresChangeUsername”, se l'attività del cambio dell'username è già stata avviata.
- L'esecuzione del metodo

```
public void showChgPass();
```

- Il pulsante tramite il quale è possibile avviare l'attività di eliminazione dell'account

```
CustButton del=new CustButton("Delete Account");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- Se non ancora fatto, viene avviato il timer relativo all'attività

```
if(getStartTimeDeleteUsername()==0)
    startTimeDeleteUsername();
```

In caso contrario viene incrementato il numero di fallimenti ed il counter relativi all'attività

```
else {
    failuresDeleteUsername++;
    increaseCounterDeleteUsername();
}
```

- L'incremento del counter "failuresChangePassword", se l'attività di cambio della password è già stata avviata.
- L'incremento del counter "failuresChangeUsername", se l'attività del cambio dell'username è già stata avviata.
- L'esecuzione del metodo

```
public void showDel();
```

- Il pulsante tramite il quale è possibile effettuare il logout

```
CustButton logout=new CustButton("Logout");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter "failuresChangeUsername", se l'attività del cambio dell'username è già stato avviato.
- L'incremento del counter "failuresDeleteUsername", se l'attività di eliminazione dell'account è già stata avviata.
- L'incremento del counter "failuresChangePassword", se l'attività del cambio della password è già stata avviata.
- L'esecuzione del metodo

```
public void logout();
```

- Il pulsante tramite il quale è possibile avviare la simulazione di KD

```
CustButton kd=new CustButton("KD Simulation");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter “failuresChangeUsername”, se l'attività del cambio dell'username è già stato avviato.
- L'incremento del counter “failuresDeleteUsername”, se l'attività di eliminazione dell'account è già stata avviata.
- L'incremento del counter “failuresChangePassword”, se l'attività del cambio della password è già stata avviata.

–

- Se non ancora fatto, viene avviato il timer relativo all'attività¹⁴

```
if(getStartTimeKD()==0)
    startTimeKD();
```

- L'esecuzione del metodo

```
public void startSimulation(String usr);
```

- Il pulsante tramite il quale è possibile mostrare i risultati della simulazione di KD

```
CustButton resultKD=new CustButton("Show Result");
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'esecuzione del metodo

```
public void showResult();
```

- Questo pulsante viene mostrato solo se, in precedenza, è già stata completata la simulazione di KD

```
if(testKD!=0){
    resultKD.setVisible(true);
    ...
}
else{
    resultKD.setVisible(false);
    ...
}
```

¹⁴Questa attività viene valutata in modo differente rispetto alle altre, dal momento che viene creata una finestra apposita, all'interno di un evento AWT separato dal processo principale. Per questo motivo, l'interazione con gli altri pulsanti non influisce negativamente sul numero di fallimenti legati alla simulazione di KD.

public void paintx(Graphics g);

Metodo che si occupa di definire la struttura della pagina personale dell'utente. Tra le altre cose viene inoltre stampato, nell'angolo in alto a destra, il nome dell'utente

```
w = g.getFontMetrics().stringWidth(usrname) + 60;  
g2d.drawString(usrname, getWidth()-w, 120);
```

public static void disableAll();

Metodo utilizzato per pulire l'interfaccia da tutte le schermate richiamabili dalla pagina personale. Viene tipicamente utilizzato prima di inizializzare una specifica attività.

public static void showHome();

Metodo che viene richiamato alla pressione del tasto "HOME". Esso si occupa di rendere visibile la schermata principale, andando a chiudere tutte le altre

```
disableAll();  
h.setVisible(true);
```

public void showDel();

L'attività di cancellazione dell'account viene richiamata dal metodo showDel() ma utilizza una classe dedicata, chiamata UserDel.java (vedi paragrafo [A.10.7](#)). In questo caso viene inizializzata la variabile "delpn" passando l'username dell'utente

```
static UserDel delpn;  
delpn=new UserDel(usrname);  
...  
delpn.setVisible(true);
```

public void startSimulation(String usr);

La simulazione di KD viene richiamata dal metodo startSimulation() ma utilizza un package apposito, chiamato kdPost (vedi paragrafo [A.11](#)). Prima di fare ciò, viene però scritto l'username dell'utente sul file temporaneo "TmpPattern.txt", che verrà poi utilizzato dal processo AWT.

public void showChgName();

L'attività di cambio dell'username viene richiamata dal metodo showChgName() ma utilizza una classe dedicata, chiamata ChgUsr.java (vedi paragrafo [A.10.4](#)). In questo caso viene inizializzata la variabile "cu" passando l'username dell'utente

```
static ChgUsr cu;  
cu=new ChgUsr(username);  
...  
cu.setVisible(true);
```

public void showChgPass();

L'attività di cambio della password viene richiamata dal metodo showChgPass() ma utilizza una classe dedicata, chiamata ChgPassPn.java (vedi paragrafo [A.10.3](#)). In questo caso viene inizializzata la variabile "passpn" passando l'username dell'utente

```
static ChgPassPn passpn;  
passpn=new ChgPassPn(username);  
...  
passpn.setVisible(true);
```

public void logout();

L'attività di logout viene richiamata dal metodo logout() ma utilizza una classe dedicata, chiamata LogoutPn.java (vedi paragrafo [A.10.5](#)). In questo caso viene inizializzata la variabile "lgpn"

```
static LogoutPn lgPn;  
lgPn=new LogoutPn();  
...  
lgPn.setVisible(true);
```

public void showResult();

Metodo che viene richiamato dal pulsante "SHOW", che permette di mostrare il risultato della simulazione

```
switch(testKD){  
case 1:  
    result = "Match";  
    break;  
case 0:  
    result = "No simulation result is founded";  
    break;  
case -1:  
    result = "No match";  
    break;  
default:  
    result = "Error #1";  
}
```

Il risultato viene analizzato dalla variabile “testKD” e scritto su una stringa “result”, che verrà mostrata poi su schermo tramite la classe ShowKd.java (vedi paragrafo [A.10.6](#))

```
static ShowKd showkd;  
showkd = new ShowKd(result);  
...  
showkd.setVisible(true);
```

A.10.2 UserFrame.java

La classe UserFrame si occupa di definire alcune caratteristiche della schermata utente come

- Il font ed il colore di alcuni elementi.
- Il comportamento che deve avere il sistema alla chiusura della finestra

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

- Il fatto che la finestra non possa essere ridimensionata

```
setResizable(false);
```

A.10.3 ChgPassPn.java

La classe ChgPassPn viene utilizzata per costruire l’attività di cambio della password, una volta premuto il relativo pulsante. La schermata (vedi figura [A.9](#)) è composta dai seguenti elementi:

- Una griglia, usata dall’utente per immettere la password.
- Un pulsante “CLEAR”, che cancella la password inserita.
- Un pulsante “SHOW PASSWORD”, che mostra l’ordine temporale con il quale è stata inserita la password.
- Un pulsante “CANCEL”, con il quale si torna alla schermata precedente.
- Un pulsante “OK”, con il quale confermare i dati inseriti.

La maggior parte degli elementi presenti nell’interfaccia, riprende quelli analizzati in precedenza all’interno della schermata di registrazione (discussi nel paragrafo [A.4](#)). In questa sezione andremo quindi ad analizzare solo i metodi che differiscono dal modello precedente.

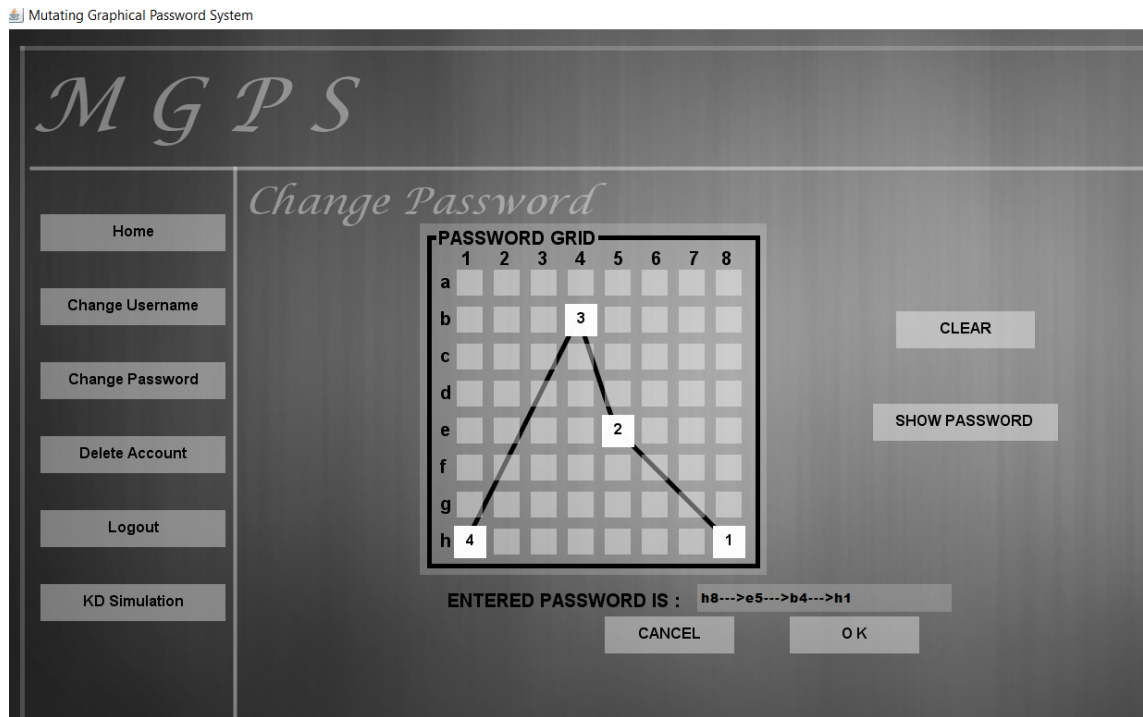


Figura A.9. Attività di cambio password del sistema di autenticazione.

```
public void procOK();
```

La procedura di conferma si compone da una prima fase in cui viene aggiornato il file “UserCredentials.txt”

```
private void changePassword(String username, String password);
```

Dopo avere aggiornato i dati dell’utente, viene stoppato il timer, terminando in questo modo i test sull’attività. I dati raccolti vengono poi inseriti all’interno di una struttura JSON, tramite la classe WriteTest.java che verrà poi discussa nel paragrafo [A.12.1](#).

```
Long duration = System.nanoTime() -
                    UserPn.getStartTimeChangePassword();
Float durationF = Float.valueOf(duration);
durationF=durationF/1000000000;
...
WriteTest.writeOnFile("ChangePassword", durationF,
    counter,UserPn.getFailuresChangePassword());
```

Viene infine chiusa la finestra corrente e richiamata la pagine iniziale

```
setVisible(false);
UserPn.showHome();
```

```
private void changePassword(String username, String password);
```

Metodo che si occupa di aggiornare il file “UserCredentials.txt” con la nuova password. Come prima cosa, viene localmente ricostruito il file all’interno di un’apposita lista. Se durante la lettura viene rilevato un match con la variabile “username”, viene aggiunta la nuova password (opportunamente cifrata tramite la classe MyCipher.java) anziché quella vecchia.

Per ogni riga letta, all’interno di un ciclo while, vi sono tre sotto-casi possibili:

1. Se la riga letta è vuota, si passa a quella successiva

```
if(sCurrentLine.compareTo("")==0){  
    break;  
}
```

2. Se la riga letta è uguale all’username di cui si sta cambiando la password, viene introdotta la nuova password e scartata quella vecchia

```
else if(username.compareTo(MyCipher.decrypt  
    (key, initVector, sCurrentLine))==0){  
    newFile.add(sCurrentLine);  
    newFile.add(MyCipher.encrypt(key, initVector, password));  
    br.readLine(); //drop the old password  
}
```

3. In tutti gli altri casi, vengono aggiunti l’username e la password senza alcuna modifica

```
else{  
    newFile.add(sCurrentLine);  
    newFile.add(br.readLine());  
}
```

A questo punto la lista viene copiata all’interno del file “UserCredentials.txt”, completando in questo modo l’aggiornamento della password.

A.10.4 ChgUsr.java

La classe ChgUsr viene utilizzata per costruire l’attività di cambio dell’username, una volta premuto il relativo pulsante. La schermata (vedi figura [A.10](#)) è composta dai seguenti elementi:

- Una scritta statica che mostra l’username corrente.
- Un form all’interno del quale inserire il nuovo username.
- Un pulsante “CANCEL”, con il quale si torna alla schermata precedente.
- Un pulsante “OK”, con il quale confermare i dati inseriti.

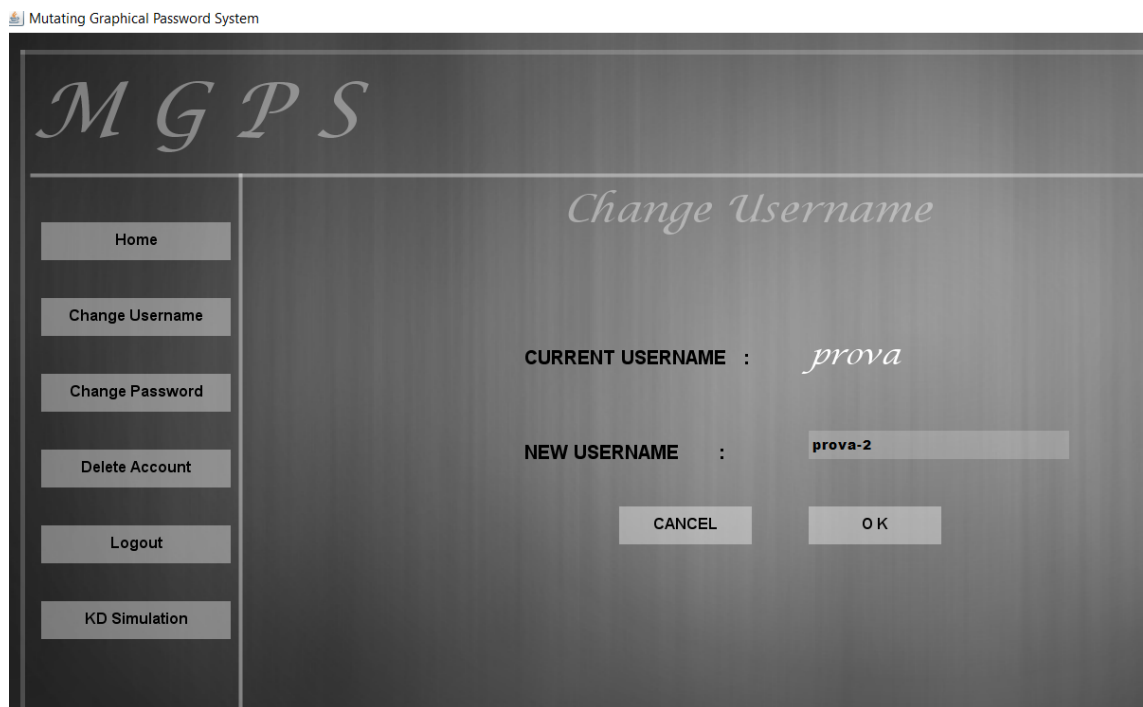


Figura A.10. Attività di cambio username del sistema di autenticazione.

```
public ChgUsr(String usr);
```

Il metodo costruttore ha il compito di inizializzare tutti gli elementi grafici presenti all'interno dell'interfaccia:

- La costruzione del form username

```
TextField txtUsr;  
txtUsr = new TextField();
```

Ad esso è associato:

- Un `KeyListener` che viene abilitato al rilascio di ogni pulsante della tastiera e che scatena il seguente evento:

```
public void keyReleased(KeyEvent e) {  
    procPassUser(e);  
}
```

- Un `MouseListener` che tiene traccia dei click del mouse effettuati, per la valutazione dei test oggettivi

```
public void mousePressed(MouseEvent e) {  
    ...  
    UserPn.increaseCounterChangeUsername();  
}
```

- Il pulsante “CANCEL” a cui è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell’azione appena eseguita all’interno di una lista (utilizzata per monitorare le azioni dell’utente durante il test).
- L’incremento del counter relativo al numero di fallimenti effettuati durante l’attività.
- L’incremento del counter relativo al numero di click effettuati durante la fase di login.
- L’esecuzione del metodo

```
public void procCancel();
```

- Il pulsante “OK” a cui è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell’azione appena eseguita all’interno di una lista (utilizzata per monitorare le azioni dell’utente durante il test).
- L’incremento del counter relativo al numero di click effettuati durante la fase di login.
- L’esecuzione dei metodi

```
if(procOK())  
    setUsr();
```

```
public void paintx(Graphics g);
```

Metodo che si occupa di definire la struttura dell’attività di cambio dell’username, secondo lo struttura mostrata nella figura [A.10](#).

```
private void procPassUser(KeyEvent e);
```

Il metodo `procPassUser` rileva, come prima cosa, se sia stato premuto il tasto `enter`

```
if(e.getKeyChar()==KeyEvent.VK_ENTER)
```

In caso affermativo, viene eseguita la stessa procedura che viene eseguita quando si preme il tasto “OK”.

```
public boolean procOK();
```

Metodo che viene utilizzato per verificare i dati inseriti dall’utente, prima dell’eventuale convalida. Esso si basa su un unico costrutto `if-else`:

- Se l’utente inserito ha una dimensione pari a 0, il campo non è considerato valido. Viene quindi settato il messaggio d’errore ed inserita l’azione all’interno della lista, prima di ritornare il valore “false”

```
if(txtUsr.getText().length()==0){  
    ...  
    return false;  
}
```

- Se il metodo findUser torna un valore “true”, significa che l’username inserito esiste già all’interno del DB. A questo punto viene settato il messaggio d’errore ed inserita l’azione all’interno della lista, prima di ritornare il valore “false”

```
else if(findUser(txtUsr.getText())){  
    ...  
    return false;  
}
```

- In tutti gli altri casi il valore inserito viene considerato valido e la funziona torna un valore “true”.

private boolean findUser(String user);

Metodo utilizzato per verificare se l’username inserito dall’utente non sia già presente all’interno del DB. La verifica viene fatta leggendo il file “UserCredentials.txt”, andando ad iterare su ogni singola riga tramite un ciclo while

```
while ((sCurrentLine = br.readLine()) != null){  
    ...  
}
```

Il metodo torna il valore “true” se viene verificata una corrispondenza, “false” nel caso opposto.

public void setUsr();

Metodo che si occupa di:

- Aggiornare il file “UserCredentials.txt” con il nuovo username

```
findAndWrite(username,txtUsr.getText());
```

- Settare il nuovo username all’interno della schermata utente

```
username=txtUsr.getText();  
UserPn temp=(UserPn)getParent();  
temp.setUsr(username);
```

- Terminare ed inviare i test relativi all’attività di cambio username

```
WriteTest.writeOnFile("ChangeUsername", durationF,  
    counter,UserPn.getFailuresChangeUsername());
```

```
private void findAndWrite(String oldUser, String newUser);
```

L'aggiornamento dell'username segue un meccanismo simile a quello visto nel metodo `changePassword` della classe `ChgPassPn.java`, discusso nel paragrafo [A.10.3](#):

- Lettura del file e sua creazione locale.
- Aggiornamento dei dati con il nuovo username.
- Scrittura dei nuovi dati sul file.

A differenza della password, l'username è un valore presente sia nel file “User-Credentials.txt” che in “UserPattern.txt” e, di conseguenza, l'azione dovrà essere replicata su entrambi i file¹⁵.

```
public void procCancel();
```

Metodo che permette di ritornare alla schermata iniziale della pagina utente

```
UserPn.showHome();
```

A.10.5 LogoutPn.java

La classe `LogoutPn` viene utilizzata per costruire l'attività di logout, una volta premuto il relativo pulsante. La schermata (vedi figura [A.11](#)) è composta dai seguenti elementi:

- Una scritta che chiede la conferma di effettuare il logout.
- Un pulsante “NO”, con il quale si torna alla schermata precedente.
- Un pulsante “YES”, con il quale confermare la propria scelta.

Il processo di logout (come quello di eliminazione dell'account [A.10.7](#)) prevede l'invio di tutti i dati raccolti durante la simulazione e rappresenta l'attività conclusiva del test.

¹⁵Dato che il modo in cui viene effettuata l'operazione è molto simile a quella presente nel metodo `changePassword`, rimando alla lettura del paragrafo [A.10.3](#) per maggiori dettagli sul funzionamento del meccanismo.

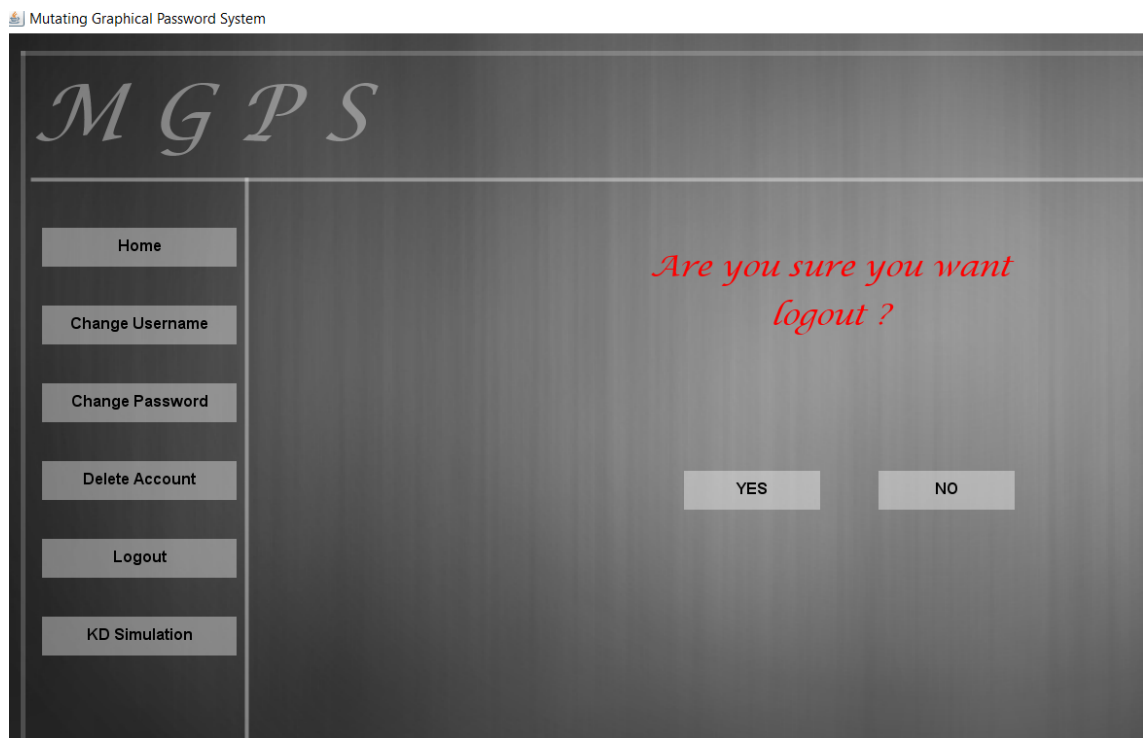


Figura A.11. Attività di logout del sistema di autenticazione.

public LogoutPn();

Il metodo costruttore della classe prevede l’inizializzazione di tutti gli elementi grafici presenti all’interno dell’interfaccia:

- Un pulsante “YES”, a cui è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell’azione appena compiuta all’interno di una lista.
- La creazione di una variabile “message”, che verrà inizializzata con la lista di tutte le azioni compiute dall’utente durante la simulazione

```
String message= WriteTest.getList();
```

- L’invio di tutti i dati raccolti, in formato JSON

```
SendDataResult.sendEmail(UserPn.usrname, message);
```

- L’esecuzione del metodo

```
public void procOK();
```

- Un pulsante “NO”, a cui è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell’azione appena compiuta all’interno di una lista.

- L’esecuzione del metodo

```
public void procCancel();
```

public void paintx(Graphics g);

Metodo che si occupa di definire gli aspetti grafici della scritta che compare a schermo.

public void procCancel()

Metodo che svolge due azioni:

1. Eliminare la parte grafica relativa al logout
`setVisible(false);`
2. Riabilitare la schermata principale della pagine utente
`UserPn.showHome();`

public void procOK();

Metodo che svolge due azioni:

1. Eliminare la parte grafica relativa al logout
`setVisible(false);`
2. L'esecuzione del metodo `disableUsr()` che chiude il programma corrente
`WelcomePanel.disableUsr();`

A.10.6 ShowKd.java

La classe `ShowKd` viene utilizzata per abilitare il pulsante “SHOW” (vedi figura A.12), una volta completata la simulazione di KD. Esso mostra unicamente il risultato dell'ultimo test e non tiene traccia di quelli precedenti.

public ShowKd(String input);

Il costruttore della classe `ShowKd`, si occupa di inizializzare la variabile “input” con il risultato della simulazione

```
this.input = input;
```

Esso verrà utilizzato dal metodo successivo per mostrare a schermo il relativo output, con la relativa formattazione.

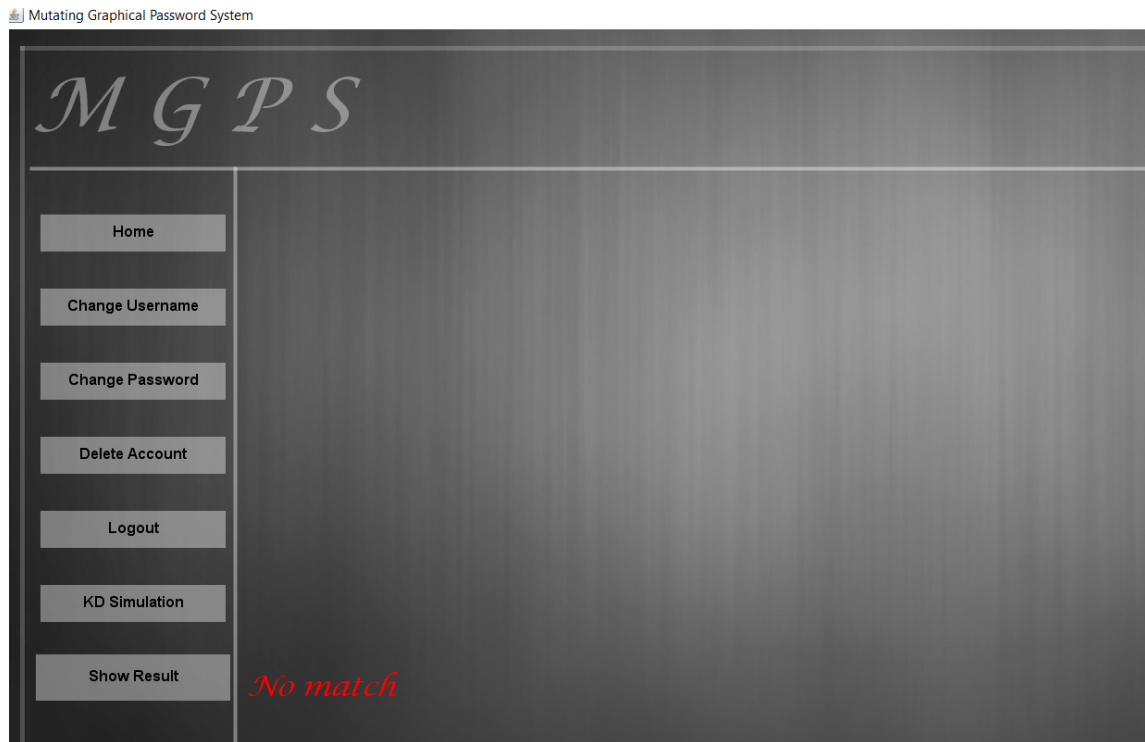


Figura A.12. Pulsante “SHOW” del sistema di autenticazione.

```
public void paintx(Graphics g);
```

In base al valore contenuto nella variabile “input” (inizializzata dal metodo costruttore), possono verificarsi due casi:

1. È avvenuto un match

```
if(input.compareTo("Match")==0)
    g2d.setColor(Color.GREEN);
```

2. Non è avvenuto un match (o si è riscontrato un errore durante la valutazione)

```
else
    g2d.setColor(Color.RED);
```

A.10.7 UserDel.java

La classe UserDel viene utilizzata per costruire l’attività di cancellazione dell’account, una volta premuto il relativo pulsante. La schermata (vedi figura A.13) è composta dai seguenti elementi:

- Una scritta che chiede la conferma per la cancellazione dell’account.
- Un pulsante “NO”, con il quale si torna alla schermata precedente.

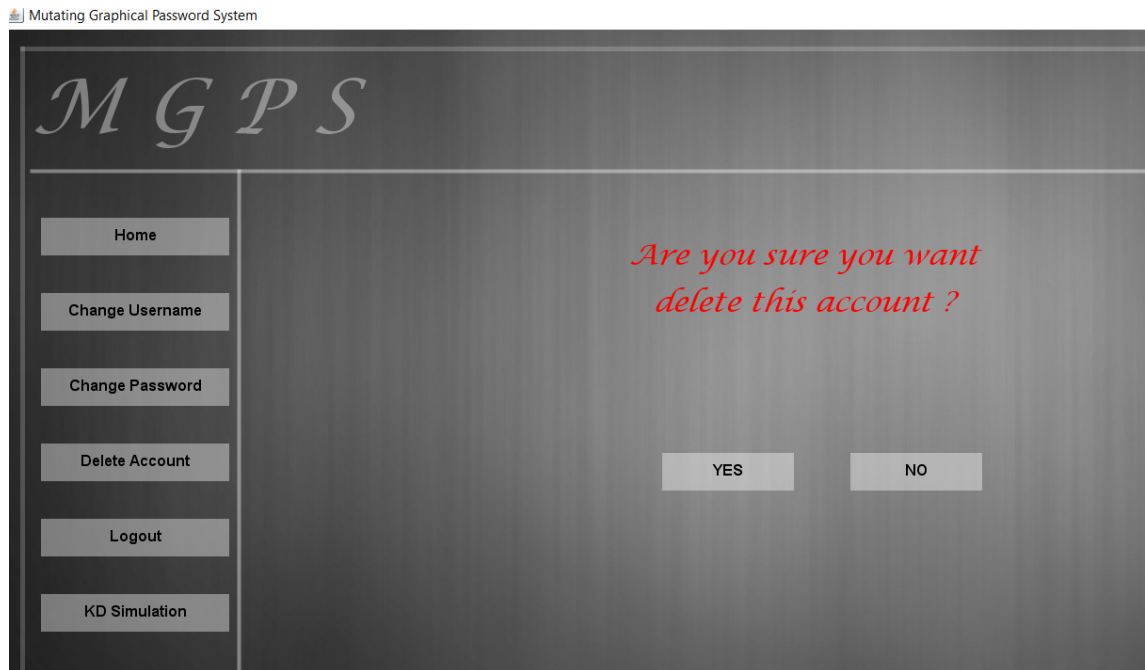


Figura A.13. Attività di cancellazione account del sistema di autenticazione.

- Un pulsante “YES”, con il quale confermare la propria scelta.

Il processo di cancellazione dell’account (come quello di logout, discusso nel paragrafo [A.10.5](#)) prevede l’invio di tutti i dati raccolti durante la simulazione e rappresenta l’attività conclusiva del test.

```
public UserDel(String usr);
```

Il metodo costruttore della classe prevede l’inizializzazione di tutti gli elementi grafici presenti all’interno dell’interfaccia:

- Un pulsante “YES”, a cui è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:
 - L’incremento del counter relativo al numero di click effettuati durante l’attività.
 - La scrittura dell’azione appena compiuta all’interno di una lista.
 - Viene stoppato il timer relativo all’attività e calcolato il tempo totale trascorso per completarla.
 - La creazione di una variabile “message”, che verrà inizializzata con la lista di tutte le azioni compiute dall’utente durante la simulazione

```
String message= WriteTest.getList();
```

- L’invio di tutti i dati raccolti, in formato JSON

```
SendDataResult.sendEmail(UserPn.username, message);
```

- L'esecuzione del metodo

```
public void procOK();
```

- Un pulsante “NO”, a cui è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena compiuta all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante l'attività.
- L'incremento del numero dei fallimenti relativi all'attività.
- L'esecuzione del metodo

```
public void procCancel();
```

```
public void paintx(Graphics g);
```

Metodo che si occupa di definire gli aspetti grafici della scritta che compare a schermo.

```
public void procCancel();
```

Metodo che svolge due azioni:

1. Eliminare la parte grafica relativa alla cancellazione dell'account

```
setVisible(false);
```

2. Riabilitare la schermata principale della pagine utente

```
UserPn.showHome();
```

```
public void procOK();
```

Metodo che svolge tre azioni:

1. Richiamare il metodo con il quale verrà poi eliminato l'account

```
delUsr(username);
```

2. Eliminare la parte grafica relativa alla cancellazione dell'account

```
setVisible(false);
```

3. L'uscita dal programma, tramite la funzione `System.exit(0)` richiamata dal metodo

```
WelcomePanel.disableUsr();
```

```
private void delUsr(String user);
```

La cancellazione dell'account segue un meccanismo simile a quello visto nel metodo `changePassword` della classe `ChgPassPn.java` (discusso nel paragrafo [A.10.3](#)) e del metodo `findAndWrite` della classe `ChgUsr.java` (discusso nel paragrafo [A.10.4](#)):

- Lettura del file e sua creazione locale.
- Eliminazione delle credenziali relative all'username "user".
- Scrittura dei nuovi dati sul file.

Come accade per la classe `ChgUsr.java`, l'username è un valore presente sia nel file "UserCredentials.txt" che in "UserPattern.txt" e, di conseguenza, l'azione dovrà essere replicata su entrambi i file¹⁶.

A.10.8 UsrHome.java

Questa classe si occupa di settare l'username all'interno della pagina personale dell'utente. Sono principalmente due i metodi che si occupano di questo compito:

- Il metodo costruttore

```
public UsrHome(String usr);
```

- La funzione `setUsr`

```
public void setUsr(String usr);
```

```
public void paintx(Graphics g);
```

Metodo che si occupa di definire gli aspetti grafici di tutte le scritte presenti nella schermata.

A.11 kdPost package

Il package `kdPost` si occupa di gestire tutta la parte di valutazione riguardante la simulazione di `Keystroke Dynamics`. Buona parte del sistema riprende la classe `AWT` usata in fase di registrazione: per questo motivo, per i dettagli sulla parte grafica e sul funzionamento del processo `AWT`, rimando alla lettura del paragrafo [A.5.1](#). In questa sezione verranno trattate unicamente le differenze o gli aspetti cruciali che intercorrono all'interno del processo di verifica.

¹⁶Dato che il modo in cui viene effettuata l'operazione è molto simile a quella presente nel metodo `changePassword`, rimando alla lettura del paragrafo [A.10.3](#) per maggiori dettagli sul funzionamento del meccanismo.

A.11.1 AWTPost.java

Una prima differenza che la classe AWTPost ha rispetto ad AWT, è la lettura del linguaggio per la costruzione dell'interfaccia grafica¹⁷. Nella classe AWTPost, infatti, è necessario come prima cosa aprire il file “TmpPattern.txt” per estrapolare l'username dell'utente:

```
user= MyCipher.decrypt(key, initVector, br.readLine());
```

Dopodiché viene utilizzata la variabile “user” per identificare all'interno del file “UserPattern.txt” quali sono il linguaggio ed il pattern associati a quell'username.

```
if(decrypt.compareTo(user)==0){  
    language = MyCipher.decrypt(key, initVector, br.readLine());  
    patternRegistration = MyCipher.decrypt(key, initVector,  
        br.readLine());  
    break;  
}
```

Dopo che l'utente ha completato la simulazione, viene richiamata la funzione `sendPostForMatch` della classe `Request` (vedi paragrafo [A.11.2](#)) per iniziare il processo di comparazione.

```
String result = Request.sendPostForMatch  
    (patternRegistration, typingPattern);
```

```
//patternRegistrazione è il pattern ricavato durante la  
    registrazione  
//typingPattern è il pattern ricavato durante la simulazione
```

Il risultato viene poi inserito all'interno di una struttura JSON

```
JSONParser parser = new JSONParser();  
JSONObject json = null;  
json = (JSONObject) parser.parse(result);
```

per poi ricavare il risultato delle query all'interno del campo “success”

```
long succes = (long) json.get("success");
```

In base al valore di quest'ultima variabile, viene infine inizializzato l'elemento “UserPn.TestKD”, che verrà utilizzato dalla classe `ShowKd.java` (vedi paragrafo [A.10.6](#)) per mostrare l'esito della simulazione all'utente.

A.11.2 Request.java

La classe `Request` si occupa di creare ed inoltrare la richiesta di matching al web-service di `typingDNA`. Per fare questo, vengono utilizzate due stringhe alfanumeriche, “apiKey” e “apiSecret”, che permettono di accedere un numero limitato di

¹⁷Il meccanismo di costruzione rimane lo stesso della classe AWT, ma cambiano le frasi che vengono adoperate.

volte al servizio e che sono legate ad uno specifico account che è stato creato in precedenza. Tra le altre variabile vi sono:

- L'uri associato al web-service

```
String uri = "https://api.typingdna.com/match";
```

- Il content-type

```
String contentType = "application/x-www-form-urlencoded";
```

- La precisione con il quale viene effettuata la comparazione

```
String quality = "2";
```

Il servizio permette di scegliere tra tre valori, dove '1' rappresenta la misurazione più accurata, ma che potrebbe generare più problematiche di falsi-negativi, mentre il valore '3' identifica la situazione opposta. La piattaforma raccomanda l'utilizzo del valore intermedio.

```
public static String sendPostForMatch(String pattern1, String pattern2);
```

Il metodo principale si occupa come prima cosa di creare la richiesta HTTP POST.

```
CloseableHttpClient client = HttpClients.createDefault();  
HttpPost httpPost = new HttpPost(uri);
```

Vengono poi aggiunte le credenziali ed il content-type all'header della richiesta.

```
...  
httpPost.addHeader(new BasicScheme().authenticate(creds,  
    httpPost, null));
```

Si passa poi alla realizzazione della lista dei parametri che verranno aggiunti al payload; essa è composta dalle due stringhe "pattern1" e "pattern2" e da un intero che rappresenta la qualità della misurazione.

```
List<BasicNameValuePair> params = new  
    ArrayList<BasicNameValuePair>();  
... //aggiunta dei parametri alla lista  
httpPost.setEntity(new UrlEncodedFormEntity(params));
```

Viene infine inoltrata la richiesta, chiusa la connessione e ritornata la risposta.

```
CloseableHttpResponse response = client.execute(httpPost);  
HttpEntity entity = response.getEntity();  
...  
client.close();  
return responseString;
```

A.12 test package

Il package test viene utilizzato per due obiettivi specifici:

1. Raccogliere all'interno di un oggetto JSON i dati raccolti durante la simulazione.
2. Inviare i dati via mail.

Ognuna di queste funzionalità viene evidenziata da una specifica funzione, di cui andremo ora a discutere i dettagli.

A.12.1 WriteTest.java

La classe WriteTest utilizza le seguenti variabili globali:

- Una variabile “mylist” che contiene l'elenco descrittivo delle azioni effettuate dall'utente durante la simulazione

```
private static List<String> mylist = new ArrayList<>();
```

- Una struttura JSON “obj”, all'interno del quale verranno inseriti tutti i dati

```
static JSONObject obj = new JSONObject();
```

- Una serie di contatori, che permettono di valutare più volte la stessa attività, all'interno della medesima simulazione

```
private static int counterRegistration = 0;
private static int counterLogin = 0;
private static int counterChangePassword = 0;
private static int counterChangeUsername = 0;
private static int counterDeleteUsername = 0;
private static int counterKDSimulation = 0;
```

```
public static void insertList(String data);
```

Metodo che inserisce un dato descrittivo (relativo ad un'azione compiuta dall'utente) all'interno della lista “mylist”

```
mylist.add(data);
```

```
public static String getList();
```

Metodo che si occupa di inserire la lista delle azioni compiute dall'utente all'interno della struttura JSON, tramite la chiave "UserActions"

```
obj.put("UserActions", mylist);
```

Prima di inviare i dati, l'oggetto JSON deve essere tradotto in una stringa, che diventerà poi l'effettivo testo della mail.

```
return obj.toString();
```

```
public static void writeOnFile(String title, float time, int counterClick,  
int failures);
```

I dati puntuali, associati ad una specifica attività, vengono elaborati tramite la funzione writeOnFile. Essa viene richiamata al termine di ogni singola attività e contiene quattro argomenti:

- "title", il nome dell'attività.
- "time", il tempo richiesto per completare l'attività.
- "counterClick", il numero di click effettuati all'interno dell'attività.
- "failures", il numero di fallimenti riscontrati prima di terminare l'attività.

La costruzione del nome, che verrà utilizzato per inserire il risultato nell'oggetto JSON, segue la sintassi introdotta nel paragrafo 6.2.8. Per fare ciò, viene come prima cosa rilevato il tipo di attività, in modo tale da poter:

- Inserire il giusto counter all'interno del nome

```
title+="_"+counterRegistration+"_";
```

- Incrementare il counter

```
counterRegistration++;
```

Viene poi aggiunto il nome del parametro valutato, andando in questo modo a completare la stringa

```
String name = title + "Time";
```

Come ultimo passaggio viene inserito il valore del test all'interno della struttura JSON, attraverso la stringa ricavata in precedenza

```
obj.put(name, time);
```

A.12.2 SendDataResult.java

La classe SendDataResult utilizza il metodo sendEmail per inviare i dati raccolti durante la simulazione. Vediamo ora quali sono i dettagli utilizzati per inviare la mail:

- La mail utilizzata per raccogliere i test è “giovanni.usable@gmail.com”; essa viene utilizzata sia come mittente che destinatario della mail.
- Il soggetto della mail è dato da:

```
message.setSubject(name + " has completed the test!");  
//name = username dell'utente
```

- Il messaggio della mail viene ricavato dal metodo writeOnFile il quale, a partire dall'oggetto JSON, ricava la stringa associata.

A.13 Creazione del file eseguibile

Vediamo ora come avviene la creazione del file eseguibile, che verrà utilizzato dall'utente per lanciare il test, a partire dal progetto eclipse.

Come prima cosa viene estratto il “Runnable JAR file”, andando a selezionare la classe Home (vedi paragrafo [A.3.1](#)) come punto di partenza per il lancio del programma.

Infine, dopo avere generato il file JAR, ho utilizzato il tool jsmooth [\[169\]](#) per ricavare il file eseguibile, andando ad associare tutte le librerie esterne di cui il programma è dipendente (vedi paragrafo [A.1.2](#)).

Appendice B

Manuale utente

In questa breve guida verranno spiegati alcuni elementi fondamentali che ti permetteranno di eseguire al meglio questa simulazione.

B.1 Requisiti minimi di sistema

Il sistema di autenticazione è stato sviluppato interamente in Java ed è compatibile unicamente con il sistema operativo Windows 8 o versioni successive. Per motivi di privacy, segnalo che durante l'esecuzione del programma verranno registrate, a scopi statistici, tutte le attività effettuate. Per il corretto invio dei risultati raccolti, è necessaria infine una connessione ad internet costante durante l'intera durata della simulazione.

B.2 Installazione dell'ambiente

L'esecuzione del programma richiede l'installazione di un Java Development Kit:

- Come prima cosa, collegati alla seguente pagina,
[-->CLICCAQUI](#)
- Scarica il file “Java SE Development Kit 8u202”, come mostrato in figura [B.1](#), scegliendo l'architettura adeguata. Ricordati di accettare gli accordi di licenza prima di scaricare il relativo file.

Java SE Development Kit 8u202		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
☐ Accept License Agreement ☒ Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.86 MB	jdk-8u202-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.75 MB	jdk-8u202-linux-arm64-vfp-hflt.tar.gz
Linux x86	173.08 MB	jdk-8u202-linux-i586.rpm
Linux x86	187.9 MB	jdk-8u202-linux-i586.tar.gz
Linux x64	170.15 MB	jdk-8u202-linux-x64.rpm
Linux x64	185.05 MB	jdk-8u202-linux-x64.tar.gz
Mac OS X x64	249.15 MB	jdk-8u202-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	125.09 MB	jdk-8u202-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	88.1 MB	jdk-8u202-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	124.37 MB	jdk-8u202-solaris-x64.tar.Z
Solaris x64	85.38 MB	jdk-8u202-solaris-x64.tar.gz
Windows x86	201.64 MB	jdk-8u202-windows-i586.exe
Windows x64	211.58 MB	jdk-8u202-windows-x64.exe

Figura B.1. Pagina di download di Oracle.

- Procedi ora con l'installazione, accettando gli accordi di licenza e lasciando come cartella di destinazione quella proposta dal sistema.

A questo punto è necessario settare la variabile d'ambiente "JAVA_HOME" all'interno del sistema operativo di Windows:

- Nella barra di ricerca di Windows, digita "Sistema" e seleziona il primo risultato.
- Fai click su "Impostazioni di sistema avanzate", situato nella parte sinistra della schermata.
- Seleziona "Variabili di ambiente", posto in basso a destra.
- A questo punto clicca sul secondo pulsante "Nuovo...", posto nella parte inferiore della schermata (all'interno della sezione "Variabili di sistema")
- Inserisci ora la variabile d'ambiente "JAVA_HOME", come mostrato in figura B.2. Il percorso da inserire rappresenta quello che si è scelto durante l'installazione del JDK che, di default, è

C:\Program Files\Java\jdk1.8.0_202

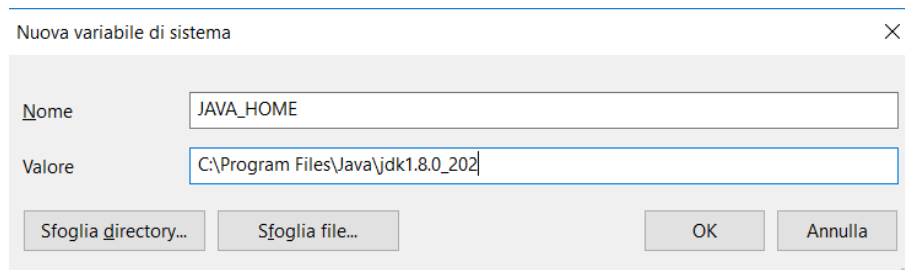


Figura B.2. Schermata “Variabili d’ambiente”.

- Premi “OK” per confermare l’operazione.

B.3 Simulazione

La simulazione si compone in tutto di tre fasi:

B.3.1 Pre-test

La prima parte è composta da un breve test di autovalutazione, che servirà ad avere un quadro generale sulle tue capacità informatiche. Il test è composto da 4 domande, a cui dovrai rispondere sinceramente, e che potrai trovare a questo link:

<https://goo.gl/forms/OUFdnK6zEUD0UjSz2>

B.3.2 Sistema di autenticazione

A questo punto inizierà la parte principale della simulazione, che consiste nella prova del sistema di autenticazione. Dopo avere scompattato l’archivio che hai trovato in allegato a questo documento, fai doppio click sul file “start.exe” per lanciare il programma.

Le attività **obbligatorie** che dovrai compiere sono:

1. Effettuare la registrazione, cliccando sul tasto “sign up” che ti apparirà nella schermata principale. All’interno di questa schermata, troverai una griglia che dovrà essere utilizzata per inserire la password grafica (composta da almeno tre punti), oltre che ad uno specifico username. Ad esempio, nella figura B.3, viene mostrato l’inserimento della password con coordinate h1 → h8 → a8. Dopo avere confermato la password, ti verrà proposto di scrivere un breve testo, in italiano o in inglese.
2. Una volta conclusa la fase di registrazione, dovrai effettuare l’accesso al sistema cliccando sul tasto “login”. Nella schermata di login ti verrà chiesto di digitare le relative coordinate nell’apposita area (come mostrato in figura B.4), oltre che l’username scelto in fase di registrazione. Ogni volta che inserisci un punto, verrà generata una griglia differente.

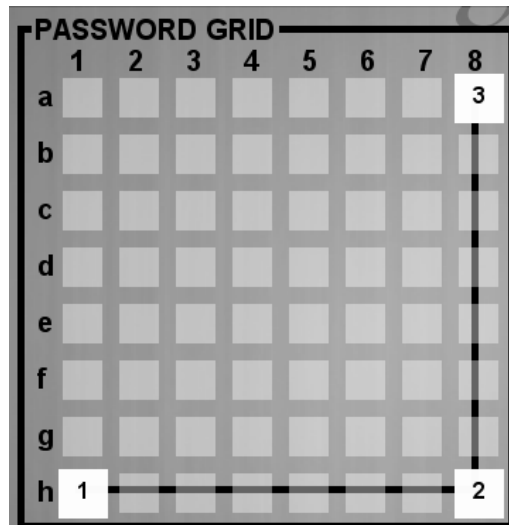


Figura B.3. Inserimento della password all'interno della schermata di registrazione.



Figura B.4. Inserimento della password all'interno della schermata di login. La sequenza inserita corrisponde a quella rappresentata nell'esempio di registrazione (vedi figura B.3).

3. Dopo che sarai entrato nella tua pagina personale, dovrai svolgere una serie di attività secondarie:
 - Cambio dell'username scelto in fase di registrazione.
 - Cambio della password scelta in fase di registrazione.
 - Simulazione di una breve attività di scrittura (usa il tasto "KD Simulation").
 - Lettura del risultato della simulazione appena completata, tramite il pulsante "SHOW KD".
4. Per terminare il programma puoi scegliere di:

- Effettuare il logout.
- Eliminare il proprio account.

Non chiudere in alcun modo il programma tramite il pulsante “X” posto in alto a destra in quanto, questa procedura, annullerebbe l’invio di tutti i dati raccolti durante la simulazione.

Puoi scegliere (facoltativo) di completare più volte tutte le attività appena citate, all’interno della stessa simulazione.

B.3.3 Test finale

L’ultima parte è composta da un test, tramite il quale dovrai esprimere un giudizio sul sistema di autenticazione che hai appena testato. Esso è composto da una serie di domande obbligatorie a risposta chiusa e da una serie di domande facoltative a risposta libera. Anche in questo caso, al fine di rendere i risultati attendibili, è necessario risponde con la massima sincerità. Di seguito, il link al test:

<https://goo.gl/forms/XccJzIOfXVn6mPXy2>

Appendice C

Applicazione del prototipo all'interno di uno scenario avanzato

A partire dal prototipo presentato nel capitolo 7, ho realizzato una simulazione avanzata che permette di verificare il comportamento del sistema all'interno di un'attività più complessa: la scrittura di una mail. Il programma non subirà un processo di testing come è accaduto per il prototipo originale, ma permetterà di fornire unicamente un'indicazione su come sia possibile integrare un meccanismo di continuous authentication all'interno di uno scenario reale. Vedremo quindi in questa appendice quali sono i dettagli implementativi e concettuali del prototipo di autenticazione avanzato.

C.1 Funzionamento generale

La modifica principale del prototipo è definita dalla simulazione di keystroke dynamics, da avviare dalla homepage principale, subito dopo avere effettuato il login. L'interfaccia che viene proposta (vedi figura C.1) è composta da:

- Un form in cui inserire il destinatario della mail.
- Un form in cui inserire il testo della mail.
- Un pulsante per confermare l'invio della mail.
- Un pulsante che permette di cancellare il testo precedentemente inserito.

Nella colonna a sinistra è presente un breve testo, il cui compito è quello di fornire alcune direttive su cosa dovrà essere inserito all'interno della mail. Questo viene fatto per impedire che l'utente possa scrivere un testo troppo breve e, quindi, non efficace al fine del riconoscimento di keystroke dynamics. A parte questo, non viene posto alcun altro limite su come deve essere composto il testo, lasciando all'utente la massima libertà.

Al termine della scrittura della mail, possono verificarsi due situazioni differenti:

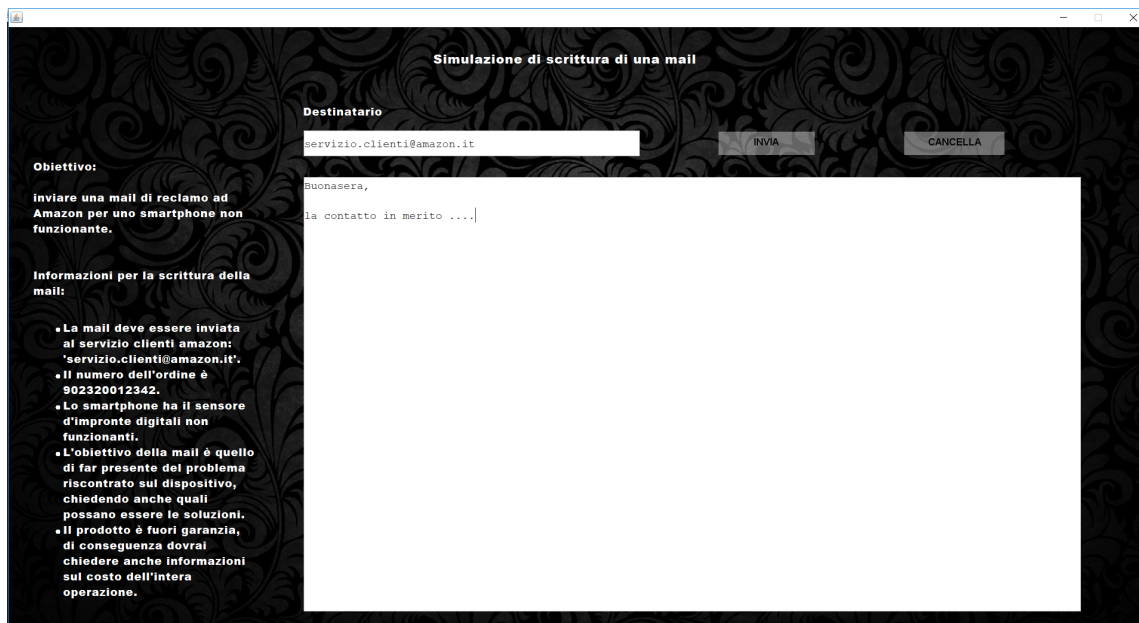


Figura C.1. Ambiente utilizzato per scrivere la mail, all'interno del sistema di autenticazione avanzato.

1. Il sistema riconosce correttamente l'utente e l'email viene inviata. In questo caso viene mostrato un messaggio di corretto invio della mail e si ritorna alla schermata principale.
2. Il sistema non riconosce l'utente e l'email non viene inviata. In questo caso viene proposta una schermata simile a quella di login (vedi figura C.2) in cui bisogna inserire la password grafica associata al proprietario: se la sequenza è corretta la mail viene spedita mentre, in caso contrario, il sistema viene bloccato. Ogni tentativo di chiusura forzata (tramite la "X" posta in alto a destra) della schermata di blocco causerà l'arresto dell'intero sistema.

È da notare che non viene segnalato alcun messaggio tra la conferma di invio della mail e la schermata di blocco. Questa scelta è intenzionale e viene fatta per aumentare il livello di sicurezza del sistema: un ipotetico malintenzionato avrà infatti molte più difficoltà a sbloccare il sistema, dal momento che non gli viene fornita alcuna informazione sui passaggi che deve fare per cercare di completare l'autenticazione¹.

¹Information leakage.

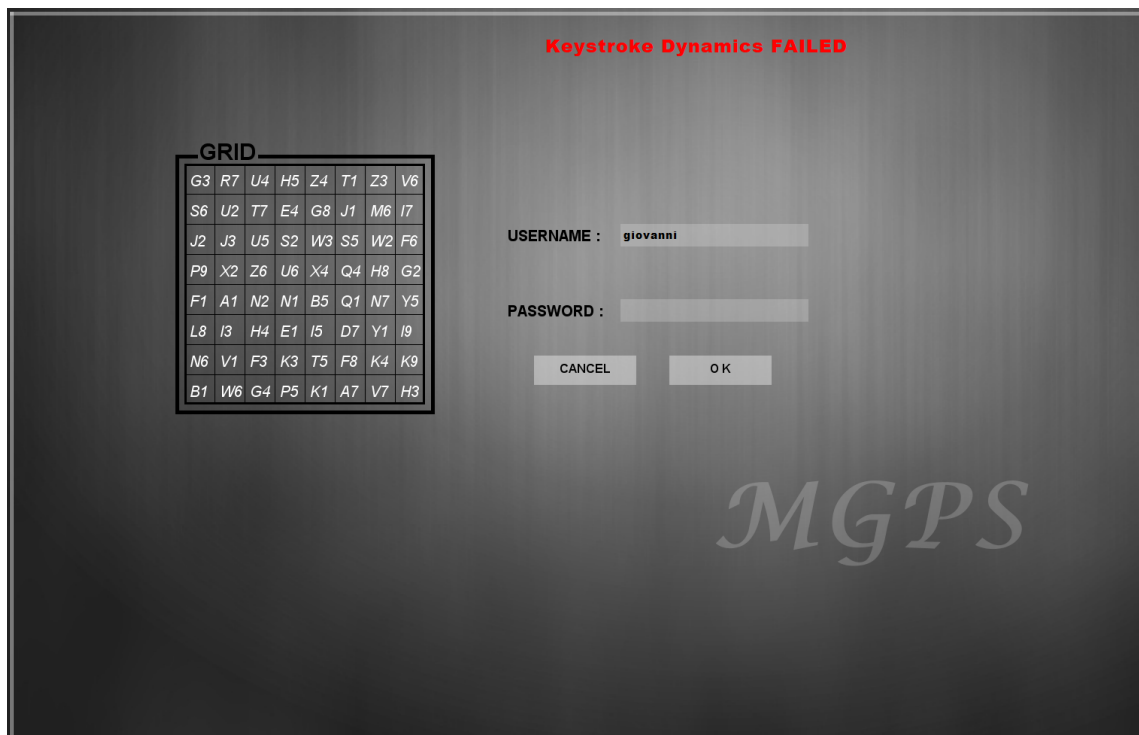


Figura C.2. Schermata di blocco del prototipo avanzato.

C.2 Manuale del programmatore del prototipo avanzato

Vediamo ora, a livello di codice, come è stata sviluppata la parte relativa alla nuova simulazione di KD².

C.2.1 Registrazione, login ed attività secondarie

Tutte ciò che riguarda la fase di registrazione, login, cambio username, cambio password, logout e cancellazione dell'account riprende fedelmente le caratteristiche già evidenziate all'interno del prototipo originale.

A livello di programmazione sono state aggiunte alcuni elementi all'interno della classe "Home.java", contenuta nel package "HomePage", che riguardano:

- La creazione e l'inizializzazione dell'oggetto "LoginEmail", utilizzato per gestire l'interfaccia di blocco quando il riconoscimento di KD fallisce

```
static LoginEmail le;
le = new LoginEmail();
```

- Il metodo utilizzato per abilitare la schermata di blocco

²I riferimenti a meccanismi o concetti già presenti nel prototipo originale verranno omessi.

```
public static void loginEmailClick();
```

- Il metodo utilizzato per disabilitare la schermata di blocco

```
public static void disableLoginEmail()
```

C.2.2 Simulazione di keystroke dynamics avanzata

Per realizzare l'ambiente di simulazione di scrittura di una email è stata modificata la classe "AWTPost.java", contenuta all'interno del package "kdPost". A livello di variabili viene mantenuto lo stesso set del prototipo originale, che permetteva di avere un'interfaccia in lingua italiana o inglese rispetto alla scelta effettuata in fase di registrazione. Vengono tuttavia cambiati i valori testuali di quelle variabili, al fine di avere un ambiente coerente con la nuova attività di scrittura.

A livello grafico, la nuova interfaccia è composta dai seguenti elementi:

- Un oggetto JLabel, che viene reso visibile nell'istante in cui non viene inserito alcun destinatario

```
JLabel ErrLabel = new JLabel("The email inserted is not  
valid");
```

La formattazione utilizzata per definire il messaggio d'errore è composta dai seguenti elementi

```
ErrLabel.setBounds(493+150, 112, 404, 59);  
ErrLabel.setForeground(Color.RED);  
ErrLabel.setFont(new Font("Lucida  
Calligraphy",Font.ITALIC,20));
```

- Un oggetto JLabel utilizzato per presentare l'attività di scrittura della mail, inizializzato in base alla lingua scelta in fase di registrazione

```
JLabel lblNewLabel = null;  
  
if(language.compareTo("ITALIANO")==0)  
    lblNewLabel = new  
        JLabel(introductionItalian,SwingConstants.CENTER);  
  
else  
    lblNewLabel = new  
        JLabel(introductionEnglish,SwingConstants.CENTER);
```

- Un oggetto JLabel utilizzato per presentare il task da seguire durante la simulazione, inizializzato anch'esso in italiano o in inglese

```
JLabel lblEnterAnyText = new JLabel();  
  
if(language.compareTo("ITALIANO")==0)
```

```
lblEnterAnyText.setText(mainItalian);
```

```
else
```

```
    lblEnterAnyText.setText(mainEnglish);
```

- Un oggetto `JTextField` all'interno del quale viene inserito il destinatario della mail

```
JTextField textField = new JTextField();
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la simulazione.
- La disabilitazione del messaggio d'errore di destinatario mancante, se presente

```
ErrLabel.setVisible(false);
```

- Un oggetto `JTextArea`, all'interno del quale viene inserito il testo della mail

```
final JTextArea textMain = new JTextArea();
```

Ad esso è associato un `MouseListener` che, una volta abilitato, scatena i seguenti eventi:

- La scrittura dell'azione appena eseguita all'interno di una lista.
- L'incremento del counter relativo al numero di click effettuati durante la simulazione.

- Un oggetto `CustButton` adibito alla cancellazione del testo precedentemente inserito e tramite il quale è possibile resettare la simulazione di KD

```
CustButton btnReset = null;
```

```
if(language.compareTo("ITALIANO")==0)
    btnReset = new CustButton(buttItalian2);
```

```
else
```

```
    btnReset = new CustButton(buttEnglish2);
```

- Un oggetto `CustButton`, con il quale è possibile confermare l'invio della mail

```
CustButton btnGetPattern = null;
```

```
if(language.compareTo("ITALIANO")==0)
    btnGetPattern = new CustButton(buttItalian1);
```

```
else
```

```
    btnGetPattern = new CustButton(buttEnglish1);
```

Rispetto al prototipo originale, dopo aver confrontato il pattern ricavato durante la scrittura della mail con quello creato durante la fase di registrazione, possono presentarsi due casi differenti:

- Se la comparazione ha avuto un esito positivo, viene creato ed inizializzato l'oggetto “EmailOK”, tramite il quale viene segnalato il corretto completamento della simulazione

```
EmailOK yes = null;
```

```
yes = new EmailOK();  
yes.main(null);
```

- Se la comparazione ha avuto esito negativo, viene proposta la schermata di blocco, tramite l'utilizzo del metodo “loginEmailClick” di cui discusso in precedenza

```
Home.loginEmailClick();
```

Oltre al reparto grafico, è presente un metodo aggiuntivo di conclusione del test che viene avviato al completamento della simulazione e che segue la logica di quanto già visto nel prototipo originale

```
public static void stopAWTStopEmail() {  
    Long duration = System.nanoTime()-UserPn.getStartTimeKD();  
    Float durationF = Float.valueOf(duration);  
    int clicks = UserPn.getCounterKD();  
    durationF=durationF/1000000000;  
    WriteTest.writeOnFile("KDSimulation",  
        durationF,clicks,UserPn.getFailuresKD());  
    UserPn.clearCounterKD();  
    UserPn.clearStartTimeKD();  
    UserPn.clearFailuresKD();  
}
```

È da segnalare che il metodo non viene avviato se il meccanismo di KD non è stato in grado di riconoscere correttamente l'utente³.

C.2.3 Schermata di blocco

L'intero processo di blocco (vedi figura C.2) viene gestito da un package a parte chiamato “LoginEmail”, composto dalle stesse classi presenti nel package “Login”⁴. Tra esse solo le classi “LoginEmail.java” e “LoginPanelEmail.java” differiscono parzialmente da quelli originali.

³In questo caso la reale terminazione della simulazione avverrà unicamente dopo l'inserimento della sequenza grafica corretta.

⁴È stato aggiunto il suffisso ‘Email’ al nome delle classi al fine di differenziarle da quelle originali.

LoginEmail.java

All'interno della classe LoginEmail vengono aggiunte le seguenti funzionalità:

- Un metodo che permette di settare l'username all'interno della schermata di blocco

```
public static void setUser() {  
    contentPane.txtUsr.setText(UsrHome.getUser());  
}
```

Vedremo, nella sezione successiva, come questa funzione sarà fondamentale per impedire che un malintenzionato possa inserire un username a piacimento all'interno della schermata di blocco.

- Un metodo che permette di cancellare tutti gli elementi presenti nel campo password

```
public static void clearPassword() {  
    contentPane.txtPass.setText("");  
    contentPane.password="";  
    contentPane.index=0;  
    contentPane.length=0;  
}
```

LoginPanelEmail.java

Per quanto riguarda la classe LoginPanelEmail, vengono inseriti i seguenti elementi:

- Un oggetto JLabel che si occupa di segnalare il fallimento del meccanismo di KD

```
JLabel lblNewLabel = null;  
lblNewLabel = new JLabel("Keystroke Dynamics  
FAILED",SwingConstants.CENTER);
```

- La costruzione di un campo username, non modificabile da parte dell'utente

```
txtUsr.setEditable(false);
```

- L'inizializzazione di un oggetto LoginEmail che viene richiamato nel caso in cui l'autenticazione sia stata completata con successo

```
EmailOK yes = null;  
yes = new EmailOK();  
  
yes.main(null);
```

C.2.4 Conferma di avvenuto invio della mail

L'ultimo elemento presente nel prototipo riguarda la schermata di corretto invio della mail (vedi figura C.3), che viene presentato nel caso in cui:

- L'utente viene riconosciuto correttamente dal meccanismo di KD durante la scrittura della mail.
- L'utente non viene riconosciuto durante la scrittura della mail, ma è in grado di autenticarsi correttamente tramite la schermata di blocco.

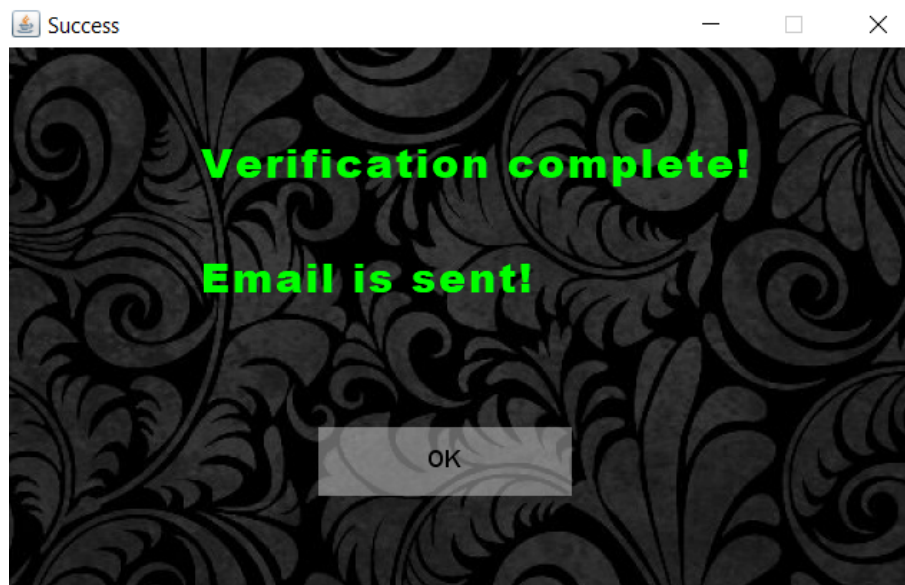


Figura C.3. Schermata di conferma di avvio della mail.

In entrambi i casi, la classe di riferimento prende il nome di “EmailOK.java”, collocata all'interno del package “kdPost”. Essa è composta da due elementi grafici:

- Un oggetto JLabel contenente il messaggio di conferma

```
JLabel lblKeystrokeDynamicOk = new
    JLabel("messaggio",SwingConstants.CENTER);
```
- Un oggetto CustButton che, una volta premuto, scatena i seguenti eventi
 - La terminazione e l'invio di tutti i parametri relativi alla simulazione di KD

```
AWTPost.stopAWTStopEmail();
```
 - L'eliminazione della schermata di simulazione

```
AWTPost.disposeFrame();
```

- L'eliminazione della schermata di blocco (se presente)

```
Home.disableLoginEmail();
```

- L'eliminazione del frame corrente

```
Home.disableLoginEmail();
```

Infine, alla chiusura di questo popup, verrà presentata nuovamente l'homepage personale dell'utente.

Appendice D

Risultati individuali grezzi

Presento in quest'appendice i risultati individuali dei singoli test, di cui verranno omesse le informazioni personali. Sarà possibile notare in alcuni casi la presenza di più simulazioni di keystroke dynamics, da attribuire alle diverse prove che i singoli utenti hanno effettuato per verificare il funzionamento del meccanismo. Di esse, solo la prima è la prova reale che è stata utilizzata per ricavare i risultati finali, di cui discusso al capitolo 8).

Utente A

```
"KDSimulation_0_Time":24.067503,
>Login_0_Failures":0,
>Registration_0_Failures":0,
>Registration_0_Time":68.99292,
>ChangeUsername_0_Failures":0,
>ChangePassword_0_Time":10.265526,
>Login_0_Click":2,
>Login_0_Time":16.6042,
>ChangeUsername_0_Time":3.670854,
>ChangePassword_0_Failures":0,
>KDSimulation_0_Failures":0,
>ChangeUsername_0_Click":1,
>KDSimulation_0_Click":1,
>ChangePassword_0_Click":6,
>Registration_0_Click":11,
>UserActions":[
  "Mon Feb 18 18:03:19 CET 2019 --> Home: click on SIGN UP button",
  "Mon Feb 18 18:03:22 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:03:23 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:03:24 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:03:25 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:03:27 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:03:28 CET 2019 --> Sign up: click on text user form",
  "Mon Feb 18 18:03:41 CET 2019 --> Sign up: click on SHOW PASSWORD
    button",
  "Mon Feb 18 18:04:05 CET 2019 --> Sign up: click on OK button",
  "Mon Feb 18 18:04:05 CET 2019 --> Sign up OK! User: userA, Password:
    h7g8g7g6h7",
  "Mon Feb 18 18:04:08 CET 2019 --> Sign up: Select the language
    ITALIANO",
```

"Mon Feb 18 18:04:28 CET 2019 --> Sign up (KD registration): confirm the text inserted",
"Mon Feb 18 18:04:30 CET 2019 --> Home: click on LOGIN button",
"Mon Feb 18 18:04:33 CET 2019 --> LOGIN: click on password form",
"Mon Feb 18 18:04:36 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:36 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:38 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:38 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:40 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:40 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:42 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:43 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:45 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:45 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:04:46 CET 2019 --> LOGIN: press ENTER button and start data validation (= press OK button)",
"Mon Feb 18 18:04:46 CET 2019 --> LOGIN userA OK",
"Mon Feb 18 18:04:56 CET 2019 --> USER PAGE: click on CHANGE USERNAME button",
"Mon Feb 18 18:04:58 CET 2019 --> USER PAGE (Change user): click on text form",
"Mon Feb 18 18:05:00 CET 2019 --> USER PAGE: (Change user) press ENTER button (= press OK button)",
"Mon Feb 18 18:05:00 CET 2019 --> USER PAGE (Change user): the username inserted is valid",
"Mon Feb 18 18:05:00 CET 2019 --> USER PAGE (Change user): the new username is userA2",
"Mon Feb 18 18:05:01 CET 2019 --> USER PAGE: click on CHANGE PASSWORD button",
"Mon Feb 18 18:05:11 CET 2019 --> USER PAGE (Change password): click on OK button",
"Mon Feb 18 18:05:11 CET 2019 --> USER PAGE (Change password): the password inserted is valid ",
"Mon Feb 18 18:05:11 CET 2019 --> USER PAGE (Change password): new password is a8--->b8--->b7--->a7--->a8",
"Mon Feb 18 18:05:15 CET 2019 --> USER PAGE: click on KD SIMULATION button",
"Mon Feb 18 18:05:39 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Mon Feb 18 18:05:41 CET 2019 --> USER PAGE: click on SHOW RESULT button",
"Mon Feb 18 18:05:42 CET 2019 --> USER PAGE: click on LOGOUT button",
"Mon Feb 18 18:05:43 CET 2019 --> USER PAGE (logout): click on YES button"

]

Utente B

```

"KDSimulation_0_Time":93.07519,
"Login_0_Failures":1,
"Registration_0_Failures":0,
"Registration_0_Time":44.696575,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":9.088453,
"Login_0_Click":1,
"Login_0_Time":47.07664,
"ChangeUsername_0_Time":7.017844,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":8,
"ChangePassword_0_Click":6,
"Registration_0_Click":13,
"UserActions":[
  "Wed Feb 13 11:33:19 CET 2019 --> Home: click on SIGN UP button",
  "Wed Feb 13 11:33:21 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 11:33:22 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 11:33:23 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 11:33:24 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 11:33:25 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 11:33:29 CET 2019 --> Sign up: click on text user form",
  "Wed Feb 13 11:33:30 CET 2019 --> Sign up: click on text user form",
  "Wed Feb 13 11:33:30 CET 2019 --> Sign up: click on text user form",
  "Wed Feb 13 11:33:30 CET 2019 --> Sign up: click on text user form",
  "Wed Feb 13 11:33:35 CET 2019 --> Sign up: click on OK button",
  "Wed Feb 13 11:33:35 CET 2019 --> Sign up OK! User: userB, Password:
    h8h7h6h5h4",
  "Wed Feb 13 11:33:38 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Wed Feb 13 11:34:04 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Wed Feb 13 11:34:06 CET 2019 --> Home: click on LOGIN button",
  "Wed Feb 13 11:34:18 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:18 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:20 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:24 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:24 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:26 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:27 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:29 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:30 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:32 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 11:34:33 CET 2019 --> LOGIN (entering the password):
    backspace",

```

```

"Wed Feb 13 11:34:37 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 11:34:38 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 11:34:39 CET 2019 --> LOGIN: press ENTER button and start
  data validation (= press OK button)",
"Wed Feb 13 11:34:39 CET 2019 --> LOGIN: the password inserted is
  wrong",
"Wed Feb 13 11:34:42 CET 2019 --> LOGIN (entering the password):
  backspace",
"Wed Feb 13 11:34:43 CET 2019 --> LOGIN (entering the password):
  backspace",
"Wed Feb 13 11:34:46 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 11:34:47 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 11:34:52 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 11:34:52 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 11:34:53 CET 2019 --> LOGIN: press ENTER button and start
  data validation (= press OK button)",
"Wed Feb 13 11:34:53 CET 2019 --> LOGIN userB OK",
"Wed Feb 13 11:35:03 CET 2019 --> USER PAGE: click on CHANGE USERNAME
  button",
"Wed Feb 13 11:35:05 CET 2019 --> USER PAGE (Change user): click on
  text form",
"Wed Feb 13 11:35:10 CET 2019 --> USER PAGE (Change user): click on OK
  button",
"Wed Feb 13 11:35:10 CET 2019 --> USER PAGE (Change user): the
  username inserted is valid",
"Wed Feb 13 11:35:10 CET 2019 --> USER PAGE (Change user): the new
  username is userB2",
"Wed Feb 13 11:35:14 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
  button",
"Wed Feb 13 11:35:23 CET 2019 --> USER PAGE (Change password): click
  on OK button",
"Wed Feb 13 11:35:23 CET 2019 --> USER PAGE (Change password): the
  password inserted is valid ",
"Wed Feb 13 11:35:23 CET 2019 --> USER PAGE (Change password): new
  password is a1--->b1--->c1--->b1--->a1",
"Wed Feb 13 11:35:30 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Wed Feb 13 11:37:03 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Wed Feb 13 11:37:05 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Wed Feb 13 11:37:16 CET 2019 --> USER PAGE: click on LOGOUT button",
"Wed Feb 13 11:37:17 CET 2019 --> USER PAGE (logout): click on YES
  button"

```

]

Utente C

```

"KDSimulation_0_Time":51.169678,
>Login_0_Failures":0,
"Registration_0_Failures":0,
"Registration_0_Time":129.18712,

```

```

"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":17.159492,
"Login_0_Click":4,
"KDSimulation_1_Click":1,
"Login_0_Time":33.041355,
"ChangeUsername_0_Time":11.176223,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"KDSimulation_1_Time":30.40569,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":1,
"ChangePassword_0_Click":6,
"Registration_0_Click":10,
"KDSimulation_1_Failures":1,
"UserActions":[
  "Thu Feb 07 15:39:07 CET 2019 --> Home: click on SIGN UP button",
  "Thu Feb 07 15:39:21 CET 2019 --> Sign up: click on grid",
  "Thu Feb 07 15:39:23 CET 2019 --> Sign up: click on grid",
  "Thu Feb 07 15:39:29 CET 2019 --> Sign up: click on grid",
  "Thu Feb 07 15:39:31 CET 2019 --> Sign up: click on grid",
  "Thu Feb 07 15:39:43 CET 2019 --> Sign up: click on text user form",
  "Thu Feb 07 15:40:17 CET 2019 --> Sign up: click on OK button",
  "Thu Feb 07 15:40:18 CET 2019 --> Sign up OK! User: userC, Password:
    a8h1h8b8",
  "Thu Feb 07 15:40:23 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Thu Feb 07 15:40:28 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Thu Feb 07 15:41:16 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Thu Feb 07 15:41:19 CET 2019 --> Home: click on LOGIN button",
  "Thu Feb 07 15:41:25 CET 2019 --> LOGIN: click on user form",
  "Thu Feb 07 15:41:33 CET 2019 --> LOGIN: click on password form",
  "Thu Feb 07 15:41:37 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:38 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:38 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:42 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:42 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:42 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:45 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:46 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:46 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:49 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:49 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 07 15:41:49 CET 2019 --> LOGIN (entering the password):
    insert a char",

```

```

"Thu Feb 07 15:41:52 CET 2019 --> LOGIN: click on OK button",
"Thu Feb 07 15:41:52 CET 2019 --> LOGIN userC OK",
"Thu Feb 07 15:42:02 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Thu Feb 07 15:42:05 CET 2019 --> USER PAGE (Change user): click on
text form",
"Thu Feb 07 15:42:13 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Thu Feb 07 15:42:13 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Thu Feb 07 15:42:13 CET 2019 --> USER PAGE (Change user): the new
username is userC2",
"Thu Feb 07 15:42:17 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Thu Feb 07 15:42:34 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Thu Feb 07 15:42:34 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Thu Feb 07 15:42:34 CET 2019 --> USER PAGE (Change password): new
password is h8--->h1--->a1--->a8--->g2",
"Thu Feb 07 15:42:43 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Thu Feb 07 15:43:34 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Thu Feb 07 15:43:38 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Thu Feb 07 15:43:53 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Thu Feb 07 15:44:24 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Thu Feb 07 15:44:25 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Thu Feb 07 15:44:28 CET 2019 --> USER PAGE: click on LOGOUT button",
"Thu Feb 07 15:44:29 CET 2019 --> USER PAGE (logout): click on YES
button"

```

]

Utente D

```

"KDSimulation_0_Time":20.322056,
>Login_0_Failures":0,
"Registration_0_Failures":0,
"Registration_0_Time":95.28114,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":12.410873,
>Login_0_Click":3,
"KDSimulation_1_Click":1,
>Login_0_Time":44.923664,
"ChangeUsername_0_Time":13.66133,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"KDSimulation_1_Time":35.491188,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":9,
"Registration_0_Click":14,
"KDSimulation_1_Failures":1,
>UserActions":[
"Mon Feb 18 18:59:18 CET 2019 --> Home: click on SIGN UP button",

```

"Mon Feb 18 18:59:51 CET 2019 --> Sign up: click on grid",
"Mon Feb 18 18:59:53 CET 2019 --> Sign up: click on grid",
"Mon Feb 18 18:59:54 CET 2019 --> Sign up: click on grid",
"Mon Feb 18 18:59:55 CET 2019 --> Sign up: click on grid",
"Mon Feb 18 18:59:56 CET 2019 --> Sign up: click on grid",
"Mon Feb 18 18:59:57 CET 2019 --> Sign up: click on grid",
"Mon Feb 18 18:59:58 CET 2019 --> Sign up: click on grid",
"Mon Feb 18 18:59:59 CET 2019 --> Sign up: click on grid",
"Mon Feb 18 19:00:00 CET 2019 --> Sign up: click on text user form",
"Mon Feb 18 19:00:10 CET 2019 --> Sign up: click on OK button",
"Mon Feb 18 19:00:10 CET 2019 --> Sign up OK! User: userD, Password:
a1b2c3d2e1f2g3h2",
"Mon Feb 18 19:00:15 CET 2019 --> Sign up: Select the language
ITALIANO",
"Mon Feb 18 19:00:19 CET 2019 --> Sign up (KD registration): click on
text form",
"Mon Feb 18 19:00:54 CET 2019 --> Sign up (KD registration): confirm
the text inserted",
"Mon Feb 18 19:00:57 CET 2019 --> Home: click on LOGIN button",
"Mon Feb 18 19:01:02 CET 2019 --> LOGIN: click on password form",
"Mon Feb 18 19:01:03 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:04 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:04 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:05 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:07 CET 2019 --> LOGIN (entering the password):
backspace",
"Mon Feb 18 19:01:07 CET 2019 --> LOGIN (entering the password):
backspace",
"Mon Feb 18 19:01:13 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:13 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:15 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:15 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Mon Feb 18 19:01:17 CET 2019 --> LOGIN (entering the password):
backspace",
"Mon Feb 18 19:01:17 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:17 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:21 CET 2019 --> LOGIN (entering the password):
backspace",
"Mon Feb 18 19:01:21 CET 2019 --> LOGIN (entering the password):
backspace",
"Mon Feb 18 19:01:22 CET 2019 --> LOGIN (entering the password):
backspace",
"Mon Feb 18 19:01:24 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:24 CET 2019 --> LOGIN (entering the password):
insert a char",

"Mon Feb 18 19:01:26 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:27 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:29 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:29 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:31 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:31 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:33 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:33 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:35 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:35 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:37 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:38 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:40 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:40 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Feb 18 19:01:42 CET 2019 --> LOGIN: click on OK button",
"Mon Feb 18 19:01:42 CET 2019 --> LOGIN userD OK",
"Mon Feb 18 19:01:51 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Mon Feb 18 19:01:52 CET 2019 --> USER PAGE (Change user): click on
text form",
"Mon Feb 18 19:02:04 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Mon Feb 18 19:02:04 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Mon Feb 18 19:02:04 CET 2019 --> USER PAGE (Change user): the new
username is userD2",
"Mon Feb 18 19:02:07 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Mon Feb 18 19:02:19 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Mon Feb 18 19:02:19 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Mon Feb 18 19:02:19 CET 2019 --> USER PAGE (Change password): new
password is h1--->g2--->f3--->e2--->d1--->c2--->b3--->a2",
"Mon Feb 18 19:02:23 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Feb 18 19:02:24 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Mon Feb 18 19:02:43 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Mon Feb 18 19:02:49 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Feb 18 19:03:33 CET 2019 --> USER PAGE: click on KD SIMULATION
button",

```

"Mon Feb 18 19:04:08 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Mon Feb 18 19:04:12 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Feb 18 19:04:16 CET 2019 --> USER PAGE: click on LOGOUT button",
"Mon Feb 18 19:04:18 CET 2019 --> USER PAGE (logout): click on YES
button"
]

```

Utente E

```

"KDSimulation_0_Time":39.17925,
"Login_0_Failures":0,
"Registration_0_Failures":0,
"Registration_0_Time":114.51231,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":12.373907,
"Login_0_Click":1,
"Login_0_Time":41.95577,
"ChangeUsername_0_Time":12.218113,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":1,
"ChangePassword_0_Click":6,
"Registration_0_Click":9,
"UserActions":[
  "Mon Feb 18 18:07:06 CET 2019 --> Home: click on SIGN UP button",
  "Mon Feb 18 18:07:08 CET 2019 --> Sign up: click on text user form",
  "Mon Feb 18 18:07:10 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:07:11 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:07:12 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:07:14 CET 2019 --> Sign up: click on text user form",
  "Mon Feb 18 18:08:38 CET 2019 --> Sign up: click on OK button",
  "Mon Feb 18 18:08:38 CET 2019 --> Sign up OK! User: userE, Password:
h1h1h1",
  "Mon Feb 18 18:08:41 CET 2019 --> Sign up: Select the language
ITALIANO",
  "Mon Feb 18 18:09:00 CET 2019 --> Sign up (KD registration): confirm
the text inserted",
  "Mon Feb 18 18:09:04 CET 2019 --> Home: click on LOGIN button",
  "Mon Feb 18 18:09:13 CET 2019 --> LOGIN (entering the password):
insert a char",
  "Mon Feb 18 18:09:13 CET 2019 --> LOGIN (entering the password):
insert a char",
  "Mon Feb 18 18:09:15 CET 2019 --> LOGIN (entering the password):
insert a char",
  "Mon Feb 18 18:09:15 CET 2019 --> LOGIN (entering the password):
insert a char",
  "Mon Feb 18 18:09:17 CET 2019 --> LOGIN (entering the password):
insert a char",
  "Mon Feb 18 18:09:18 CET 2019 --> LOGIN (entering the password):
insert a char",
  "Mon Feb 18 18:09:45 CET 2019 --> LOGIN: press ENTER button and start
data validation (= press OK button)",
  "Mon Feb 18 18:09:45 CET 2019 --> LOGIN userE OK",
  "Mon Feb 18 18:09:48 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",

```

```

"Mon Feb 18 18:09:50 CET 2019 --> USER PAGE (Change user): click on
text form",
"Mon Feb 18 18:10:01 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Mon Feb 18 18:10:01 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Mon Feb 18 18:10:01 CET 2019 --> USER PAGE (Change user): the new
username is userE2",
"Mon Feb 18 18:10:02 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Mon Feb 18 18:10:14 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Mon Feb 18 18:10:14 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Mon Feb 18 18:10:14 CET 2019 --> USER PAGE (Change password): new
password is e4--->e5--->d5--->d4--->e4",
"Mon Feb 18 18:10:17 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Feb 18 18:10:56 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Mon Feb 18 18:10:56 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Feb 18 18:10:58 CET 2019 --> USER PAGE: click on LOGOUT button",
"Mon Feb 18 18:10:59 CET 2019 --> USER PAGE (logout): click on YES
button"

```

]

Utente F

```

"KDSimulation_0_Time":33.63348,
>Login_0_Failures":0,
"Registration_0_Failures":0,
"Registration_0_Time":101.20205,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":8.439199,
>Login_0_Click":2,
"KDSimulation_1_Click":1,
>Login_0_Time":35.549995,
"ChangeUsername_0_Time":7.416888,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"KDSimulation_1_Time":25.408726,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":5,
"Registration_0_Click":9,
"KDSimulation_1_Failures":1,
"UserActions":[
  "Mon Feb 18 18:57:42 CET 2019 --> Home: click on SIGN UP button",
  "Mon Feb 18 18:57:52 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:57:55 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:58:01 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:58:02 CET 2019 --> Sign up: click on grid",
  "Mon Feb 18 18:58:35 CET 2019 --> Sign up: click on OK button",
  "Mon Feb 18 18:58:35 CET 2019 --> Sign up OK! User: UserF, Password:
a4b4b3c3",
  "Mon Feb 18 18:58:46 CET 2019 --> Sign up: Select the language
ITALIANO",

```

"Mon Feb 18 18:58:47 CET 2019 --> Sign up (KD registration): click on text form",
"Mon Feb 18 18:59:23 CET 2019 --> Sign up (KD registration): confirm the text inserted",
"Mon Feb 18 18:59:25 CET 2019 --> Home: click on LOGIN button",
"Mon Feb 18 18:59:41 CET 2019 --> LOGIN: click on password form",
"Mon Feb 18 18:59:50 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:59:51 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:59:53 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:59:54 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:59:56 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:59:57 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:59:59 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 18:59:59 CET 2019 --> LOGIN (entering the password): insert a char",
"Mon Feb 18 19:00:00 CET 2019 --> LOGIN: press ENTER button and start data validation (= press OK button)",
"Mon Feb 18 19:00:00 CET 2019 --> LOGIN UserF OK",
"Mon Feb 18 19:00:13 CET 2019 --> USER PAGE: click on CHANGE USERNAME button",
"Mon Feb 18 19:00:17 CET 2019 --> USER PAGE (Change user): click on text form",
"Mon Feb 18 19:00:20 CET 2019 --> USER PAGE (Change user): click on OK button",
"Mon Feb 18 19:00:20 CET 2019 --> USER PAGE (Change user): the username inserted is valid",
"Mon Feb 18 19:00:20 CET 2019 --> USER PAGE (Change user): the new username is UserF2",
"Mon Feb 18 19:00:28 CET 2019 --> USER PAGE: click on CHANGE PASSWORD button",
"Mon Feb 18 19:00:36 CET 2019 --> USER PAGE (Change password): click on OK button",
"Mon Feb 18 19:00:36 CET 2019 --> USER PAGE (Change password): the password inserted is valid ",
"Mon Feb 18 19:00:36 CET 2019 --> USER PAGE (Change password): new password is b2--->b3--->c3--->d3",
"Mon Feb 18 19:00:44 CET 2019 --> USER PAGE: click on KD SIMULATION button",
"Mon Feb 18 19:00:45 CET 2019 --> USER PAGE (KD simulation): click on text frame",
"Mon Feb 18 19:01:17 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Mon Feb 18 19:01:20 CET 2019 --> USER PAGE: click on SHOW RESULT button",
"Mon Feb 18 19:01:52 CET 2019 --> USER PAGE: click on KD SIMULATION button",
"Mon Feb 18 19:02:17 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Mon Feb 18 19:02:21 CET 2019 --> USER PAGE: click on SHOW RESULT button",
"Mon Feb 18 19:02:30 CET 2019 --> USER PAGE: click on LOGOUT button",

```
"Mon Feb 18 19:02:33 CET 2019 --> USER PAGE (logout): click on YES
button"
```

```
]
```

Utente G

```
"KDSimulation_2_Failures":0,
"KDSimulation_0_Time":47.836933,
"Login_0_Failures":1,
"Registration_0_Failures":0,
"KDSimulation_2_Click":2,
"Registration_0_Time":125.4542,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":12.411187,
"Login_0_Click":7,
"KDSimulation_1_Click":1,
"Login_0_Time":91.51827,
"ChangeUsername_0_Time":6.3558283,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"KDSimulation_1_Time":52.586914,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":5,
"Registration_0_Click":15,
"KDSimulation_1_Failures":1,
"KDSimulation_2_Time":44.04261,
"UserActions":[
  "Wed Feb 06 13:26:11 CET 2019 --> Home: click on SIGN UP button",
  "Wed Feb 06 13:26:14 CET 2019 --> Sign up: click on text user form",
  "Wed Feb 06 13:26:40 CET 2019 --> Sign up: click on grid",
  "Wed Feb 06 13:26:42 CET 2019 --> Sign up: click on grid",
  "Wed Feb 06 13:26:45 CET 2019 --> Sign up: click on grid",
  "Wed Feb 06 13:26:54 CET 2019 --> Sign up: click on grid",
  "Wed Feb 06 13:26:56 CET 2019 --> Sign up: click on grid",
  "Wed Feb 06 13:26:57 CET 2019 --> Sign up: click on grid",
  "Wed Feb 06 13:27:19 CET 2019 --> Sign up: click on SHOW PASSWORD
button",
  "Wed Feb 06 13:27:25 CET 2019 --> Sign up: click on OK button",
  "Wed Feb 06 13:27:25 CET 2019 --> Sign up OK! User: UserG, Password:
a1a6a7h1h2h3",
  "Wed Feb 06 13:27:32 CET 2019 --> Sign up: click on languages label",
  "Wed Feb 06 13:27:34 CET 2019 --> Sign up: click on languages label",
  "Wed Feb 06 13:27:36 CET 2019 --> Sign up: Select the language
ENGLISH",
  "Wed Feb 06 13:27:45 CET 2019 --> Sign up (KD registration): click on
text form",
  "Wed Feb 06 13:28:17 CET 2019 --> Sign up (KD registration): confirm
the text inserted",
  "Wed Feb 06 13:28:41 CET 2019 --> Home: click on LOGIN button",
  "Wed Feb 06 13:28:54 CET 2019 --> LOGIN: click on password form",
  "Wed Feb 06 13:28:55 CET 2019 --> LOGIN (entering the password):
insert a char",
  "Wed Feb 06 13:28:56 CET 2019 --> LOGIN (entering the password):
insert a char",
  "Wed Feb 06 13:29:06 CET 2019 --> LOGIN (entering the password):
insert a char",
```

"Wed Feb 06 13:29:08 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:20 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:20 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:27 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:28 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:30 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:30 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:32 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:34 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:35 CET 2019 --> LOGIN: click on OK button",
"Wed Feb 06 13:29:35 CET 2019 --> LOGIN: the password inserted is
wrong",
"Wed Feb 06 13:29:40 CET 2019 --> LOGIN: click on CANCEL button",
"Wed Feb 06 13:29:42 CET 2019 --> Home: click on LOGIN button",
"Wed Feb 06 13:29:48 CET 2019 --> LOGIN: click on password form",
"Wed Feb 06 13:29:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:59 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:29:59 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:30:01 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:30:02 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:30:04 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:30:05 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:30:07 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:30:08 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:30:10 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:30:10 CET 2019 --> LOGIN (entering the password):
insert a char",
"Wed Feb 06 13:30:12 CET 2019 --> LOGIN: click on OK button",
"Wed Feb 06 13:30:12 CET 2019 --> LOGIN UserG OK",
"Wed Feb 06 13:30:38 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Wed Feb 06 13:30:40 CET 2019 --> USER PAGE (Change user): click on
text form",
"Wed Feb 06 13:30:44 CET 2019 --> USER PAGE (Change user): click on OK
button",

```

"Wed Feb 06 13:30:44 CET 2019 --> USER PAGE (Change user): the
  username inserted is valid",
"Wed Feb 06 13:30:44 CET 2019 --> USER PAGE (Change user): the new
  username is userG2",
"Wed Feb 06 13:30:46 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
  button",
"Wed Feb 06 13:30:59 CET 2019 --> USER PAGE (Change password): click
  on OK button",
"Wed Feb 06 13:30:59 CET 2019 --> USER PAGE (Change password): the
  password inserted is valid ",
"Wed Feb 06 13:30:59 CET 2019 --> USER PAGE (Change password): new
  password is h1--->h3--->h5--->f5",
"Wed Feb 06 13:31:02 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Wed Feb 06 13:31:05 CET 2019 --> USER PAGE (KD simulation): click on
  text frame",
"Wed Feb 06 13:31:50 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Wed Feb 06 13:31:53 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Wed Feb 06 13:31:57 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Wed Feb 06 13:32:50 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Wed Feb 06 13:32:51 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Wed Feb 06 13:32:57 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Wed Feb 06 13:32:59 CET 2019 --> USER PAGE (KD simulation): click on
  text frame",
"Wed Feb 06 13:33:41 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Wed Feb 06 13:33:43 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Wed Feb 06 13:34:16 CET 2019 --> USER PAGE: click on LOGOUT button",
"Wed Feb 06 13:34:17 CET 2019 --> USER PAGE (logout): click on YES
  button"

```

]

Utente H

```

"KDSimulation_0_Time":41.991688,
>Login_0_Failures":0,
"Registration_0_Failures":1,
"Registration_0_Time":59.731308,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":20.52162,
>Login_0_Click":4,
>Login_0_Time":39.964657,
"ChangeUsername_0_Time":13.998487,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":4,
"Registration_0_Click":10,
"UserActions":[
  "Tue Feb 05 23:41:02 CET 2019 --> Home: click on SIGN UP button",
  "Tue Feb 05 23:41:07 CET 2019 --> Sign up: click on grid",
  "Tue Feb 05 23:41:08 CET 2019 --> Sign up: click on grid",

```

"Tue Feb 05 23:41:10 CET 2019 --> Sign up: click on grid",
"Tue Feb 05 23:41:13 CET 2019 --> Sign up: click on OK button",
"Tue Feb 05 23:41:13 CET 2019 --> Sign up: insert a wrong username
(null name)",
"Tue Feb 05 23:41:17 CET 2019 --> Sign up: click on text user form",
"Tue Feb 05 23:41:24 CET 2019 --> Sign up: click on OK button",
"Tue Feb 05 23:41:24 CET 2019 --> Sign up OK! User: UserH, Password:
ald1h1",
"Tue Feb 05 23:41:28 CET 2019 --> Sign up: Select the language
ITALIANO",
"Tue Feb 05 23:41:29 CET 2019 --> Sign up (KD registration): click on
text form",
"Tue Feb 05 23:42:02 CET 2019 --> Sign up (KD registration): confirm
the text inserted",
"Tue Feb 05 23:42:24 CET 2019 --> Home: click on LOGIN button",
"Tue Feb 05 23:42:29 CET 2019 --> LOGIN: click on user form",
"Tue Feb 05 23:42:38 CET 2019 --> LOGIN: click on password form",
"Tue Feb 05 23:42:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 05 23:42:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 05 23:42:52 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 05 23:42:56 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 05 23:42:56 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 05 23:42:58 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 05 23:43:01 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 05 23:43:01 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 05 23:43:01 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 05 23:43:04 CET 2019 --> LOGIN: click on OK button",
"Tue Feb 05 23:43:04 CET 2019 --> LOGIN userH OK",
"Tue Feb 05 23:43:24 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Tue Feb 05 23:43:25 CET 2019 --> USER PAGE (Change user): click on
text form",
"Tue Feb 05 23:43:38 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Tue Feb 05 23:43:38 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Tue Feb 05 23:43:38 CET 2019 --> USER PAGE (Change user): the new
username is userH2",
"Tue Feb 05 23:43:39 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Tue Feb 05 23:44:00 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Tue Feb 05 23:44:00 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Tue Feb 05 23:44:00 CET 2019 --> USER PAGE (Change password): new
password is a1--->d1--->g1",
"Tue Feb 05 23:44:13 CET 2019 --> USER PAGE: click on KD SIMULATION
button",

```

    "Tue Feb 05 23:44:15 CET 2019 --> USER PAGE (KD simulation): click on
      text frame",
    "Tue Feb 05 23:44:55 CET 2019 --> USER PAGE (KD simulation): MATCH",
    "Tue Feb 05 23:45:08 CET 2019 --> USER PAGE: click on SHOW RESULT
      button",
    "Tue Feb 05 23:45:21 CET 2019 --> USER PAGE: click on LOGOUT button",
    "Tue Feb 05 23:45:24 CET 2019 --> USER PAGE (logout): click on YES
      button"
  ]

```

Utente I

```

"Login_0_Failures":0,
"Registration_0_Failures":0,
"ChangeUsername_0_Failures":0,
"Login_0_Click":4,
"Login_0_Time":32.098637,
"KDSimulation_1_Time":18.930712,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":4,
"KDSimulation_1_Failures":1,
"KDSimulation_0_Time":47.075764,
"Registration_0_Time":91.22648,
"ChangePassword_0_Time":9.270627,
"KDSimulation_1_Click":1,
"ChangeUsername_0_Time":10.993789,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"Registration_0_Click":10,
"UserActions":[
  "Thu Feb 21 14:53:36 CET 2019 --> Home: click on SIGN UP button",
  "Thu Feb 21 14:53:39 CET 2019 --> Sign up: click on text user form",
  "Thu Feb 21 14:53:40 CET 2019 --> Sign up: click on text user form",
  "Thu Feb 21 14:53:52 CET 2019 --> Sign up: click on grid",
  "Thu Feb 21 14:53:53 CET 2019 --> Sign up: click on grid",
  "Thu Feb 21 14:53:54 CET 2019 --> Sign up: click on grid",
  "Thu Feb 21 14:54:01 CET 2019 --> Sign up: click on OK button",
  "Thu Feb 21 14:54:02 CET 2019 --> Sign up OK! User: userI, Password:
    a2a4c4",
  "Thu Feb 21 14:54:06 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Thu Feb 21 14:54:08 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Thu Feb 21 14:55:07 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Thu Feb 21 14:55:11 CET 2019 --> Home: click on LOGIN button",
  "Thu Feb 21 14:55:12 CET 2019 --> LOGIN: click on user form",
  "Thu Feb 21 14:55:17 CET 2019 --> LOGIN: click on password form",
  "Thu Feb 21 14:55:22 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 21 14:55:23 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 21 14:55:23 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 21 14:55:23 CET 2019 --> LOGIN (entering the password): the
    char inserted is not valid",

```

```

"Thu Feb 21 14:55:31 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Thu Feb 21 14:55:31 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Thu Feb 21 14:55:39 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Thu Feb 21 14:55:40 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Thu Feb 21 14:55:43 CET 2019 --> LOGIN: click on OK button",
"Thu Feb 21 14:55:43 CET 2019 --> LOGIN userI OK",
"Thu Feb 21 14:56:03 CET 2019 --> USER PAGE: click on CHANGE USERNAME
  button",
"Thu Feb 21 14:56:04 CET 2019 --> USER PAGE (Change user): click on
  text form",
"Thu Feb 21 14:56:14 CET 2019 --> USER PAGE (Change user): click on OK
  button",
"Thu Feb 21 14:56:14 CET 2019 --> USER PAGE (Change user): the
  username inserted is valid",
"Thu Feb 21 14:56:14 CET 2019 --> USER PAGE (Change user): the new
  username is userI2",
"Thu Feb 21 14:56:24 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
  button",
"Thu Feb 21 14:56:34 CET 2019 --> USER PAGE (Change password): click
  on OK button",
"Thu Feb 21 14:56:34 CET 2019 --> USER PAGE (Change password): the
  password inserted is valid ",
"Thu Feb 21 14:56:34 CET 2019 --> USER PAGE (Change password): new
  password is a2--->a4--->a6",
"Thu Feb 21 14:57:34 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Thu Feb 21 14:57:36 CET 2019 --> USER PAGE (KD simulation): click on
  text frame",
"Thu Feb 21 14:58:21 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Thu Feb 21 14:58:25 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Thu Feb 21 14:58:36 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Thu Feb 21 14:58:55 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Thu Feb 21 14:58:56 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Thu Feb 21 14:59:06 CET 2019 --> USER PAGE: click on LOGOUT button",
"Thu Feb 21 14:59:08 CET 2019 --> USER PAGE (logout): click on YES
  button",

```

]

Utente J

```

"Login_0_Failures":0,
"KDSimulation_4_Click":2,
"KDSimulation_3_Failures":0,
"Registration_0_Failures":0,
"KDSimulation_2_Click":2,
"ChangeUsername_0_Failures":0,
"Login_0_Click":3,
"Login_0_Time":20.59934,
"KDSimulation_1_Time":24.877434,
"ChangeUsername_0_Click":2,

```

```

"KDSimulation_0_Click":2,
"ChangePassword_0_Click":4,
"KDSimulation_4_Time":39.29207,
"KDSimulation_4_Failures":0,
"KDSimulation_1_Failures":1,
"KDSimulation_2_Failures":0,
"KDSimulation_3_Time":24.160597,
"KDSimulation_0_Time":29.253672,
"KDSimulation_3_Click":2,
"Registration_0_Time":51.05769,
"ChangePassword_0_Time":4.7250695,
"KDSimulation_1_Click":1,
"ChangeUsername_0_Time":12.152135,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"Registration_0_Click":9,
"KDSimulation_2_Time":23.68468,
"UserActions":[
  "Fri Feb 22 14:38:02 CET 2019 --> Home: click on SIGN UP button",
  "Fri Feb 22 14:38:06 CET 2019 --> Sign up: click on text user form",
  "Fri Feb 22 14:38:16 CET 2019 --> Sign up: click on grid",
  "Fri Feb 22 14:38:17 CET 2019 --> Sign up: click on grid",
  "Fri Feb 22 14:38:18 CET 2019 --> Sign up: click on grid",
  "Fri Feb 22 14:38:21 CET 2019 --> Sign up: click on OK button",
  "Fri Feb 22 14:38:22 CET 2019 --> Sign up OK! User: userJ, Password:
    h1h2g2",
  "Fri Feb 22 14:38:25 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Fri Feb 22 14:38:31 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Fri Feb 22 14:38:53 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Fri Feb 22 14:38:56 CET 2019 --> Home: click on LOGIN button",
  "Fri Feb 22 14:39:04 CET 2019 --> LOGIN: click on password form",
  "Fri Feb 22 14:39:06 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Fri Feb 22 14:39:06 CET 2019 --> LOGIN (entering the password): the
    char inserted is not valid",
  "Fri Feb 22 14:39:06 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Fri Feb 22 14:39:07 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Fri Feb 22 14:39:09 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Fri Feb 22 14:39:10 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Fri Feb 22 14:39:12 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Fri Feb 22 14:39:13 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Fri Feb 22 14:39:13 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Fri Feb 22 14:39:13 CET 2019 --> LOGIN (entering the password): the
    char inserted is not valid",
  "Fri Feb 22 14:39:17 CET 2019 --> LOGIN: click on OK button",
  "Fri Feb 22 14:39:17 CET 2019 --> LOGIN userJ OK",

```

"Fri Feb 22 14:39:21 CET 2019 --> USER PAGE: click on CHANGE USERNAME button",
"Fri Feb 22 14:39:23 CET 2019 --> USER PAGE (Change user): click on text form",
"Fri Feb 22 14:39:33 CET 2019 --> USER PAGE (Change user): click on OK button",
"Fri Feb 22 14:39:33 CET 2019 --> USER PAGE (Change user): the username inserted is valid",
"Fri Feb 22 14:39:33 CET 2019 --> USER PAGE (Change user): the new username is userJ2",
"Fri Feb 22 14:39:37 CET 2019 --> USER PAGE: click on CHANGE PASSWORD button",
"Fri Feb 22 14:39:41 CET 2019 --> USER PAGE (Change password): click on OK button",
"Fri Feb 22 14:39:41 CET 2019 --> USER PAGE (Change password): the password inserted is valid ",
"Fri Feb 22 14:39:41 CET 2019 --> USER PAGE (Change password): new password is h1--->g1--->g2",
"Fri Feb 22 14:39:45 CET 2019 --> USER PAGE: click on KD SIMULATION button",
"Fri Feb 22 14:39:47 CET 2019 --> USER PAGE (KD simulation): click on text frame",
"Fri Feb 22 14:40:14 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Fri Feb 22 14:40:17 CET 2019 --> USER PAGE: click on SHOW RESULT button",
"Fri Feb 22 14:40:26 CET 2019 --> USER PAGE: click on KD SIMULATION button",
"Fri Feb 22 14:40:51 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Fri Feb 22 14:40:52 CET 2019 --> USER PAGE: click on SHOW RESULT button",
"Fri Feb 22 14:40:59 CET 2019 --> USER PAGE: click on KD SIMULATION button",
"Fri Feb 22 14:41:00 CET 2019 --> USER PAGE (KD simulation): click on text frame",
"Fri Feb 22 14:41:23 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Fri Feb 22 14:41:30 CET 2019 --> USER PAGE: click on SHOW RESULT button",
"Fri Feb 22 14:41:40 CET 2019 --> USER PAGE: click on KD SIMULATION button",
"Fri Feb 22 14:41:41 CET 2019 --> USER PAGE (KD simulation): click on text frame",
"Fri Feb 22 14:42:04 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Fri Feb 22 14:42:05 CET 2019 --> USER PAGE: click on SHOW RESULT button",
"Fri Feb 22 14:42:15 CET 2019 --> USER PAGE: click on KD SIMULATION button",
"Fri Feb 22 14:42:16 CET 2019 --> USER PAGE (KD simulation): click on text frame",
"Fri Feb 22 14:42:54 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Fri Feb 22 14:42:56 CET 2019 --> USER PAGE: click on SHOW RESULT button",
"Fri Feb 22 14:43:01 CET 2019 --> USER PAGE: click on LOGOUT button",
"Fri Feb 22 14:43:03 CET 2019 --> USER PAGE (logout): click on YES button"

Utente K

```

"KDSimulation_0_Time":40.376106,
"Login_0_Failures":0,
"Registration_0_Failures":0,
"Registration_0_Time":124.35913,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":10.723126,
"Login_0_Click":3,
"KDSimulation_1_Click":1,
"Login_0_Time":27.56449,
"ChangeUsername_0_Time":6.375863,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"KDSimulation_1_Time":27.381544,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":7,
"Registration_0_Click":19,
"KDSimulation_1_Failures":1,
"UserActions":[
  "Wed Feb 13 10:43:52 CET 2019 --> Home: click on SIGN UP button",
  "Wed Feb 13 10:43:57 CET 2019 --> Sign up: click on text user form",
  "Wed Feb 13 10:44:26 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:28 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:29 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:31 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:32 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:32 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:33 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:42 CET 2019 --> Sign up: click on CLEAR button",
  "Wed Feb 13 10:44:43 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:44 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:45 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:45 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:46 CET 2019 --> Sign up: click on grid",
  "Wed Feb 13 10:44:57 CET 2019 --> Sign up: click on OK button",
  "Wed Feb 13 10:44:57 CET 2019 --> Sign up OK! User: userK, Password:
    a1b2c3d2e1",
  "Wed Feb 13 10:45:02 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Wed Feb 13 10:45:06 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Wed Feb 13 10:45:57 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Wed Feb 13 10:46:03 CET 2019 --> Home: click on LOGIN button",
  "Wed Feb 13 10:46:10 CET 2019 --> LOGIN: click on password form",
  "Wed Feb 13 10:46:12 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 10:46:13 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 10:46:17 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 10:46:18 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Feb 13 10:46:20 CET 2019 --> LOGIN (entering the password):
    insert a char",

```

```

"Wed Feb 13 10:46:21 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 10:46:24 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 10:46:24 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 10:46:28 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 10:46:28 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Wed Feb 13 10:46:30 CET 2019 --> LOGIN: click on OK button",
"Wed Feb 13 10:46:30 CET 2019 --> LOGIN userK OK",
"Wed Feb 13 10:46:48 CET 2019 --> USER PAGE: click on CHANGE USERNAME
  button",
"Wed Feb 13 10:46:50 CET 2019 --> USER PAGE (Change user): click on
  text form",
"Wed Feb 13 10:46:55 CET 2019 --> USER PAGE (Change user): click on OK
  button",
"Wed Feb 13 10:46:55 CET 2019 --> USER PAGE (Change user): the
  username inserted is valid",
"Wed Feb 13 10:46:55 CET 2019 --> USER PAGE (Change user): the new
  username is userK2",
"Wed Feb 13 10:47:07 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
  button",
"Wed Feb 13 10:47:18 CET 2019 --> USER PAGE (Change password): click
  on OK button",
"Wed Feb 13 10:47:18 CET 2019 --> USER PAGE (Change password): the
  password inserted is valid ",
"Wed Feb 13 10:47:18 CET 2019 --> USER PAGE (Change password): new
  password is a2--->b3--->c4--->d3--->e2--->f1",
"Wed Feb 13 10:47:24 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Wed Feb 13 10:47:26 CET 2019 --> USER PAGE (KD simulation): click on
  text frame",
"Wed Feb 13 10:48:05 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Wed Feb 13 10:48:08 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Wed Feb 13 10:48:14 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Wed Feb 13 10:48:41 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Wed Feb 13 10:48:45 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Wed Feb 13 10:49:58 CET 2019 --> USER PAGE: click on LOGOUT button",
"Wed Feb 13 10:50:00 CET 2019 --> USER PAGE (logout): click on YES
  button"

```

]

Utente L

```

"Login_0_Failures":0,
"KDSimulation_3_Failures":1,
"Registration_0_Failures":0,
"KDSimulation_2_Click":1,
"ChangeUsername_0_Failures":0,
"Login_0_Click":4,
"Login_0_Time":97.868774,
"KDSimulation_1_Time":27.368298,

```

```

"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":8,
"KDSimulation_1_Failures":1,
"KDSimulation_2_Failures":0,
"KDSimulation_3_Time":30.41859,
"KDSimulation_0_Time":32.219772,
"KDSimulation_3_Click":1,
"Registration_0_Time":87.92463,
"ChangePassword_0_Time":17.65774,
"KDSimulation_1_Click":1,
"ChangeUsername_0_Time":8.991256,
"ChangePassword_0_Failures":1,
"KDSimulation_0_Failures":0,
"Registration_0_Click":14,
"KDSimulation_2_Time":20.799286,
"UserActions":[
  "Thu Feb 21 18:40:00 CET 2019 --> Home: click on SIGN UP button",
  "Thu Feb 21 18:40:04 CET 2019 --> Sign up: click on text user form",
  "Thu Feb 21 18:40:12 CET 2019 --> Sign up: click on grid",
  "Thu Feb 21 18:40:19 CET 2019 --> Sign up: click on CLEAR button",
  "Thu Feb 21 18:40:21 CET 2019 --> Sign up: click on grid",
  "Thu Feb 21 18:40:23 CET 2019 --> Sign up: click on grid",
  "Thu Feb 21 18:40:29 CET 2019 --> Sign up: click on grid",
  "Thu Feb 21 18:40:29 CET 2019 --> Sign up: click on grid",
  "Thu Feb 21 18:40:32 CET 2019 --> Sign up: click on grid",
  "Thu Feb 21 18:40:37 CET 2019 --> Sign up: click on OK button",
  "Thu Feb 21 18:40:38 CET 2019 --> Sign up OK! User: userL, Password:
    c1a1c1c1a1",
  "Thu Feb 21 18:40:41 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Thu Feb 21 18:40:43 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Thu Feb 21 18:40:44 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Thu Feb 21 18:41:28 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Thu Feb 21 18:41:31 CET 2019 --> Home: click on LOGIN button",
  "Thu Feb 21 18:41:51 CET 2019 --> LOGIN: click on password form",
  "Thu Feb 21 18:41:57 CET 2019 --> LOGIN: click on password form",
  "Thu Feb 21 18:41:58 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 21 18:41:59 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 21 18:41:59 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 21 18:42:00 CET 2019 --> LOGIN (entering the password):
    backspace",
  "Thu Feb 21 18:42:00 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 21 18:42:02 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Thu Feb 21 18:42:04 CET 2019 --> LOGIN (entering the password):
    backspace",
  "Thu Feb 21 18:42:06 CET 2019 --> LOGIN (entering the password):
    insert a char",

```

"Thu Feb 21 18:42:06 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:10 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:10 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:10 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:11 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:11 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:14 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:17 CET 2019 --> LOGIN (entering the password):
backspace",
"Thu Feb 21 18:42:19 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:19 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:19 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:22 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:22 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:24 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:25 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:25 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:26 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:35 CET 2019 --> LOGIN (entering the password):
backspace",
"Thu Feb 21 18:42:37 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:37 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:38 CET 2019 --> LOGIN (entering the password):
backspace",
"Thu Feb 21 18:42:38 CET 2019 --> LOGIN (entering the password):
backspace",
"Thu Feb 21 18:42:39 CET 2019 --> LOGIN (entering the password):
backspace",
"Thu Feb 21 18:42:42 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:42 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:45 CET 2019 --> LOGIN (entering the password):
backspace",
"Thu Feb 21 18:42:45 CET 2019 --> LOGIN (entering the password):
backspace",
"Thu Feb 21 18:42:45 CET 2019 --> LOGIN (entering the password):
backspace",

"Thu Feb 21 18:42:45 CET 2019 --> LOGIN (entering the password):
backspace",
"Thu Feb 21 18:42:47 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:47 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:49 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:50 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:52 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:53 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:56 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:57 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:59 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:42:59 CET 2019 --> LOGIN (entering the password):
insert a char",
"Thu Feb 21 18:43:09 CET 2019 --> LOGIN: click on OK button",
"Thu Feb 21 18:43:09 CET 2019 --> LOGIN userL OK",
"Thu Feb 21 18:43:20 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Thu Feb 21 18:43:22 CET 2019 --> USER PAGE (Change user): click on
text form",
"Thu Feb 21 18:43:29 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Thu Feb 21 18:43:29 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Thu Feb 21 18:43:29 CET 2019 --> USER PAGE (Change user): the new
username is userL2",
"Thu Feb 21 18:43:34 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Thu Feb 21 18:43:35 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Thu Feb 21 18:43:52 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Thu Feb 21 18:43:52 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Thu Feb 21 18:43:52 CET 2019 --> USER PAGE (Change password): new
password is a1--->c1--->c1--->c1--->a1--->a2",
"Thu Feb 21 18:43:56 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Thu Feb 21 18:43:58 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Thu Feb 21 18:44:28 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Thu Feb 21 18:44:31 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Thu Feb 21 18:44:31 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Thu Feb 21 18:44:39 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Thu Feb 21 18:45:06 CET 2019 --> USER PAGE (KD simulation): NO MATCH",

```

"Thu Feb 21 18:45:08 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Thu Feb 21 18:45:11 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Thu Feb 21 18:45:12 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Thu Feb 21 18:45:33 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Thu Feb 21 18:45:35 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Thu Feb 21 18:45:39 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Thu Feb 21 18:46:09 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Thu Feb 21 18:46:10 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Thu Feb 21 18:46:19 CET 2019 --> USER PAGE: click on LOGOUT button",
"Thu Feb 21 18:46:21 CET 2019 --> USER PAGE (logout): click on YES
button"

```

]

Utente M

```

"KDSimulation_0_Time":54.75751,
"Login_0_Failures":1,
"Registration_0_Failures":0,
"Registration_0_Time":172.77426,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":13.572057,
"Login_0_Click":7,
"KDSimulation_1_Click":1,
"Login_0_Time":66.15041,
"ChangeUsername_0_Time":15.76284,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"KDSimulation_1_Time":21.27226,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":5,
"Registration_0_Click":9,
"KDSimulation_1_Failures":1,
"UserActions":[
  "Fri Feb 22 17:57:21 CET 2019 --> Home: click on SIGN UP button",
  "Fri Feb 22 17:57:30 CET 2019 --> Sign up: click on text user form",
  "Fri Feb 22 17:58:23 CET 2019 --> Sign up: click on grid",
  "Fri Feb 22 17:58:30 CET 2019 --> Sign up: click on grid",
  "Fri Feb 22 17:58:32 CET 2019 --> Sign up: click on grid",
  "Fri Feb 22 17:58:42 CET 2019 --> Sign up: click on OK button",
  "Fri Feb 22 17:58:42 CET 2019 --> Sign up OK! User: UserM, Password:
a1c3a3",
  "Fri Feb 22 17:58:45 CET 2019 --> Sign up: Select the language
ITALIANO",
  "Fri Feb 22 17:58:50 CET 2019 --> Sign up (KD registration): click on
text form",
  "Fri Feb 22 18:00:14 CET 2019 --> Sign up (KD registration): confirm
the text inserted",
  "Fri Feb 22 18:00:17 CET 2019 --> Home: click on LOGIN button",
  "Fri Feb 22 18:00:37 CET 2019 --> LOGIN: click on password form",

```

"Fri Feb 22 18:00:39 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:00:39 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:00:42 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:00:43 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:00:46 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:00:47 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:00:50 CET 2019 --> LOGIN: click on OK button",
"Fri Feb 22 18:00:50 CET 2019 --> LOGIN: the password inserted is
wrong",
"Fri Feb 22 18:00:53 CET 2019 --> LOGIN: click on password form",
"Fri Feb 22 18:00:53 CET 2019 --> LOGIN: click on password form",
"Fri Feb 22 18:00:54 CET 2019 --> LOGIN: click on password form",
"Fri Feb 22 18:00:55 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Feb 22 18:00:57 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:00:58 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Feb 22 18:00:58 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Feb 22 18:00:59 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Feb 22 18:00:59 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Feb 22 18:01:02 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:03 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:06 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:07 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:11 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Feb 22 18:01:11 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Feb 22 18:01:14 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:15 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:17 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:18 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:20 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:21 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Feb 22 18:01:23 CET 2019 --> LOGIN: click on OK button",
"Fri Feb 22 18:01:23 CET 2019 --> LOGIN userM OK",

```

"Fri Feb 22 18:01:38 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Fri Feb 22 18:01:41 CET 2019 --> USER PAGE (Change user): click on
text form",
"Fri Feb 22 18:01:54 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Fri Feb 22 18:01:54 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Fri Feb 22 18:01:54 CET 2019 --> USER PAGE (Change user): the new
username is userM2",
"Fri Feb 22 18:01:58 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Fri Feb 22 18:02:11 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Fri Feb 22 18:02:11 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Fri Feb 22 18:02:11 CET 2019 --> USER PAGE (Change password): new
password is a1--->a3--->a5--->a7",
"Fri Feb 22 18:02:23 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Fri Feb 22 18:02:25 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Fri Feb 22 18:03:18 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Fri Feb 22 18:03:21 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Fri Feb 22 18:03:33 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Fri Feb 22 18:03:54 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Fri Feb 22 18:03:55 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Fri Feb 22 18:04:10 CET 2019 --> USER PAGE: click on LOGOUT button",
"Fri Feb 22 18:04:12 CET 2019 --> USER PAGE (logout): click on YES
button"

```

]

Utente N

```

"Login_0_Failures":0,
"Registration_0_Failures":0,
"ChangeUsername_0_Failures":0,
"Login_0_Click":3,
"Login_0_Time":22.96388,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":1,
"ChangePassword_0_Click":5,
"KDSimulation_0_Time":35.552402,
"Registration_0_Time":91.3881,
"ChangePassword_0_Time":8.701988,
"ChangeUsername_0_Time":13.144216,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"Registration_0_Click":10,
"UserActions":[
  "Tue Feb 26 22:06:22 CET 2019 --> Home: click on SIGN UP button",
  "Tue Feb 26 22:06:35 CET 2019 --> Sign up: click on grid",
  "Tue Feb 26 22:06:39 CET 2019 --> Sign up: click on grid",
  "Tue Feb 26 22:06:40 CET 2019 --> Sign up: click on grid",

```

"Tue Feb 26 22:06:42 CET 2019 --> Sign up: click on grid",
"Tue Feb 26 22:06:44 CET 2019 --> Sign up: click on text user form",
"Tue Feb 26 22:07:03 CET 2019 --> Sign up: click on OK button",
"Tue Feb 26 22:07:03 CET 2019 --> Sign up OK! User: userN, Password:
a1a8h8h1",
"Tue Feb 26 22:07:05 CET 2019 --> Sign up: Select the language
ITALIANO",
"Tue Feb 26 22:07:08 CET 2019 --> Sign up (KD registration): click on
text form",
"Tue Feb 26 22:07:53 CET 2019 --> Sign up (KD registration): confirm
the text inserted",
"Tue Feb 26 22:08:27 CET 2019 --> Home: click on LOGIN button",
"Tue Feb 26 22:08:34 CET 2019 --> LOGIN: click on password form",
"Tue Feb 26 22:08:36 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 26 22:08:36 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 26 22:08:40 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 26 22:08:40 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 26 22:08:43 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 26 22:08:43 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 26 22:08:47 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 26 22:08:47 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Feb 26 22:08:50 CET 2019 --> LOGIN: click on OK button",
"Tue Feb 26 22:08:50 CET 2019 --> LOGIN userN OK",
"Tue Feb 26 22:09:58 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Tue Feb 26 22:10:00 CET 2019 --> USER PAGE (Change user): click on
text form",
"Tue Feb 26 22:10:11 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Tue Feb 26 22:10:11 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Tue Feb 26 22:10:11 CET 2019 --> USER PAGE (Change user): the new
username is userN2",
"Tue Feb 26 22:10:13 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Tue Feb 26 22:10:22 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Tue Feb 26 22:10:22 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Tue Feb 26 22:10:22 CET 2019 --> USER PAGE (Change password): new
password is a1--->h1--->h8--->a8",
"Tue Feb 26 22:10:27 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Tue Feb 26 22:11:02 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Tue Feb 26 22:11:04 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Tue Feb 26 22:12:06 CET 2019 --> USER PAGE: click on LOGOUT button",
"Tue Feb 26 22:12:09 CET 2019 --> USER PAGE (logout): click on YES
button"

]

Utente O

```

"Login_0_Failures":0,
"KDSimulation_4_Click":2,
"KDSimulation_3_Failures":1,
"Registration_0_Failures":0,
"KDSimulation_2_Click":1,
"ChangeUsername_0_Failures":0,
"Login_0_Click":3,
"Login_0_Time":25.665709,
"KDSimulation_1_Time":6.314484,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":4,
"ChangePassword_0_Click":4,
"KDSimulation_4_Time":40.755707,
"KDSimulation_4_Failures":0,
"KDSimulation_1_Failures":1,
"KDSimulation_2_Failures":1,
"KDSimulation_3_Time":27.77671,
"KDSimulation_0_Time":58.936565,
"KDSimulation_3_Click":2,
"Registration_0_Time":127.09094,
"ChangePassword_0_Time":13.724079,
"KDSimulation_1_Click":1,
"ChangeUsername_0_Time":7.277485,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"Registration_0_Click":11,
"KDSimulation_2_Time":4.441756,
"UserActions":[
  "Sat Mar 02 12:43:11 CET 2019 --> Home: click on SIGN UP button",
  "Sat Mar 02 12:43:26 CET 2019 --> Sign up: click on grid",
  "Sat Mar 02 12:43:39 CET 2019 --> Sign up: click on CLEAR button",
  "Sat Mar 02 12:43:41 CET 2019 --> Sign up: click on grid",
  "Sat Mar 02 12:43:42 CET 2019 --> Sign up: click on grid",
  "Sat Mar 02 12:43:49 CET 2019 --> Sign up: click on grid",
  "Sat Mar 02 12:43:55 CET 2019 --> Sign up: click on text user form",
  "Sat Mar 02 12:44:07 CET 2019 --> Sign up: click on OK button",
  "Sat Mar 02 12:44:07 CET 2019 --> Sign up OK! User: user0, Password:
    h1a8a4",
  "Sat Mar 02 12:44:10 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Sat Mar 02 12:44:11 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Sat Mar 02 12:45:18 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Sat Mar 02 12:45:27 CET 2019 --> Home: click on LOGIN button",
  "Sat Mar 02 12:45:39 CET 2019 --> LOGIN: click on password form",
  "Sat Mar 02 12:45:41 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Sat Mar 02 12:45:42 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Sat Mar 02 12:45:47 CET 2019 --> LOGIN (entering the password):
    insert a char",

```

"Sat Mar 02 12:45:47 CET 2019 --> LOGIN (entering the password):
insert a char",
"Sat Mar 02 12:45:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Sat Mar 02 12:45:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Sat Mar 02 12:45:53 CET 2019 --> LOGIN: click on OK button",
"Sat Mar 02 12:45:53 CET 2019 --> LOGIN user0 OK",
"Sat Mar 02 12:46:05 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Sat Mar 02 12:46:07 CET 2019 --> USER PAGE (Change user): click on
text form",
"Sat Mar 02 12:46:12 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Sat Mar 02 12:46:12 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Sat Mar 02 12:46:12 CET 2019 --> USER PAGE (Change user): the new
username is user02",
"Sat Mar 02 12:46:15 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Sat Mar 02 12:46:29 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Sat Mar 02 12:46:29 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Sat Mar 02 12:46:29 CET 2019 --> USER PAGE (Change password): new
password is c1--->c8--->h1",
"Sat Mar 02 12:46:31 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Sat Mar 02 12:46:33 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Sat Mar 02 12:47:23 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Sat Mar 02 12:47:25 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Sat Mar 02 12:47:30 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Sat Mar 02 12:47:37 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Sat Mar 02 12:47:56 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Sat Mar 02 12:48:02 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Sat Mar 02 12:48:05 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Sat Mar 02 12:48:08 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Sat Mar 02 12:48:12 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Sat Mar 02 12:48:14 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Sat Mar 02 12:48:18 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Sat Mar 02 12:48:20 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Sat Mar 02 12:48:46 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Sat Mar 02 12:48:48 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Sat Mar 02 12:48:57 CET 2019 --> USER PAGE: click on SHOW RESULT
button",

```

"Sat Mar 02 12:48:58 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Sat Mar 02 12:48:59 CET 2019 --> USER PAGE (KD simulation): click on
  text frame",
"Sat Mar 02 12:49:38 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Sat Mar 02 12:49:41 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Sat Mar 02 12:50:00 CET 2019 --> USER PAGE: click on LOGOUT button",
"Sat Mar 02 12:50:02 CET 2019 --> USER PAGE (logout): click on YES
  button"
]

```

Utente P

```

"KDSimulation_0_Time":102.579384,
"Login_0_Failures":0,
"Registration_0_Failures":0,
"Registration_0_Time":104.77431,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":6.12322,
"Login_0_Click":3,
"KDSimulation_1_Click":1,
"Login_0_Time":18.850311,
"ChangeUsername_0_Time":7.2346897,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"KDSimulation_1_Time":93.71572,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":4,
"Registration_0_Click":9,
"KDSimulation_1_Failures":1,
"UserActions":[
  "Sun Mar 03 19:10:50 CET 2019 --> Home: click on SIGN UP button",
  "Sun Mar 03 19:11:01 CET 2019 --> Sign up: click on grid",
  "Sun Mar 03 19:11:07 CET 2019 --> Sign up: click on grid",
  "Sun Mar 03 19:11:09 CET 2019 --> Sign up: click on grid",
  "Sun Mar 03 19:11:12 CET 2019 --> Sign up: click on text user form",
  "Sun Mar 03 19:11:18 CET 2019 --> Sign up: click on OK button",
  "Sun Mar 03 19:11:21 CET 2019 --> Sign up OK! User: userP, Password:
    a1a8h1",
  "Sun Mar 03 19:11:23 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Sun Mar 03 19:11:26 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Sun Mar 03 19:12:34 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Sun Mar 03 19:12:38 CET 2019 --> Home: click on LOGIN button",
  "Sun Mar 03 19:12:45 CET 2019 --> LOGIN: click on password form",
  "Sun Mar 03 19:12:46 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Sun Mar 03 19:12:47 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Sun Mar 03 19:12:49 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Sun Mar 03 19:12:51 CET 2019 --> LOGIN (entering the password):
    insert a char",

```

```

"Sun Mar 03 19:12:53 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Sun Mar 03 19:12:54 CET 2019 --> LOGIN (entering the password):
  insert a char",
"Sun Mar 03 19:12:57 CET 2019 --> LOGIN: click on OK button",
"Sun Mar 03 19:12:57 CET 2019 --> LOGIN userP OK",
"Sun Mar 03 19:13:04 CET 2019 --> USER PAGE: click on CHANGE USERNAME
  button",
"Sun Mar 03 19:13:06 CET 2019 --> USER PAGE (Change user): click on
  text form",
"Sun Mar 03 19:13:11 CET 2019 --> USER PAGE (Change user): click on OK
  button",
"Sun Mar 03 19:13:11 CET 2019 --> USER PAGE (Change user): the
  username inserted is valid",
"Sun Mar 03 19:13:11 CET 2019 --> USER PAGE (Change user): the new
  username is userP2",
"Sun Mar 03 19:13:14 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
  button",
"Sun Mar 03 19:13:20 CET 2019 --> USER PAGE (Change password): click
  on OK button",
"Sun Mar 03 19:13:20 CET 2019 --> USER PAGE (Change password): the
  password inserted is valid ",
"Sun Mar 03 19:13:20 CET 2019 --> USER PAGE (Change password): new
  password is a1--->a2--->a3",
"Sun Mar 03 19:13:23 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Sun Mar 03 19:14:18 CET 2019 --> USER PAGE (KD simulation): click on
  CLEAR button",
"Sun Mar 03 19:15:05 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Sun Mar 03 19:15:07 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Sun Mar 03 19:15:11 CET 2019 --> USER PAGE: click on KD SIMULATION
  button",
"Sun Mar 03 19:16:44 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Sun Mar 03 19:16:48 CET 2019 --> USER PAGE: click on SHOW RESULT
  button",
"Sun Mar 03 19:17:08 CET 2019 --> USER PAGE: click on LOGOUT button",
"Sun Mar 03 19:17:11 CET 2019 --> USER PAGE (logout): click on YES
  button"

```

]

Utente Q

```

"Login_0_Failures":0,
"KDSimulation_4_Click":1,
"KDSimulation_3_Failures":1,
"Registration_0_Failures":0,
"KDSimulation_2_Click":2,
"ChangeUsername_0_Failures":0,
"Login_0_Click":3,
"Login_0_Time":18.90038,
"KDSimulation_1_Time":22.934391,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":1,
"ChangePassword_0_Click":4,
"KDSimulation_4_Time":27.383995,
"KDSimulation_4_Failures":0,

```

```

"KDSimulation_1_Failures":1,
"KDSimulation_2_Failures":1,
"KDSimulation_3_Time":24.061773,
"KDSimulation_0_Time":34.69611,
"KDSimulation_3_Click":2,
"Registration_0_Time":92.58418,
"ChangePassword_0_Time":8.35103,
"KDSimulation_1_Click":1,
"ChangeUsername_0_Time":4.175726,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"Registration_0_Click":13,
"KDSimulation_2_Time":23.035543,
"UserActions":[
  "Mon Mar 04 13:45:33 CET 2019 --> Home: click on SIGN UP button",
  "Mon Mar 04 13:45:46 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 13:45:48 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 13:45:50 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 13:46:06 CET 2019 --> Sign up: click on CLEAR button",
  "Mon Mar 04 13:46:08 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 13:46:10 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 13:46:11 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 13:46:13 CET 2019 --> Sign up: click on text user form",
  "Mon Mar 04 13:46:20 CET 2019 --> Sign up: click on OK button",
  "Mon Mar 04 13:46:21 CET 2019 --> Sign up OK! User: userQ, Password:
    c1c8h1",
  "Mon Mar 04 13:46:24 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Mon Mar 04 13:46:26 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Mon Mar 04 13:47:06 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Mon Mar 04 13:47:08 CET 2019 --> Home: click on LOGIN button",
  "Mon Mar 04 13:47:17 CET 2019 --> LOGIN: click on password form",
  "Mon Mar 04 13:47:19 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 13:47:19 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 13:47:20 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 13:47:23 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 13:47:23 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 13:47:24 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 13:47:26 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 13:47:26 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 13:47:26 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 13:47:27 CET 2019 --> LOGIN: click on OK button",
  "Mon Mar 04 13:47:27 CET 2019 --> LOGIN userQ OK",
  "Mon Mar 04 13:47:32 CET 2019 --> USER PAGE: click on CHANGE USERNAME
    button",

```

```

"Mon Mar 04 13:47:33 CET 2019 --> USER PAGE (Change user): click on
text form",
"Mon Mar 04 13:47:36 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Mon Mar 04 13:47:36 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Mon Mar 04 13:47:36 CET 2019 --> USER PAGE (Change user): the new
username is userQ2",
"Mon Mar 04 13:47:40 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Mon Mar 04 13:47:48 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Mon Mar 04 13:47:48 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Mon Mar 04 13:47:48 CET 2019 --> USER PAGE (Change password): new
password is a1--->a8--->h1",
"Mon Mar 04 13:47:51 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Mar 04 13:48:25 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Mon Mar 04 13:48:28 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Mar 04 13:48:37 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Mar 04 13:49:00 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Mon Mar 04 13:49:01 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Mar 04 13:49:06 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Mar 04 13:49:07 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Mon Mar 04 13:49:29 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Mon Mar 04 13:49:33 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Mar 04 13:49:36 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Mar 04 13:49:37 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Mon Mar 04 13:50:00 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Mon Mar 04 13:50:01 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Mar 04 13:50:03 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Mar 04 13:50:30 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Mon Mar 04 13:50:31 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Mar 04 13:50:47 CET 2019 --> USER PAGE: click on LOGOUT button",
"Mon Mar 04 13:50:49 CET 2019 --> USER PAGE (logout): click on YES
button"

```

]

Utente R

In questo caso le prime due simulazioni di KD sono state eseguite da persone differenti dall'utente R. La simulazione attendibile, eseguita dall'utente R, è quella denominata "KDSimulation_2_*".

```
"KDSimulation_2_Failures":0,
```

```

"KDSimulation_0_Time":33.663303,
"Login_0_Failures":0,
"Registration_0_Failures":0,
"KDSimulation_2_Click":2,
"Registration_0_Time":46.58409,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":4.727454,
"Login_0_Click":8,
"KDSimulation_1_Click":1,
"Login_0_Time":66.94663,
"ChangeUsername_0_Time":9.017416,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":1,
"KDSimulation_1_Time":19.47898,
"ChangeUsername_0_Click":4,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":4,
"Registration_0_Click":12,
"KDSimulation_1_Failures":1,
"KDSimulation_2_Time":23.312721,
"UserActions":[
  "Mon Mar 04 15:38:12 CET 2019 --> Home: click on SIGN UP button",
  "Mon Mar 04 15:38:18 CET 2019 --> Sign up: click on CANCEL button",
  "Mon Mar 04 15:38:20 CET 2019 --> Home: click on LOGIN button",
  "Mon Mar 04 15:38:21 CET 2019 --> LOGIN: click on user form",
  "Mon Mar 04 15:38:21 CET 2019 --> LOGIN: click on user form",
  "Mon Mar 04 15:38:26 CET 2019 --> LOGIN: click on CANCEL button",
  "Mon Mar 04 15:38:27 CET 2019 --> Home: click on SIGN UP button",
  "Mon Mar 04 15:38:29 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 15:38:30 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 15:38:31 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 15:38:32 CET 2019 --> Sign up: click on grid",
  "Mon Mar 04 15:38:33 CET 2019 --> Sign up: click on text user form",
  "Mon Mar 04 15:38:36 CET 2019 --> Sign up: click on OK button",
  "Mon Mar 04 15:38:36 CET 2019 --> Sign up OK! User: userR, Password:
    a1a8h1h8",
  "Mon Mar 04 15:38:37 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Mon Mar 04 15:38:38 CET 2019 --> Sign up (KD registration): click on
    text form",
  "Mon Mar 04 15:38:58 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Mon Mar 04 15:39:00 CET 2019 --> Home: click on LOGIN button",
  "Mon Mar 04 15:39:00 CET 2019 --> LOGIN: click on user form",
  "Mon Mar 04 15:39:04 CET 2019 --> LOGIN: click on password form",
  "Mon Mar 04 15:39:05 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 15:39:05 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 15:39:05 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 15:39:11 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 15:39:11 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Mon Mar 04 15:39:11 CET 2019 --> LOGIN (entering the password):
    insert a char",

```

"Mon Mar 04 15:39:21 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Mar 04 15:39:22 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Mar 04 15:39:22 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Mar 04 15:39:24 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Mar 04 15:39:24 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Mar 04 15:39:24 CET 2019 --> LOGIN (entering the password):
insert a char",
"Mon Mar 04 15:39:27 CET 2019 --> LOGIN: click on OK button",
"Mon Mar 04 15:39:27 CET 2019 --> LOGIN userR OK",
"Mon Mar 04 15:39:29 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Mon Mar 04 15:39:30 CET 2019 --> USER PAGE (Change user): click on
text form",
"Mon Mar 04 15:39:33 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Mon Mar 04 15:39:33 CET 2019 --> USER PAGE (Change user): the
username inserted already exists",
"Mon Mar 04 15:39:34 CET 2019 --> USER PAGE (Change user): click on
text form",
"Mon Mar 04 15:39:38 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Mon Mar 04 15:39:38 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Mon Mar 04 15:39:38 CET 2019 --> USER PAGE (Change user): the new
username is userR2",
"Mon Mar 04 15:39:40 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Mon Mar 04 15:39:44 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Mon Mar 04 15:39:44 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Mon Mar 04 15:39:44 CET 2019 --> USER PAGE (Change password): new
password is a1--->h8--->h1",
"Mon Mar 04 15:39:46 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Mar 04 15:39:47 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Mon Mar 04 15:40:20 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Mon Mar 04 15:40:20 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Mar 04 15:40:22 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Mar 04 15:40:41 CET 2019 --> USER PAGE (KD simulation): NO MATCH",
"Mon Mar 04 15:40:42 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Mon Mar 04 15:40:43 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Mon Mar 04 15:40:44 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Mon Mar 04 15:41:06 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Mon Mar 04 15:41:08 CET 2019 --> USER PAGE: click on SHOW RESULT
button",

```

"Mon Mar 04 15:41:14 CET 2019 --> USER PAGE: click on LOGOUT button",
"Mon Mar 04 15:41:16 CET 2019 --> USER PAGE (logout): click on YES
button"
]

```

Utente S

```

"Login_0_Click":18,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":2,
"ChangePassword_0_Click":5,
"KDSimulation_0_Time":46.59394,
"Registration_0_Time":150.85803,
"ChangePassword_0_Time":18.131908,
"ChangeUsername_0_Time":19.448437,
"KDSimulation_0_Failures":0,
"Registration_0_Click":15,
"Login_0_Failures":4,
"Registration_0_Failures":0,
"ChangeUsername_0_Failures":0,
"Login_0_Time":330.7795,
"ChangePassword_0_Failures":0,
"UserActions":[
  "Fri Mar 01 17:49:25 CET 2019 --> Home: click on SIGN UP button",
  "Fri Mar 01 17:49:32 CET 2019 --> Sign up: click on text user form",
  "Fri Mar 01 17:49:34 CET 2019 --> Sign up: click on SHOW PASSWORD
button",
  "Fri Mar 01 17:49:34 CET 2019 --> Sign up: click on SHOW PASSWORD
button",
  "Fri Mar 01 17:49:38 CET 2019 --> Sign up: click on grid",
  "Fri Mar 01 17:49:48 CET 2019 --> Sign up: click on grid",
  "Fri Mar 01 17:49:50 CET 2019 --> Sign up: click on grid",
  "Fri Mar 01 17:49:52 CET 2019 --> Sign up: click on grid",
  "Fri Mar 01 17:49:55 CET 2019 --> Sign up: click on grid",
  "Fri Mar 01 17:49:59 CET 2019 --> Sign up: click on grid",
  "Fri Mar 01 17:50:46 CET 2019 --> Sign up: click on text user form",
  "Fri Mar 01 17:50:51 CET 2019 --> Sign up: click on OK button",
  "Fri Mar 01 17:50:51 CET 2019 --> Sign up OK! User: userS, Password:
a1b4e4g2g7c2",
  "Fri Mar 01 17:51:05 CET 2019 --> Sign up: Select the language
ITALIANO",
  "Fri Mar 01 17:51:11 CET 2019 --> Sign up (KD registration): click on
text form",
  "Fri Mar 01 17:51:56 CET 2019 --> Sign up (KD registration): confirm
the text inserted",
  "Fri Mar 01 17:51:58 CET 2019 --> Home: click on LOGIN button",
  "Fri Mar 01 17:54:06 CET 2019 --> LOGIN: click on user form",
  "Fri Mar 01 17:54:10 CET 2019 --> LOGIN: click on password form",
  "Fri Mar 01 17:54:15 CET 2019 --> LOGIN: click on password form",
  "Fri Mar 01 17:54:15 CET 2019 --> LOGIN: click on password form",
  "Fri Mar 01 17:54:19 CET 2019 --> LOGIN: click on password form",
  "Fri Mar 01 17:54:21 CET 2019 --> LOGIN (entering the password):
insert a char",
  "Fri Mar 01 17:54:21 CET 2019 --> LOGIN (entering the password):
insert a char",

```

"Fri Mar 01 17:54:23 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:54:24 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:54:24 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:25 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:27 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:27 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:35 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:35 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:44 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:45 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:47 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:54:47 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:47 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:49 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:49 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:53 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:53 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:54 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:54 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:55 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:54:55 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:08 CET 2019 --> LOGIN: click on OK button",
"Fri Mar 01 17:55:08 CET 2019 --> LOGIN: the password inserted is
wrong",
"Fri Mar 01 17:55:11 CET 2019 --> LOGIN: click on password form",
"Fri Mar 01 17:55:13 CET 2019 --> LOGIN: click on password form",
"Fri Mar 01 17:55:19 CET 2019 --> LOGIN: click on OK button",
"Fri Mar 01 17:55:19 CET 2019 --> LOGIN: the password inserted is
wrong",
"Fri Mar 01 17:55:20 CET 2019 --> LOGIN: click on OK button",
"Fri Mar 01 17:55:20 CET 2019 --> LOGIN: the password inserted is
wrong",
"Fri Mar 01 17:55:21 CET 2019 --> LOGIN: click on OK button",
"Fri Mar 01 17:55:21 CET 2019 --> LOGIN: the password inserted is
wrong",

"Fri Mar 01 17:55:25 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:25 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:29 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:55:32 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:32 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:43 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:43 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:43 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:43 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:43 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:43 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:44 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:44 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:44 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:44 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:44 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:44 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:47 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:55:47 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:55:49 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:49 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:51 CET 2019 --> LOGIN (entering the password): the
point inserted is not valid",
"Fri Mar 01 17:55:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:54 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:55:54 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:55:54 CET 2019 --> LOGIN (entering the password):
backspace",

"Fri Mar 01 17:55:55 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:55:55 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:55:56 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:56 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:55:56 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:55:57 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:00 CET 2019 --> LOGIN (entering the password):
backspace",
"Fri Mar 01 17:56:01 CET 2019 --> LOGIN: click on CANCEL button",
"Fri Mar 01 17:56:04 CET 2019 --> Home: click on LOGIN button",
"Fri Mar 01 17:56:10 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:10 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:10 CET 2019 --> LOGIN (entering the password): the
point inserted is not valid",
"Fri Mar 01 17:56:10 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:15 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:16 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:16 CET 2019 --> LOGIN (entering the password): the
point inserted is not valid",
"Fri Mar 01 17:56:16 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:34 CET 2019 --> LOGIN: click on CANCEL button",
"Fri Mar 01 17:56:44 CET 2019 --> Home: click on LOGIN button",
"Fri Mar 01 17:56:50 CET 2019 --> LOGIN: click on password form",
"Fri Mar 01 17:56:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:51 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:51 CET 2019 --> LOGIN (entering the password): the
point inserted is not valid",
"Fri Mar 01 17:56:52 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:57 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:57 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:56:57 CET 2019 --> LOGIN (entering the password): the
point inserted is not valid",
"Fri Mar 01 17:56:57 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:08 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:08 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:57:08 CET 2019 --> LOGIN (entering the password):
insert a char",

"Fri Mar 01 17:57:08 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:17 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:17 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:17 CET 2019 --> LOGIN (entering the password): the
point inserted is not valid",
"Fri Mar 01 17:57:17 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:22 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:22 CET 2019 --> LOGIN (entering the password): the
char inserted is not valid",
"Fri Mar 01 17:57:22 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:23 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:28 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:28 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:28 CET 2019 --> LOGIN (entering the password): the
point inserted is not valid",
"Fri Mar 01 17:57:28 CET 2019 --> LOGIN (entering the password):
insert a char",
"Fri Mar 01 17:57:30 CET 2019 --> LOGIN: click on OK button",
"Fri Mar 01 17:57:30 CET 2019 --> LOGIN userS OK",
"Fri Mar 01 17:57:53 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Fri Mar 01 17:57:55 CET 2019 --> USER PAGE (Change user): click on
text form",
"Fri Mar 01 17:58:12 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Fri Mar 01 17:58:12 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Fri Mar 01 17:58:12 CET 2019 --> USER PAGE (Change user): the new
username is userS2",
"Fri Mar 01 17:58:19 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Fri Mar 01 17:58:38 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Fri Mar 01 17:58:38 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Fri Mar 01 17:58:38 CET 2019 --> USER PAGE (Change password): new
password is a1--->a8--->h1--->h8",
"Fri Mar 01 17:58:51 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Fri Mar 01 17:58:52 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Fri Mar 01 17:59:37 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Fri Mar 01 17:59:49 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Fri Mar 01 18:00:14 CET 2019 --> USER PAGE: click on LOGOUT button",
"Fri Mar 01 18:00:16 CET 2019 --> USER PAGE (logout): click on YES
button",

Utente T

```

"KDSimulation_0_Time":23.792833,
"Login_0_Failures":0,
"Registration_0_Failures":0,
"Registration_0_Time":44.570625,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":6.825569,
"Login_0_Click":1,
"Login_0_Time":16.257935,
"ChangeUsername_0_Time":8.9098215,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"ChangeUsername_0_Click":1,
"KDSimulation_0_Click":1,
"ChangePassword_0_Click":5,
"Registration_0_Click":11,
"UserActions":[
  "Wed Mar 06 12:42:06 CET 2019 --> Home: click on SIGN UP button",
  "Wed Mar 06 12:42:09 CET 2019 --> Sign up: click on text user form",
  "Wed Mar 06 12:42:15 CET 2019 --> Sign up: click on grid",
  "Wed Mar 06 12:42:16 CET 2019 --> Sign up: click on grid",
  "Wed Mar 06 12:42:17 CET 2019 --> Sign up: click on grid",
  "Wed Mar 06 12:42:19 CET 2019 --> Sign up: click on grid",
  "Wed Mar 06 12:42:20 CET 2019 --> Sign up: click on SHOW PASSWORD
    button",
  "Wed Mar 06 12:42:23 CET 2019 --> Sign up: click on OK button",
  "Wed Mar 06 12:42:25 CET 2019 --> Sign up OK! User: userT, Password:
    ala8h1h8",
  "Wed Mar 06 12:42:25 CET 2019 --> Sign up: click on OK button",
  "Wed Mar 06 12:42:26 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Wed Mar 06 12:42:50 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Wed Mar 06 12:42:54 CET 2019 --> Home: click on LOGIN button",
  "Wed Mar 06 12:43:00 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Mar 06 12:43:01 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Mar 06 12:43:04 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Mar 06 12:43:04 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Mar 06 12:43:06 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Mar 06 12:43:07 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Mar 06 12:43:09 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Mar 06 12:43:09 CET 2019 --> LOGIN (entering the password):
    insert a char",
  "Wed Mar 06 12:43:10 CET 2019 --> LOGIN: press ENTER button and start
    data validation (= press OK button)",
  "Wed Mar 06 12:43:10 CET 2019 --> LOGIN userT OK",
  "Wed Mar 06 12:43:14 CET 2019 --> USER PAGE: click on CHANGE USERNAME
    button",

```

```

"Wed Mar 06 12:43:17 CET 2019 --> USER PAGE (Change user): click on
text form",
"Wed Mar 06 12:43:23 CET 2019 --> USER PAGE: (Change user) press ENTER
button (= press OK button)",
"Wed Mar 06 12:43:23 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Wed Mar 06 12:43:23 CET 2019 --> USER PAGE (Change user): the new
username is userT2",
"Wed Mar 06 12:43:24 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Wed Mar 06 12:43:31 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Wed Mar 06 12:43:31 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Wed Mar 06 12:43:31 CET 2019 --> USER PAGE (Change password): new
password is h8--->h1--->a8--->a1",
"Wed Mar 06 12:43:33 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Wed Mar 06 12:43:56 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Wed Mar 06 12:43:57 CET 2019 --> USER PAGE: click on SHOW RESULT
button",
"Wed Mar 06 12:43:58 CET 2019 --> USER PAGE: click on LOGOUT button",
"Wed Mar 06 12:44:00 CET 2019 --> USER PAGE (logout): click on YES
button"

```

]

Utente U

```

"KDSimulation_0_Time":63.54886,
>Login_0_Failures":0,
"Registration_0_Failures":0,
"Registration_0_Time":115.98837,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":16.061707,
>Login_0_Click":3,
>Login_0_Time":59.39257,
"ChangeUsername_0_Time":13.993912,
"ChangePassword_0_Failures":0,
"KDSimulation_0_Failures":0,
"ChangeUsername_0_Click":2,
"KDSimulation_0_Click":3,
"ChangePassword_0_Click":5,
"Registration_0_Click":9,
"UserActions":[
  "Tue Mar 05 22:27:23 CET 2019 --> Home: click on SIGN UP button",
  "Tue Mar 05 22:27:30 CET 2019 --> Sign up: click on text user form",
  "Tue Mar 05 22:27:45 CET 2019 --> Sign up: click on grid",
  "Tue Mar 05 22:27:48 CET 2019 --> Sign up: click on grid",
  "Tue Mar 05 22:27:48 CET 2019 --> Sign up: click on grid",
  "Tue Mar 05 22:27:49 CET 2019 --> Sign up: click on grid",
  "Tue Mar 05 22:27:55 CET 2019 --> Sign up: click on OK button",
  "Tue Mar 05 22:27:55 CET 2019 --> Sign up OK! User: userU, Password:
a1a2a3a4",
  "Tue Mar 05 22:27:59 CET 2019 --> Sign up: Select the language
ITALIANO",
  "Tue Mar 05 22:29:18 CET 2019 --> Sign up (KD registration): confirm
the text inserted",

```

"Tue Mar 05 22:32:26 CET 2019 --> Home: click on LOGIN button",
"Tue Mar 05 22:32:36 CET 2019 --> LOGIN: click on password form",
"Tue Mar 05 22:32:42 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:32:44 CET 2019 --> LOGIN (entering the password):
backspace",
"Tue Mar 05 22:32:46 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:32:48 CET 2019 --> LOGIN (entering the password):
backspace",
"Tue Mar 05 22:32:52 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:32:52 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:32:58 CET 2019 --> LOGIN (entering the password):
backspace",
"Tue Mar 05 22:33:03 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:33:04 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:33:08 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:33:08 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:33:14 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:33:14 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:33:21 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:33:21 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 22:33:25 CET 2019 --> LOGIN: click on OK button",
"Tue Mar 05 22:33:25 CET 2019 --> LOGIN userU OK",
"Tue Mar 05 22:33:35 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Tue Mar 05 22:33:41 CET 2019 --> USER PAGE (Change user): click on
text form",
"Tue Mar 05 22:33:49 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Tue Mar 05 22:33:49 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Tue Mar 05 22:33:49 CET 2019 --> USER PAGE (Change user): the new
username is userU2",
"Tue Mar 05 22:33:55 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Tue Mar 05 22:34:11 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Tue Mar 05 22:34:11 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Tue Mar 05 22:34:11 CET 2019 --> USER PAGE (Change password): new
password is b1--->b2--->b3--->b4",
"Tue Mar 05 22:36:08 CET 2019 --> USER PAGE: click on KD SIMULATION
button",
"Tue Mar 05 22:36:42 CET 2019 --> USER PAGE (KD simulation): click on
text frame",
"Tue Mar 05 22:37:11 CET 2019 --> USER PAGE (KD simulation): MATCH",

```

    "Tue Mar 05 22:37:16 CET 2019 --> USER PAGE: click on SHOW RESULT
      button",
    "Tue Mar 05 22:37:47 CET 2019 --> USER PAGE: click on LOGOUT button",
    "Tue Mar 05 22:37:49 CET 2019 --> USER PAGE (logout): click on YES
      button"
  ]

```

Utente V

L'utente V ha effettuato le attività secondarie (cambio username e cambio password) in una simulazione differente da quella primaria. Verranno quindi presentate due risultati parziali anziché uno unico.

```

"KDSimulation_0_Time":43.688324,
"Login_0_Failures":0,
"Registration_0_Failures":0,
"Registration_0_Time":72.07261,
"Login_0_Click":3,
"Login_0_Time":19.958157,
"KDSimulation_0_Failures":0,
"KDSimulation_0_Click":2,
"Registration_0_Click":9,
"UserActions":[
  "Tue Mar 05 22:02:40 CET 2019 --> Home: click on SIGN UP button",
  "Tue Mar 05 22:02:43 CET 2019 --> Sign up: click on text user form",
  "Tue Mar 05 22:02:44 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:44 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:45 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:45 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:45 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:47 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:47 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:47 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:48 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:48 CET 2019 --> Sign up: Press ENTER on user form (=
    OK button)",
  "Tue Mar 05 22:02:55 CET 2019 --> Sign up: click on grid",
  "Tue Mar 05 22:02:57 CET 2019 --> Sign up: click on grid",
  "Tue Mar 05 22:02:59 CET 2019 --> Sign up: click on grid",
  "Tue Mar 05 22:03:02 CET 2019 --> Sign up: click on grid",
  "Tue Mar 05 22:03:05 CET 2019 --> Sign up: click on OK button",
  "Tue Mar 05 22:03:05 CET 2019 --> Sign up OK! User: colette2,
    Password: a1a8h1h8",
  "Tue Mar 05 22:03:08 CET 2019 --> Sign up: Select the language
    ITALIANO",
  "Tue Mar 05 22:03:53 CET 2019 --> Sign up (KD registration): confirm
    the text inserted",
  "Tue Mar 05 22:03:58 CET 2019 --> Home: click on LOGIN button",

```

```

"Tue Mar 05 22:04:04 CET 2019 --> LOGIN: click on password form",
"Tue Mar 05 22:04:06 CET 2019 --> LOGIN (entering the password):
    insert a char",
"Tue Mar 05 22:04:06 CET 2019 --> LOGIN (entering the password):
    insert a char",
"Tue Mar 05 22:04:09 CET 2019 --> LOGIN (entering the password):
    insert a char",
"Tue Mar 05 22:04:09 CET 2019 --> LOGIN (entering the password):
    insert a char",
"Tue Mar 05 22:04:13 CET 2019 --> LOGIN (entering the password):
    insert a char",
"Tue Mar 05 22:04:13 CET 2019 --> LOGIN (entering the password):
    insert a char",
"Tue Mar 05 22:04:16 CET 2019 --> LOGIN (entering the password):
    insert a char",
"Tue Mar 05 22:04:16 CET 2019 --> LOGIN (entering the password):
    insert a char",
"Tue Mar 05 22:04:18 CET 2019 --> LOGIN: click on OK button",
"Tue Mar 05 22:04:18 CET 2019 --> LOGIN colette2 OK",
"Tue Mar 05 22:04:25 CET 2019 --> USER PAGE: click on KD SIMULATION
    button",
"Tue Mar 05 22:04:26 CET 2019 --> USER PAGE (KD simulation): click on
    text frame",
"Tue Mar 05 22:05:08 CET 2019 --> USER PAGE (KD simulation): MATCH",
"Tue Mar 05 22:05:13 CET 2019 --> USER PAGE: click on SHOW RESULT
    button",
"Tue Mar 05 22:05:22 CET 2019 --> USER PAGE: click on LOGOUT button",
"Tue Mar 05 22:05:25 CET 2019 --> USER PAGE (logout): click on YES
    button",
]

```

```

"Login_0_Click":2,
"Login_0_Failures":0,
"Login_0_Time":22.033035,
"ChangeUsername_0_Time":17.4302,
"ChangePassword_0_Failures":2,
"ChangeUsername_0_Click":2,
"ChangePassword_0_Click":9,
"ChangeUsername_0_Failures":0,
"ChangePassword_0_Time":42.093536,
"UserActions":[
    "Tue Mar 05 23:23:04 CET 2019 --> Home: click on LOGIN button",
    "Tue Mar 05 23:23:13 CET 2019 --> LOGIN (entering the password):
        insert a char",
    "Tue Mar 05 23:23:13 CET 2019 --> LOGIN (entering the password):
        insert a char",
    "Tue Mar 05 23:23:18 CET 2019 --> LOGIN (entering the password):
        insert a char",
    "Tue Mar 05 23:23:18 CET 2019 --> LOGIN (entering the password):
        insert a char",
    "Tue Mar 05 23:23:21 CET 2019 --> LOGIN (entering the password):
        insert a char",
    "Tue Mar 05 23:23:22 CET 2019 --> LOGIN (entering the password):
        insert a char",
    "Tue Mar 05 23:23:24 CET 2019 --> LOGIN (entering the password):
        insert a char",
]

```

"Tue Mar 05 23:23:24 CET 2019 --> LOGIN (entering the password):
insert a char",
"Tue Mar 05 23:23:26 CET 2019 --> LOGIN: click on OK button",
"Tue Mar 05 23:23:26 CET 2019 --> LOGIN colette2 OK",
"Tue Mar 05 23:23:29 CET 2019 --> USER PAGE: click on CHANGE USERNAME
button",
"Tue Mar 05 23:23:32 CET 2019 --> USER PAGE (Change user): click on
text form",
"Tue Mar 05 23:23:47 CET 2019 --> USER PAGE (Change user): click on OK
button",
"Tue Mar 05 23:23:47 CET 2019 --> USER PAGE (Change user): the
username inserted is valid",
"Tue Mar 05 23:23:47 CET 2019 --> USER PAGE (Change user): the new
username is miticacole",
"Tue Mar 05 23:23:51 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Tue Mar 05 23:24:13 CET 2019 --> USER PAGE (Change password): click
on CANCEL button",
"Tue Mar 05 23:24:15 CET 2019 --> USER PAGE: click on CHANGE PASSWORD
button",
"Tue Mar 05 23:24:33 CET 2019 --> USER PAGE (Change password): click
on OK button",
"Tue Mar 05 23:24:33 CET 2019 --> USER PAGE (Change password): the
password inserted is valid ",
"Tue Mar 05 23:24:33 CET 2019 --> USER PAGE (Change password): new
password is g2--->g7--->b7--->f3",
"Tue Mar 05 23:24:42 CET 2019 --> USER PAGE: click on LOGOUT button",
"Tue Mar 05 23:24:47 CET 2019 --> USER PAGE (logout): click on YES
button",
"Tue Mar 05 23:24:50 CET 2019 --> USER PAGE (logout): click on YES
button",
"Tue Mar 05 23:25:56 CET 2019 --> USER PAGE (logout): click on YES
button"

]

Bibliografia

- [1] 2016: Current State of Cybercrime, <https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf>
- [2] Internet Security Threat Report, <https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf>
- [3] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast, <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>
- [4] J. H. Huh, S. Verma, S. S. V. Rayala, R. B. Bobba, K. Beznosov, and H. Kim, “I Don’t Use Apple Pay Because It’s Less Secure ...: Perception of Security and Usability in Mobile Tap-and-Pay”, NDSS Symposium 2017, San Diego, California, 2017, DOI [10.14722/usec.2017.23021](https://doi.org/10.14722/usec.2017.23021)
- [5] iOS Security, https://www.apple.com/business/docs/iOS_Security_Guide.pdf
- [6] ISO 9241-210:2010, “Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems”, 2010, DOI [10.3403/30186637](https://doi.org/10.3403/30186637)
- [7] M. O. Leavitt and B. Shneiderman, “Research-based web design & usability guidelines”, U.S. Dept. of Health and Human Services, 2006, ISBN: 0-16-076270-7
- [8] ISO/IEC 2012. ISO/IEC 27000, “Information technology, Security techniques, Information security management systems”, 2014, DOI [10.4236/jis.2013.42011](https://doi.org/10.4236/jis.2013.42011)
- [9] M. Bishop and D. V. Klein, “Improving system security via proactive password checking”, Computers & Security, vol. 14, December 1995, pp. 233–249, DOI [10.1016/0167-4048\(95\)00003-Q](https://doi.org/10.1016/0167-4048(95)00003-Q)
- [10] “A Roadmap for Cybersecurity Research.” <https://www.dhs.gov/sites/default/files/publications/CSD-DHS-Cybersecurity-Roadmap.pdf>
- [11] M. Bishop, “UNIX Security: Threats and Solutions”, 1996, <http://seclab.cs.ucdavis.edu/projects/vulnerabilities/scriv/1996-share86.pdf>
- [12] S. Moller, N. Ben-Asher, K.-P. Engelbrecht, R. Englert, and J. Meyer, “Modeling the behavior of users who are confronted with security mechanisms”, Computers & Security, vol. 30, June 2011, pp. 242–256, DOI [10.1016/j.cose.2011.01.001](https://doi.org/10.1016/j.cose.2011.01.001)
- [13] P. Ketola and V. Roto, “Exploring user experience measurement needs”, vol. 5, January 2008, DOI [10.4018/jthi.2009070104](https://doi.org/10.4018/jthi.2009070104)
- [14] V. Roto, M. Obrist, and K. Väänänen-Vainio-Mattila, “User experience evaluation methods in academic and industrial contexts”, Workshop on User Experience Evaluation Methods, in conjunction with Interact’09 conference, Uppsala (Sweden), June, 2018. <https://doi.org/10.1145/3200000.3200001>

- [//www.researchgate.net/publication/228555728_User_Experience_Evaluation_Methods_in_Academic_and_Industrial_Contexts](http://www.researchgate.net/publication/228555728_User_Experience_Evaluation_Methods_in_Academic_and_Industrial_Contexts)
- [15] F. šuba, “Usability and Security of DRM architectures”, 2007
 - [16] D. K. Mulligan and A. Perzanowski, “The Magnificence of the Disaster: Reconstructing the Sony BMG Rootkit Incident”, *Berkeley Technology Law Journal*, vol. 22, June 2007, pp. 1158–1232, DOI [10.15779/Z38P41V](https://doi.org/10.15779/Z38P41V)
 - [17] A. Jumratjaroenvanit and Y. Teng-amnuay, “Probability of attack based on system vulnerability life cycle”, 2008 International Symposium on Electronic Commerce and Security, Guangzhou City (China), August 03-05, 2008, pp. 531–535, DOI [10.1109/ISECS.2008.212](https://doi.org/10.1109/ISECS.2008.212)
 - [18] Schneier, “Secrets and Lies: Digital Security in a Networked World”, Wiley, 2015, ISBN: 978-0-471-25311-2
 - [19] J. Bonneau and E. Shutova, “Linguistic properties of multi-word passphrases”, *Financial Cryptography and Data Security*, Berlin (Heidelberg), 2012, pp. 1–12, DOI [10.1007/978-3-642-34638-5_1](https://doi.org/10.1007/978-3-642-34638-5_1)
 - [20] B. Ives, K. R. Walsh, and H. Schneider, “The domino effect of password reuse”, *Commun. ACM*, vol. 47, April 2004, pp. 75–78, DOI [10.1145/975817.975820](https://doi.org/10.1145/975817.975820)
 - [21] M. Urueña and C. Busquiel, “Analysis of a privacy vulnerability in the openid authentication protocol”, 3rd International Conference on Multimedia Communications, Services and Security, Krakow (Poland), May, 2010
 - [22] N. Haller, C. Metz, P. Nesser, and M. Straw, “A One-Time Password System.” RFC-5246, February 1998, DOI [10.17487/RFC2289](https://doi.org/10.17487/RFC2289)
 - [23] D. M’Raihi, S. Machani, M. Pei, and J. Rydell, “Totp: Time-based one-time password algorithm”, May 2011, DOI [10.17487/RFC6238](https://doi.org/10.17487/RFC6238)
 - [24] N. Haller, “The s/key one-time password system”, February 1995, DOI [10.17487/RFC1760](https://doi.org/10.17487/RFC1760)
 - [25] G. Blonder, “Blonder scheme”, 1996, <https://www.google.com/patents/US5559961>
 - [26] S. Chiasson, E. Stobert, A. Forget, R. Biddle, and P. C. V. Oorschot, “Persuasive cued click-points: Design, implementation, and evaluation of a knowledge-based authentication mechanism”, *IEEE Transactions on Dependable and Secure Computing*, vol. 9, March 2012, pp. 222–235, DOI [10.1109/TDSC.2011.55](https://doi.org/10.1109/TDSC.2011.55)
 - [27] H. M. Sun, S. T. Chen, J. H. Yeh, and C. Y. Cheng, “A shoulder surfing resistant graphical authentication system”, *IEEE Transactions on Dependable and Secure Computing*, vol. 15, March 2018, pp. 180–193, DOI [10.1109/TDSC.2016.2539942](https://doi.org/10.1109/TDSC.2016.2539942)
 - [28] A. Danish, L. Sharma, H. Varshney, and A. M. Khan, “Alignment based graphical password authentication system”, 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, March, 2016, pp. 2950–2954
 - [29] S. Akula and V. Devisetty, “Image based registration and authentication system”, 2004, <https://www.semanticscholar.org/paper/Image-Based-Registration-and-Authentication-System-Akula-Devisetty/c3f08f67e50b880ecdb24d4f4a5c9578ea4d378c>

- [30] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, “User authentication via prnu-based physical unclonable functions”, *IEEE Transactions on Information Forensics and Security*, vol. 12, August 2017, pp. 1941–1956, DOI [10.1109/TIFS.2017.2697402](https://doi.org/10.1109/TIFS.2017.2697402)
- [31] A. Altinok and M. Turk, “Temporal integration for continuous multimodal biometrics”, In *Multimodal User Authentication*, ‘03, 2003, pp. 11–12
- [32] S. Egelman, S. Jain, R. S. Portnoff, K. Liao, S. Consolvo, and D. Wagner, “Are you ready to lock?”, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale (Arizona), November 03–07, 2014, pp. 750–761, DOI [10.1145/2660267.2660273](https://doi.org/10.1145/2660267.2660273)
- [33] L. Coventry, A. De Angeli, and G. Johnson, “Usability and biometric verification at the atm interface”, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, April 05–10, 2003, pp. 153–160, DOI [10.1145/642611.642639](https://doi.org/10.1145/642611.642639)
- [34] “Chaos computer club breaks iris recognition system if the samsung galaxy s8”, 2017, <https://www.ccc.de/en/updates/2017/iriden>
- [35] S. Eberz, K. B. Rasmussen, V. Lenders, and I. Martinovic, “Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics”, *Conference: network and Distributed System Security*, San Diego (California), February 07, 2015, DOI [10.14722/ndss.2015.23203](https://doi.org/10.14722/ndss.2015.23203)
- [36] I. Sluganovic, M. Roeschlin, K. B. Rasmussen, and I. Martinovic, “Using reflexive eye movements for fast challenge-response authentication”, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna (Austria), October 24–28, 2016, pp. 1056–1067, DOI [10.1145/2976749.2978311](https://doi.org/10.1145/2976749.2978311)
- [37] J. Villalba and E. Lleida, “Detecting replay attacks from far-field recordings on speaker verification systems”, *Biometrics and ID Management*, Heidelberg (Berlin), March 08–10, 2011, pp. 274–285, DOI [10.1007/978-3-642-19530-3_25](https://doi.org/10.1007/978-3-642-19530-3_25)
- [38] L. Zhang, S. Tan, J. Yang, and Y. Chen, “Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones”, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna (Austria), October 24–28, 2016, pp. 1080–1091, DOI [10.1145/2976749.2978296](https://doi.org/10.1145/2976749.2978296)
- [39] J. Modi, H. G. Upadhyay, and M. Thakor, “Password less authentication using keystroke dynamics: A survey”, *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, November 2014
- [40] E. Yu and S. Cho, “Keystroke dynamics identity verification, its problems and practical solutions”, *Computers & Security*, vol. 23, July 2004, pp. 428–440, DOI [10.1016/j.cose.2004.02.004](https://doi.org/10.1016/j.cose.2004.02.004)
- [41] A. Alsultan, K. Warwick, and H. Wei, “Non-conventional keystroke dynamics for user authentication”, *Pattern Recognition Letters*, vol. 89, April 2017, pp. 53–59, DOI [10.1016/j.patrec.2017.02.010](https://doi.org/10.1016/j.patrec.2017.02.010)
- [42] H. Crawford and E. Ahmadzadeh, “Authentication on the go: Assessing the effect of movement on mobile device keystroke dynamics”, *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, Santa Clara (California), July 12–14, 2017, pp. 163–173

- [43] F. Monroe, M. K. Reiter, and S. Wetzel, "Password hardening based on keystroke dynamics", *International Journal of Information Security*, vol. 1, February 2002, pp. 69–83, DOI [10.1007/s102070100006](https://doi.org/10.1007/s102070100006)
- [44] A. Roy, T. Halevi, and N. Memon, "An hmm-based behavior modeling approach for continuous mobile authentication", 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence (Italy), May 04-09, 2014, pp. 3789–3793, DOI [10.1109/ICASSP.2014.6854310](https://doi.org/10.1109/ICASSP.2014.6854310)
- [45] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication", *IEEE Transactions on Information Forensics and Security*, vol. 8, January 2013, pp. 136–148, DOI [10.1109/TIFS.2012.2225048](https://doi.org/10.1109/TIFS.2012.2225048)
- [46] H. Zhang, V. M. Patel, M. Fathy, and R. Chellappa, "Touch gesture-based active user authentication using dictionaries", 2015 IEEE Winter Conference on Applications of Computer Vision, January 05-09, 2015, pp. 207–214, DOI [10.1109/WACV.2015.35](https://doi.org/10.1109/WACV.2015.35)
- [47] L. Li, X. Zhao, and G. Xue, "Unobservable reauthentication for smart phones", 20th Network and Distributed Syst. Security Symp., San Diego (California), February 24-27, 2014
- [48] T. Feng, Z. Liu, K. A. Kwon, W. Shi, B. Carburnar, Y. Jiang, and N. Nguyen, "Continuous mobile authentication using touchscreen gestures", 2012 IEEE Conference on Technologies for Homeland Security (HST), Waltham (Massachusetts), November 13-15, 2012, pp. 451–456, DOI [10.1109/THS.2012.6459891](https://doi.org/10.1109/THS.2012.6459891)
- [49] A. Serwadda, V. V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms", 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), 29 September - 02 October, 2013, pp. 1–8
- [50] X. Zhao, T. Feng, W. Shi, and I. A. Kakadiaris, "Mobile user authentication using statistical touch dynamics images", *IEEE Transactions on Information Forensics and Security*, vol. 9, November 2014, pp. 1780–1789, DOI [10.1109/TIFS.2014.2350916](https://doi.org/10.1109/TIFS.2014.2350916)
- [51] C. Kuroda, M. Kobayashi, M. Watanabe, and M. Okamoto, "Poster: Authentication by web browsing history", 36th IEEE Symposium on Security and Privacy, San Jose (California), May 18-30, 2015
- [52] M. Abramson and D. W. Aha, "User authentication from web browsing behavior", Florida Artificial Intelligence Research Society Conference: The 26th International FLAIRS Conference, May, May 22-24, 2013
- [53] P. Zhao, C. Yan, and C. Jiang, "Authenticating web user's identity through browsing sequences modeling", 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona (Spain), December 12-15, 2016, pp. 335–342, DOI [10.1109/ICDMW.2016.0054](https://doi.org/10.1109/ICDMW.2016.0054)
- [54] F. Li, N. Clarke, M. Papadaki, and P. Dowland, "Active authentication for mobile devices utilising behaviour profiling", *International Journal of Information Security*, vol. 13, June 2014, pp. 229–244, DOI [10.1007/s10207-013-0209-6](https://doi.org/10.1007/s10207-013-0209-6)
- [55] J. Tian, Y. Cao, W. Xu, and S. Wang, "Challenge-response authentication using in-air handwriting style verification", *IEEE Transactions on Dependable and Secure Computing*, September 2017, DOI [10.1109/TDSC.2017.2752164](https://doi.org/10.1109/TDSC.2017.2752164)

- [56] A. Ceccarelli, L. Montecchi, F. Brancati, P. Lollini, A. Marguglio, and A. Bon-davalli, “Continuous and transparent user identity verification for secure in-ternet services”, *IEEE Transactions on Dependable and Secure Computing*, vol. 12, May 2015, pp. 270–283, DOI [10.1109/TDSC.2013.2297709](https://doi.org/10.1109/TDSC.2013.2297709)
- [57] “Qualcomm announces advanced fingerprint scanning and authentication tech-nology”, 2017, <https://www.qualcomm.com/news/releases/2017/06/28/qualcomm-announces-advanced-fingerprint-scanning-and-authentication>
- [58] C. Song, F. Lin, Y. Zhuang, W. Xu, C. Li, and K. Ren, “Cardiac scan: A non-contact and continuous heart-based user authentication system”, *MobiCom ’17 Proceedings of the 23rd Annual International Conference on Mobile Com-puting and Networking, Snowbird (Utah)*, October 16-20, 2017, pp. 315–328, DOI [10.1145/3117811.3117839](https://doi.org/10.1145/3117811.3117839)
- [59] A. Ross and A. K. Jain, “Multimodal biometrics: An overview”, 2004 12th European Signal Processing Conference, Vienna (Austria), September 06-10, 2004, pp. 1221–1224
- [60] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, “Active authentica-tion on mobile devices via stylometry, application usage, web browsing, and gps location”, *IEEE Systems Journal*, vol. 11, June 2017, pp. 513–521, DOI [10.1109/JSYST.2015.2472579](https://doi.org/10.1109/JSYST.2015.2472579)
- [61] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong, “Senguard: Passive user identification on smartphones using multiple sensors”, 2011 IEEE 7th Interna-tional Conference on Wireless and Mobile Computing, Networking and Com-munications (WiMob), Wuhan (China), October 10-12, 2011, pp. 141–148, DOI [10.1109/WiMOB.2011.6085412](https://doi.org/10.1109/WiMOB.2011.6085412)
- [62] C. McCool, S. Marcel, A. Hadid, M. Pietikäinen, P. Matejka, J. Cernocký, N. Poh, J. Kittler, A. Larcher, C. Lévy, D. Matrouf, J. F. Bonastre, P. Tresa-der, and T. Cootes, “Bi-modal person recognition on a mobile phone: Using mobile phone data”, 2012 IEEE International Conference on Multimedia and Expo Workshops, Melbourne (Australia), July 09-13, 2012, pp. 635–640, DOI [10.1109/ICMEW.2012.116](https://doi.org/10.1109/ICMEW.2012.116)
- [63] Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, “Hmog: New behavioral biometric features for continuous authentication of smartphone users”, *IEEE Transactions on Information Forensics and Security*, vol. 11, May 2016, pp. 877–892, DOI [10.1109/TIFS.2015.2506542](https://doi.org/10.1109/TIFS.2015.2506542)
- [64] H. Saevanee, N. Clarke, S. Furnell, and V. Biscione, “Text-based active authentication for mobile devices”, *ICT Systems Security and Privacy Protection*, Marrakech (Morocco), June 02-04, 2014, pp. 99–112, DOI [10.1007/978-3-642-55415-5_9](https://doi.org/10.1007/978-3-642-55415-5_9)
- [65] T.-Y. Chang, C.-J. Tsai, and J.-H. Lin, “A graphical-based password keystroke dynamic authentication system for touch screen handheld mobile devices”, *Journal of Systems and Software*, vol. 85, May 2012, pp. 1157–1165, DOI [10.1016/j.jss.2011.12.044](https://doi.org/10.1016/j.jss.2011.12.044)
- [66] J. Brainard, A. Juels, R. L. Rivest, M. Szydło, and M. Yung, “Fourth-factor authentication: Somebody you know”, *Proceedings of the 13th ACM Con-ference on Computer and Communications Security*, Alexandria (Virginia), October 30 - November 03, 2006, pp. 168–178, DOI [10.1145/1180405.1180427](https://doi.org/10.1145/1180405.1180427)

- [67] Y. Kuamazawa, A. Ito, and M. Okamoto, “Poster: Authentication with plural servers in the correct order”, 36th IEEE Symposium on Security and Privacy, San Jose (California), May 18-20, 2015
- [68] T. Ishizuka, H. Iwata, and M. Okamoto, “Poster: Multi-sign-on; authentication collector”, 35th IEEE Symposium on Security and Privacy, San Jose (California), May 18-21, 2014
- [69] A. Gutmann, K. Renaud, J. Maguire, P. Mayer, M. Volkamer, K. Matsuu-
ra, and J. Müller-Quade, “Zeta-zero-trust authentication: Relying on innate human ability, not technology”, 2016 IEEE European Symposium on Security and Privacy (EuroS P), Saarbrücken (Germany), March 21-24, 2016, pp. 357–371, DOI [10.1109/EuroSP.2016.35](https://doi.org/10.1109/EuroSP.2016.35)
- [70] M. J. B. Campbell, C. Mortimore, “Security assertion markup language (saml) 2.0 profile for oauth 2.0 client authentication and authorization grants”, May 2015, DOI [10.17487/RFC7522](https://doi.org/10.17487/RFC7522)
- [71] J. Rosenberg, “The extensible markup language (xml) configuration access protocol (xcap)”, May 2007, DOI [10.17487/RFC4825](https://doi.org/10.17487/RFC4825)
- [72] “OATH Reference Architecture, Release 2.0.” <https://openauthentication.org/wp-content/uploads/2015/09/ReferenceArchitectureVersion2.pdf>
- [73] M. Nystrom, A. Doherty, M. Pei, and S. Machani, “Dynamic Symmetric Key Provisioning Protocol (DSKPP).” RFC-6063, December 2010, DOI [10.17487/RFC6063](https://doi.org/10.17487/RFC6063)
- [74] M. View, D. M’Raihi, F. Hoornaert, D. Naccache, M. Bellare, and O. Rannen, “HOTP: An HMAC-Based One-Time Password Algorithm.” RFC-4226, December 2005, DOI [10.17487/RFC4226](https://doi.org/10.17487/RFC4226)
- [75] D. H. Krawczyk, M. Bellare, and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication.” RFC-2104, February 1997, DOI [10.17487/RFC2104](https://doi.org/10.17487/RFC2104)
- [76] M. View, D. Naccache, J. Rydell, S. Bajaj, and S. Machani, “OCRA: OATH challenge-response Algorithm.” RFC-6287, June 2011, DOI [10.17487/RFC6287](https://doi.org/10.17487/RFC6287)
- [77] S. Machani, M. Pei, and P. Hoyer, “Portable Symmetric Key Container (PSKC).” RFC-6030, October 2010, DOI [10.17487/RFC6030](https://doi.org/10.17487/RFC6030)
- [78] “Vault by hashicorp.” <https://www.vaultproject.io/>
- [79] “Hashicorp.” <https://www.hashicorp.com/>
- [80] M. Ito, A. Saito, and T. Nishizeki, “Secret sharing scheme realizing general access structure”, Electronics and Communications in Japan (Part III: Fundamental Electronic Science), vol. 72, no. 9, 1989, pp. 56–64, DOI [10.1002/ecjc.4430720906](https://doi.org/10.1002/ecjc.4430720906)
- [81] “Entrust IdentityGuard Adaptive Authentication”, 2015, <https://www.entrustdatacard.com/products/authentication/identityguard-adaptive-authentication>
- [82] “IBM Security Access Manager.” <https://www.ibm.com/it-it/marketplace/access-management#product-header-top>
- [83] “ISO/IEC 27002:2013.” <http://www.iso27001security.com/html/27002.html>

- [84] E. R. Lindgreen and I. Herschberg, “On the validity of the bell-la padula model”, *Computers & Security*, vol. 13, September 1994, pp. 317–333, DOI [10.1016/0167-4048\(94\)90023-X](https://doi.org/10.1016/0167-4048(94)90023-X)
- [85] N. Wang, Z. Liu, and Y. Huang, “Applying dynamic labels of the biba model in forensic evidence analysis”, 2009 International Conference on Computational Intelligence and Security, Beijing (China), December 11-14, 2009, pp. 478–482, DOI [10.1109/CIS.2009.92](https://doi.org/10.1109/CIS.2009.92)
- [86] R. Sandhu and Q. Munawer, “How to do discretionary access control using roles”, *Proceedings of the Third ACM Workshop on Role-based Access Control*, Fairfax (Virginia), October 22-23, 1998, pp. 47–54, DOI [10.1145/286884.286893](https://doi.org/10.1145/286884.286893)
- [87] S. Osborn, “Mandatory access control and role-based access control revisited”, *Proceedings of the Second ACM Workshop on Role-based Access Control*, Fairfax (Virginia), November 06-07, 1997, pp. 31–40, DOI [10.1145/266741.266751](https://doi.org/10.1145/266741.266751)
- [88] R. S. Sandhu, “Role-based access control”, 11 Portions of this chapter have been published earlier in Sandhu et al. (1996), Sandhu (1996), Sandhu and Bhamidipati (1997), Sandhu et al. (1997) and Sandhu and Feinstein (1994) (M. V. Zelkowitz, ed.), pp. 237–286, Elsevier, 1998, DOI [10.1016/S0065-2458\(08\)60206-5](https://doi.org/10.1016/S0065-2458(08)60206-5)
- [89] J. Crampton and J. Sellwood, “Path conditions and principal matching: A new approach to access control”, *Proceedings of the 19th ACM Symposium on Access Control Models and Technologies*, London, Ontario (Canada), June 25-27, 2014, pp. 187–198, DOI [10.1145/2613087.2613094](https://doi.org/10.1145/2613087.2613094)
- [90] W. Enck, M. Ongtang, and P. McDaniel, “Understanding android security”, *IEEE Security Privacy*, vol. 7, January 2009, pp. 50–57, DOI [10.1109/MSP.2009.26](https://doi.org/10.1109/MSP.2009.26)
- [91] “Android fragmentation”, February 2018, <https://developer.android.com/about/dashboards>
- [92] B. Shebaro, O. Oluwatimi, and E. Bertino, “Context-based access control systems for mobile devices”, *IEEE Transactions on Dependable and Secure Computing*, vol. 12, March 2015, pp. 150–163, DOI [10.1109/TDSC.2014.2320731](https://doi.org/10.1109/TDSC.2014.2320731)
- [93] Z. Fang, W. Han, and Y. Li, “Permission based android security: Issues and countermeasures”, *Computers & Security*, vol. 43, 2014, pp. 205–218, DOI [10.1016/j.cose.2014.02.007](https://doi.org/10.1016/j.cose.2014.02.007)
- [94] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, “Attribute-based access control”, *Computer*, vol. 48, February 2015, pp. 85–88, DOI [10.1109/MC.2015.33](https://doi.org/10.1109/MC.2015.33)
- [95] D. R. Kuhn, E. J. Coyne, and T. R. Weil, “Adding attributes to role-based access control”, *Computer*, vol. 43, June 2010, pp. 79–81, DOI [10.1109/MC.2010.155](https://doi.org/10.1109/MC.2010.155)
- [96] S. Ruj, M. Stojmenovic, and A. Nayak, “Decentralized access control with anonymous authentication of data stored in clouds”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, February 2014, pp. 384–394, DOI [10.1109/TPDS.2013.38](https://doi.org/10.1109/TPDS.2013.38)
- [97] “eXtensible Access Control Markup Language(XACML)”, 2013, <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>

- [98] A. Mourad and H. Jebbaoui, “SBA-XACML: Set-based approach providing efficient policy decision process for accessing Web services”, *Expert Systems with Applications*, vol. 42, January 2015, pp. 165–178, DOI [10.1016/j.eswa.2014.07.031](https://doi.org/10.1016/j.eswa.2014.07.031)
- [99] F. Marfia, M. Arrigoni Neri, F. Pellegrini, and M. Colombetti, “Using owl reasoning for evaluating xacml policies”, *International Conference on E-Business and Telecommunications (ICETE)*, Colmar (France), July 20-22, 2016, pp. 343–363, DOI [10.1007/978-3-319-30222-5_16](https://doi.org/10.1007/978-3-319-30222-5_16)
- [100] K. Seol, Y. G. Kim, E. Lee, Y. D. Seo, and D. K. Baik, “Privacy-preserving attribute-based access control model for xml-based electronic health record system”, *IEEE Access*, vol. 6, February 2018, pp. 9114–9128, DOI [10.1109/ACCESS.2018.2800288](https://doi.org/10.1109/ACCESS.2018.2800288)
- [101] A. Anderson, “Xacml profile for role based access control (rbac)”, 2004, <https://docs.oasis-open.org/xacml/cd-xacml-rbac-profile-01.pdf>
- [102] M. A. Shibli, R. Masood, and U. Habiba, “Realization of fgac model using xacml policy specification”, *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Takamatsu (Japan), June 01-03, 2015, pp. 1–6, DOI [10.1109/SNPD.2015.7176199](https://doi.org/10.1109/SNPD.2015.7176199)
- [103] F. Turkmen, J. d. Hartog, and N. Zannone, “Poster: Analyzing access control policies with smt”, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale (Arizona), November 03-07, 2014, pp. 1508–1510, DOI [10.1145/2660267.2662366](https://doi.org/10.1145/2660267.2662366)
- [104] H. X. Son, T. K. Dang, and F. Massacci, “Rew-smt: A new approach for rewriting xacml request with dynamic big data security policies”, *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage (SpaCCS)*, Guangzhou (China), December 12-15, 2017, pp. 501–515, DOI [10.1007/978-3-319-72389-1_40](https://doi.org/10.1007/978-3-319-72389-1_40)
- [105] S. Raj, “Neo4j high performance”, Packt Publishing, 2015, ISBN: 9781783555154
- [106] F. P. Diez, A. C. Vasu, D. S. Touceda, and J. M. S. Cámara, “Modeling xacml security policies using graph databases”, *IT Professional*, vol. 19, November 2017, pp. 52–57, DOI [10.1109/MITP.2017.4241465](https://doi.org/10.1109/MITP.2017.4241465)
- [107] J. Park and R. Sandhu, “The uconabc usage control model”, *ACM Trans. Inf. Syst. Secur.*, vol. 7, February 2004, pp. 128–174, DOI [10.1145/984334.984339](https://doi.org/10.1145/984334.984339)
- [108] X. Zhang, F. Parisi-Presicce, R. Sandhu, and J. Park, “Formal model and policy specification of usage control”, *ACM Trans. Inf. Syst. Secur.*, vol. 8, November 2005, pp. 351–387, DOI [10.1145/1108906.1108908](https://doi.org/10.1145/1108906.1108908)
- [109] M. Colombo, A. Lazouski, F. Martinelli, and P. Mori, “A proposal on enhancing xacml with continuous usage control features”, *Grids, P2P and Services Computing* (F. Desprez, V. Getov, T. Priol, and R. Yahyapour, eds.), pp. 133–146, Springer, 2010, DOI [10.1007/978-1-4419-6794-7_11](https://doi.org/10.1007/978-1-4419-6794-7_11)
- [110] F. Martinelli, I. Matteucci, P. Mori, and A. Saracino, “Enforcement of u-xacml history-based usage control policy”, *Security and Trust Management* (G. Barthe, E. Markatos, and P. Samarati, eds.), pp. 64–81, Springer International Publishing, 2016, DOI [10.1007/978-3-319-46598-2_5](https://doi.org/10.1007/978-3-319-46598-2_5)

- [111] K. Z. Yin and H. H. Wang, “Mcaacm: A cloud storage access control model for multi-clouds environment based on xacml”, *Mechatronics Engineering and Modern Information Technologies in Industrial Engineering* (F. Shao, F. Long, J. Liang, H. Chen, and M. Yuan, eds.), pp. 2451–2454, Trans Tech Publications, 2015, DOI [10.4028/www.scientific.net/AMM.713-715.2451](https://doi.org/10.4028/www.scientific.net/AMM.713-715.2451)
- [112] “Core and hierarchical role based access control (RBAC) profile of XACML v2.0”, 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf
- [113] “XACML v3.0 Core and Hierarchical Role Based Access Control (RBAC) ”, 2014, <http://docs.oasis-open.org/xacml/3.0/rbac/v1.0/cs02/xacml-3.0-rbac-v1.0-cs02.html>
- [114] K. Alissa, J. Reid, E. Dawson, and F. Salim, “Bp-xacml an authorisation policy language for business processes”, *Information Security and Privacy*, Brisbane (Australia), June 29 - July 1, 2015, pp. 307–325, DOI [10.1007/978-3-319-19962-7_18](https://doi.org/10.1007/978-3-319-19962-7_18)
- [115] S. P. Kaluvuri, A. I. Egner, J. den Hartog, and N. Zannone, “Safax - an extensible authorization service for cloud environments”, *Frontiers in ICT*, vol. 2, May 2015, DOI [10.3389/fict.2015.00009](https://doi.org/10.3389/fict.2015.00009)
- [116] E. Hammer-Lahav, “The OAuth 1.0 Protocol.” RFC-5849, April 2010, DOI [10.17487/RFC5849](https://doi.org/10.17487/RFC5849)
- [117] K. Kiani, “Four Attacks on OAuth - How to Secure Your OAuth Implementation”, 2013, <https://www.sans.org/reading-room/whitepapers/application/attacks-oauth-secure-oauth-implementation-33644>
- [118] D. Hardt, “The OAuth 2.0 Authorization Framework.” RFC-6749, October 2012, DOI [10.17487/RFC6749](https://doi.org/10.17487/RFC6749)
- [119] “Vault - policies.” <https://www.vaultproject.io/docs/concepts/policies.html>
- [120] “Hcl.” <https://github.com/hashicorp/hcl>
- [121] W. Melicher, D. Kurilova, S. M. Segreti, P. Kalvani, R. Shay, B. Ur, L. Bauer, N. Christin, L. F. Cranor, and M. L. Mazurek, “Usability and security of text passwords on mobile devices”, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, May 07-12, 2016, pp. 527–539, DOI [10.1145/2858036.2858384](https://doi.org/10.1145/2858036.2858384)
- [122] R. Surana, “Evolution of OTP interface”, 2017, <https://medium.com/user-experience-design-review/a-change-in-otp-interface-design-over-time-ada3ee1fdf67>
- [123] T. Curran and J. Doyle, “Picture superiority doubly dissociates the erp correlates of recollection and familiarity”, *J. Cognitive Neuroscience*, vol. 23, May 2011, pp. 1247–1262, DOI [10.1162/jocn.2010.21464](https://doi.org/10.1162/jocn.2010.21464)
- [124] M. Anwar and A. Imran, “A Comparative Study of Graphical and Alphanumeric Passwords for Mobile Device Authentication”, 2015, https://www.researchgate.net/publication/277476774_A_Comparative_Study_of_Graphical_and_Alphanumeric_Passwords_for_Mobile_Device_Authentication
- [125] C. Braz and J.-M. Robert, “Security and usability: The case of the user authentication methods”, *Proceedings of the 18th Conference on L’Interaction*

- Homme-Machine, Montreal (Canada), April 18-21, 2006, pp. 199–203, DOI [10.1145/1132736.1132768](https://doi.org/10.1145/1132736.1132768)
- [126] H. Khan, U. Hengartner, and D. Vogel, “Usability and security perceptions of implicit authentication: Convenient, secure, sometimes annoying”, Eleventh Symposium On Usable Privacy and Security (SOUPS 2015), Ottawa (Canada), July 22-24, 2015, pp. 225–239
- [127] N. Clarke, S. Karatzouni, and S. Furnell, “Flexible and transparent user authentication for mobile devices”, IFIP International Information Security Conference: Emerging Challenges for Security, Privacy and Trust, Pafos (Cyprus), May 18-20, 2009, pp. 1–12, DOI [10.1007/978-3-642-01244-0_1](https://doi.org/10.1007/978-3-642-01244-0_1)
- [128] H. Crawford and K. Renaud, “Understanding user perceptions of transparent authentication on a mobile device”, Journal of Trust Management, vol. 1, June 2014, DOI [10.1186/2196-064X-1-7](https://doi.org/10.1186/2196-064X-1-7)
- [129] H. H. Cao, B. Han, D. Hirshleifer, and H. H. Zhang, “Fear of the unknown: Familiarity and economic decisions”, Review of Finance, vol. 15, January 2011, pp. 173–206, DOI [10.1093/rof/rfp023](https://doi.org/10.1093/rof/rfp023)
- [130] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbell, “Continuous user authentication on mobile devices: Recent progress and remaining challenges”, IEEE Signal Processing Magazine, vol. 33, July 2016, pp. 49–61, DOI [10.1109/MSP.2016.2555335](https://doi.org/10.1109/MSP.2016.2555335)
- [131] M. L. Johnson, S. M. Bellovin, R. W. Reeder, and S. Schechter, “Laissez-faire file sharing: Access control designed for individuals at the endpoints”, Proceedings New Security Paradigms Workshop (NSPW), Oxford (United Kingdom), September 08-11, 2009, DOI [10.1145/1719030.1719032](https://doi.org/10.1145/1719030.1719032)
- [132] E. Lieberman and R. Miller, “Facemail: showing faces of recipients to prevent misdirected email”, SOUPS ’07: Proceedings of the 3rd Symposium on Usable Privacy and Security, Pittsburgh (Pennsylvania), July 18-20, 2007, DOI [10.1145/1280680.1280696](https://doi.org/10.1145/1280680.1280696)
- [133] M. L. Mazurek, J. P. Arsenault, J. Bresee, N. Gupta, I. Ion, C. Johns, D. Lee, Y. Liang, J. Olsen, B. Salmon, R. Shay, K. Vaniea, L. Bauer, L. F. Cranor, G. R. Ganger, and M. K. Reiter, “Access control for home data sharing: Attitudes, needs and practices”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta (Georgia), April 10-15, 2010, pp. 645–654, DOI [10.1145/1753326.1753421](https://doi.org/10.1145/1753326.1753421)
- [134] R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, K. Bacon, K. How, and H. Strong, “Expandable grids for visualizing and authoring computer security policies”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Florence (Italy), April 05-10, 2008, pp. 1473–1482, DOI [10.1145/1357054.1357285](https://doi.org/10.1145/1357054.1357285)
- [135] M. E. Zurko, R. Simon, and T. Sanfilippo, “A user-centered, modular authorization service built on an rbac foundation”, Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344), Oakland (California), May 14, 1999, pp. 57–71, DOI [10.1109/SECPRI.1999.766718](https://doi.org/10.1109/SECPRI.1999.766718)
- [136] D. K. Smetters and N. Good, “How users use access control”, Proceedings of the 5th Symposium on Usable Privacy and Security, Mountain View (California), July 15-17, 2009, pp. 1–12, DOI [10.1145/1572532.1572552](https://doi.org/10.1145/1572532.1572552)

-
- [137] B. Salmon, F. Hady, and J. Melican, “Learning to Share: A Study of Sharing Among Home Storage Devices”, 2007, <http://repository.cmu.edu/pdl/28/>
 - [138] R. A. Maxion and R. W. Reeder, “Improving user-interface dependability through mitigation of human error”, *International Journal of Human-Computer Studies*, vol. 63, July 2005, pp. 25–50, DOI [10.1016/j.ijhcs.2005.04.009](https://doi.org/10.1016/j.ijhcs.2005.04.009)
 - [139] R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, and K. Vaniea, “More than skin deep: Measuring effects of the underlying model on access-control system usability”, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, BC (Canada), May 07-12, 2011, pp. 2065–2074, DOI [10.1145/1978942.1979243](https://doi.org/10.1145/1978942.1979243)
 - [140] S. Garfinkel and H. R. Lipford, “Usable security”, - History, Themes and Challenges, pp. 17–38, Morgan & Claypool Publishers, 2014, DOI [10.2200/S00594ED1V01Y201408SPT011](https://doi.org/10.2200/S00594ED1V01Y201408SPT011)
 - [141] A. K. Karlson, A. B. Brush, and S. Schechter, “Can i borrow your phone?: Understanding concerns when sharing mobile phones”, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Boston (Massachusetts), April 04-09, 2009, pp. 1647–1650, DOI [10.1145/1518701.1518953](https://doi.org/10.1145/1518701.1518953)
 - [142] S. Volda, W. K. Edwards, M. W. Newman, R. E. Grinter, and N. Ducheneaut, “Share and share alike: Exploring the user interface affordances of file sharing”, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, April 22-27, 2006, pp. 221–230, DOI [10.1145/1124772.1124806](https://doi.org/10.1145/1124772.1124806)
 - [143] Y. Wang, G. Norcie, S. Komanduri, A. Acquisti, P. G. Leon, and L. F. Cranor, ““i regretted the minute i pressed share”: A qualitative study of regrets on facebook”, *Proceedings of the Seventh Symposium on Usable Privacy and Security*, Pittsburgh (Pennsylvania), July 20-22, 2011, DOI [10.1145/2078827.2078841](https://doi.org/10.1145/2078827.2078841)
 - [144] J. King, A. Lampinen, and A. Smolen, “Privacy: Is there an app for that?”, *Proceedings of the Seventh Symposium on Usable Privacy and Security*, July 20-22, 2011, DOI [10.1145/2078827.2078843](https://doi.org/10.1145/2078827.2078843)
 - [145] K. Vaniea, L. Bauer, L. F. Cranor, and M. K. Reiter, “Out of sight, out of mind: Effects of displaying access-control information near the item it controls”, *2012 Tenth Annual International Conference on Privacy, Security and Trust*, Paris (France), 2012, pp. 128–136, DOI [10.1109/PST.2012.6297929](https://doi.org/10.1109/PST.2012.6297929)
 - [146] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, and D. Wagner, “Android permissions remystified: A field study on contextual integrity”, *24th USENIX Security Symposium*, Washington, August 12-14, 2010
 - [147] P. Wijesekera, A. Baokar, L. Tsai, J. Reardon, S. Egelman, D. Wagner, and K. Beznosov, “The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences”, *2017 IEEE Symposium on Security and Privacy (SP)*, San Jose (California), May 22-26, 2017, pp. 1077–1093, DOI [10.1109/SP.2017.51](https://doi.org/10.1109/SP.2017.51)
 - [148] L. Tsai, P. Wijesekera, J. Reardon, I. Reyes, S. Egelman, D. Wagner, N. Good, and J.-W. Chen, “Turtle guard: Helping android users apply contextual privacy preferences”, *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, Santa Clara (California), July 12-14, 2017, pp. 145–162

- [149] A. Cockburn, “Writing effective use cases”, Addison-Wesley Professional, 2000, ISBN: 978-0201702255
- [150] Fulvio Corno, Marco Torchiano, “Sistemi Informativi Aziendali.” <https://elite.polito.it/files/courses/02CIX/libro/cap-5-standalone.pdf>
- [151] B. Welch, “Exploiting the weaknesses of ss7”, Network Security, vol. 2017, January 2017, pp. 17–19, DOI [10.1016/S1353-4858\(17\)30008-9](https://doi.org/10.1016/S1353-4858(17)30008-9)
- [152] Unicredit, “Password usa e getta.” <https://www.unicredit.it/it/privati/internet-e-mobile/tutti-i-servizi-internet-e-mobile/domande-utili-e-assistenza/Passwordusaegetta.html>
- [153] FinecoBank, “Regolamento del servizio SMS PIN.” <https://finecobank.com/public/popup/informativa-sms-pin>
- [154] Android, “Automatic SMS Verification with the SMS Retriever AP.” <https://developers.google.com/identity/sms-retriever/overview>
- [155] T. Oliveira, R. Mendes, and A. Bessani, “Sharing files using cloud storage services”, Euro-Par 2014: Parallel Processing Workshops, Porto (Portugal), August 25-26, 2014, pp. 13–25, DOI [10.1007/978-3-319-14313-2_2](https://doi.org/10.1007/978-3-319-14313-2_2)
- [156] MEGA, “Main.” <https://mega.nz/>
- [157] E. Frøkjær, M. Hertzum, and K. Hornbæk, “Measuring usability: Are effectiveness, efficiency, and satisfaction really correlated?”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, The Hague (The Netherlands), April 01-06, 2000, pp. 345–352, DOI [10.1145/332040.332455](https://doi.org/10.1145/332040.332455)
- [158] J. Fierrez-Aguilar, D. Garcia-Romero, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, “Bayesian adaptation for user-dependent multimodal biometric authentication”, Pattern Recognition, vol. 38, August 2005, pp. 1317–1319, DOI [10.1016/j.patcog.2005.01.013](https://doi.org/10.1016/j.patcog.2005.01.013)
- [159] Jacob Nielsen, “First Rule of Usability? Don’t Listen to Users.” <https://www.nngroup.com/articles/first-rule-of-usability-dont-listen-to-users/>
- [160] W. J. Doll and G. Torkzadeh, “The measurement of end-user computing satisfaction”, Management Information Systems Research Center, University of Minnesota, vol. 12, June 1988, pp. 259–274, DOI [10.2307/248851](https://doi.org/10.2307/248851)
- [161] E. Geisen and J. R. Bergstrom, “Usability testing for survey research”, Morgan Kaufmann Publishers, 2017, ISBN: 9780128036563
- [162] J. E. Ware, “Effects of acquiescent response set on patient satisfaction ratings”, Medical Care, vol. 16, April 1978, pp. 327–336
- [163] Prajwal Kondawar, “Mutating Graphical Password System.” <https://github.com/prajwal051290/Mutating-Graphical-Password-System>
- [164] TypingDNA, “TypingDnaRecorder.” <https://www.typingdna.com/>
- [165] TypingDNA, “Repository github TypingDnaRecorder in Java.” <https://github.com/TypingDNA/TypingDnaRecorder-Java>
- [166] TypingDNA, “Match typing patterns.” https://api.typingdna.com/index.html#api-API_Services-matchTypingPattern
- [167] S. Frankel, R. Glenn, and S. Kelly, “The AES-CBC Cipher Algorithm and Its Use with IPsec.” RFC-3602, September 2003, DOI [10.17487/RFC3602](https://doi.org/10.17487/RFC3602)
- [168] K. Moriarty, B. Kaliski, and A. Rusch, “Pkcs #5: Password-based cryptography specification”, January 2017, DOI [10.17487/RFC8018](https://doi.org/10.17487/RFC8018)
- [169] “jsmooth.” <http://jsmooth.sourceforge.net/>