POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

Design of the membership concept (off-chain) between IoT nodes

Relatori e Tutori
Danilo Bazzanella
Davide Margaria
Andrea Vesco

Candidato
Alessandro Pino

Anno Accademico 2022-2023

Summary

The title of this thesis is explanatory of its content. The purpose was to find a construction or method or procedure to design a concept of membership. The concept of (dynamic) membership is central in this digression, and can be expressed in different ways. The proposed solutions are built for Internet of Things (IoT) nodes, with all the constraints and the consequences that this entails.
This thesis was carried out with the collaboration of a supervisor from the Politecnico di Torino and two tutors from the LINKS foundation (`https://linksfoundation.com/`), who contributed ideas, advices, and supervision of the work done.

The development of Internet of Things sector promise to revolutionize our everyday life. The availability of high-speed connection, the affordability of low-power, low-cost sensor technologies, the growing popularity of cloud computing, and the growing usage of data processing and analytics are the main drivers of the Internet of Things market. Moreover, the expansion of smart city projects around the globe, a rise in connected devices and the emergence of 5G technology will facilitate IoT adoption internationally.

The vast diffusion of connected devices in the IoT has created enormous demand for robust security. It must be a top priority to address these issues and guarantee security in IoT goods and services. The promotion of trust of these technologies is of primary importance.

In this context were born protocols for Self-Sovereign Identity (SSI). The core concept of SSI is to move the control of digital identity from third-party identity providers directly to individuals. This is achieved through Verifiable Credentials (VCs) supporting anonymity and selective disclosure.

One platform that can be used to implement these protocols is IOTA Tangle, on which was designed a layer two protocol called L2Sec.

The problem which this work is willing to explore is how a member of a Self Sovereign Identity scheme is supposed to verify that another member, with who is in contact, is a node authorized to write its own Decentralized Identifier (DID) (since in an environment without any central authority anyone can create his own identity) on the Root of Trust (RoT) which can be a distribute ledger technology (DLT) or directly a blockchain.

The aforementioned problem is modelled as a proof of membership in a dynamic group. Three different methods are proposed to begin the research that has the scope to solve the problem based on Merkle trees, BBS Group Signature (BBS from the authors Boneh, Boyen and Shacham) and Dynamic Accumulators.

For this thesis, the author had to go through a process summarized here in a list:

- What is the idea under the Self Sovereign Identity and its current state, also linked with the vision of Trust over IP (ToIP) and the evolution of human trust in a digital domain.
- What is a Verifiable Credential and a DID, tools in a SSI environment.
- The proposed layer two of the DLT IOTA, L2SEC, that is a solution for enabling secure data exchange for IoT constrained devices.
- Understand the problem of using a Pre-Shared Key (PSK) for authentication and encryption.
- Explored the IoT concept and its applications connected with the importance of security.
- Formalization of the problem in a proof of membership problem.
- Detailed analysis of the state of the art of different cryptographic methods.
- Studied a construction based on Merkle Tree proposed by the tutors of the thesis' author. Formalized and hypothesized a construction. Studied pro and cons.
- Studied one of the state of the art of group signature schemes: the BBS group signature. Thought, with the purpose to use that as a solution of the problem. Edit of the main construction searching for improvements or variants. Studied pro and cons.
- Studied the accumulators in the improvement of dynamic accumulators. Then found some variant, interesting for the solution of the problem. Formalized and hypothesized a construction. Studied pro and cons.

The work for this thesis resulted in some innovative contributions that can be summarized as follows:

- The formalization of the problem in a dynamic proof of membership problem in chapter 4.
- The application of abstract concepts (or methods used for a different purpose) to the proof of membership problem. The entire second part of the thesis is an examination of the use of these constructions.
- The critical review of this methods in the context of this work (sections 5.6, 6.5, 7.6).
- Variants of already existing methods. Specifically, a variation of the Merkle Tree verification (sections 5.2, 5.3, 5.4, 5.5), an implementation of a Join algorithm in the group signature BBS (section 6.3), a modification of an application of membership testing regarding the dynamic accumulator without central authority (section 7.5).
- Comparative analysis of these variants related to the problem. Mainly in chapter 8, and highlighting the table 8.1.

Acknowledgements

Un primo ringraziamento al mio relatore e ai miei tutori per avermi aiutato a svolgere questa tesi.

Un grande ringraziamento va alla mia famiglia, la mia ragazza e i miei amici per avermi accompagnato durante questo lungo e difficile percorso.

Contents

List of Tables

List of Figures

We can only see a short distance ahead,
but we can see plenty there
that needs to be done.

[Alan Turing]

Part I

Contextualization of the problem

Chapter 1

IoT introduction and security

This chapter is divided into four sections that aim to introduce IoT technology, its areas of application, the security issues involved, and the benefits that blockchain technology can give used in an IoT context.
This chapter is helpful in understanding the environment in which all the work of this thesis is embedded.

1.1 Internet of Things

The term "Internet of Things" describes a growing network of actual physical items, including devices, cars, and buildings, that are provided with sensors, software, and other technologies to connect to other devices and systems through the internet and exchange data with them. By allowing unprecedented levels of automation, efficiency, and intelligence, the IoT has the potential to revolutionize a variety of industries, from healthcare and manufacturing to transportation and energy.

The Internet of Things is facilitated by a number of technologies, including big data analytics, cloud computing, and low-power wireless networks. Large volumes of data may be collected and transmitted by devices and systems thanks to these technologies, which can subsequently be analyzed to draw conclusions and start automated processes. A smart thermostat, for instance, can utilize information about a user's heating and cooling preferences to automatically alter the temperature in a house or workplace.

According to Li et al. [2014], IoT was initially conceptualized by Kevin Ashton in 1999. He defined IoT as a network of uniquely identified, interconnected items using radio frequency identification (RFID) technology. IoT was then generally intended as “dynamic global network infrastructure with self-configuring capabilities based on standards and interoperable communication protocols; physical and virtual ‘things’ in an IoT have identities and attributes and are capable of using intelligent interfaces and being integrated as an information network” (IERC [2013]).

In essence, the Internet of Things may be thought of as a superset of interconnecting objects that can be individually identified using current near field communication (NFC) technologies (ETSI [2013]). The terms "Internet" and "Things" refer to a global network

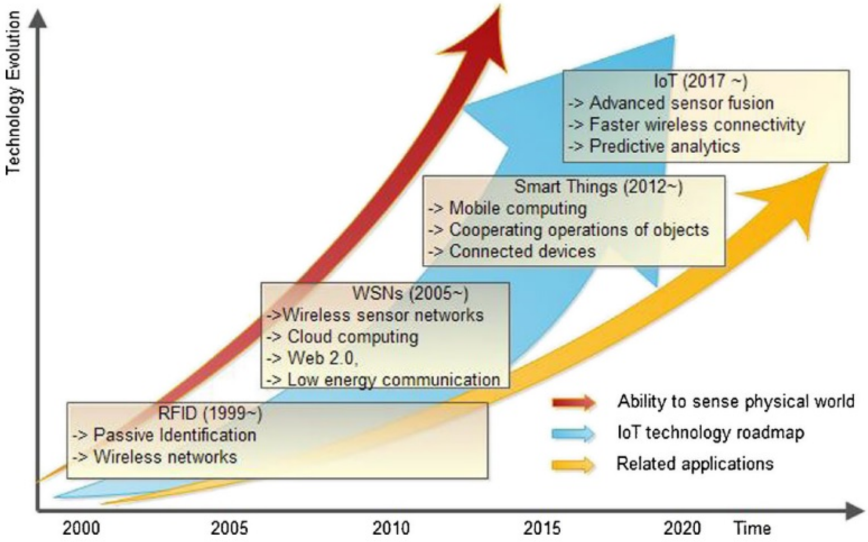


Figure 1.1. Evolution of IoT (Li et al. [2014])

with connections based on technologies for sensing, communicating, networking, and processing information, which may be the newest iteration of information and communications technology (ICT).

The initial notion of the Internet of Things is now expanding to include ambient intelligence and autonomous control as a result of the growing wirelessly sensing technologies, which have considerably increased the sensory capabilities of devices.

But why is IoT important?
According to Gillis [2022], people who use the internet of things live and work more efficiently and have total control over their life. IoT is crucial to business as well as providing smart home automation devices. With the help of IoT, organizations can see in real time how their systems function, gaining insights into anything from equipment performance to supply chain and logistics activities.

Businesses may automate procedures and save money on labor thanks to IoT. Additionally, it reduces waste, enhances service delivery, lowers the cost of manufacturing and delivering items, and provides transparency into consumer interactions.

As a result, IoT is among the most significant technologies of modern life, and it will gain momentum as more companies recognize how interconnected devices can help them stay competitive.

As stated by a MarketsandMarkets [2022] report, it is predicted that at a compound annual growth rate of 16.7% from 2021 to 2026, the worldwide IoT market would increase from $300.3 billion in 2021 to $650.5 billion by 2026. The availability of high-speed connection, the affordability of low-power, low-cost sensor technologies, the growing popularity of cloud computing, and the growing usage of data processing and analytics are

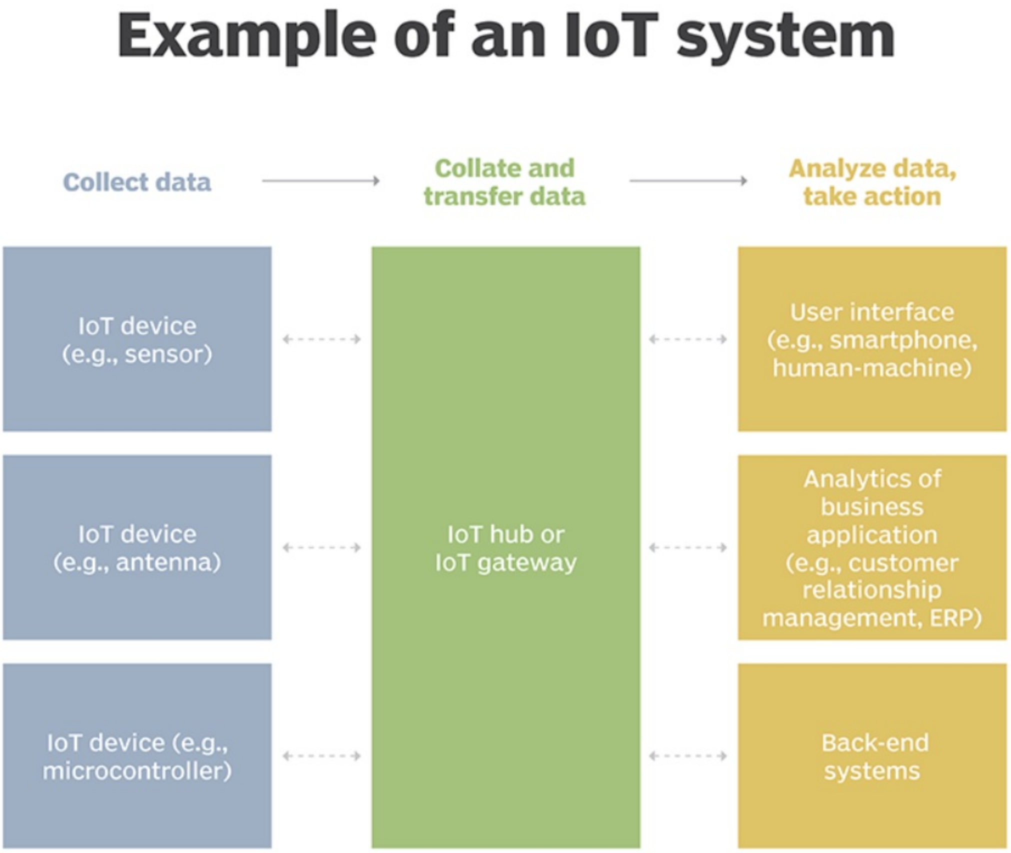

Example of an IoT system
Collect data
Collate and transfer data
Analyze data, take action
IoT device (e.g., sensor)
IoT device (e.g., antenna)
IoT device (e.g., microcontroller)
IoT hub or IoT gateway
User interface (e.g., smartphone, human-machine)
Analytics of business application (e.g., customer relationship management, ERP)
Back-end systems

Figure 1.2. Example of an IoT system (Gillis [2022])

the main drivers of the Internet of Things market. Moreover, the expansion of smart city projects around the globe, a rise in connected devices that will fuel IoT development, and the emergence of 5G technology that will facilitate IoT adoption internationally will present attractive prospects for IoT producers.

All things considered, the IoT has the potential to significantly advance several fields and applications. But it also brings up significant difficulties and issues, such as security, privacy, and the ethical consequences of growing automation and data collecting.

1.2 Application areas of IoT

Almost all IoT applications that have been implemented or are being deployed place a high priority on security. IoT applications are expanding quickly and are now present in the majority of the established sectors. Several of these IoT applications require more strict security protection from the technologies they employ, even while operators provide these applications over conventional networking technologies. This section discusses a variety of IoT applications that are security-sensitive following the structure of Hassija et al. [2019].

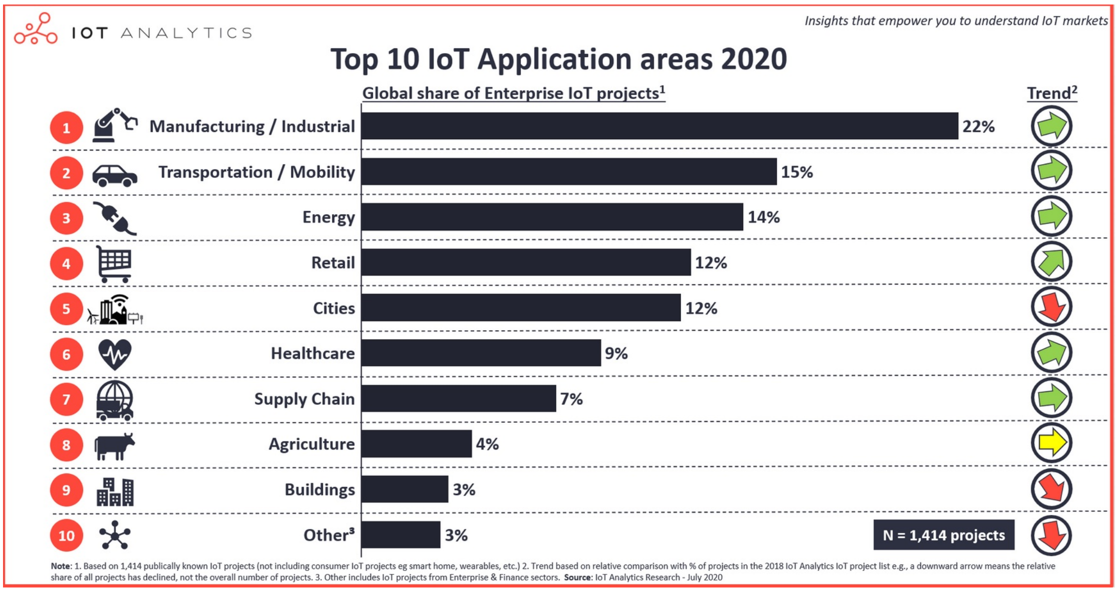
1. **Smart Cities:** To improve the general quality of life for the population, Smart Cities make considerable use of newly developed computing and communication technologies (Gharaibeh et al. [2017]). It includes smart homes, smart traffic management, smart disaster management, smart utilities, etc.
   Governments throughout the world are promoting the creation of smarter cities through a variety of incentives (Eckhoff and Wagner [2018]).
   Although using smart applications is meant to enhance residents' general quality of life, it also poses a danger to their privacy. Citizens' credit card information and purchasing habits are frequently at danger while using smart card services. The location history of users may be leaked by smart mobility applications. There are applications that parents may use to monitor their children. The child's safety, though, may be in danger if these applications were compromised.

2. **Smart Environment:** The term "smart environment" refers to a variety of IoT applications, such as the detection of forest fires, monitoring snow levels in high-altitude areas, preventing landslides, early earthquake detection, pollution monitoring, etc.
   All of these IoT applications are directly tied to how people and animals live in those regions. The information from these IoT applications will be used by the government organizations working in these domains as well.
   Serious repercussions may result from security lapses and vulnerabilities in any sector connected to such IoT applications. False findings, whether false positives or false negatives, might have fatal consequences. For instance, if the application starts incorrectly identifying earthquakes, the government and companies may suffer financial damages. However if the program is unable to forecast the earthquake, it will results in the loss of both property and lives. Applications for the smart environment must be extremely exact in order to prevent security breaches and data manipulation.

3. **Smart Metering and Smart Grids:** Applications for varied measures, monitoring, and administration are included in smart metering. Smart metering is most frequently used in smart grids, which measure and track power use. Another issue that might be solved with smart meters is electricity theft (Xia et al. [2019]). Monitoring the amounts of water, oil, and gas in storage tanks and cisterns is an use for smart meters as well. By constantly adjusting the angle of solar panels to capture the most solar energy possible, smart meters are also employed to monitor and improve the performance of solar energy facilities. There are IoT applications that measure the weight of items or the water pressure in water transportation networks using smart meters.
   Contrary to traditional meters, which can only be interfered with through physical attacks, smart metering systems are susceptible to both physical and cyber-attacks. Furthermore, smart meters, also known as advanced metering infrastructure (AMI),

are made to do more than just monitor energy use. All of the home's electric appliances are connected to smart meters as part of a smart home area network (HAN), and the data gathered from these appliances may be utilized to regulate loads and costs. Consumers or adversaries might intentionally interfere with these communication networks and change the information collected, costing service providers or customers money (Namboodiri et al. [2014]).

4. **Security and Emergencies:** Many IoT applications are being used in the field of security and emergency, which is another crucial sector. It includes applications such as allowing only authorized people in restricted areas etc.
   Another use in this field is finding hazardous gas leaks in industrial areas in the vicinity of chemical factories. Nuclear power plants and cellular base stations may also have their radiation levels checked, and when the radiation level is high, alarms can be transmitted. Different structures house sensitive commodities or have sensitive data systems. Security programs can be used to safeguard sensitive data and items.
   IoT applications that can identify different liquids can also be used to stop corrosion and malfunctions in certain delicate structures. Security breaches may potentially have many severe repercussions in these applications. Criminals could attempt to access restricted regions, for instance, by abusing these programs' weaknesses. False radiation level warnings can also have detrimental short- and long-term effects. For example, prolonged exposure to high doses of radiation may cause serious, life-threatening disorders in babies.



Note: 1. Based on 1,414 publically known IoT projects (not including consumer IoT projects eg smart home, wearables, etc.) 2. Trend based on relative comparison with % of projects in the 2018 IoT Analytics IoT project list e.g., a downward arrow means the relative share of all projects has declined, not the overall number of projects. 3. Other includes IoT projects from Enterprise & Finance sectors. **Source**: IoT Analytics Research - July 2020

Figure 1.3. Top 10 IoT application areas (Lueth [2020])

5. **Smart Retail:** Applications for the Internet of Things are widely employed in the

retail industry. Devices of all kinds have been created to keep track of how the commodities are stored as they go through the supply chain.
IoT is also being utilized to manage product tracking in warehouses so that replenishment may be carried out as efficiently as possible. Various intelligent shopping applications are also being developed to assist the customers based on their preferences, habits, allergies to particular substances, etc. Additionally, have been created methods for employing augmented reality technology to provide offline merchants the same online purchasing experience.
Security vulnerabilities have arisen when various retail companies adopted and utilized some IoT applications. Some of these companies include Apple, Home Depot, JP Morgan Chase and Sony (Dlamini and Johnston [2016]). In order to maximize sales, adversaries can try to breach IoT devices related to the storage conditions of the items and provide users inaccurate information about the products. Customers and merchants might suffer financial losses if security elements are not incorporated in smart retail. Attackers could acquire consumer debit and credit card information, phone numbers, email addresses, and other personal information.

6. **Smart Agriculture and Animal Farming:** Monitoring soil moisture, regulating microclimates, irrigating selectively in dry areas, and regulating humidity and temperature are all aspects of smart agriculture. By utilizing such cutting-edge features, farmers may avoid financial losses while reaching excellent yields. Fungus and other microbial pollutants can be avoided by controlling the temperature and humidity levels throughout different grain and vegetable production processes. Increasing the quantity and quality of vegetables and crops can also benefit from climate control.
Similar to crop monitoring, there are IoT applications that use sensors attached to farm animals to track their movements and general health. If these apps are hacked, animals may be stolen from farms and crops may be harmed by competitors.

7. **Home Automation:** One of the most frequently used and implemented IoT applications is home automation. This includes programs used to remotely regulate electrical devices in order to conserve energy, systems installed on windows and doors in order to detect intruders, etc. Energy and water supply use is being tracked by monitoring devices, and customers are being encouraged to conserve money and resources.
Jose and Malekian [2017] have proposed the use of logic based security algorithms to enhance security level in homes. By contrasting user activity at key locations in the home with expected user behavior at these locations, intrusions are identified. Attackers may, however, be able to access IoT devices in the house without authorization and attempt to damage the users. For instance, with the installation of numerous home automation systems, the number of house thefts has dramatically increased (Jose and Malekian [2017]).
Moreover, there have been a number of instances in the past where adversaries attempted to determine the type and volume of Internet data going to and coming from the smart house in order to assess the occupants' presence and activity.

1.3 The importance of security

The Internet of Things, which has recently experienced rapid development and can provide a variety of services, is now one of the fastest-growing technology and has a significant influence on both commercial environments and social contexts. As showed in the previous section, IoT has increasingly impacted many parts of modern living, including education, healthcare, and business. It now involves the storing of private data about people and businesses, the exchange of financial data, and the development and marketing of products. To meet the rising need of millions or maybe billions of connected devices and services throughout the world, the massive proliferation of IoT devices has generated a huge demand for strong security. (Abomhara and Køien [2015]).

Although security issues are not new in the context of information technology, many IoT implementations' characteristics create new and distinct security concerns. It must be a top priority to address these issues and guarantee security in IoT goods and services. Particularly as this technology becomes more prevalent and incorporated into our daily lives, users must have confidence that IoT devices and related data services are safe against threats (Singhania [2015]).

Inadequately protected data streams on IoT devices and services might act as possible entry points for cyber attacks and expose user data to theft. Because IoT devices are linked, every unsecure device that is connected to the Internet has the ability to influence the global security and resilience of the network.

This challenge is amplified by other considerations like the mass-scale deployment of homogenous IoT devices, the ability of some devices to automatically connect to other devices, and the likelihood of fielding these devices in unsecure environments.

As explained in Abomhara and Køien [2015], threats are increasing constantly, and attacks are becoming more frequent and sophisticated. The scale of networks and the number of possible attackers are both expanding, and the technologies available to potential attackers are likewise getting more advanced, effective, and efficient (Schneier [2000], Kizza [2017]). Because of this, IoT need security from threats and vulnerabilities in order to realize its full potential (Taneja [2013]).

Security is a procedure that preserves an object against physical damage, illegal access, theft, or loss by upholding the confidentiality and integrity of information about the object and making that information accessible whenever it is required. No item, tangible or not, can ever be in a totally safe condition and still be useful, hence according to Kizza [2017], there is no such thing as the secure state of any object. If a procedure can preserve an object's maximal inherent value under many scenarios, it is considered secure. The IoT environment has the same security needs as any other ICT systems. Hence, preserving the maximum intrinsic value of both tangible things (devices) and intangible ones is necessary for assuring IoT security (services, information and data).

For numerous reasons, connected devices or machines are very attractive to cyber-attackers (Abomhara and Køien [2015]):

1. Since most IoT devices are operated without human supervision, it is simple for an attacker to physically access them.

2. The majority of Internet of Things components interact through wireless networks, where an attacker might eavesdrop and collect private information.

3. Due to their limited power and computational capacity, the most of IoT devices components cannot implement complicated security systems.

Additionally, cyberattacks might be conducted against any IoT facilities and assets, perhaps resulting in system failure or damage, endangering the general population, or causing owners and users to suffer significant economic loss. Attacks against home automation systems, as well as unauthorized access to lighting, heating, air conditioning, and physical security systems, are a few examples. Cyberattacks might be launched, among other things, against any public infrastructure, such as utility systems (power systems or water treatment facilities), to cut off the flow of water or energy to residents. Previous section presented many relevant examples of attacks that IoT devices could face.

With the transition to the IoT, security and privacy concerns are becoming more and more important to consumers and providers. The extent of the harm brought on by an attack or a compromised device connected to the network is definitely simple to comprehend. It is widely acknowledged that integrating any IoT device into our personal, professional, or commercial surroundings might lead to new security issues. Users and vendors must take these security and privacy issues into account and exercise caution.

As stated in Singhania [2015], ensuring the security, reliability, resilience, and stability of Internet applications and services is critical to promoting trust and use of the Internet. When it comes to the risk tolerance associated with the online activities we wish to engage in, as Internet users, we must have a high level of confidence that the Internet, its applications, and the devices connected to it are secure enough to support such activities.

This is also true with the Internet of Things, because users' capacity to trust their environment is intrinsically tied to security. People are less likely to utilize the Internet if they don't think their connected devices and personal information are sufficiently safe from abuse or danger.

Electronic commerce, technological advancement, free expression, and pretty much every other facet of internet activity are all affected globally by these aspects. In fact, the industry should consider guaranteeing security in IoT devices and services a key priority.

1.4 IoT and Blockchain

IoT and blockchain are remarkable technologies that will significantly affect the IT and telecom sectors. These two technologies concentrate on enhancing consumers' overall levels of comfort, visibility, transparency, and trust. The IoT devices receive in a continuous mode data related to the environment and therefore also relative to the individuals in it, thus collecting sensitive data and providing real-time data from sensors and blockchain provides the key for data security using a distributed, decentralized and shared ledger (Miller [2018]). Data integrity is essential for decision-making in a range of contexts, from clinical diagnosis to environmental protection, from changing machine behavior to

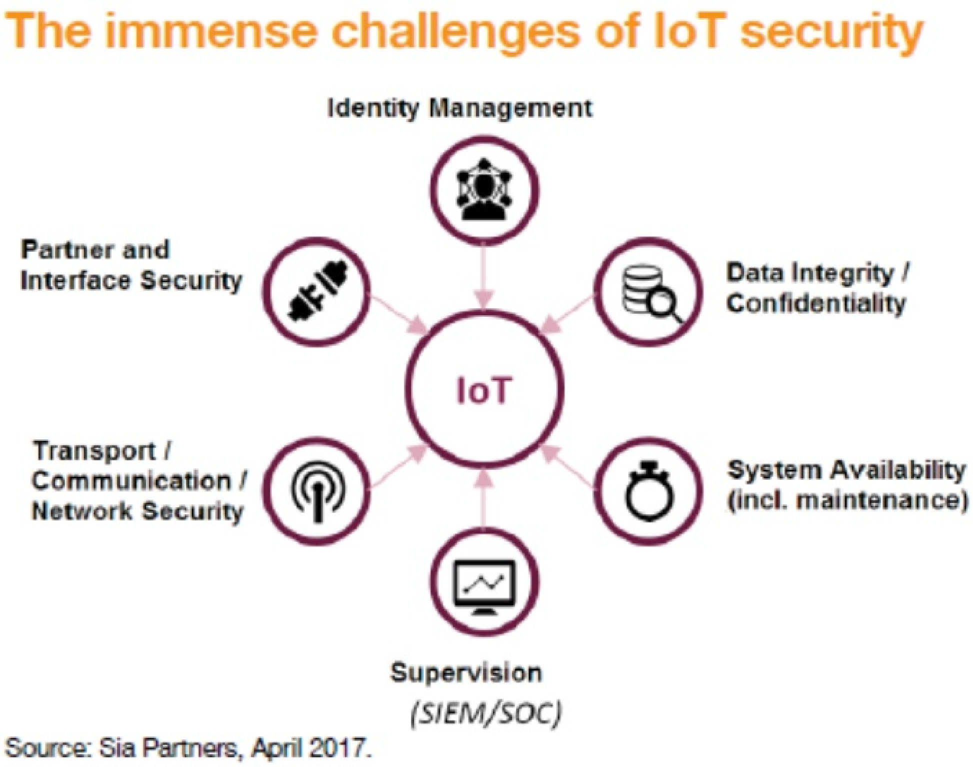

The immense challenges of IoT security
Identity Management
Partner and Interface Security
Data Integrity / Confidentiality
IoT
Transport / Communication / Network Security
System Availability (incl. maintenance)
Supervision (SIEM/SOC)
Source: Sia Partners, April 2017.

Figure 1.4. The immense challenges of security (Raphaël Dereymez [2018])

identification and access control. These are just a few IoT characteristics that blockchain technology can address.

In the following are introduced some advantages that comes from the usage of blockchain in IoT applications. Some key benefits, collected by Hassija et al. [2019] are:

1. **Data coming from IoT devices can be stored in Blockchain:** Applications for the Internet of Things use a wide range of interconnected devices. Other devices are connected to and in control of these devices as well. The usage of IoT applications from any place is made possible by a setup's additional connection to the cloud. Blockchain is a potential method for storing data and guarding against its exploitation because of the wide space for data flow it provides. Blockchain can serve as an appropriate option to store and transport data, regardless of the layer in an IoT application.

2. **Distributed nature of blockchain allowing secure data storage:** Due to the distributed nature of the blockchain design, it can minimize the possibility of having a single point of failure, which is a problem for many cloud-based IoT systems. Regardless of the distance between the devices, it is simple and safe to store the

data they produce on the blockchain (Sarkar and Misra [2016]). Also, using the blockchain network for data storage, duplicated among large numbers of computers and devices would allow for a redundant infrastructure that reduces latency times and risks related to server failures.

3. **Data encryption using the hash key and verified by miners:** Blockchain allows for the storage of only the hash key for the data, not the actual data itself. The hash key may be associated with the original data, and the real data can be kept on the cloud. The data's hash will change if the data changes in any way. The data are now private and secure. Since only the hash values are saved in the chain, the size of the data will not have an impact on the size of the blockchain. Using the hash of the data, only the intended parties and those who are permitted to use that data may access the data from the cloud. Besides, because every set of data saved on blockchain is correctly validated in the network, utilizing blockchain as a solution decreases the likelihood of storing faulty data from the devices.

4. **Prevention from data loss and spoofing attacks:** Spoofing attacks on IoT applications include the introduction of a new adversary node into the IoT network, which begins acting like it is a component of the original network. The attacker can quickly collect, observe, or inject data into the network using spoofing. Blockchain offers a possible defense against these threats. Since every valid user and device is registered on the blockchain, there is no need for central brokers or certification bodies because devices can readily identify and verify one another (Dickson [2016]). IoT devices carry a risk of data loss due to their low power nature. There may be situations where both the sender and the receiver lose the data as a result of certain external environmental problems. Utilizing blockchain can avoid these losses since there is no way to delete a block after it has been put to the chain (He et al. [2018]).

5. **Blockchain to prevent unauthorized access:** There is a lot of regular connection between different nodes in many IoT applications. Since public and private keys are used for communication in blockchain, only the intended person or node may access the data. The data is encrypted with keys, so even if the undesired person is able to view it, the contents will be unintelligible. As a result, the blockchain data format aims to address the many security problems that IoT applications encounter.

6. **Proxy-based architecture in blockchain for resource-constrained devices:** Although blockchain offers numerous security characteristics for a distributed system, resource limitations provide a unique difficulty for IoT. IoT devices are unable to hold huge ledgers due to their severe resource limitations. To make the usage of blockchain in IoT easier, numerous efforts have been made in this area. One of the possible approaches for enabling IoT devices to use blockchain is proxy-based architecture. To save the resources in an encrypted format, proxy servers can be installed on the network. The client can download the encrypted resources from the proxy servers (Alphand et al. [2018]).

7. **Elimination of centralized cloud servers:** Blockchain can make IoT devices

> more secure by finally getting rid of centralized cloud servers and making the network peer-to-peer. The main focus of data thieves is centralized cloud servers. A cryptographic hash function will be used to encrypt the data before it is distributed throughout all of the network's nodes utilizing blockchain.

A suggestion for a specific solution comes again from Hassija et al. [2019] were it is pointed IOTA as another upcoming and highly promising solution to secure IoT. IOTA is a DLT that was created specifically for IoT devices with limited resources. The previous two requests must be verified for every incoming request in the network. IOTA is able to offer a high level of security at the device or edge level by using this process of cumulative validations. Request verification uses the tip selection algorithm. For each request, a cumulative weight is produced. A device is more secure the more weight it has in the network. In contrast to blockchain, which employs a chain data structure, IOTA uses a tangle data structure (Oodles).

Chapter 2

Evolution of human trust, from physical to digital trust: the Trust over IP stack

In this chapter, we will go through different phases of the history of human trust following the ToIP [2020] white paper from which started the research that ended up to the drafting of this work. These phases are:

- **Trust in the Pre-Internet Era:** the basic, universal techniques humans developed to build trust in relationships before the advent of the internet.
- **The Internet Era and the "Trust Gap":** what took place when we went online and why there is such a big "trust gap" when compared to real-world trust.
- **The New Era of Digital Trust:** how open standard digital credentials and governance frameworks might help us close the trust gap.
- **The Trust over IP Stack:** what the TCP/IP stack achieved for the peer-to-peer exchange of data packets, the four-layer architecture created by ToIP has the potential to do for the peer-to-peer exchange of reliable digital credentials.

2.1 Trust in the Pre-Internet Era

Before the advent of digital networks, when all interpersonal and professional encounters took place in person, we had developed a straightforward, all-encompassing, decentralized process for establishing trust. We made use of a variety of credentials.

Notably, when we refer to "credentials," we don't only mean the documents you keep in your wallet to confirm your identification, such as your driver's license, government ID, job card, credit card, etc. We refer to any document, regardless of size, that enables you or your business to demonstrate a fact that promotes the formation of trust.

The basic "trust triangle" depicted in Fig. 2.1. is the reason credentials have become a universal technique for creating real-world trust.

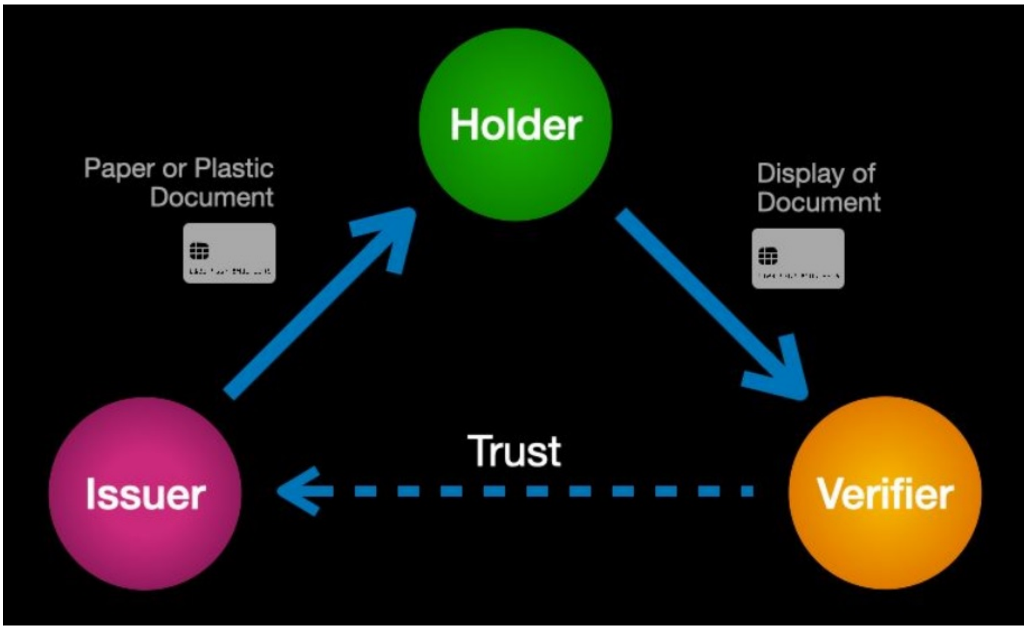


Figure 2.1. The credential trust triangle (ToIP [2020])

No matter what type of credential, the triangle involves the same three primary roles:

1. Credentials come from issuers since each credential has an issuer. The majority are issued by organizations, including governments (passports), banks (credit cards), colleges (degrees), businesses (employment IDs), churches (awards), etc. However, issuers can also be people.

2. Holders obtain credentials from issuers, keep them in their wallets or file cabinets, and provide them to verifiers upon request (and approved by the holder). Although we typically conceive of holders as being people, they may also be businesses or even objects (such as the registration for a car).

3. Verifiers might be anyone looking for some form of trust guarantee regarding the credential holder. Verifiers employ their own procedures to check the credentials' validity and authenticity after requesting the credentials they require. For instance, a TSA agent at an airport will examine a passport or driver's license for certain characteristics to determine whether it is legitimate, and will then check to make sure it is not expired.

Some credentials can only be issued by a single issuer, but others can be issued by a variety of issuers. For instance, hundreds of nations offer passports, while tens of thousands of banks and credit unions issue credit cards. The governance trust triangle is a second trust triangle that applies to any credential that will be extensively utilized by many holders and respected by many verifiers, as shown in Fig. 2.2.

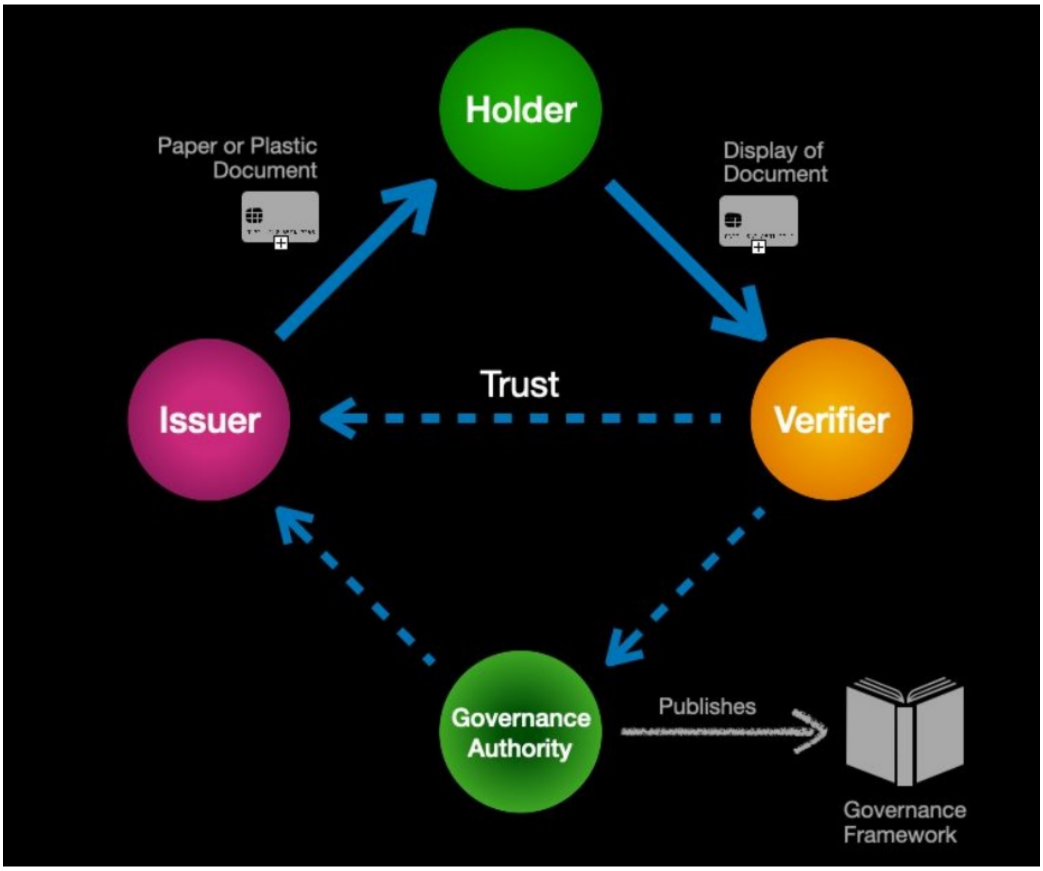


Figure 2.2. The governance trust triangle (ToIP [2020])

Any group of issuers that wish to standardize the commercial, legal, and technological guidelines for granting, maintaining, and validating a set of credentials may be represented by a governing authority. The goal of a governance authority, regardless of its structure (government, consortium, cooperative), is to publish a governance framework that details the guidelines that the participants in a trust community agreed to follow.

2.2 The Internet Era and the “Trust Gap”

Since the earliest computers were still refrigerator-sized, access to the "login" terminals operated in the same way as everything else in the real world: a guard would open the door after checking the credentials you have in your wallet. But as soon as we entered a networked environment, physical security was no longer able to regulate login access. Therefore, we developed computer-based login account access restrictions. The dreadful username and password were created in this way. We attempted to govern everything through login accounts since they served as a virtual "door" to servers, which in turn served as the gateway to networks.

By examining the trust model, or how trust truly moves between the parties, it is simple to identify the basic issue with intermediaries. All trustworthy interactions in the present account-based client-server paradigm must be mediated by a server, and all parties involved must be integrated with that server. All the participants in the interaction must have faith in whoever manages this server. The model in Fig. 2.3 is this one. Compare this to the paradigm of peer-to-peer trust on the right. There is no need for middlemen. Server integration is not necessary. Direct bonds of trust are established between every peer and every other peer. Each peer establishes its own rules for who it will trust.

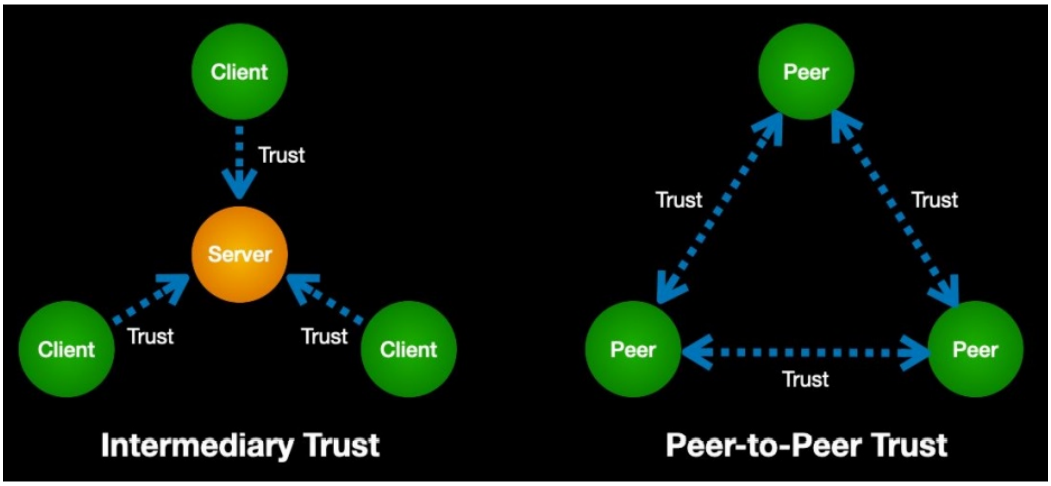


Figure 2.3. Client-server trust model (ToIP [2020])

Ironically, this is exactly how the trust mechanism for credentials from the actual world operates. Each peer holds its own credentials and verifies the credentials of other peers. When necessary, any peer can grant credentials. This is what is to blame for our trust deficit. In a decentralized, peer-to-peer trust model, intermediaries are not necessary, but our existing Internet trust model calls for them.

2.3 The New Era of Digital Trust

We delayed bringing our long-established physical credentials-based real-world trust paradigm to the digital realm for obvious reasons. Physical identification documents are reasonably simple to make (using traditional printing and stamping techniques) and validate (via human inspection, if we accept a reasonable degree of error). Digital credentials are significantly more difficult.

Introducing digital credentials would have several advantages. Like we do now with real credentials, each of us might acquire credentials in a digital wallet.

Thankfully, early adopters at the World Wide Web Consortium realized the potential of digital credentials a number of years ago (W3C). They started the process of standardizing the necessary file formats and digital signatures. The Verifiable Credentials Data Model v1.1 (W3C [2022b]) specification was the end outcome, and it was accepted as a complete W3C standard in September 2019. Figure 2.4 illustrates the operation of Verifiable Credentials.

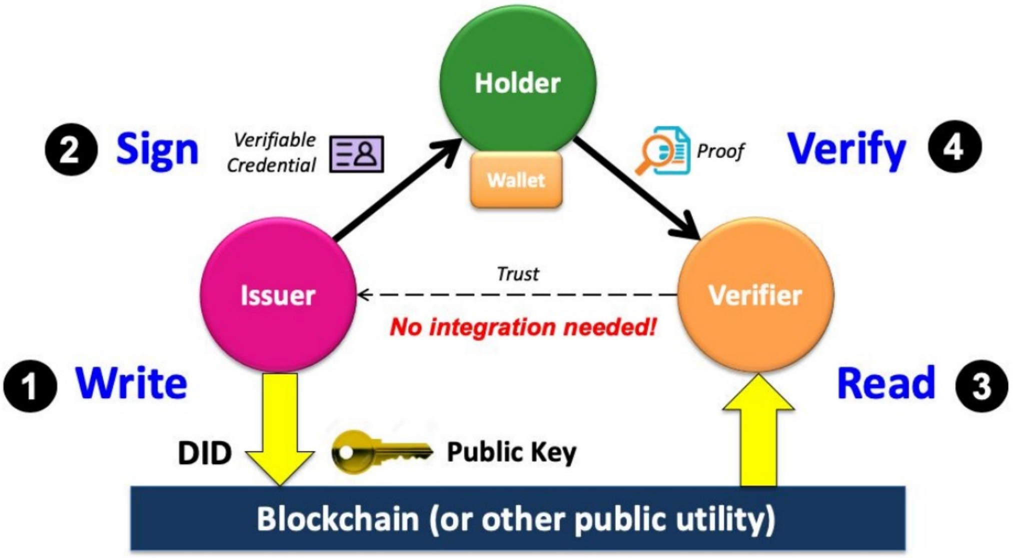


Figure 2.4. Verifiable credential (ToIP [2020])

1. The issuer first writes a Decentralized Identifier to a blockchain (or some other trusted public utility) along with its public key (and any additional cryptographic data required for the issuer's verifiable credentials).

2. A verifiable credential is issued to a qualified holder who then keeps it in her own digital wallet after the issuer uses its private key to digitally sign it. Keep in mind that the whole issuing procedure happens off-chain to protect privacy.

3. Then, a verifier asks the holder to provide a digital proof of one or more credentials. If the holder agrees, the wallet produces the proofs and sends them back to the verifier. The verifier utilizes the issuer's DID, which is present in the proofs, to read the public key and other cryptographic information from the blockchain.

4. The last step involves the verifier confirming the validity of the proofs and the integrity of the digital credential using the issuer's public key.

We may apply the same trust model—and mental model—to Verifiable Credentials and digital wallets as we do to physical credentials and wallets. Furthermore, we can expand this model to any size trust network and modify it to any trust community using governance frameworks. Fig. 2.5 depicts the trust triangle for digital governance.

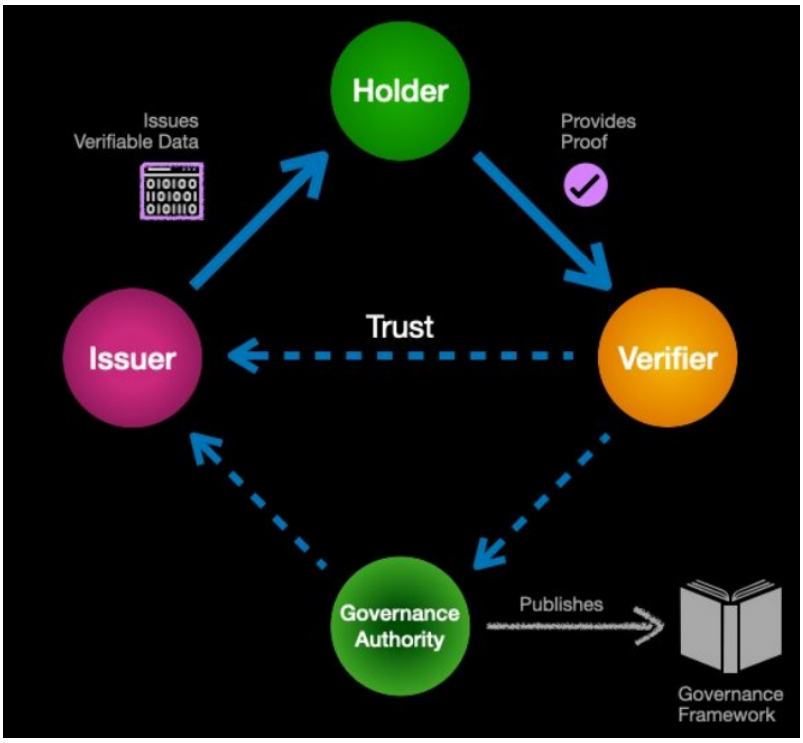


Figure 2.5. Digital governance trust triangle (ToIP [2020])

The frameworks for digital governance form the foundation of this new era of digital trust, as this diagram illustrates. Every digital credential in your wallet ought to be supported by a governance structure that outlines the operational parameters of the company, law, and technology. We can now usher in a new age of Internet-scale digital trust infrastructure by fusing the technical trust of W3C Verifiable Credentials and DIDs with the human trust embedded in these governance frameworks.

2.3.1 What is a DID?

The W3C [2022a] defines the Decentralized Identifiers as a new type of identifier that enables verifiable, decentralized digital identity. A DID can represent any subject that the DID's controller specifies (e.g., a person, group, object, data model, abstract entity, etc.). DIDs have been created to be independent of centralized registries, identity providers, and certificate authorities, in contrast to conventional, federated identifiers. In particular, the design enables the controller of a DID to demonstrate control over it without requesting authorization from any other party, even while other parties may be utilized to aid in the finding of information pertaining to a DID. DIDs are URIs (Uniform Resource Identifier, the standard identifier format for all resources on the World Wide Web) that link a DID subject to a DID document, enabling trusted interactions with that subject.

Each DID document may contain cryptographic information, verification techniques, or other services that offer a variety of ways for a DID controller to demonstrate control over the DID. Services allow for secure communication involving DID subjects. If the DID subject is an information resource, such as a data model, then a DID could offer the capability to return the DID subject itself.

2.4 The Trust over IP Stack

Developers community realized this new peer-to-peer trust model could support an entire layer of Internet-scale digital trust infrastructure when they started integrating DIDs and verifiable credentials. The ToIP Technology Stack shown in Fig. 2.6 was designed in this context.

Apart the ToIP Stack, it is also worth to mention the Governance Stack, the natural governance and policy issues that need to be resolved to promote commercial, legal, and societal adoption. However, following paragraphs will focus only on the ToIP Stack, while the Governance Stack can be considered outside of the scope of this work.

- **Layer One: Public Utilities**
  The first two layers of the ToIP stack are intended to offer technical trust, or the confidence that two machines can connect securely and privately. You must be able to firmly validate the public key of the party you are connecting to in order to do this utilizing public key cryptography. The W3C Decentralized Identifier protocol, which standardizes how you may permanently identify and verify a public key held on a blockchain or other distributed system, resolves this issue without the use of centralized certificate authorities.

  The public utilities created by this method act as solid cryptographic roots-of-trust for the public keys and DIDs of Verifiable Credential issuers. Any technology that can offer the required trust guarantees, such as blockchains (of any sort), distributed ledgers, decentralized file systems, distributed hash tables, and so on, can be used to develop ToIP Layer One services.

- **Layer Two: Peer-to-Peer Protocol**
  Layer Two is about the branches, the digital wallets and digital agents required to establish secure, private peer-to-peer connections using either peer DIDs (from Layer Two) or public DIDs (from Layer One). If Layer One is about the solid cryptographic roots of technical trust, then Layer Two is about the roots. Since they never need to touch a blockchain, the latter may be traded directly between peers, which has a huge benefit for both scalability and privacy.

- **Layer Three: Data Exchange Protocols**

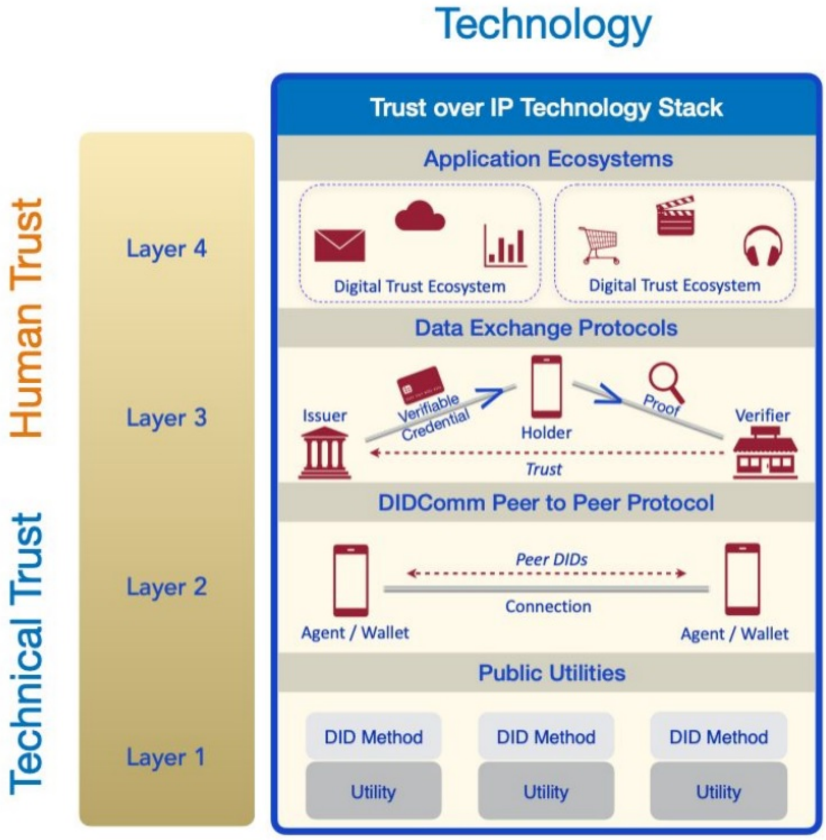


Figure 2.6. Technology stack (ToIP [2020])

Human trust is created and sustained at layers three and four. The Verifiable Credential trust triangle covered in Part Three is located in Layer Three of the technological stack. This layer is where DIDComm-based credential exchange protocols are used by issuers, holders, and verifiers to exchange credentials and proofs. Remember that there are several additional secure messaging and workflow automation protocols that may be implemented at Layer Three; these are just a few examples of the trusted data exchange protocols that can function at this layer.

- **Layer Four: Application Ecosystems**
  The fourth layer is the application layer, where users engage in trusted interactions with programs to further a particular corporate, legal, or social goal. ToIP-enabled applications call the ToIP stack in the same way that Internet-enabled applications call the TCP/IP stack to communicate over the Internet in order to register DIDs, establish connections, acquire and exchange Verifiable Credentials, and engage in trusted data exchange using the Layers One, Two, and Three protocols.

Chapter 3

Secure data exchange through the IOTA Tangle

This chapter traces Carelli et al. [2022]'s work and proposes a framework for exchanging data securely through IOTA Tangle. Its construction is laid out in detail.

3.1 Security regarding data transmission in IoT technologies

IoT technologies make it possible to gather data from an expanding range of sensors for analysis and reasoned decision-making. These systems now need to execute (near) in real-time in order to prevent offline data analysis, and they also need to have end-to-end security with data source authentication, data confidentiality, and data integrity from sensors to remote sites where data is stored and processed. End-to-end security concerns are of the utmost significance and frequently influence the choice of the solution. The work of Carelli et al. [2022] contains a solution for the aforementioned problem and is considered a starting point for the work contained in the second part of this master's thesis.

In the modern day, Distributed Ledger Technologies are another pertinent and practical choice to support data transmission while using crucial security aspects like data verifiability and immutability. The IOTA Tangle in its latest Chrysalis version is a sensible option in the Carelli et al. [2022] perspective for dealing with widely dispersed sensor systems that produce data at high throughput.

3.2 The IOTA Tangle

The Tangle (Wiki [2022]) is a data structure that replicates itself over a network of computers (also known as "nodes") and contains all the information required to monitor token ownership.

The Tangle network's nodes continuously validate transactions at maximum speed without charging any fees. When a transaction refers to a unique one known as a milestone, it is regarded as legitimate. Proof-of-Work (PoW) is simply intended to reduce spam transactions on the Tangle and is not intended to be a component of the validation process, which would protect the network as in blockchains. A node must validate two earlier transactions before it is allowed to broadcast a new transaction over the network. As a result, consensus and transaction validation occur more quickly as there are more incoming transactions. Though the aggregate throughput is theoretically limitless, in reality it is determined by the consensus process and low-complexity PoW. The Tangle's operating principles allow it to expand with the volume of incoming transactions, making it a viable alternative for the IoT and sensor industries.

Moreover, the Tangle is built to provide the ability to store data with a transaction of a particular message carrying an indexation-type payload (Foundation [2022a]). The data is anchored to the Tangle via a transaction that contains an indexation message. Any node ready to use that information can find it at the given index. Data is organized via the Tangle using a protocol that operates at Layer 2 (L2), which is on top of IOTA Layer 1 and is responsible for interacting with the Tangle. This allows for (i) the transmission of complex data streams and (ii) simple data retrieval. A L2 protocol of this type uses cryptography to protect data transfer over the Tangle from beginning to finish. It offers the primitives to quickly collect and consume the data stream while confirming it, cipher and decode data, prove data source and ownership, and organize a stream of data across the Tangle. Data integrity and immutability are provided by the Tangle itself.

A L2 cryptographic protocol coupled with the Tangle can be a safe transport method for secure data transfer. In order to communicate data safely and in (near) real-time, any distributed sensor system can use this combination as a trust layer. Furthermore, because the data are constantly attached to the Tangle, they may be checked and used a posteriori at any moment.

It is worth emphasizing that the Tangle might be a conventional data exchange interface. Without the need to redesign and create new unique data exchange interfaces, heterogeneous and independent systems, each built to serve a particular function, may yet communicate with one another. Any data source merely has to allow the other system access to its data stream.

The IOTA Foundation has developed two L2 solutions. The first is the Masked Authenticated Messaging (MAM) (Foundation [2022b]) and the second is STREAM (Foundation [2022c]).

These frameworks' fundamental flaw is that they fall short of the promise made by the IOTA ecosystem to enable the IoT and sensor worlds. Due to the programming languages used, they are really best suited for desktop applications. Additionally, they were not created with the usual IoT constrained devices' processing capacity, memory availability, or low-energy consumption needs in mind. Overall, these techniques can't be implemented in actual constraint sensing systems even though.

L2Sec, a cryptographic protocol designed in Carelli et al. [2022], will be discussed in the part that follows. It is intended for IoT constrained devices based on microcontrollers.

L2Sec gives a constrained IoT device all the tools necessary to organize a data stream across the Tangle and allow safe data exchange. This cryptographic protocol is used to organize, secure, and move data throughout the Tangle.

3.3 L2Sec—A Cryptographic Protocol for Constraint IoT

Design Principles and Features: for organizing and moving safe data over the IOTA Chrysalis Tangle, L2Sec is a simple cryptographic protocol. It is intended to be (i) light enough to run on constrained IoT devices (i.e., MCU-based platforms without operating systems), (ii) suitable for sensors application data model, such as one single publisher producing time sequenced data, and (iii) modular so that the building block can be easily used in other applications and extendable to make it easier to integrate additional features and fields.

The L2Sec protocol uses the indexation payload, a particular Chrysalis message payload, to tie the data to the Tangle. The indexation payload is made up of an index and some random data (i.e., application data). Therefore, the indexation payload encapsulates any L2Sec protocol message.

A data stream is organized by L2Sec as a single-link chain over the Tangle. Each item of data in the stream is connected to the one after it using a different index. Any subscriber to the data stream has the ability to rebuild it by beginning reading at any point on the Tangle and moving along the chain of data that connects each piece of information. In essence, each data packet includes the index for the current message and the index for the following message.

The L2Sec protocol's architecture makes it possible to incorporate a hardware secure element and outsource cryptographic functions to it. Also, the secure element may be used as a hardware Root-of-Trust and as the source of the IoT device's distinct electronic identity.

Payload structure: a L2Sec message is encapsulated in the indexation payload of a IOTA Chrysalis message, Fig. 3.1.

Message Chaining: a series of data must be connected in order for continuous data transmission and data that is longer than the duration of a single L2Sec message. The NEXT_IDX field, which contains the index of the following message in the stream to search for in the Tangle, allows for the chaining of these messages. Figure 3.2 show the chaining mechanism.

Any subscriber may read a data stream in only one way by realizing the chain through a single connection between messages. This characteristic is deliberate as it prevents the recovery of earlier data (i.e., past messages belonging to the same data stream).

The Figure 3.3 depicts the flow for the generation of INDEX and NEXT_IDX. L2Sec deterministically creates a secret key and associated public key from a random seed. The output of the public key's hash function is then used to determine the index of an L2Sec message. In addition, each L2Sec message includes a link to the following index

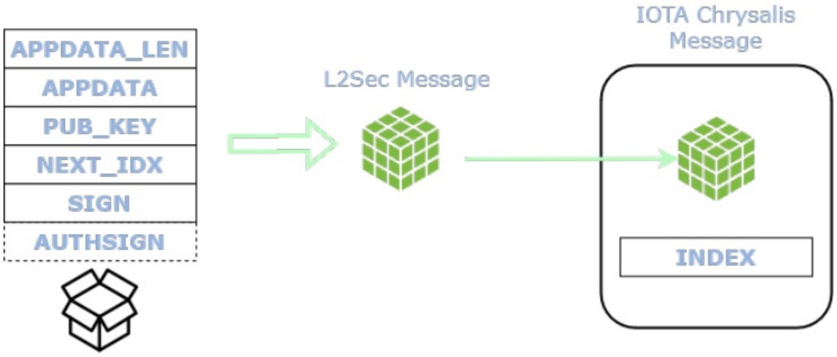


Figure 3.1. Payload Carelli et al. [2022]

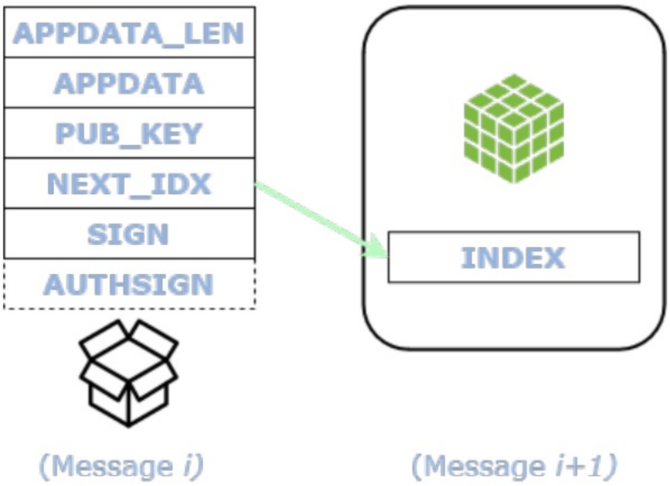


Figure 3.2. Message chaining (Carelli et al. [2022])

(NEXT_IDX), which enables the implementation of a continuous data stream. A different key-pair is used as the starting point for computing the NEXT_IDX in the same way.

Encryption: to preserve the confidentiality of the data, which will be public in the Tangle, every L2Sec message is encrypted as shown in Figure 3.4. A nonce is used as the initialization vector and a symmetric cryptographic key is used to carry out the encryption. The message's sender and the multiple subscribers pre-share the encryption key (Pre-Shared Key).

A Pre-Shared Key is used to encrypt the data between the parties participating in the conversation (i.e., author and subscribers). The work Carelli et al. [2022] does not address how the PSK is transferred between the parties, but this work's focus is on investigating

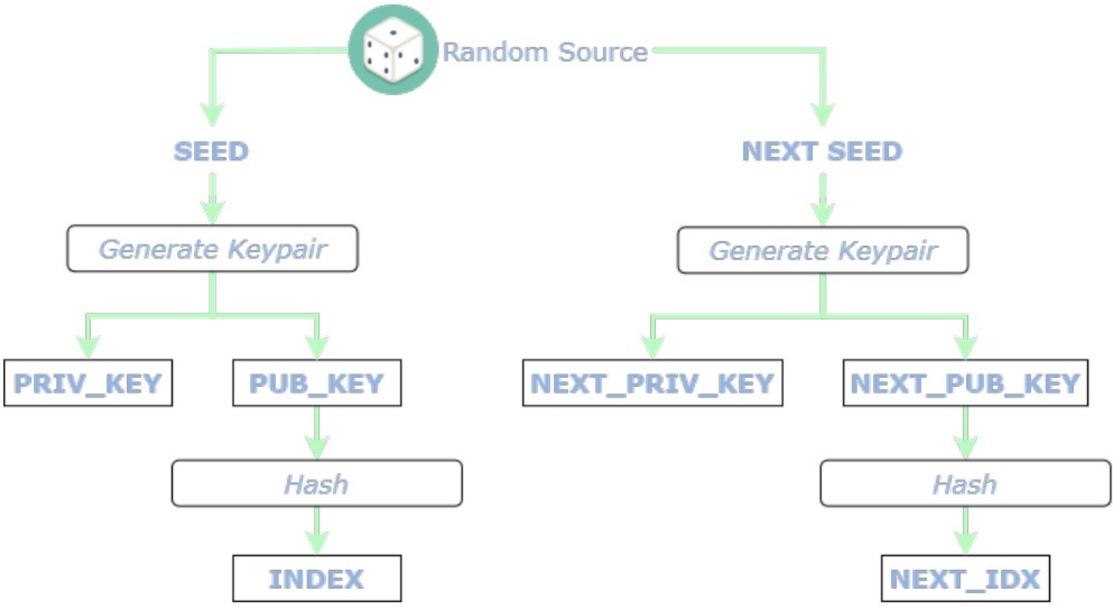


Figure 3.3. Index generation (Carelli et al. [2022])

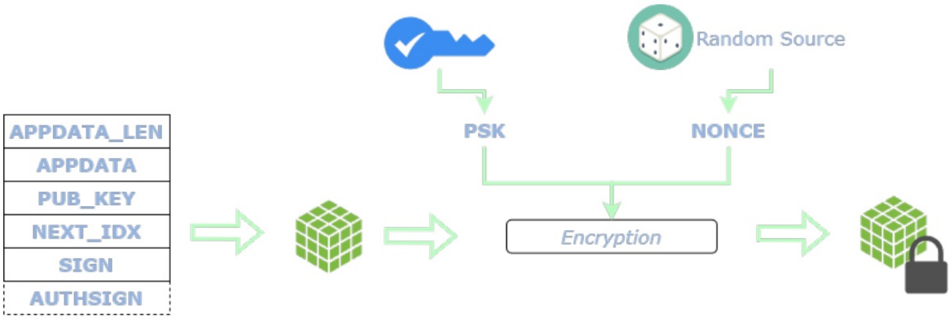


Figure 3.4. Encryption (Carelli et al. [2022])

this specific aspect searching for the best approaches.
The description of **Data Ownership** and **Authentication** of this method are out of the scope of this work, but are explained in details in Carelli et al. [2022].

Part II

Problem formalization and proposed solutions

Chapter 4

Problem formalization: A membership problem between IoT nodes

The purpose of this section is to sum up the research that brought to this work and to model the membership problem.

4.1 Context and motivation of the work

The ToIP [2020] shed light on an issue of trust in the Digital Era and provided a new vision to overcome this trust problem by introducing the Verifiable Credential Data Model (W3C [2022b]) and then creating a new structure intended to embed this data model and the DIDs: the ToIP Technology Stack (Fig. 2.6).

Subsequently, a technique of securing data transmission in sensor systems based on distributed ledger technology was put out. It promises great security, user-friendliness, and the possibility of extensive integration of heterogeneous sensor systems. IOTA Tangle is one such DLT that has a lot of promise to enhance the sharing of sensor data. Carelli et al. [2022] introduced L2Sec, a cryptographic system that can protect data transmission across the IOTA Tangle. This protocol can be implemented on constrained devices, like typical IoT devices, which will increase scalability.

At this point, the scope of this work regard the layer 2 of the ToIP technology stack, the peer to peer protocol.

In Carelli et al. [2022] this is made using a PSK among the author of the messages written in the DLT and the various subscribers, however how this PSK is exchanged between parties is not mentioned in that work.

But there are several potential disadvantages to using a PSK for authentication and encryption:

- Security: the security of the entire system depends on the security of the key. If the key is weak or is somehow compromised, the security of the system is at risk. For example, if the key is easily guessable or has been discovered by an attacker, they will be able to gain unauthorized access to the system or decrypt communications.

- Manageability: in a large network or system with many devices or users, it can be difficult to manage and distribute the key securely. This can be particularly challenging if the key needs to be updated frequently or rotated for security reasons.
- Limited scalability: a PSK system is generally not well-suited to large networks or systems with many devices or users. As the number of devices or users grows, the difficulty of securely distributing and managing the key also increases.
- Limited flexibility: a PSK system is typically less flexible than other methods, as it does not provide the same level of granular control over access or encryption. For example, it may not be possible to easily revoke access for a specific device or user without changing the key for the entire system.
- Vulnerability to brute force attacks: if an attacker is able to obtain a copy of the encrypted data, they may be able to use a brute force attack to try to determine the key by trying every possible combination. This can be particularly problematic if the key is short or not sufficiently random, as it may be more susceptible to such an attack.

Here the problem is formalised to be as general as possible and with the objective to overcome the idea of using a PSK and exploring new and better solutions.

4.2 A membership problem between IoT nodes

The problem which this work is willing to explore is how a member of a Self Sovereign Identity scheme is supposed to verify that another member, with who is in contact, is a node authorised to write its own Decentralized Identifier on the Root of Trust which can be a distribute ledger technology or directly a blockchain.

In this decentralized environment everyone can write a DID regardless he has the right to do it or is willing to tell the truth. The challenge is to find a method that can assure that the members of the group can verify if one node is a member of the same group.

In few words this work focus on finding a proof of membership of a dynamic group.

A scheme of the context in which the solution is seek is shown in Fig. 4.1: the nodes N_1 and N_2 can write their own DID on the DLT using a preceding agreed method, the DID method (W3C [2022a]), with all its rules. Every node also owns a tuple of asymmetric keys (P_k, S_k) that are used to sign the DID written to claim its property. When they are willing to communicate, every node has to verify if the other with who is in contact is authorized to write its own DID. The problem is shifted to the peer to peer communication between nodes (layer 2 of ToIP [2020] technology stack) and is modelled as a membership problem. Every node has to prove that it is part of the group which has the right to write its own DID.

Some possible solutions to overcome this problem are explored in the next sections.

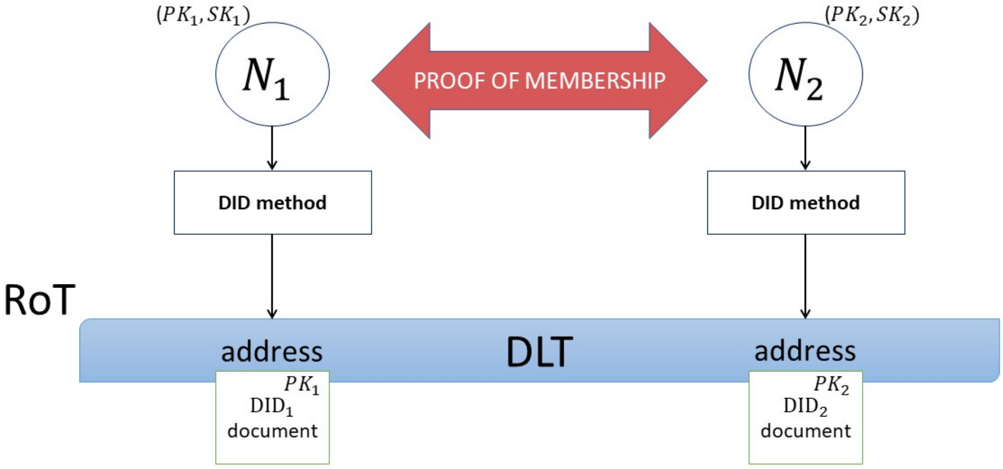


Figure 4.1. SSI scheme: before establishing a peer to peer communication, there should be a proof of membership between the parts.

Chapter 5

Merkle trees

This chapter introduces and analyses a first proposed solution for the previously introduced problem, based on Merkle trees. After a brief introduction is explained the construction, the verification phase and then the critical review of this solution.

5.1 Introduction to Merkle tree

A Merkle tree, also known as a hash tree, is a data structure that is used to efficiently verify the integrity of large amounts of data. It is named after its inventor, Ralph Merkle, who introduced the concept in a paper first published in 1979 (Merkle [1979]).

It is a binary tree, in which each leaf node contains a data block and each non-leaf node contains the hash of the concatenation of its child nodes.

One of the key advantages of Merkle trees is that it allows for efficient and secure data verification. A verifier only needs to receive the root hash of the tree, along with the hashes of a few leaf nodes (the siblings), in order to verify the integrity of the entire data set. This means that the verifier does not need to receive the entire data set, which can be very large, in order to perform the verification.

Merkle trees are often used in the context of blockchain technology, where it is used to efficiently verify the integrity of the data stored in the blockchain. As a result, this structure serves as an appropriate example of a cryptographic commitment scheme in which the tree's root is regarded as a commitment and its leaf nodes may be exposed and shown to be a part of the original commitment (Wikipedia [2022a]).

The described construction will be used in a way such that the commitment scheme provided by the Merkle tree allows performing a proof of membership.

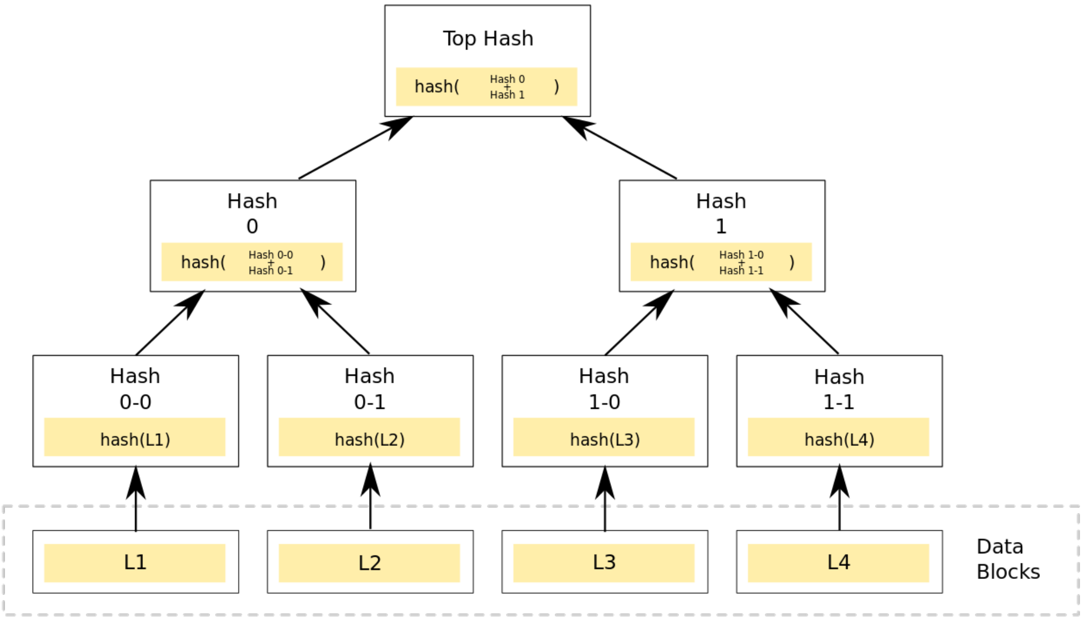


Figure 5.1. Merkle Tree (Wikipedia [2022a])

5.2 Merkle tree as proof of membership

This proposed solution comes with a precise structure of the SSI scheme, that is specifically the one introduced by Carelli et al. [2022].

The DID written on the RoT can be subjected to various operations: Create, Revoke, Update, Delete (i.e., CRUD, as illustrated in Fig. 5.2). So a created DID, exists until it's revoked or deleted, and every time it is updated it changes its index.

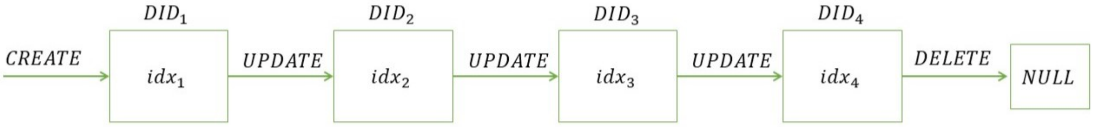


Figure 5.2. Example of a structure of a DID lifetime using the CRUD method.

A DID index is created with this method (visual representation on Fig. 5.3): starting from a simple key derivation function, based on HMAC message authentication code, is generated a seed C. From this seed is extrapolated a couple of secret and public key (Sk, Pk). From the public key is computed the index: $idx = H(Pk)$ where H is an hash function. When a DID is updated, a new index is created in the same way. Every index is linked with the following with a one way link, such that it is impossible to retrace the previous indexes.

For a more in-depth description, look at section 3.3.
The proposed solution imply knowing previously how many times a member is willing to update his DID before the deletion. This because it needs to publish the root of the Merkle tree built upon its DID indexes as in figure 5.3, which is used to verify the fairness of the indexes of the DID.

5.3 Verification phase

In general, Merkle proofs are used to decide upon the following factors (Prahalad [2018]):

- If the data belongs in the Merkle tree.

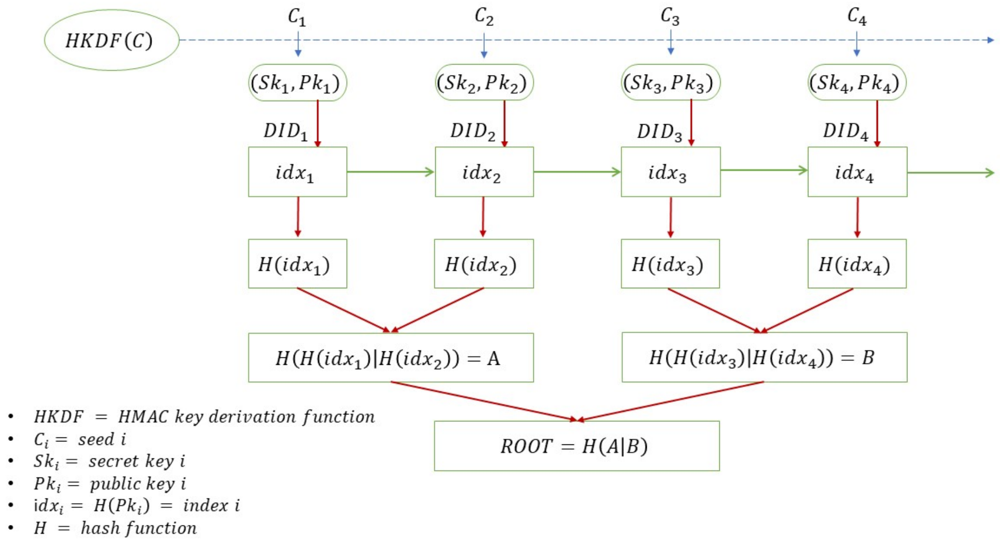


Figure 5.3. A member Merkle tree of indexes for DID.

- To concisely prove the validity of data being part of a dataset without storing the whole dataset.
- To ensure the validity of a certain dataset, being inclusive in a larger dataset without revealing either the complete dataset or its subset.

As stated before, Merkle trees rely heavily on one-way hashing. No two plaintext hashes can or should be the same, since one-way hashes are deterministic algorithms that are meant to be collision-free.

Merkle proofs are produced by hashing together a hash's corresponding hash and ascending the tree until you reach the root hash, or Merkle root, which is or may be made publicly available.

A Merkle root is the hash of all the hashes of the indexes in a block of a blockchain. Verifying a Merkle root involves checking that it is equal to the hash of all the hashes of the individual transactions in the block. This is done to ensure the integrity of the data in the block and to confirm that no transactions have been altered or tampered with. To verify a Merkle root, the hashes of all the transactions in the block are first calculated and then hashed together to produce a single hash value, which is compared to the Merkle root in the block header. If the two values are the same, the Merkle root has been successfully verified.

Similar to the case of blockchain, in our specific case, during the verification phase a verifier is given the siblings of the Merkle tree, such that together with the index that wants to check he can compute the ROOT and verify that is the same as the one published. The procedure is explained in Fig. 5.4.

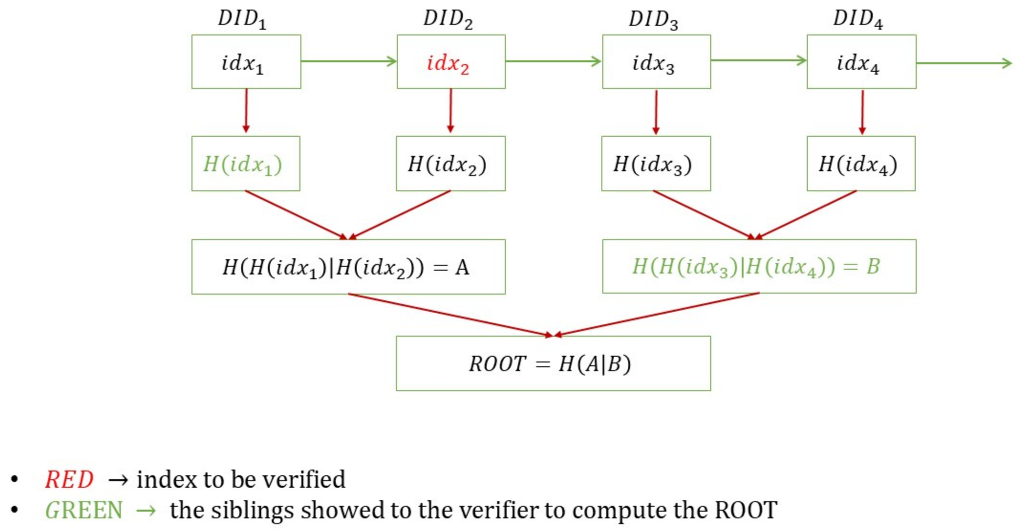

DID_1
DID_2
DID_3
DID_4
idx_1
idx_2
idx_3
idx_4
$H(idx_1)$
$H(idx_2)$
$H(idx_3)$
$H(idx_4)$
$H(H(idx_1)|H(idx_2)) = A$
$H(H(idx_3)|H(idx_4)) = B$
$ROOT = H(A|B)$
• RED → index to be verified
• GREEN → the siblings showed to the verifier to compute the ROOT

Figure 5.4. Verification of the second index. Starting from idx_2 and using the siblings in green, the verifier can compute the ROOT and verify that is the same as the one published.

5.4 Data ownership

The verifiers need a trusted source where every ROOT of each Merkle tree is available and verifiable in plain text: in this way, the verifiers can check if their computed ROOT is the correct one. In fact, to assure and link a ROOT to a node, all the ROOTs are published in a public list including the Group Manager (GM) signature and the ID of a member $ID|ROOT|SIGNATURE$:

- $ID_1|ROOT_1|SIGNATURE$
- $ID_2|ROOT_2|SIGNATURE$
- $ID_3|ROOT_3|SIGNATURE$
- $ID_4|ROOT_4|SIGNATURE$
- ...

where the subscript attached to the ID and $ROOT$ denote the member's index.

The GM signature, useful to assure the validity of the ROOT published, makes this method more centralized because it is needed an authority that secures the procedure.

5.5 Add and revoke from the list

Adding or revoking members is solved by adding or removing *ROOT* from the formerly introduced list.

Adding a member to the list of the participants is simple. The new member that is willing to join the scheme has to compute his indexes by previously deciding how many times to update his DID. After that he has to compute his own Merkle tree starting from indexes. In the end, he has to post his ROOT in the list with the GM signature attached.

The revoke procedure is simple as well: the GM has only to remove the ROOT linked to the member compromised, from the list of all the ROOTs.

5.6 Critical review of the solution

In this section there is a sum up on the introduced method that uses Merkle tree.

This is a method that solves the proposed problem and that suits well the construction that is used in the L2SEC protocol in Carelli et al. [2022], so it can be used in a self-sovereign identity schemes to provide several benefits:

- Improved security in respect to the PSK where the security of the entire system depends on the security of the key. In that case, if the key is weak or is somehow compromised, the security of the system is at risk.

  With the Merkle tree method, an attacker should guess all the next and previous indexes of the DID to generate the Merkle tree of that specific member and prove to an eventual verifier that he possesses the necessary siblings to compute the ROOT.

- Simplified key management, because unlike the PSK, it is not necessary to manage or distribute any secret key (i.e., PSK), but just a public list of signed ROOTs of Merkle trees. That is a big advantage, especially for large networks or systems with many devices or users, as in an IoT scenario.

- Improved scalability, as stated before, using PSK system is generally not well-suited to large networks or systems with many devices or users. But this method has new users only to compute their Merkle tree without exchanging anything with each other (except for communicating the ROOT to the GM) so the network system can be as large as it needs.

- Decentralization, every member has its own Merkle tree and ROOT, there is not the same shared key between all the members.

- Flexibility for dynamic groups, using this method comes with an efficient way to add or delete a member without affecting all the system.

- Quantum-Safe, cryptographic hashes are only slightly affected by quantum computers (Nakov [2018]).

Despite these advantages, this method has also some impactful issues:

- Single point of failure, despite being a more decentralized process in respect to the PSK, it needs a central authority in the figure of the Group Manager.
  The GM has the power to add or revoke every member, so if it is compromised, the system results compromised in its entirety.
  Without compromising directly the GM, another issue comes with the list of the ROOTs that can be compromised without the GM noticing. This list needs to be protected by some security requirements.

- Poor DID flexibility management, a member has to know in advance how many times its DID will be updated before being deleted. It is a prerequisite to compute the Merkle tree.

Overall, the decision to use a Merkle tree in a self-sovereign identity scheme will depend on the specific requirements and goals of the system, as well as the potential trade-offs between the benefits and drawbacks of this approach. Other solutions will be presented in the next chapters.

Chapter 6

BBS group signature

In this chapter of the second part will be explored a second proposed solution for the previously introduced problem, based on the group signature BBS. The scheme is exposed in its entirety, then is proposed a variation of the original scheme suitable for our purpose, it is explained how this scheme can be used as a proof of membership and, in the end, it is made a critical review of the solution.

6.1 Group signature schemes

In this work, it is explored a solution to the previously exposed problem, using the group signature schemes.

In the following it is presented a definition of group signature (Wikipedia [2022b]) with some of its characteristics.

A group signature scheme is a technique that enables a group member to secretly sign a message on the group's behalf. Chaum and van Heyst [1991] were the ones who initially conceived the idea. A group manager, who is in control of adding group members and has the authority to expose the original signer in the event of disputes, is crucial to a group signing system. In certain systems, a membership manager and a revocation manager, respectively, are in charge of adding new members and removing signature anonymity. Multiple different schemes have been presented, but they should all follow to these fundamental characteristics:

Soundness and completeness: invalid signatures never pass verification, while the authentic signatures of group members always do.

Unforgeability: valid group signatures can only be produced by group members.

Anonymity: without the group manager's secret key, given a message and its signature, it is impossible to know who signed it.

Traceability: given any valid signature, it should be possible for the group manager to determine whose user signed it (this and the preceding criterion indicate that only the group manager may compromise user anonymity).

Unlinkability: we cannot determine if the signatures came from the same signer or not, given two messages and their respective signatures.

No framing: even if everyone in the group (including the managers) worked together, they couldn't create a forged signature for someone who wasn't in the group.

Unforgeable tracing verification: a signer cannot be fraudulently accused of producing a signature by the revocation manager.

Coalition resistance: a conspiring subset of group members cannot provide a legitimate signature that the group manager cannot link with one of the colluding group members.

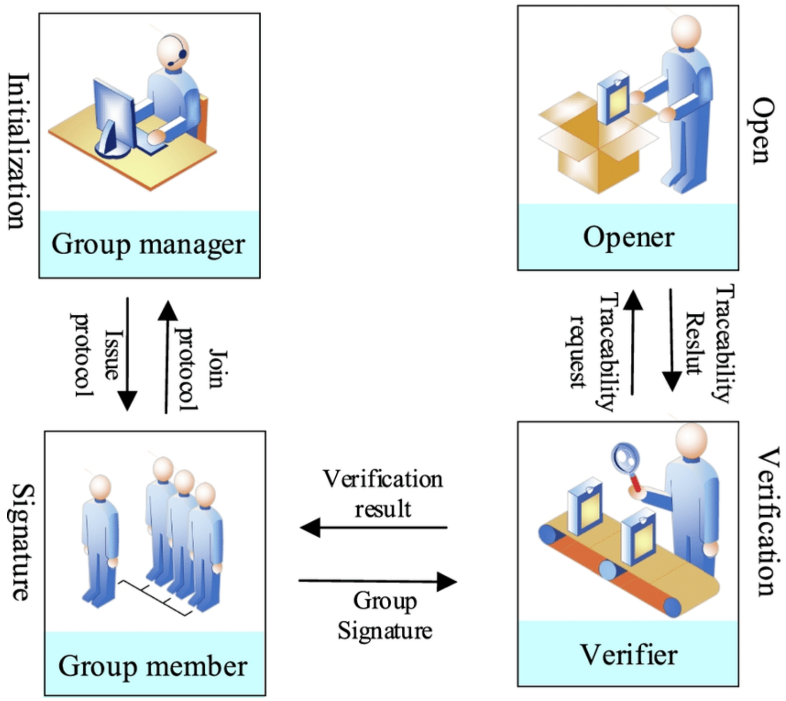


Figure 6.1. Group signature scheme. The opener is often the group manager (Huang et al. [2022]).

Group signature schemes may have a variety of applications, including anonymous e-voting systems, anonymous message boards, and privacy-preserving electronic commerce. They are an important tool for protecting the privacy of individuals in situations where it is necessary or desirable to allow a group to sign messages on behalf of the group, while still maintaining some degree of traceability and accountability.

Among of the most cutting-edge group signature schemes, there are the Ateniese et al. [2000] and Boneh et al. [2004] ones.

The Boneh et al. [2004] group signature based on bilinear groups is the one explored and analyzed for the purpose of managing a membership problem.

6.2 Introduction to the scheme

The purpose of this section is to sum up in the most compact way the procedures useful to build a group signature scheme, present in Boneh et al. [2004], and viewing them mainly from the mathematical point of view.

It begins with some preliminaries necessary to build the scheme, then it prosecutes with the algorithms of Key Generation, Sign, Verify, Open, Join and Revoke, after are presented the assumptions on which the scheme rely, and, in the end, some modifications are mentioned.

Preliminaries: Bilinear groups

Notation of Boneh et al. [2001].

1. G_1 and G_2 are two (multiplicative) cyclic groups of prime order p;
2. g_1 is a generator of G_1 and g_2 is a generator of G_2;
3. ψ is a computable one way isomorphism from G_2 to G_1, with $\psi(g_2) = g_1$; and
4. e is a computable map e: $G_1 \times G_2 \rightarrow G_T$ with the following properties:
    - Bilinearity: for all $u \in G_1$, $v \in G_2$ and $a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$.
    - Non-degeneracy: $e(g_1, g_2) \neq 1$.

The terminology is from Bellare et al. [2003].
Consider a bilinear group pair (G_1, G_2) with a computable isomorphism ψ, as in the previous section. Suppose further that the Strong Diffie-Hellman (SDH) assumption (Appendix A.1) holds on (G_1, G_2), and the Linear assumption (LA) (Appendix A.2) holds on G_1. The scheme employs a hash function $H : \{0,1\}^* \longrightarrow \mathbb{Z}_p$, treated as a random oracle in the proof of security.

KeyGen(n)

This randomized algorithm takes as input a parameter n, the number of members of the group, and proceeds as follows:

1. Select a generator g_2 in G_2 uniformly at random, and set $g_1 \longleftarrow \psi(g_2)$.
2. Select $h \xleftarrow{R} G_1 \backslash \{1_{G_1}\}$ and ζ_1, $\zeta_2 \xleftarrow{R} \mathbb{Z}_p^*$, and set $u, v \in G_1$ such that $u^{\zeta_1} = v^{\zeta_2} = h$.

3. Select $\gamma \xleftarrow{R} \mathbb{Z}_p^*$, and set $w = g_2^\gamma$.

4. Using γ, generate for each user i, $1 \leq i \leq n$, an SDH tuple (A_i, x_i): select $x_i \xleftarrow{R} \mathbb{Z}_p^*$, and set $A_i \longleftarrow g_1^{1/(\gamma+x_i)} \in G_1$.

The group public key is *gpk* $= (g_1, g_2, h, u, v, w)$. The private key of the group manager (the party able to trace signatures) is *gmsk* $= (\zeta_1, \zeta_2)$. Each user's private key is her tuple *gsk[i]* $= (A_i, x_i)$. No party is allowed to possess γ; it is only known to the private-key issuer.

Sign(gpk, gsk[i], M)

The protocol $Sign - Verify$ is a zero-knowledge proof based on SDH problem where the signer proves to the verifier the possession of a pair (A, x), where $A \in G_1$ and $x \in Z_p$, such that $A^{x+\gamma} = g_1$. Such a pair satisfies $e(A, wg_2^x) = e(g_1, g_2)$. This is the proof of knowledge of discrete logarithm in a group of prime order.

Given a group public key *gpk* $= (g_1, g_2, h, u, v, w)$, a user's key *gsk* $= (A, x)$, and a message $M \in \{0,1\}^*$, compute the signature as follows:

1. Compute the values $T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5$:

   The signer selects exponents $\alpha, \beta \xleftarrow{R} \mathbb{Z}_p^*$ and computes

$$T_1 \longleftarrow u^\alpha \quad T_2 \longleftarrow v^\beta \quad T_3 \longleftarrow Ah^{\alpha+\beta} \tag{6.1}$$

   Then also computes two helper values $\delta_1 \longleftarrow x\alpha$ and $\delta_2 \longleftarrow x\beta \in \mathbb{Z}_p^*$.

   Picks blinding values $r_\alpha, r_\beta, r_x, r_{\delta_1}$, and r_{δ_2} at random from $\mathbb{Z}_p$ and computes

$$\begin{aligned} &R_1 \longleftarrow u^{r_\alpha} \qquad\qquad R_2 \longleftarrow v^{r_\beta} \\ &R_3 \longleftarrow e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}} \\ &R_4 \longleftarrow T_1^{r_x} \cdot u^{-r_{\delta_1}} \qquad R_5 \longleftarrow T_2^{r_x} \cdot v^{-r_{\delta_2}} \end{aligned} \tag{6.2}$$

2. Compute a challenge c using the hash function as:

$$c \longleftarrow H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5) \in \mathbb{Z}_p. \tag{6.3}$$

3. Using c construct the values $s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2}$:

$$s_\alpha \longleftarrow r_\alpha + c\alpha \quad s_\beta \longleftarrow r_\beta + c\beta \quad s_x \longleftarrow r_x + cx \quad s_{\delta_1} \longleftarrow r_{\delta_1} + c\delta_1 \quad s_{\delta_2} \longleftarrow r_{\delta_2} + c\delta_2 \tag{6.4}$$

4. Output the signature σ, computed as $\sigma \longleftarrow (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta 2})$.

Verify(gpk, M, σ)

Given a group public key $gpk = (g_1, g_2, h, u, v, w)$, a message M, and a group signature σ, verify that σ is a valid signature as follows:

1. Re-derive R_1, R_2, R_3, R_4, and R_5:

$$\begin{aligned}
&\tilde{R}_1 \longleftarrow u^{s_\alpha} \cdot T_1^{-c} \qquad\qquad \tilde{R}_2 \longleftarrow v^{s_\beta} \cdot T_2^{-c} \\
&\tilde{R}_4 \longleftarrow T_1^{s_x} \cdot u^{-s_{\delta_1}} \qquad\quad \tilde{R}_5 \longleftarrow T_2^{s_x} \cdot v^{-s_{\delta_2}} \\
&\tilde{R}_3 \longleftarrow e(T_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} [e(T_3, w)/e(g_1, g_2)]^c
\end{aligned} \tag{6.5}$$

2. Check that these, along with the other first-round values included in σ, give the challenge c, i.e., that

$$c \stackrel{?}{=} H(M, T_1, T_2, T_3, \tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \tilde{R}_4, \tilde{R}_5) \tag{6.6}$$

accepts if this check succeeds and reject otherwise.

Open(gpk, gmsk, M, σ)

This algorithm is used for tracing a signature to a signer and can only be performed by the group manager.
It takes as input a group public key $gpk = (g_1, g_2, h, u, v, w)$ and the corresponding group manager's private key $gmsk = (\zeta_1, \zeta_2)$, together with a message M and a signature $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$ to trace, and proceeds as follows:

1. Verify that σ is a valid signature on M.
2. Consider the first three elements (T_1, T_2, T_3) as a Linear encryption (LE) (Appendix A.3), and recover the user's A as $A \longleftarrow T_3/(T_1^{\zeta_1} \cdot T_2^{\zeta_2})$.

If the group manager is given the elements $\{Ai\}$ of the user's private keys, he can look up the user index corresponding to the identity A recovered from the signature.

Join

In the paper Boneh et al. [2004] there is not an explicit protocol for the Join procedure, but it is implied in the generation of the keys. When a new member wants to join the group signature scheme, the group manager simply gives him a tuple (A_i, x_i) as happens in the point 4 of the KeyGen.

Revoke

It is now discussed how to revoke users.
Here, is described a revocation mechanism along the lines of Camenisch and Lysyanskaya [2002].
Another applicable revocation mechanisms for group signatures have been proposed in Ateniese et al. [2000].

Recall that the group's public key in this system is (g_1, g_2, h, u, v, w) where $w = g_2^{\gamma} \in G_2$ for random $\gamma \in \mathbb{Z}_p^*$ and random $h, u, v \in G_1$. User i's private key is a pair (A_i, x_i) where $A_i = g_1^{1/(\gamma+x_i)} \in G_1$.

Now, suppose we wish to revoke users $1, \ldots, r$ without affecting the signing capability of other users. To do so, the Revocation Authority (RA) publishes a Revocation List (RL) containing the private keys of all revoked users. More precisely, $RL = \{(A_1^*, x_1), ..., (A_r^*, x_r)\}$, where $A_i^* = g_2^{1/(\gamma+x_i)} \in G_2$. Note that $A_i = \psi(A_i^*)$. Here, the SDH secret γ is needed to compute the A_i^* 's. In case G_1 equals G_2 then $A_i = A_i^*$ and consequently the Revocation List can be derived directly from the private keys of revoked users without having to use γ. The list RL is given to all signers and verifiers in the system. It is used to update the group public key used to verify signatures.

Let $y = \prod_{i=1}^{r}(\gamma + x_i) \in \mathbb{Z}_p^*$. The new public key is $(\bar{g}_1, \bar{g}_2, h, u, v, \bar{w})$ where $\bar{g}_1 = g_1^{1/y}, \bar{g}_2 = g_2^{1/y}$, and $\bar{w} = (\bar{g}_2)^{\gamma}$. We show that, given RL, anyone can compute this new public key, and any unrevoked user can update her private key locally so that it is well-formed with respect to this new public key. Revoked users are unable to do so.

Updating the Group Public key

It is showed how to revoke one private key at a time. By repeating the process r times (as the revocation list grows over time) can be revoked all the private keys on the Revocation List. It is first showed how given the public key (g_1, g_2, h, u, v, w) and one revoked private key, $(A_1^*, x_1) \in RL$ anyone can construct the new public key $(\hat{g}_1, \hat{g}_2, h, u, v, \hat{w})$ where $\hat{g}_1 = g_1^{1/(\gamma+x_1)}$, $\hat{g}_2 = g_2^{1/(\gamma+x_1)}$, and $\hat{w} = (\hat{g}_2)^{\gamma}$. This new public key is constructed simply as:

$$\hat{g}_1 \longleftarrow \psi(A_1^*) \quad \hat{g}_2 \longleftarrow A_1^* \quad \textit{and} \quad \hat{w} \longleftarrow g_2 \cdot (A_1^*)^{-x_1}; \tag{6.7}$$

then $\hat{g}_1 = \psi(A_1^*) = g_1^{1/(\gamma+x_1)}$ *and* $\hat{w} = g_2 \cdot (A_1^*)^{-x_1} = g_2^{1-\frac{x_1}{\gamma+x_1}} = (A_1^*)^{\gamma} = \hat{g}_2^{\gamma}$, as required.

Updating the user's Private Key

Next, is showed how unrevoked users update their own private keys. Consider an unrevoked user whose private key is (A, x). Given a revoked private key, (A_1^*, x_1) the user computes $\hat{A} \longleftarrow \frac{\psi(A_1^*)^{1/(x-x_1)}}{A^{1/(x-x_1)}}$ and sets his new private key to be $(\hat{A}, x)$. Then, indeed,

$$(\hat{A})^{\gamma+x} = \frac{\psi(A_1^*)^{(\gamma+x)/(x-x_1)}}{A^{(\gamma+x)/(x-x_1)}} = \frac{\psi(A_1^*)^{[(\gamma+x_1)+(x-x_1)]/[x-x_1]}}{g_1^{1/(x-x_1)}} = \psi(A_1^*) = \hat{g}_1, \tag{6.8}$$

as required. Hence, $(\hat{A}, x)$ is a valid private key with respect to $(\hat{g}_1, \hat{g}_2, h, u, v, \hat{w})$. By repeating this process r times (once for each revoked key in RL) anyone can compute the updated public key $(\bar{g}_1, \bar{g}_2, h, u, v, \bar{w})$ defined above. Similarly, an unrevoked user with private key (A, x) can compute his updated private key $(\bar{A}, x)$ where $\bar{A} = (\bar{g}_1)^{1/(\gamma+x)}$.

6.3 A modification of Join protocol

If it is necessary the acquirement of the Strong Exculpability property, it can be followed the suggestion in the paper Boneh et al. [2004] that is using the protocol $JOIN$ present in Ateniese et al. [2000] that can be adapted to adjust the scheme. In the following there is a recap on what are the suggestions in BBS, some hint on how to build the protocol from Camenisch and Lysyanskaya [2004] (with link to similar protocols) and some hypothetical construction of the scheme.

6.3.1 Improvement in the group signature scheme BBS

Boneh, Boyen and Shacham presented a group signature scheme secure under the strong Diffie-Hellman and the Linear assumptions. They showed that, under these assumptions in groups with bilinear pairings, it is hard, on input $(g_1, g_2 = g_1^\gamma)$ to sample tuples of the form (A, x) where $A = g_1^{1/(\gamma+x)}$ (in other words, $A^{\gamma+x} = g_1$), even given a polynomial number of such samples. In their group signature scheme, such a tuple (A, x) is a user's group membership certificate, while (g_1, g_2) is the public key of the group. At the heart of their construction are (1) a zero-knowledge proof of knowledge of such a tuple; and (2) a scheme for encrypting x. They prove the resulting construction secure under a slightly weaker variant of the Bellare, Micciancio, and Warinschi definition of security (Bellare et al. [2003]).

The BBS group signature scheme can be extended to provide Strong Exculpability. Note that this does not prevent the group manager from generating group signatures using fraudulent signers (i.e., nonexistent group members). A closely related property is that of non framing (Chen and Pedersen [1995]); it captures the notion of a group member not being made responsible for a signature she did not produce. A stronger notion of exculpability (Strong Exculpability) is considered in Ateniese et al. [2000], where one requires that even the entity that issues user keys cannot forge signatures on behalf of users.

Boneh, Boyen, and Shacham modify their main group signature scheme to achieve exculpability, as follows. The public key of the group is augmented by an additional value h; it is now (g_1, g_2, h). The membership certificate of a group member is (A, x, y) such that $A^{\gamma+x} h^y = g_1$. This membership certificate is created via a protocol in which the group manager only learns the value h^y, but not the value y. The unforgeability of membership certificates in this modified scheme can be derived from that of their main scheme. They achieve exculpability because a proof of knowledge of a membership certificate requires the knowledge of the value y.

The ZKPK of the $Sign-Verify$ procedure of BBS can be modified to prove knowledge of such a triple. The resulting system is a short group signature with strong exculpability. For similar group signatures with the same mechanism, you can look for scheme A, or for a slight modification of schemes B and C described in Camenisch and Lysyanskaya [2004].

6.3.2 Hypothesis: JOIN protocol for BBS

After some research in the literature, not an explicit implementation of the ACJT JOIN protocol into the BBS group signature scheme was found. Because of that, here are hypothesized by the author of this document two different algorithms to implement this JOIN protocol in the BBS group signature scheme. These hypotheses are not officially reviewed, so they should not be trusted before the adequate verifications.

The first is simpler, while the second introduce the user i performing a zero-knowledge proof of knowledge of his secret y_i. The KeyGen algorithm is similar to the one described in the paper, with a difference at the 4^{th} step where this JOIN protocol is performed.

First hypothesis:

1. the GM selects at random $h_1 \xleftarrow{R} G_1$, thus adding an element in the public key: $gpk = (g_1, g_2, h, u, v, w, h_1)$;
2. the user i selects at random $y \xleftarrow{R} \mathbb{Z}_p^*$ and sends to the GM $Y = h_1^{-y_i}$;
3. the GM selects $x_i \xleftarrow{R} \mathbb{Z}_p^*$ and sends to the user i $\left((g_1 Y)^{\frac{1}{\gamma+x_i}} = A_i, x_i\right)$.

The new user's public key will be (A_i, x_i, y_i).
y_i is only known by the user, and is protected from being discovered by the GM (who only knows $Y = h_1^{-y_i}$) by the discrete logarithm problem.

Second hypothesis:

1. the GM selects at random $h_1 \xleftarrow{R} G_1$, thus adding a member in the public key: $gpk = (g_1, g_2, h, u, v, w, h_1)$;
2. the user i selects at random $y \xleftarrow{R} \mathbb{Z}_p^*$ and sends to the GM $Y = h_1^{-y_i}$;
3. the GM selects $x_i \xleftarrow{R} \mathbb{Z}_p^*$ and compute $(g_1 Y)^{\frac{1}{\gamma+x_i}} = A_i$, then sends to the user $h_1^{\frac{1}{\gamma+x_i}}$;
4. the user sends back $B_i = (h_1^{\frac{1}{\gamma+x_i}})^{-y_i}$;

5. the GM compute $A'_i = B_i(g_1^{\frac{1}{\gamma+x_i}})$ and verifies that is equal to A_i, if it's the case, user i proves to possess y_i;

6. the GM sends to the user (A_i, x_i).

The new user's public key will be (A_i, x_i, y_i). This is not the only modification to improve the group signature. As stated before, there should be a change of the $Sign - Verify$ procedure that is not analyzed in this document.

6.4 BBS04 as membership scheme

The way a group signature scheme is intended to solve the problem is that the set of all the signers can be seen as the set of the members of the group authorized to write a DID, and the private key of a group member is its certificate that he is a member. A member can prove his membership using a zero knowledge proof of his private key (signature) of the scheme. Group signatures also allow maintaining privacy while still being able to revoke a signing key in case a node present in the group is compromised.

The Boneh et al. [2004] scheme can begin with the keys' generation, an algorithm performed by the group manager that creates his own keys and creates and distributes the keys to all the members.

The Sign-Verify protocol can be used by a member to prove (and by another to verify) its membership to the group.

The algorithm Open, that can only be performed by the group manager, is used for tracing a signature to a signer. This protocol is not necessary to solve the intended problem, but it is an interesting feature that a group signature scheme can give.

The revocation mechanism is performed by a revocation authority (can be the group manager or someone different) that has the power to remove the membership of a compromised node. All the unrevoked members has to update their private key and the group public key (revoked members can't do this).

6.5 Critical review of the solution

A BBS group signature is a type of digital signature that allows a member of a group to sign a message on behalf of the group, while still providing a verifiable proof of the individual member's identity. This can be useful in a self-sovereign identity scheme, as it allows members to prove their membership in the group without revealing their individual identities. In the following is presented a list of advantages provided:

- The digital signature is intended for member of a group to sign a message on behalf of the group, this allows members to prove their membership signing a DID, so it solves the problem.

- Can be used in schemes that require anonymity of the individual identities. This is an interesting feature that can be extremely useful in some cases.

- Offers stronger security against attacks. BBS group signatures use a stronger cryptographic mechanism than PSK, which makes them more resistant to attacks.
- BBS group signature is a more decentralized method in respect to managing a PSK.

Despite these benefits, this method has also some drawbacks:

- The scheme needs a group manager that is a central authority, and having that in a decentralized environment is not the best scenario. In addition, the group manager knows every private key of every member, resulting in a single point of failure. This last problem exists also during the JOIN protocol, where a new member is given its private key (by the group manager) to join the scheme. This second problem can be overcome taking inspiration from the JOIN protocol presents in Ateniese et al. [2000] where the group manager knows only a part of the private keys of the members, the rest is only known by its owner.
- The Open algorithm can be a big resource, but can be also an issue. If the group manager is compromised, it can be a very serious problem, because it has the power to deanonymize all the signatures' authors.
- There is also a weakness regarding scalability. With thousands or even millions of users, if everyone has to update his private key and the group public key every time a user's private key is revoked, the system becomes slow and heavy. Another complication is due to the fact that all the nodes must be connected to have its keys updated.
  The revocation protocol of Boneh et al. [2004] is inspired by Camenisch and Lysyanskaya [2002]. In Camenisch and Lysyanskaya [2002] are studied the Dynamic Accumulators, that inspired the revocation protocol. Such a construction will be studied in the next chapter to find another solution to the membership problem.
- The security of the BBS group signature scheme relies on the hardness of the discrete logarithm problem, which is believed to be secure against classical computers but is vulnerable to attack by quantum computers (Shor [1997]). Therefore, the BBS group signature scheme is not post-quantum secure.

In general, as said for the previous method, this solution may be a suitable option for some applications, but it may not be the most practical or secure choice for others.

BBS is the group signature that is being analyzed for being one of the state of the art, but the author doesn't exclude that exist some other group signature that suits better the problem.

Chapter 7

Dynamic accumulators

In this chapter of the second part will be explored the third proposed solution, for the previously introduced problem, based on dynamic accumulators. In the beginning will be explained and explored the structure of dynamic accumulators, with a focus on dynamic accumulators without central authority. Then will be explained how these constructions will be useful to solve the membership problem. In the end, a critical review of this method is made.

7.1 Introduction

In 1993, Benaloh and de Mare [1994] proposed the idea of accumulator schemes as a decentralized replacement for digital signatures in the design of secure distributed protocols and outlined the fundamental functionalities and security characteristics that such schemes should offer to do away with the need for a trusted authority in applications like time-stamping and membership testing.

A simple definition of an accumulator scheme is a method that creates a single, short accumulator out of a huge collection of values, providing a quick proof that each value was included. The more difficult concept of dynamic accumulators, which allows for the dynamic addition and deletion of members from/to the initial set, was presented by Camenisch and Lysyanskaya [2002]. The proposed variant is particularly intriguing because it achieves such a higher degree of flexibility with a work per deletion and addition independent of the number of accumulated values, and because it does not require knowledge of any sensitive information to update old witnesses to be consistent with the new accumulator.

7.2 Dynamic accumulators

It's introduced the notion of a dynamic accumulator from Camenisch and Lysyanskaya [2002]. An accumulator scheme makes it possible to hash many inputs into a single short value, providing a quick proof that a given input was used to generate the result. Using a dynamic accumulator, one may add and remove values dynamically with a cost that is independent of the total sum of accumulated values. Here is the construction of a dynamic accumulator built using the definitions and algorithms from Camenisch and Lysyanskaya [2002].

Definition. A secure accumulator for a family of inputs $\{\chi_k\}$ is a family of families of functions $\mathcal{G} = \{\mathcal{F}_k\}$ with the following properties:
Efficient generation: There is an efficient probabilistic algorithm G that on input 1^k produces a random element f of $\mathcal{F}_k$. Moreover, along with f, G also outputs some auxiliary information about f, denoted t_f.
Efficient evaluation: $f \in \mathcal{F}_k$ is a polynomial-size circuit that, on input $(u, x) \in \mathcal{U}_f \times \mathcal{X}_k$, outputs a value $v \in \mathcal{U}_f$, where $\mathcal{U}_f$ is an efficiently-samplable input domain for the function f; and $\mathcal{X}_k$ is the intended input domain whose elements are to be accumulated.
Quasi-commutative: For all k, for all $f \in \mathcal{F}_k$, for all $u \in \mathcal{U}_f$, for all $x_1, x_2 \in \mathcal{X}_k$, $f(f(u, x_1), x_2) = f(f(u, x_2), x_1)$. If $X = x_1, ..., x_m \subset \mathcal{X}_k$, then by $f(u, X)$ we denote $f(f(...(u, x_1), ...), x_m)$.
Witnesses: Let $v \in \mathcal{U}_f$ and $x \in \mathcal{X}_k$. A value $w \in \mathcal{U}_f$ is called a witness for x in v under f if $v = f(w, x)$.
Security: Let $\mathcal{U'}_f \times \mathcal{X'}_k$ denote the domains for which the computational procedure for the function $f \in \mathcal{F}_k$ is defined (thus $\mathcal{U}_f \subseteq \mathcal{U'}_f$, $\mathcal{X}_k \subseteq \mathcal{X'}_k$). For all probabilistic polynomial-time adversaries $\mathcal{A}_k$,

$$\begin{aligned}&Pr[f \longleftarrow G(1^k);\ (x, w, X) \longleftarrow \mathcal{A}_k(f, U_f, u) :\\ &X \subset \chi_k;\ w \in \mathcal{U'}_f;\ x \in \mathcal{X'}_k;\ x \notin X;\ f(w, x) = f(u, X)] = neg(k).\end{aligned}$$

Note that only the legitimate accumulated values, $(x_1, ..., x_m)$, must belong to $\mathcal{X}_k$; the forged value x can belong to a possibly larger set $\mathcal{X'}_k$.

This is basically the definition of Barić and Pfitzmann [1997], with the difference that they do not require that the accumulator be quasi-commutative.

In this document, however, the interest is focused on a dynamic use where there is a manager controlling the accumulator, and several users. It can be showed now that dynamic addition of a value is done at unit cost in this setting, exhibiting how a user update his witness when a new member joins.

Let $f \in \mathcal{F}_k$. Let $v = f(u, X)$ be the accumulator so far. Let $v' = f\,(v, x') = f\,(u, X')$ be the value of the accumulator when x' is added to the accumulated set, $X' = X \cup \{x'\}$. Let $x \in X$ and w be the witness for x in v. The computation of w' which is the witness for x in v', is independent of the size of X.
w' is computed as follows: $w' = f\,(w, x')$. Let us show correctness using the quasi-commutative property: $f\,(w', x) = f\,(f\,(w, x')\,, x) = f\,(f(w, x), x') = f\,(v, x') = v'$.

We must also be able to handle dynamic deletions of a value from the accumulator.

Definition. A secure accumulator is dynamic if it has the following property:
Efficient deletion: there exist efficient algorithms D and W such that, if $v = f(u, X)$, x, $x' \in X$, and $f(w, x) = v$, then

1. $D(t_f, v, x') = v'$ such that $v' = f(u, X \setminus \{x'\})$; and
2. $W(f, v, v', x, x', w) = w'$ such that $f(w', x) = v'$.

Note that D is given the trap-door information t_f while W is not.
Finally, in the application desired, it is required that the accumulator allows for an efficient proof that a secret value given by some commitment is contained in a given accumulator value. That is, it is required that the accumulator be efficiently provable with respect to some commitment scheme ($Commit$).

Zero-knowledge proof of member knowledge: there exists an efficient zero-knowledge proof of knowledge system where the common inputs are c (where $c = Commit(x, r)$ with a r being a randomly chosen string), the accumulating function f and $v \in U_f$, and the prover's inputs are $(r, x \in \mathcal{X}_k, u \in \mathcal{U}_f)$ for proving knowledge of x, w, r such that $c = Commit(x, r)$ and $v = f(w, x)$.

7.3 Dynamic accumulator without central authority

In this section is presented a new definition for dynamic accumulator that is actually slightly different from that of Camenisch and Lysyanskaya [2002]: a few changes were made to attain a formalization more adherent to the original motivation of Benaloh and de Mare [1994], i.e., avoiding the need for a trusted central authority.

In fact, to meet the efficiency requirement for the element deletion algorithm **Del**, in Camenisch and Lysyanskaya [2002] is considered a scheme where the accumulator key generation algorithm **Gen** outputs, along with the accumulator key k, some secret information t_k that enables an efficient implementation of the Del algorithm, but at the same time opens a potential hole in the security of the scheme itself. Thus, the trapdoor t_k should only be available to an "accumulator manager", who is trusted to use this knowledge exclusively for the purpose of updating the accumulator after the removal of some elements, and not for deriving fake witnesses for values which have not been accumulated.

The following definition is from Fazio and Nicolosi [2003].

Definition (Dynamic Accumulator Scheme)
A dynamic accumulator scheme is a 7-tuple of polynomial time algorithms (Gen, Eval, Wit, Ver, Add, Del, Upd), where:

- **Gen**, the key generation algorithm, is a probabilistic algorithm used to set up the parameters of the accumulator. Gen takes as input a security parameter 1^λ and an accumulation threshold N (i.e., an upper bound on the total number of values that can be securely accumulated) and returns an accumulator key k from an appropriate key space $\mathcal{K}_{\lambda,N}$;
- **Eval**, the evaluation algorithm, is a probabilistic algorithm used to accumulate a set $L \doteq \{y_1, \ldots, y_{N'}\}$ of $N' \leq N$ elements from an efficiently-samplable domain Y_k,

where k is some accumulator key from $\mathcal{K}_{\lambda,N}$. **Eval** receives as input $(k, y_1, \ldots, y_{N'})$ and returns an accumulated value (or accumulator) $z \in Z_k$ and some auxiliary information aux, which will be used by other algorithms. Notice that every execution of **Eval** on the same input $(k, y_1, \ldots, y_{N'})$ must yield the same accumulated value z, whereas the auxiliary information aux can differ;

- **Wit**, the witness extraction algorithm, is a probabilistic algorithm that takes as input an accumulator key $k \in \mathcal{K}_{\lambda,N}$, a value $y_i \in Y_k$ and the auxiliary information aux previously output (along with the accumulator z) by Eval $(k, y_1, \ldots, y_{N'})$, and returns either a witness w_i (from an efficiently-samplable witness space $\mathcal{W}_k$) that "proves" that y_i was accumulated within z if this is indeed the case, or the special symbol $\perp$ if $y_i \notin \{y_1, \ldots, y_{N'}\}$.

- **Ver**, the verification algorithm, is a deterministic algorithm that, on input (k, y_i, w_i, z), returns a Yes/No answer according to whether the witness w_i constitutes a valid proof that y_i has been accumulated within z or not.

- **Add**, the element addition algorithm, is a (usually deterministic) algorithm that given an accumulator key k, a value $z \in Z_k$ obtained as the accumulation of some set L of less than N elements of Y_k, and another element $y' \in Y_k$, returns a new accumulator z' corresponding to the set $L \cup \{y'\}$, along with a witness $w' \in \mathcal{W}_k$ for y' and some update information aux $_{\text{Add}}$ which will be used by the **Upd** algorithm;

- **Del**, the element deletion algorithm, is a (usually deterministic) algorithm that given an accumulator key k, a value $z \in Z_k$ obtained as the accumulation of some set L of elements of Y_k, and an element $y' \in L$, returns a new accumulator z' corresponding to the set $L \backslash \{y'\}$, along with some update information aux Del which will be used by the **Upd** algorithm;

- **Upd**, the witness update algorithm, is a deterministic algorithm used to update the witness $w \in \mathcal{W}_k$ for an element $y \in Y_k$ previously accumulated within an accumulator $z \in Z_k$, after the addition (or deletion) of an element $y' \in Y_k \backslash \{y\}$ in (or from) z. Upd takes as input $(k, y, w, b,\ \text{aux}_{\text{op}})$ (where op is either **Add** or **Del**), and returns an updated witness w' that "proves" the presence of y within the updated accumulator z'.

This section was about a formal definition of an accumulator. That was not a theoretic construction without an effective use, the proof of that is presented in Fazio and Nicolosi [2003] where are showed some practical implementations.

Two different approaches are mentioned, one is based on (a variant of) a well-known number theoretic assumption, and the other based on families of functions with strong (pseudo-)random properties.

7.4 Applications: from time-stamping to membership testing

The original goal of Benaloh and deMare's study of cryptographic accumulators was to provide a primitive for the creation of distributed protocols that are space-efficient and do not require a trusted third party.

The first application that Benaloh and de Mare [1994] take into consideration is Time-Stamping (Haber and Stornetta [1991]), which is a protocol that allows a "publication" date to be added to any document in order to give an ordering criterion for determining the relative locations of any two documents. A straightforward solution may be reached in the presence of a reliable central authority C, by having C signing the collection of all papers created by all the system's m users during each round, at discrete time instants known as rounds.

Some improvements are considered in Fazio and Nicolosi [2003] and now presented. It is feasible to improve the above system by requesting active involvement from all participants who provided documents to be published, in order to decrease the amount of confidence to be put on the central authority. It can be demonstrated that this method still needs storage per user every round that is logarithmic in the total number of participants, even if it can do away with the necessity for a trusted third party. It turns out that an accumulator technique may be used to make this space overhead constant.

This is how the protocol would work.
At round t, a new accumulator key k_t is generated, and each of the m participants encodes the messages he/she wishes to publish as an element $y_{t,i}$ of the input domain $Y_{k_t}, i = 1, \ldots, m$. All the $y_{t,i}$'s values are then accumulated together, computing $(z_t, \text{aux}_t) \doteq \text{Eval}\,(k_t, y_{t,1}, \ldots, y_{t,m})$, and the participants store the resulting accumulator value z_t, along with the witness $w_{t,i} \doteq \text{Wit}\,(k_t, w_{t,i}, \text{aux}_t)$ for their own value $y_{t,i}$. In this way, to later show that a given document was time-stamped at the round t corresponding to the accumulator value z_t, the user i just needs to show that such document is encoded within the value $y_{t,i}$, and then provide the witness $w_{t,i}$ to prove that $y_{t,i}$ is indeed one of the values accumulated within z_t.

The prior architecture may be easily modified to produce membership testing. Each group essentially represents a round of the timestamping protocol, with the group members functioning as the time-stamped documents. Additionally, each member can demonstrate to non-members that he or she is a part of the group without having to reveal the full list of members if the accumulated value z, which can be thought of as a very compact representation of the membership list, is made available to users outside the group.

7.5 Membership testing scheme

A new group that needs a membership test scheme has to follow the protocol proposed in Fazio and Nicolosi [2003], that it is here schematized from the preceding section:

- A new accumulator key k is generated, and each of the m members encodes his certificate as an element y_i of the input domain Y_k, $i = 1, \ldots, m$.

- All the y_i 's values are then accumulated together, computing $(z, \text{aux}) \doteq \text{Eval}\,(k, y_1, \ldots, y_m)$.
- The participants store the resulting accumulator value z, along with the witness $w_i \doteq \text{Wit}\,(k, w_i, \text{aux})$ for their own value y_i.
- Every time n members have to be added or deleted from the accumulator, iterate the first three points with the new number of members $m + n$ or $m - n$.

7.5.1 Add and delete functionality: decentralized version

For the purpose of this thesis, starting from the previous idea of Fazio and Nicolosi [2003], are here proposed some additions to the scheme by the author of this work.

To overcome the necessity to re-compute a new accumulator every time a new member is added or deleted, a new method that includes the functionality of addition and deletion of the accumulator is explored.

The addition or deletion of new members comes in a decentralized way. The members of the group have to participate in a consensus algorithm to decide if a member has the right to be added to the group or if a member has to be deleted from the group. After the running of such algorithm, every member has to run the **Add** or **Del** algorithm to update z and **Upd** to update every w_i.

To better clarify the procedure for the reader, an example of an algorithm to formalize the previous process of addition and deletion is provided.

The Addition algorithm:

- A member asks to be added to the group.
- Every member participate in the consensus algorithm.
- The consensus algorithm output a **Yes/No** depending on every member's vote or the type of algorithm.
- If it is a **Yes**, every member runs the **Add** algorithm, including the new member certificate into the accumulator, thus rendering him part of the group. If it is a **No**, nothing happens.

The Deletion algorithm:

- A member proposes to the group to expel another member.
- Every member participate in the consensus algorithm.
- The consensus algorithm output a **Yes/No** depending on every member's vote or the type of algorithm.
- If it is a **Yes**, every member runs the **Del** algorithm, excluding the old member certificate from the accumulator, thus getting him out of the group. If it is a **No**, nothing happens.

The most trivial example of a consensus algorithm might be of majority voting. But the choice of the specific algorithm is crucial, and it depends on the context of the IoT devices involved, how much voting power it is right to give to every device, if every device has the same voting power and many other considerations. However, the literature is plenty of different consensus algorithms that can be chosen depending on the specific needs.

7.5.2 Ownership of the membership certificate

Again, this section is designed by the author for the purpose of this work.

A problem with these schemes is that the membership certificates y_i (and its witnesses w_i) are public, so everyone can claim the ownership of one certificate and consequently the inclusion in the group.

A solution can be that during the **Eval** procedure, another protocol should be considered such that every member can put his y_i as input without revealing it to anyone. Hiding y_i allows only the owner of the certificate to compute his own w_i.

Another solution can be that y_i is the (encoded) public key of a tuple of asymmetric keys, such that the real owner of the y_i can prove it, using zero knowledge proof by knowing the related private key.

This means that in the verification phase, as well as the verification algorithm, the verifier has to check that the member is able to decrypt (with his private key) a message encrypted with the certificate y_i (which is the public key).

7.6 Critical review of the solution

In this chapter, two different types of dynamic accumulator were presented. One built by Camenisch and Lysyanskaya [2002] that were the ones who first proposed it. The second, instead, a modification of the first, was built by Fazio and Nicolosi [2003], and seems more promising for our aim, mainly because is made such that there is no need for a central authority.

In a nutshell, in a self-sovereign identity scheme, a dynamic accumulator is a data structure that can be used to prove membership in a set of items without revealing the individual items themselves. This is achieved by allowing a user to append new items to the accumulator and then generating a proof that can be used to verify the membership of a specific item in the set. It provides many useful advantages to the proof of membership scheme:

- One advantage of using a dynamic accumulator for proof of membership is that it is relatively efficient, as it allows a user to prove membership in a large set of items with a relatively small proof.

- It is well-suited for use in situations where it is important to maintain the privacy of the accumulated items (certificates) in the set, as the proof does not reveal any information about the specific items themselves.

- The flexibility of the numeric construction it is also a useful feature. Provided the structure, many numeric constructions that satisfy the structure can be built.
- Scalability: dynamic accumulators can support large sets of data and numerous users without requiring excessive computational resources. This can help self-sovereign identity systems scale to meet the needs of numerous users.
- Decentralization: with the Fazio and Nicolosi [2003] version, the dynamic accumulator lacks of a central authority and that makes this method potentially decentralized in its entirety. The best thing in a self sovereign identity scheme.
- Quantum-safe? Despite not being present in the suggested constructions in Fazio and Nicolosi [2003], there exist some dynamic accumulators constructions that are based on problems not easily solvable using quantum computing.

However, there are also some potential drawbacks to using a dynamic accumulator for proof of membership.

- If the accumulator is compromised, it could potentially reveal the membership of all items in the set, which could compromise the privacy of the users.
- Vulnerability to attacks: while dynamic accumulators can provide improved security in respect to a PSK, they are not immune to attacks. For example, a malicious actor could attempt to compromise the system by introducing false data into the accumulator.
- Scalability: every time a new element is accumulated, every previously computed witness of any member has to be updated, resulting in the members being forced to be connected to check if they have to update their values.
- There is not an already created suitable construction for the problem researched. To achieve a satisfying scheme, many works can be combined and modified to reach the best solution, starting from that of Fazio and Nicolosi [2003].

This was the last proposed method, interesting mainly for its decentralization property.

Chapter 8

Conclusions

Now, the conclusion of this research work should be the answer to the question: “what is the best method for a proof of membership in a self sovereign identity scheme between IoT nodes?". But the answer that comes up from this work is that there is no one-size-fits-all answer to this question, as the best method for proving membership in a self-sovereign identity scheme will depend on the specific requirements and goals of the scheme.

This was a new research that tried to adapt some existing constructions present in literature to the presented problem. It was not an easy research since after the in-depth study of a new process, the author of this digression had to find a way to innovate the method, to reshape it and convert it to an eligible solution.

It’s important to carefully evaluate the security, efficiency, and usability of the chosen method to ensure that it meets the needs of the scheme.

Summing up, the Merkle tree method is the most scalable and dynamic, because unlike the others’ methods, it doesn’t need to update every member certificate every time a new member is added or revoked. But it is the less flexible regarding the DID management.

The method using dynamic accumulators is certainly the most decentralized, because in the version of Fazio and Nicolosi [2003] it doesn’t need a central authority to work, and that is a key advantage. But it needs a consensus algorithm to work, that can be not so efficient to run, and also needs to update every member’s certificate during an addition or a revoke.

The BBS method is a middle ground in terms of functionality: it is less decentralized than the dynamic accumulator method, but needs to update the certificate only for a revoke. So it is less scalable than the Merkle tree method, but possess the interesting feature of anonymity of the group signatures.

In the followings there is a table (8.1) where are summarized some characteristics of the different methods, comparing them also with the PSK method that is considered the benchmark. For an in-depth analysis, please refer to the previous chapter.

One last notable aspect that need to be considered is that we are going through a period where quantum computers will rewrite some aspects of the modern cryptography.

Particularly, according to Nakov [2018], quantum computing is a paradigm of computation based on quantum mechanics. It operates differently from classical computers and is capable of tasks that conventional computers are unable to perform. But quantum computers cannot do any computing task more quickly and are not "faster computers" or all-powerful. For some tasks, quantum computers are quite effective, whereas for others, they are relatively ineffective.

It is commonly known in computer science that various cryptographic systems, such as RSA, ECC, and ECDSA, which rely on the IFP (integer factorization issue), the DLP (discrete logarithm's problem), and the ECDLP (elliptic-curve discrete logarithm problem), will be broken by quantum computers.

Despite these considerations, the advent of quantum algorithms won't spell the end of cryptography because only some cryptosystems are quantum-unsafe (RSA, DHKE, ECC, ECDSA and ECDH). Some cryptosystems will barely be impacted since they are quantum-safe (like cryptographic hashes, MAC, algorithms and symmetric key ciphers).

Said that, and contextualized in our digression, some presented method could have some problem with quantum computing. The Merkle tree method should not be affected, because hashing will resist this revolution. But BBS group signature results weak for it being founded on the discrete logarithm problem, so, to overcome this problem, it needs to be modified relying on some different problem or, instead, could be used a different group signature with similar characteristic but quantum safe. Also, dynamic accumulators have to be constructed with a base problem that is quantum safe.

The research does not end with this work, so here are presented three different paths to traverse depending on the specific requisite that the specific self sovereign identity scheme need to pursue. In fact, some methods explored are already built and ready to be used, other needs more or less modification to be suitable, all should be analyzed finding the best solution for a specific use case.

This is also to say that some features have not been addressed in depth, even though they are fundamental, such as security or the complexity of the suggested methods. These features can be analyzed once the best method has been chosen in a specific case, which will then be explored in its entirety.

In the end, this work should be considered a starting point in finding a solution (or many, it depends on the future researches) for a proof of membership in a self sovereign identity scheme between IoT nodes.

8.1 Potential Evolutions

As said before, the aim of this research is to try to do the first steps in the direction of a solution for the presented proof of membership problem for dynamic groups in a self sovereign identity scheme.

All of these introduced methods were not reviewed, so the next step in the direction of this research could be to try to formalize one (or more) of the presented methods to create a new protocol that can be finally reviewed. Some scientific papers could also be produced in the process.

Then, this can end up in trying to build the first implementation of the method to try it practically and check if it is really suitable or arise new issue that weren't seen theoretically.

Property \ *Method*	*PSK (benchmark)*	*Merkle Trees*	*BBS Group Signature*	*Dynamic Accumulators*
Decentralization	Low	Medium	Medium	High
Scalability	Low	High	Medium	Low
Single Point of Failure	Yes (every node)	Yes (only the GM)	Yes (only the GM)	No
Anonymity features	No	No	Yes	No
DID flexibility	High	Low	High	High
Modifications	No	Some	Few	Some
Quantum Safe	Yes	Yes	With Adjustments	With Adjustments

Table 8.1. Pros and Cons of proposed methods

Appendix A

Assumptions and Definitions

Here are some assumptions and definition from Boneh et al. [2004], that are useful in understanding what the group signature is based on.

A.1 The Strong Diffie-Hellman Assumption

q-Strong Diffie-Hellman Problem. The q-SDH problem in (G_1, G_2) is defined as follows: given a $(q+2)$-tuple $(g_1, g_2, g_2^{\gamma}, g_2^{\gamma^2}, ..., g_2^{\gamma^q})$ as input, output a pair $(g_1^{1/(\gamma+x)}, x)$ where $x \in Z_p^*$, g_1 is a generator of G_1, g_2 is a generator of G_2 and $g_1 \longleftarrow \psi(g2)$.

Definition. We say that the (q, t, ϵ)-SDH assumption holds in (G_1, G_2) if no t-time algorithm has advantage at least ϵ in solving the q-SDH problem in (G_1, G_2).

A.2 Decision Linear Assumption

Decision Linear Problem in G_1. Given $u, v, h, u^a, v^b, h^c \in G_1$ as input, output **yes** if $a + b = c$ and **no** otherwise.

Definition. We say that the (t, ϵ)-Decision Linear Assumption (LA) holds in G_1 if no t-time algorithm has advantage at least ϵ in solving the Decision Linear problem in G_1.

A.3 Linear Encryption

The Decision Linear problem gives rise to the Linear encryption (LE) scheme, a natural extension of ElGamal encryption. Unlike ElGamal encryption, Linear encryption can be secure even in groups where a DDH-deciding algorithm exists. In this scheme, a user's public key is a triple of generators $u, v, h \in G_1$; her private key is the exponents $x, y \in Z_p$ such that $u^x = v^y = h$. To encrypt a message $M \in G_1$, choose random values $a, b \in Z_p$, and output the triple $(u^a, v^b, M \cdot h^{a+b})$. To recover the message from an encryption (T_1, T_2, T_3), the user computes $T_3/(T_1^x \cdot T_2^y)$. By a natural extension of the proof of security of ElGamal, LE is semantically secure against a chosen-plaintext attack, assuming Decision-LA holds.

A.4 Strong Exculpability

In Bellare et al. [2003], exculpability is informally defined as follows: no member of the group and not even the group manager — the entity that is given the tracing key— can produce signatures on behalf of other users. Thus, no user can be framed for producing a signature he did not produce. They argue that a group signature, secure in the sense of full-traceability, also has the exculpability property. Thus, in the terminology of Bellare et al. [2003], our group signature has the exculpability property.

A stronger notion of exculpability is considered in Ateniese et al. [2000], where one requires that even the entity that issues user keys cannot forge signatures on behalf of users.

Bibliography

Mohamed Abomhara and Geir M Køien. Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security and Mobility*, pages 65–88, 2015.

Olivier Alphand, Michele Amoretti, Timothy Claeys, Simone Dall 'Asta, Andrzej Duda, Gianluigi Ferrari, Franck Rousseau, Bernard Tourancheau, Luca Veltri, and Francesco Zanichelli. IoTChain: A Blockchain Security Architecture for the Internet of Things. In *IEEE Wireless Communications and Networking Conference*, Barcelona, Spain, April 2018. URL `https://hal.archives-ouvertes.fr/hal-01705455`.

Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2000. doi: 10.1007/3-540-44598-6_16. URL `https://www.iacr.org/archive/crypto2000/18800256/18800256.pdf`.

Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In Walter Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, pages 480–494, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg. ISBN 978-3-540-69053-5.

Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, pages 614–629, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-39200-2.

Josh Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital signatures. In Tor Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, pages 274–285, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg. ISBN 978-3-540-48285-7.

Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 514–532, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-45682-7.

Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 41–55, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-28628-8.

Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, pages 61–76, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45708-4.

Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 56–72, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-28628-8.

Alberto Carelli, Andrea Palmieri, Antonio Vilei, Fabien Castanier, and Andrea Vesco. Enabling secure data exchange through the iota tangle for iot constrained devices. *Sensors*, 22(4), 2022. ISSN 1424-8220. doi: 10.3390/s22041384. URL `https://www.mdpi.com/1424-8220/22/4/1384`.

David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, pages 257–265, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg. ISBN 978-3-540-46416-7.

L. Chen and T. P. Pedersen. New group signature schemes. In Alfredo De Santis, editor, *Advances in Cryptology — EUROCRYPT'94*, pages 171–181, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. ISBN 978-3-540-44717-7.

Ben Dickson. How blockchain can change the future of iot., 2016. URL `https://venturebeat.com/business/how-blockchain-can-change-the-future-of-iot/`.

Nomusa Nomhle Dlamini and Kevin Allan Johnston. The use, benefits and challenges of using the internet of things (iot) in retail businesses: A literature review. *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pages 430–436, 2016.

David Eckhoff and Isabel Wagner. Privacy in the smart city—applications, technologies, challenges, and solutions. *IEEE Communications Surveys & Tutorials*, 20:489–516, 2018.

ETSI. The european telecommunications standards institute. 2013. URL `http://www.etsi.org/`.

Nelly Fazio and A Nicolosi. Cryptographic accumulators: Definitions, constructions and applications. 01 2003.

IOTA Foundation. Iota wiki. the complete reference for iota., 2022a. URL `https://wiki.iota.org/`.

IOTA Foundation. mam.js, 2022b. URL `https://github.com/iotaledger/mam.js`.

IOTA Foundation. Iota streams, 2022c. URL `https://www.iota.org/solutions/streams`.

Ammar Gharaibeh, Mohammad A. Salahuddin, Sayed Jahed Hussini, Abdallah Khreishah, Issa Khalil, Mohsen Guizani, and Ala Al-Fuqaha. Smart cities: A survey on data management, security, and enabling technologies. *IEEE Communications Surveys and Tutorials*, 19(4):2456–2501, October 2017. ISSN 1553-877X. doi: 10.1109/COMST.2017.2736886. Publisher Copyright: © 1998-2012 IEEE.

Alexander S. Gillis. What is the internet of things (iot)?, 2022. URL `https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT`.

Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *J. Cryptology*, 3:99–111, 1991. doi: 10.1007/BF00196791.

Vikas Hassija, Vinay Chamola, Vikas Saxena, Divyansh Jain, Pranav Goyal, and Biplab Sikdar. A survey on iot security: Application areas, security threats, and solution architectures. *IEEE Access*, PP:1–1, 06 2019. doi: 10.1109/ACCESS.2019.2924045.

Daojing He, Sammy Chan, and Mohsen Guizani. Security in the internet of things supported by mobile edge computing. *IEEE Communications Magazine*, 56(8):56–61, August 2018. ISSN 0163-6804. doi: 10.1109/MCOM.2018.1701132.

Jing Huang, Hui-Juan Zhang, Shen He, Jia Chen, and Zhe-Yuan Sun. A remote attestation mechanism using group signature for the perception layer in centralized networking. *EURASIP Journal on Wireless Communications and Networking*, 2022, 02 2022. doi: 10.1186/s13638-022-02092-9.

IERC. Coordinating and building a broadly based consensus on the ways to realise the internet of things in europe. 2013. URL `http://www.internet-of-things-research.eu/pdf/Poster_IERC_A0_V01.pdf`.

Arun Cyril Jose and Reza Malekian. Improving smart home security: Integrating logical sensing into smart home. *IEEE Sensors Journal*, 17:4269–4286, 2017.

Joseph Kizza. *Guide to Computer Network Security*. 01 2017. ISBN 978-3-319-55605-5. doi: 10.1007/978-3-319-55606-2.

Shancang Li, Li Xu, and Shanshan Zhao. The internet of things: A survey. *Information Systems Frontiers*, 17, 04 2014. doi: 10.1007/s10796-014-9492-7.

Knud Lasse Lueth. Top 10 iot application areas, 2020. URL `https://iot-analytics.com/top-10-iot-applications-in-2020/`.

MarketsandMarkets. Internet of things (iot) market by software solution (real-time streaming analytics, security solution, data management, remote monitoring, and network bandwidth management), service, platform, application area, and region - global forecast to 2026, 2022. URL `https://www.marketsandmarkets.com/Market-Reports/internet-of-things-market-573.html`.

Ralph C. Merkle. Method of providing digital signatures, 1979. URL `https://patents.google.com/patent/US4309569A/en?oq=US4309569A`. US4309569A.

Dennis Miller. Blockchain and the internet of things in the industrial sector. *IT Professional*, 20(3):15–18, 2018. doi: 10.1109/MITP.2018.032501742.

Svetlin Nakov. Quantum-safe cryptography, 2018. URL `https://cryptobook.nakov.com/quantum-safe-cryptography`.

Vinod Namboodiri, Visvakumar Aravinthan, Surya Narayan Mohapatra, Babak Karimi, and Ward Jewell. Toward a secure wireless-based home area network for metering in smart grids. *IEEE Systems Journal*, 8:509–520, 2014.

Oodles. Will iota blockchain solution secure internet of things ecosystem? URL `https://blockchain.oodles.io/blog/blockchain-solution-iota-iot-security/`.

Belavadi Prahalad. Merkle proofs explained., 2018. URL `https://medium.com/crypto-0-nite/merkle-proofs-explained-6dd429623dc5`.

Security Raphaël Dereymez. Securing the iot: a real challenge, 2018. URL `https://www.orange-business.com/en/blogs/securing-iot-real-challenge`.

Subhadeep Sarkar and Sudip Misra. Theoretical modelling of fog computing: a green computing paradigm to support iot applications. *IET Networks*, 5(2):23–29, 2016. doi: https://doi.org/10.1049/iet-net.2015.0034. URL `https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-net.2015.0034`.

Bruce Schneier. *Secrets &amp; Lies: Digital Security in a Networked World*. John Wiley &amp; Sons, Inc., USA, 1st edition, 2000. ISBN 0471253111.

Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, oct 1997. doi: 10.1137/s0097539795293172. URL `https://doi.org/10.1137%2Fs0097539795293172`.

Vivek Singhania. The internet of things: An overview understanding the issues and challenges of a more connected world. *The Internet Society (ISOC)*, 2015.

Mukesh Taneja. An analytics framework to detect compromised iot devices using mobility behavior. *2013 International Conference on ICT Convergence (ICTC)*, pages 38–43, 2013.

ToIP. Toip foundation whitepaper, 2020. URL `https://trustoverip.github.io/WP0010-toip-foundation-whitepaper/`.

W3C. World wide web consortium. URL `https://www.w3.org/`.

W3C. Decentralized identifiers (dids) v1.0, 2022a. URL `https://www.w3.org/TR/did-core/`.

W3C. Verifiable credentials data model v1.1, 2022b. URL https://www.w3.org/TR/vc-data-model/#what-is-a-verifiable-credential.

IOTA Wiki. The tangle, 2022. URL https://wiki.iota.org/learn/about-iota/tangle.

Wikipedia. Merkle tree, 2022a. URL https://en.wikipedia.org/wiki/Merkle_tree.

Wikipedia. Group signature, 2022b. URL https://en.wikipedia.org/wiki/Group_signature.

Xiaofang Xia, Yang Xiao, and Wei Liang. Absi: An adaptive binary splitting algorithm for malicious meter inspection in smart grid. *IEEE Transactions on Information Forensics and Security*, 14:445–458, 2019.