Industry-Grade Self-Sovereign Identity

On the Realisation of a Fully Distributed Self-Sovereign Identity Architecture

TU Delft

R.M. Chotkan

DELFT UNIVERSITY OF TECHNOLOGY

MINISTRY OF THE INTERIOR AND KINGDOM RELATIONS

TO OBTAIN THE DEGREE OF MASTER OF SCIENCE
AT THE DELFT UNIVERSITY OF TECHNOLOGY,
TO BE DEFENDED PUBLICLY ON MONDAY AUGUST 30, 2021 AT 10:00 AM.

Industry-Grade Self-Sovereign Identity

Author
R.M. CHOTKAN

Supervisors
Dr. J.A. POUWELSE, TU DELFT
A. DE KOK, RVIG

Student number:	4570243	
Project duration:	November 9, 2020 – August 30, 2021	
Thesis committee:	Dr. ir. J.A. Pouwelse,	TU Delft, supervisor
	Dr. ir. F.A. Kuipers,	TU Delft

This thesis is confidential and cannot be made public until August 31, 2021.

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Contents

Abstract

This research has been performed in pursuit of the MSc Computer Science at Delft University of Technology in collaboration with the Dutch National Office for Identity Data (RvIG), part of the Dutch Ministry of the Interior and Kingdom Relations.

Self-Sovereign Identity (SSI) is a relatively new concept part of a movement aspiring to create a universal identity layer for the Internet. SSI aims to put the citizen at the centre of their data, making them the sovereign over their digital presence. Wherein the current ecosystem personal information is stored in centralised or federated settings, SSI delegates this responsibility entirely to the user.

Functioning SSI schemes have been proposed and deployed, even with governmental support. However, we identify that the key issue that remains to be solved is revocation: the invalidation of credentials. Proposed revocation mechanisms typically rely on centralised infrastructure for revocations, defying the principles of SSI itself and, furthermore, lack offline verification capabilities.

This research addresses these issues and proposes the first fully distributed revocation mechanism in SSI, using a gossip-based propagation algorithm. Our revocation mechanism requires no centralised infrastructure or strict network requirements and enables offline verification of credentials in case of disaster. Propagation is handled by honest clients, requires no direct communication with authorities and is shown to be robust in case of unreliable communication links. Furthermore, revocation acceptance is at the discretion of individual clients, making our mechanism fully adhere to the principles of Self-Sovereignty.

This revocation and verification structure is part of our Industry-Grade Self-Sovereign Identity (IG-SSI) architecture. IG-SSI is a purely academic fully distributed SSI scheme with intrinsic equality across the network. Furthermore, communication is facilitated peer-to-peer, requiring no specialised infrastructure. The architecture allows for the signing, verification and presentation of credentials using Zero-Knowledge Proofs. We believe that the characteristics of our system provide it with use for decades to come, hence, we deem it to be *industry-grade*.

Our simulation portrays that a network comprised of 10,000 clients gossips 1 million revocations within 25 seconds. Feasibility on smartphones is shown through a government-backed real-life trial. Based on our results, we claim that IG-SSI is a viable candidate for facilitating the needs for a digital identity of the European Union.

I

Article

Distributed Attestation Revocation in Self-Sovereign Identity

R.M. Chotkan and J.A. Pouwelse
R.M.Chotkan@student.tudelft.nl, J.A.Pouwelse@tudelft.nl

***Abstract*— Current digital identities can be revoked at the discretion of platform owners. The Self-Sovereign Identity (SSI) concept is part of a movement to create a standardised identity layer for the Internet. SSI has the ability to remove the grip of big tech on digital identities and to place the citizen at the centre of their data. However, millions of physical identities are lost annually and must be revoked, as such, digital identities and credentials must also be able to be revoked. Our research addresses the key issue of revocation in SSI systems. We believe that revocation is hampering the uprise of SSI: existing attempts critically rely on communication with central authorities and introduce inequalities into the architecture. Such architectures violate the principles of Self-Sovereign Identity itself. We present a fully distributed SSI architecture with the first fully distributed SSI revocation mechanism requiring no specialised nodes, in which equality and offline usability are at the core. A novel gossip-based propagation algorithm propagates revocations throughout the network, enabling offline verification. Simulations portray that a million revocations are able to propagate in a system of up to 10 thousand clients in under 25 seconds, all whilst voided of any centralised infrastructure. Furthermore, the resulting architecture allows for attestation signing, presentation and verification using Zero-Knowledge Proofs. Our results show improvements with respect to the state of the art. We claim that our architecture is a viable candidate for the upcoming European-wide identity standard. Our small-scale trial shows that this is a promising direction to further explore.**

I. Introduction

The European Union has announced to provide each European citizen with a trusted and secure digital identity [1]. This objective is further fuelled by the urgency of COVID-19 vaccination passports [2]. The majority of current digital identities are held by big tech [3], resulting in big privacy issues for the citizen's digital presence [4]. Furthermore, these digital identities can be revoked at the platform owner's discretion [5], resulting in the loss of access to a plethora of other connected services.

The *Self-Sovereign Identity* (SSI) concept can prove to overcome these digital and societal issues. The Internet has no native method for knowing with whom you communicate [6]. As such, the SSI movement aims to create a standardised identity layer for the Internet, generating digital trust through verifiable identities and putting the citizen at the centre of their data. Relevant principles and architectures of SSI have been laid out [7, 8], however, an often overlooked issue is that of theft, loss, and data breaches. For the past five years in the USA, more than a million data breaches have occurred annually [9], resulting in the loss of billions of credentials [10]. Furthermore, 0.8% of UK passports are lost annually [11]. Revocation of these credentials is required to minimise further consequences. However, *revocation* remains a key issue in Self-Sovereign Identity. As portrayed by Table I (further discussed in section III), distributed revocation in SSI is to our knowledge yet to be solved. As such, we believe that revocation is hampering the mass deployment of Self-Sovereign Identities. Existing SSI and digital identity solutions such as Sovrin[1], Veramo[2] (formerly known as uPort) and Irma[3] violate the principles of SSI itself by solving the issue of revocation through the introduction of authorities. The cardinal requirement for SSI is an authoritarian-free ecosystem. Furthermore, recent natural disasters portray that the reliance on always available digital infrastructure is a fatal prerequisite [12]. Digital identities should be disaster-proof. Dependence on authorities for verification disallows offline usability and, moreover, introduces inherent inequalities in the network. This may lead to censorship or privacy issues [13].

This research introduces an academic Self-Sovereign Identity architecture focusing on distributed revocation of attestations, offline verification, and intrinsic equality of clients across the network. The scheme is based on the previous works by [14, 15]. The following contributions are made: (1) the first fully distributed revocation mechanism for SSI, achieving reliable revocation over unreliable communication links and (2) offline verification of credentials. Furthermore, a reference implementation of the architecture is created using the IPv8 protocol stack [16, 17] as well as a proof-of-concept application portraying the usability of our SSI architecture and distributed revocation on smartphones. These implementations of the architecture have been validated in a small-scale trial.

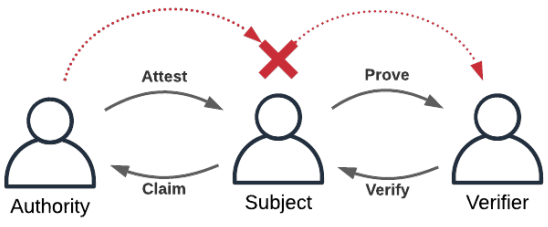


Fig. 1: Interactions in Self-Sovereign Identity

[1]For Sovrin, see: https://sovrin.org/
[2]For Veramo, see: https://veramo.io/
[3]For Irma, see: https://irma.app/?lang=en

TABLE I: Revocation comparison with related works

	Domain	Type	Maturity[1]	Description	No network operators	Offline availability	No authority interactivity	Offline verification	No SPOF	No fragility[2]
Our work (section V)	SSI	Attestation	✓	First fully distributed SSI revocation mechanism.	✓	✓	✓	✓	✓	✓
Xu et al. [18]	SSI	Node	✗	List of accepted nodes stored on blockchain.	✗	✓	✓	✗	✓	✓
Abraham et al. [19]	SSI	Attestation	✓	Revocations stored on public permissioned blockchain.	✗	✓	✓	✓	✓	✓
Lasla et al. [20]	C-ITS	Node	✗	Revocations stored on blockchain and RSUs.	✗	✓	✓	✗	✓	✓
Popescu et al. [21]	DS	Certificate	✗	Revocations handled locally by authority.	✓	✗	✗	✓	✗	✗
Liau et al. [22]	P2P	Certificate	✗	Uses distribution points and P2P communication.	✗	✓	✗	✓	✓	✓
Haas et al. [23]	VANET	Certificate	✗	RSUs and v2v propagation.	✗	✓	✓	✓	✓	✗
Laberteaux et al. [24]	VANET	Certificate	✗	RSUs and v2v propagation.	✗	✓	✓	✓	✓	✓
Eschenauer and Gligor [25]	DSN	Node	✗	Single authority propagates revocations.	✗	✓	✗	✓	✗	✓
IRMA [26]	SSI	Attestation	✓	Uses public permissioned blockchain.	✗	✗	✓	✗	✓	✗
SOVRIN [27]	SSI	Attestation	✓	Uses public permissioned blockchain.	✗	✗	✓	✗	✓	✓
Veramo (uPort) [28]	SSI	Attestation	✗	Uses public permissionless blockchain.	✓	✗	✓	✗	✓	✓

[1] Refers to the solution having reached a certain degree of maturity (e.g. through an implementation) or governmental backing in case of SSI.
[2] Refers to reliability of the revocation mechanism (e.g. the occurrence of false positives).

II. Problem Description

Because of theft or loss, digital identities may become compromised. In order to mitigate further damage, compromised credentials must be revoked. Furthermore, revocation is required in the instance that a credential becomes (prematurely) voided. E.g., an employee may no longer be employed by a company.

Figure 1 portrays the interactions between the three relevant parties in an SSI system. An Authority attests to a claim [29] of a Subject, that can be verified by a Verifier in order to prove the claim. However, in the instance that the attestation is to be revoked, the verification of the credential must lead to failure. I.e., the credential must no longer be accepted by the Verifier. The Subject can not be trusted to make the revocation apparent to the Verifier, as this sensibly goes against its best interest. Furthermore, revocations must be made apparent to any party that expects to verify the corresponding credential.

Bringing the Verifier in direct contact with the Issuer would go against the principles of Self-Sovereign Identity as this would defeat the purpose of attestations. We refer the reader to [7, 8] or the supplementary material to this article for the principles of SSI. However, the cardinal requirement of SSI is that no third party is required or able to observe or otherwise interfere with the creation or verification of identity data [15].

Finally, in order to adhere to the SSI principles, confidentiality, integrity, and availability must be ensured for the revoked credentials.

Existing revocation mechanisms typically introduce centralised infrastructure to handle revocations such as in [30, 31] or require expensive Proof-of-Work blockchains [32]. The usage of centralised infrastructure may lead to censorship or privacy issues [13]. Blockchains suffer from privacy issues [33], low throughput and limited flexibility [34], and may lead to deanonymisation [35]. Furthermore, they are prone to legislation limiting their use [36].

We believe that the lack of a fully distributed revocation mechanism is limiting the mass deployment of Self-Sovereign Identity. Hence, the problem remains how to create a revocation mechanism that is devoid of any central infrastructure, dependency on third parties during verification and allows clients to independently draw a conclusion on the validity of credentials in order to prevent any misuse due to censorship attempts whilst adhering to the SSI principles.

We formulate the following problem description: an Authority (often referred to as an Issuer [37]) revokes its attestation for a credential. This revocation must be made apparent to all clients that may verify the credential and acknowledge the Authority for attestations. All clients are in a network in which they have equal rights. As such, communication channels are established decentralised and messages are communicated peer-to-peer. As direct communication with any Authority and reliance on centralised infrastructure for verification go against the principles of Self-Sovereign Identity, the propagation of revocations must be performed decentralised. Furthermore, neither any Authority nor any receiving party can be expected to be online at all times, however, all revocations are to be spread across the network in order to reach Verifiers. Furthermore, as each client is equal, the acceptance of revocations is decided individually.

III. Related Work

Table I portrays that revocation in SSI systems is to our knowledge yet to be solved. In the table, related work is compared to our proposed revocation mechanism (see section V). This comparison is performed on the following characteristics: the maturity of the solution, the requirement of network operators, the offline availability of revocations, the reliance on interactions with authorities, the ability for offline verification of information (e.g. certificates or credentials), the existence of single points of failure (SPOFs), and the fragility of the verification mechanism (e.g. false positives or false negatives). As visible, existing and proposed solutions suffer, amongst others, from immaturity, reliance on authorities or lack of offline verification. We note that the usage of blockchains does allow for the realisation of distributed revocation, however, existing blockchains suffer from, as discussed previously, obstacles such as privacy and security issues and low throughput [33, 34, 35]. Hence, we do not deem blockchains a good fit.

As our key contribution addresses revocation, we focus on related work discussing this topic as opposed to focusing solely on SSI. We note that literature on revocation in Self-Sovereign Identity systems is not a widely discussed topic in academia, as such, the selected works address distributed revocation on a broader scale. We group related works in the revocation of SSI credentials, certificates, and nodes.

IRMA [26, 38] and Sovrin [27, 13, 39, 40] propose the usage of cryptographic accumulators [41, 42] for *revocation*

in SSI. Cryptographic accumulators are a probabilistic data structure allowing large sets of values to accumulate to a short witness value that allows for proving certain membership operations (e.g. inclusion checks) [43]. In the aforementioned solutions, a Subject provides a prove of non-revocability of their credentials through this witness value. A Verifier then verifies this proof through the witness value published on the blockchain. Sovrin does not allow for offline verification of credentials as they require both the Subject and the Verifier to retrieve the latest witness value during the verification of a credential. Similarly, IRMA disallows offline verification due to requiring communication with its infrastructure. Furthermore, cryptographic accumulators can be computationally expensive to the extent that they are discouraged to be used at each verification in IRMA [38] and their probabilistic nature leads to the possibility of false positives. Veramo (uPort) [28, 44] use a single Ethereum smart contract [45] for marking attestations as revoked. The usage of the Ethereum blockchain requires synchronisation of blocks in order to guarantee certainty on stored revocations. Furthermore, the single smart contract may introduce a security risk [46]. [18] use a blockchain for storing legitimate Subjects, indirectly disallowing access for revoked Subjects in the SSI system. Updating this set of Subjects is performed by the operators of the blockchain, which introduces centralised authorities for revocations. [19] propose the usage of a revocation list stored on a blockchain, on which consensus is reached through the nodes of the blockchain. Offline verification is achieved through the storage of this revocation list. As the revocation list is not stored per authority, clients require full storage of this list, leading to storage overhead. We note that all revocations in an SSI system can grow up to gigabytes of storage, which hinders the deployment on devices with low memory (e.g. smartphones). Furthermore, the usage of a blockchain introduces further overhead as clients have to synchronise blocks.

Mechanisms for the *revocation of PKI certificates* are present in traditional Public Key Infrastructures (PKIs) such as PKIX [47]. Broadly speaking, a PKI uses a Certificate Authority (CA) to publish a Certificate Revocation List (CRL), containing revoked certificates. In this structure, CAs are inherently central authorities, having relatively absolute power over revocations. These CAs, acting as trusted third parties, are central points of failure, suffer from MITM attacks, and are corruptable [48].

PGP's web of trust [49, 50] attempted to overcome this by handling revocation in a decentralised fashion, in which revocation of keys was handled by the owner through revocation certificates. These certificates indicate that the key was compromised and should therefore no longer be used. However, PGP and the web of trust has been shown to be impractical [51] and require central key servers. Another alternative to PKI is the *Decentralised Public Key Infrastructure* (DPKI) [52, 48]. DPKI proposes the usage of alternative storage structures such as blockchains for storing revocations of public keys. However, these proposed solutions require synchronisation with the used blockchain for verification, introducing overhead and possibly low throughput as discussed previously.

[24] discuss the revocation of PKI certificates in Vehicular ad hoc networks (VANETS) through the distribution of CRLs. Distribution is handled through Road Side Units (RSUs), serving as specialised nodes propagating the CRLs, and through epidemic spread between vehicles. The revocations are stored in Bloom filters [53]. [23] build upon this work by guaranteeing a certain degree of privacy by using group signatures [54] when requesting certificates from the CA. However, the revocations are handled by a single CA and the reliance on Bloom filters introduces the possibility for false positives. [22] propose the distribution of CRLs through direct peer updates, reducing the communication overhead caused by periodic CRL synchronisation. Signatures over CRLs allow nodes to build trust in others. However, direct peer updates may prove to be suboptimal in the case of highly adaptive networks such as that of mobile devices. [21] discuss the revocation of certificates based on the clustering of clients and probabilistic auditing for honesty of distribution points. This probabilistic auditing ensures that distributors of revocations are honest. It is probabilistic in order to reduce performance requirements, however, this allows for malicious nodes to possibly exist for quite some time.

[25] discuss the *revocation of nodes* in distributed sensor networks. Revocation is handled by a single node serving as an authority, delegating revocations to regular sensor clients. We note that the introduction of a single authority goes against the principles of SSI. [20] discuss the revocation of malicious vehicles in Cooperative Intelligent Transportation Systems (CITS). Their solution uses a blockchain for storing revocations through a distributed vehicle admission and revocation scheme. Again, we note that blockchains suffer from the aforementioned hurdles such as privacy and security issues.

IV. System & Threat Model

The followings sections describe the revocation of attestations and further SSI interactions in an *identity network*. Here, an identity network refers to a network consisting of clients that use the identity architecture that is to be laid out. Each client is equal and the network can be joined by any actor. Depending on the interaction with other clients, each client can be a Subject, an Authority or a Verifier. Note that these roles are not mutually exclusive as each client can hold credentials, making it a Subject, attest to a claim, making it an Authority, and verify a credential, making it a Verifier. Clients are assumed to be able to communicate directly with each other client, may it be eventually. Moreover, clients are deemed to be sporadically online, meaning that they are not necessarily online in all instances. However, when they are online they can be communicated with.

Adversaries or malicious actors may be present in the network. We assume that adversaries attempt to cheat the network by the usage of false credentials (i.e., stolen or fabricated) and do not aid in the health of the network (via the spread of revocations discussed in section V). These actors

are not able to drop arbitrary messages but are able to send fabricated messages (e.g. replayed) to any client. For lack of space, attacks on the network itself are omitted here and are discussed in the supplementary material.

V. Architecture & Theoretical Analysis

Authoritarian nodes for managing revocations, present in e.g. [26, 27, 28], deteriorate equality in the network, possibly leading to censorship or collusion in case these nodes are compromised [13]. As in our proposed architecture, each node is equal, the trivial solution for revocation is to actively query the Authority of an attestation in order to verify that they still attest to a claim [14]. This requires interactivity with Authorities, thus disallowing offline verification. Furthermore, whilst availability often is a key characteristic in distributed systems, there is no guarantee that the specific Authorities are online during verification. As our architecture (see section VI), theoretically, allows for an unbound number of attestations for a single claim, interactivity with the Authorities of attestations can prove to become rather impracticable as it introduces additional verification time due to network traffic and response times of Authorities.

Our revocation mechanism overcomes the hurdle of interactivity with Authorities whilst enabling offline verification. During verification of credentials, clients do not require to be online, they merely require occasional synchronisation of revoked attestations through communication with other nodes in the network.

A. Trusted Authorities

In real life, a person has (relatively speaking) a choice whether to acknowledge a certain authority. Following the principles of SSI, this choice is also possible in our revocation architecture. Each client manages, what we coin, a *Trusted Authority Storage* (TAS). The TAS is a set containing the public keys of the Authorities that are trusted by the client. Each of these Authorities is referred to as a *Trusted Authority* (TA). Hence, a distinction is drawn by each client, individually, on the Authorities in the network. As a consequence, it is up to a client to determine whether an Authority is trusted and, therefore, considered a TA or not. With respect to revocation: a client aims to accept and thus store exclusively the revocations made by TAs. The results of acceptance are the storage of the revocations by the client and further propagation of the revocations towards the rest of the network. This may significantly reduce storage requirements under the assumption that, generally, a client is not interested in revocations made by an Authority in e.g. another continent.

B. Attestation Revocation List

All received revocations are stored by a client for later reference in, what we coin, the *Attestation Revocation List* (ARL). The ARL is a list holding the revocations made by TAs. It is, similarly to the TAS, stored and managed by each client individually. In the ARL, revocations are grouped by the TA that revoked the attestation and by a unique version label that is assigned by the TA. This version can be a simple incremental integer and is unique per Authority, thus, not across revocations made by other Authorities. A signature is created over the set of revocations made by the Authority and the label in order to guarantee authenticity.

As the number of revocations can grow to a large amount, we propose the usage of probabilistic data structures, e.g. a Bloom filter [53] or a Cuckoo filter [55], for verifying whether an attestation belongs to the ARL, after which the definitive search is performed. Bloom and Cuckoo filters are memory- and time-efficient probabilistic data structures, which allows for efficient membership operations [53, 55]. [56, 57] discuss the benefits of Bloom filters in Certificate Revocation Lists, which can provide similar speed improvements for the ARL, as both require validation of whether an item is part of a set of revoked items.

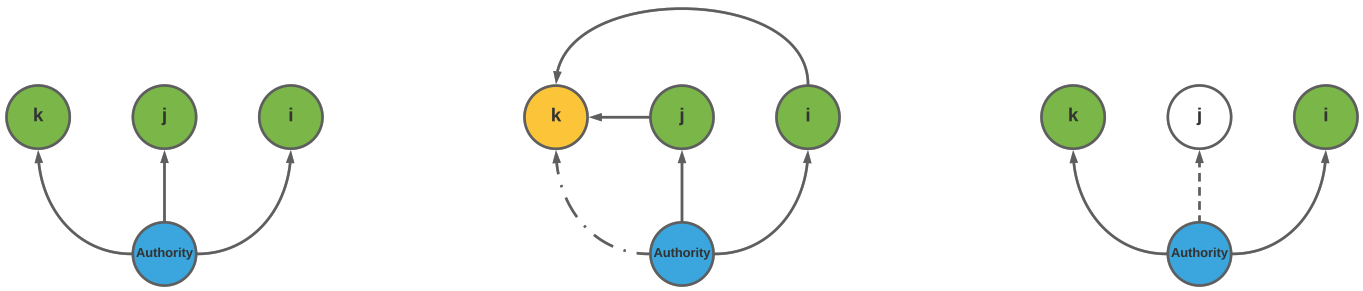
Furthermore, we note that the ARL can be replaced exclusively by a probabilistic data structure. A client may choose to accept the probabilistic nature of Bloom or Cuckoo filters over the exact membership check from storage. Such clients may not be able to aid in the propagation of the revocations, though the low memory requirements may prove to make the protocol suitable for e.g. IoT devices. However, as a result, verification of credentials on the client may be affected by false positives. Whilst this does not explicitly impact security, it could lead to the false rejection of non-revoked credentials.

C. Propagation

In order to achieve propagation of the revocations, the architecture requires a protocol that ensures information is spread across the entire network, whilst also ensuring that unavailable nodes receive the information at a later instance. For this, we propose the usage of a gossip protocol with static re-transmissions. Gossip protocols are communication protocols that allow for a periodic exchange of data with (random) peers [58]. They are originally modelled after epidemic spread [59].

In order to counteract overhead and allow for selective revocation updates, the revocations are propagated using advertisements. Gossiping nodes advertise their known revocations, after which a receiving client is able to selectively request revocations. The aforementioned unique labels assigned by the revoking Authority enable this selective gossip. Authenticity is guaranteed using the signatures. Selective requests can be performed through lower bounds on labels or through the request of a set of labels.

The gossip between clients has been visualised in Figure 2, portraying the communication of revocations from an initial Authority to a select set of clients (i, j and k). Figure 2a portrays the situation in which all revocations are received by the clients, indicated by the solid line and green coloured clients. This is the case when all clients are honest, acknowledge the Authority, and are online during the propagation window (i.e., the time at which the Authority has its first gossip iteration).



(a) Full gossip propagation (b) Indirect gossip propagation (c) uninterested/malicious node

Fig. 2: Fully distributed revocation

After which, the clients continue to propagate the revocations to clients in the remainder of the network, following the same fashion (this is omitted in favour of clarity). Hence, after the initial gossip by the Authority, no further interaction with said Authority is required for the propagation of the revocations.

Figure 2b portrays the instance in which a client is only sporadically online or suffers from a poor connection with the gossiping Authority. As a consequence, client k only receives a subset of the revocations, indicated by the dashed and dotted line and by the yellow colour of the client. However, the remainder of the clients, which are aware of all revocations, are able to further propagate the revocations to the partially informed client k, resulting in eventual propagation across the network.

Finally, Figure 2c showcases the situation in which a client i is either malicious or does not acknowledge the gossiping Authority. With the result that the client ignores the advertisement and does not aid in further propagation of the revocations. The disregard for this advertisement is indicated by the dotted line and by the white colour of the node. As becomes apparent from this description, dishonest nodes pose no large threat to the propagation of revocations. They could introduce a slight delay due to fewer clients gossiping information or due to the spread of fabricated revocations, which will be discovered by the receiving client. We do note that depending on the network topology, Eclipse attacks [60, 61] are a possibility. This is discussed more thoroughly in the supplementary material of this article.

The message flow between a gossiping and receiving client is visualised in Figure 3. The gossip is split up into two phases: firstly, a gossiping client gives notice to another client that it possesses specific revocations. Next, in case the receiving client is unaware of certain revocations, it can request an update by sending back the latest versions of the revocations stored in their TAS. This allows a client to selectively send updates, as the receiving party makes a lower bound on the known versions apparent. Finally, the gossiping client sends the revocations to the updating client. This additional step loosens network requirements for receiving clients. Clients may become spontaneously online or go sporadically offline, resulting in missing revocations (as is also modelled in Figure 2b and Figure 2c). As such, this mechanism allows partial updates. Furthermore, overhead is further reduced as clients are not interested in revocations belonging to unacknowledged Authorities or a client may already be aware of all revocations.

We note that this procedure may be fine-tuned through the usage of revocation dates. Revocation dates may allow clients to ignore old revocations, optimising storage usage as they may no longer be relevant in the system due to expired validity terms of the corresponding attestations. Furthermore, as opposed to selecting a lower bound on revocation versions, a client may request specific versions in order to reduce network usage. Furthermore, we note that this procedure can be fine-tuned by only propagating Bloom filter contents as proposed by [23].

D. Theoretical Analysis

For the remainder of this section, we construct an upper bound on the propagation time of revocations in a network and prove theoretically that eventual propagation is guaranteed in our architecture.

We consider a network of distributed clients denoted by a graph $G = (V, E, w)$. Where V is the set of clients represented by nodes, E is the set of edges between nodes,

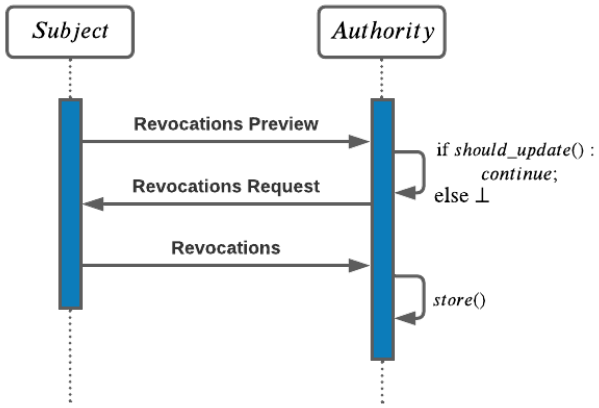


Fig. 3: Revocation gossip message flow

and w represents the network delays between nodes. An edge $(i, j) \in E$ represents a throughput link of information from node i to j with weight $w(i, j)$. Nodes do not necessarily have full knowledge of G at each instance, but eventually, an edge i, j for any $(i, j) \in V$ can be formed, following the aforementioned system model (see section IV). However, nodes always have knowledge on a subset of G, representative as their neighbours.

The propagation of the revocations is dependent on both delays imposed by the protocol itself and by the network. For protocol delays, the propagation time is dependent on the parameters imposed, being:

- **Gossip interval** (t_g): the time interval between gossip to nodes.
- **Gossip amount** (n_g): the number of nodes who are gossiped to after a time interval.
- **Node selection** ($\mathcal{F}_g(x)$): the function used to determine which nodes are gossiped to, where x is the set of nodes known by a node.

Definition V.1. *(Protocol delays). Let n_p be the size of V then it follows that at least $\frac{n_p}{n_g}$ gossip iterations are required to gossip to all nodes. Let $g = t_g \cdot \frac{n_p}{n_g}$ be the time required for this minimal number of interval iterations. The node selection function $\mathcal{F}_g(x)$ may result in overlapping subsets. I.e., let $f_i = F_g(V)$ be the subset of nodes generated at iteration i and let $f_{i+j} = F_g(V)$ be the subset generated at iteration $i + j$, then it does not necessarily hold that $f_i \cap f_{i+j} = \emptyset$. Hence, let $V_f = \{v_0, ..., v_{n-1}\}$ be the multiset of nodes of size $m_p \geq n_p$ selected throughout each iteration until propagation. I.e., $\mathcal{F}_g(x)$ selected at least $m_p \geq n_p$ nodes, leading to at least $t_g \cdot \frac{m_p}{n_g}$ iterations. The time of the additional iterations can be modelled by $h = t_g \cdot \frac{m_p - n_p}{n_g}$. This leads to the propagation time for the protocol delays for a single node v_i, denoted as $\mathcal{T}_{p_i}$, attempting to gossip a single update to the (to it) entire visible network to be as summarised in Equation 1.*

$$\begin{aligned} \mathcal{T}_{p_i} &= h + g \\ &= t_g \cdot \frac{n_p}{n_g} + t_g \cdot \frac{m_p - n_p}{n_g} \\ &= t_g \cdot \left(\frac{n_p}{n_g} + \frac{m_p - n_p}{n_g} \right) \\ &= t_g \cdot \frac{m_p}{n_g} \end{aligned} \tag{1}$$

As nodes are not aware of their position in the network (relatively to others) or of the nodes already contacted by other nodes, only an upper bound on the expected runtime of the algorithm can be set, as each node attempts to gossip all information to all other nodes. Hence, the propagation time for the entire network, denoted by $\mathcal{T}_p$, can be summarised to the formula presented in Equation 2, where $t_{g_i}, m_{p_i}, n_{g_i}$ are the gossip interval, the maximum number of gossiped nodes, and gossip amount per iteration for node v_i, respectively.

$$\begin{aligned} \mathcal{T}_p &\leq \sum_{i=0}^{n_p - 1} \mathcal{T}_{p,i} \\ &\leq \sum_{i=0}^{n_p - 1} \left(t_{g_i} \cdot \frac{m_{p_i}}{n_{g_i}} \right) \end{aligned} \tag{2}$$

Definition V.2. *(Network delays). Next, we generalise the delays imposed by the network. Let $\delta_{i,j}$ be the propagation delay from node v_i to node v_j and let function $\Delta(v_j)$ compute the smallest propagation delay for node v_j to be gossiped to. I.e., $\forall (v_i, v_k) \in V$ it holds that $\delta_{i,j} \leq \delta_{k,j}$. Finally, let $\mathcal{C} = \{c_0, \ldots, c_{n-1}\}$ be the set of delays imposed by processing times on the node on reception of messages (e.g. delays imposed by writing revocations to storage). This leads to the network delay for a single node v_i, denoted by $\mathcal{T}_{n_i}$, updating the entirety of the (to it) visible network with size n as summarised in Equation 3*

$$\mathcal{T}_{n_i} = \sum_{j=0}^{n-1} (\delta_{i,j} + c_j) \tag{3}$$

Then, the total network delays in a system with a set of $V = \{v_0, \ldots, v_{n-1}\}$ nodes of size n can be modelled as visible in Equation 4.

$$\mathcal{T}_n = \sum_{i=0}^{n-1} (\Delta(v_i) + c_i) \tag{4}$$

Definition V.3. *(Propagation time). Definition V.1 and Definition V.2 lead to a total propagation time $\mathcal{T}$ for a network of size n as visible in Equation 5. Again, due to the distributed nature and possible randomness of node selection, we only assign an upper bound.*

$$\begin{aligned} \mathcal{T} &= \mathcal{T}_p + \mathcal{T}_n \\ &\leq \left(\sum_{i=0}^{n-1} \left(t_{g_i} \cdot \frac{m_{p_i}}{n_{g_i}} \right) \right) + \left(\sum_{i=0}^{n-1} \Delta(v_i) + c_i \right) \\ &\leq \sum_{i=0}^{n-1} \left(t_{g_i} \cdot \frac{m_{p_i}}{n_{g_i}} + \Delta(v_i) + c_i \right) \end{aligned} \tag{5}$$

Equation 5 leads to a runtime of $\mathcal{O}(n)$, as in the worst case a single node updates all other nodes. However, it is expected to be logarithmic with respect to the number of nodes, as each gossiped-to node can gossip to yet uninformed nodes. More specifically, a node n_i can gossip to node n_j whilst node n_k gossips to node n_l. As such, the more nodes become informed, the faster the remaining nodes are gossiped to.

Theorem V.1. *Each node will eventually receive all revocations. Nodes may be sporadically online and still receive all revocations, albeit possibly in non-consecutive order, without affecting the availability for other nodes.*

Proof. Consider the set $R = \{r_0, \ldots, r_{l-1}\}$ of size l to be the revocations released by a node v_i at an arbitrary time instance t_i. We assume that each revocation r_i is transferred in a separate message and that messages may be received non-sequentially. I.e., revocation r_{i+1} may be received prior to r_i, this is possible due to arbitrary increases of weights. Furthermore, we assume a naive node selection algorithm $\mathcal{F}(x)$, arbitrarily generating random subsets $V_s \subseteq V$ of size equal to the gossip amount n_g. We consider four scenarios in which we assume that the referenced nodes are interested in R:

1) Node v_j that is online at t_i.
2) Node v_k that comes online at t_{i+j}, i.e., $\forall t_m \in \mathbb{Z}^+ < t_{i+j} \rightarrow w(i,k) = \infty$.
3) Node v_l that is sporadically online at t_i. Hence, at arbitrary time instance t_{i+k} for some $k \geq 0$ the edge between v_i and v_j has $w(i,j) = \infty$ for message r_i.
4) Node v_m that becomes sporadically online at t_{i+j} and further behaves as v_l.

Next, we prove that in all cases the revocations are received.

Part I

We defined the minimum number of gossip iterations to be dependent on the used node selection algorithm $\mathcal{F}(x)$ (see Definition V.1). As subsets of gossiped to nodes are generated randomly, it is impossible that node v_j is not eventually selected. As such, there exists a gossip iteration $k \geq 0$ on which v_j is gossiped to. As such, node v_j receives R at time $t_i + k \cdot t_g$, for some $k \geq 0$.

Part II

Similarly, using the above, node v_k will be gossiped to on a gossip iteration k for some $k \geq 0$. However, in case $t_i + k \cdot t_g < t_{i+j}$, node v_k will never receive R. As we assumed a naive selection algorithm, it is impossible for $\mathcal{F}(x)$ to not generate a subset $V_r \subseteq R$ at a later iteration for which it holds that $v_k \in V_r$. Hence, after additional iterations, there is an iteration $k + l$ for some $l \geq 0$ for which it holds that $t_i + (k + l) \cdot t_g \geq t_{i+j}$. As such, v_i gossips R to v_k at time $t_i + (k + l) \cdot t_g$ for some $k \geq 0$ and $l \geq 0$.

Part III

From node v_j follows that node v_i will gossip R to node v_l. However, the arbitrary loss of messages results in the reception of $R_k \subseteq R$ at gossip iteration k. Inductively, it follows from node v_k that v_l will be gossiped to at a later instance. Hence, at a later iteration $k + l$, for an $l \geq 0$ ($l = 0$ in case $R_k = R$ after the first gossip iteration), v_l receives another $R_l \subseteq R$. As messages are arbitrarily dropped, it follows that there exists some $n \geq 0$ for which it holds that after $n + k + l$ gossip attempts, all revocations have been successfully transferred at least once. I.e., let $\mathcal{R} = \{R_0, \ldots, R_{n-1}\}$ be the set of all subsets of R received after $n + k + l$ gossip attempts, then it holds that $\left(\bigcup_{X \in \mathcal{R}} X\right) \cap R = \emptyset$. As such, we can conclude that v_l possesses R at $t_i + (n + k + l) \cdot t_g$, for some $n \geq 0$, $k \geq 0$ and $l \geq 0$.

Part IV

From v_k it follows that a node coming online at t_{i+j} receives R and from v_l it follows that eventually R is received by a sporadically online node. Hence, transitively we conclude that a node coming sporadically online at t_{i+j} receives R from v_i.

We have proven for each scenario that all nodes, regardless of edge weights, receive all revocations. Furthermore, gossip to nodes is independent, as gossip sent over edge with weight $w(i,j)$ does not affect gossip sent over edge with weight $w(i,k)$. Hence, we conclude that each node, eventually, receives all revocations, albeit possibly in nonconsecutive order, without affecting availability for other nodes. □

Theorem V.2. *Revocations in any network with at least 1 honest node will propagate to each node in $\mathcal{O}(n)$ time in at most $\sum_{i=0}^{n-1} \left(t_{g_i} \cdot \frac{m_{p_i}}{n_{g_i}} + \Delta(v_i) + c_i \right)$ seconds.*

Proof. Consider a network with n nodes $V = \{v_0, \ldots, v_{n-1}\}$, of which a subset $V_m \subset V$ of size m is not aware of the latest revocations. Of the $n - m$ nodes, which are aware of the latest revocations, all but one node v_i is malicious. We assume that dishonest nodes cannot affect network traffic. Even in case there is no complete graph, we assume that there will eventually exist at least a single edge $w(i,j)$ with $v_j \in V_m$. Using Theorem V.1 we conclude that v_i is able to eventually gossip revocations to a single node $v_j \in V_m$.

Inductively, it follows that $\{v_i, v_j\}$ eventually gossip the revocations to the remainder nodes in $V_m \backslash \{v_j\}$. In the worst case, v_i gossips to each node belonging to the m nodes, resulting in a runtime of $O(n)$ and a propagation time of $\sum_{i=0}^{n-1} \left(t_{g_i} \cdot \frac{m_{p_i}}{n_{g_i}} + \Delta(v_i) + c_i \right)$.

□

Theorem V.3. *The revoking Authority does not need to be online to guarantee the propagation of its revocations across the network in case the collective knowledge of its revocations exists across honest nodes.*

Proof. Consider a network of n nodes $V = \{v_0, \ldots, v_{n-1}\}$ and a set of revocations $R = \{r_0, \ldots, r_{i-1}\}$ released by node v_i at t_i. Node v_i gossips subsets $R_i \subseteq R$ to a subset of nodes $V_s \subseteq V$. Let $\mathcal{R} = \{R_0, \ldots, R_s\}$ be the set of all such subsets. We assume that collectively all revocations are known across the subsets, i.e., let $\bigcup_{X \in \mathcal{R}} X$ be the union of each subset of $\mathcal{R}$ then it holds that $\left(\bigcup_{X \in \mathcal{R}} X\right) \cap R = \emptyset$. After this first gossip iteration, v_i goes offline, hence $\forall j \in \{0, n-1\}$ it holds that $w(i,j) = \infty \wedge w(j,i) = \infty$. Next we prove that all revocations are propagated across the network.

Base case

In the base cases it holds that $\exists R_i \in \mathcal{R}$ such that $R_i \cap R = \emptyset$. As shown by Theorem V.2, a network with at least 1 honest node will propagate all revocations. Hence, we can conclude that the revocations are further propagated by the honest

node(s).

Inductive step
Next, we prove that the Authority does not need to be online after having gossiped its revocations to a subset of any network $n > 1$ for which it holds that $\neg\exists R_i \in \mathcal{R} \rightarrow R_i \cap R = \emptyset$.

Consider a honest node v_j that holds some knowledge on R, i.e., possesses $R_j \subset R$. Note that this may be the empty set and that whether $v_j \in V_s$ is of no impact. As shown by Theorem V.2 a network with at least 1 honest node will eventually gossip its revocations to all other nodes. From this it follows that eventually each $v_k \in V_s$ gossips to v_j. From $(\bigcup_{X \in \mathcal{R}} X) \cap R = \emptyset$ it follows that v_j receives enough revocation subsets to reconstruct R. As such, there will exist at least a single honest node v_j possessing R. Now, using Theorem V.1, we can conclude that each node will receive all revocations.

We have shown that in both the base case and the inductive step all revocations are propagated, therefore, we conclude that the revoking Authority does not need to be online to guarantee the propagation of its revocations in the case the revocations are at least collectively known across honest nodes. □

VI. Attestation Architecture

Self-Sovereign Identity is built around *Verifiable Claims* (VCs) [8], which are composed of several types of information. Firstly, a *claim* is made by a Subject [62]. Authorities can attest to a claim, making it a VC. When metadata is added to a VC, we speak of an *attribute*. Finally, a set of related attributes is referred to as a *credential* [8], although this term is also used to describe an attribute. This has been visualised in Figure 4.

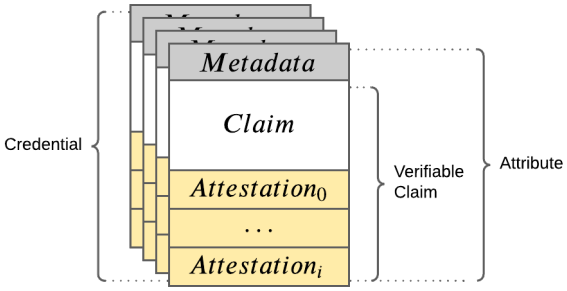


Fig. 4: Credential structure

In the proposed design, each *claim* is a Zero-Knowledge Proof (ZKP) [63] represented by an anonymised *token*, which stores a reference to a claim via its hash. A token can be referenced by multiple *metadata* structures which assign different properties to a claim (e.g. a validity term). Furthermore, multiple *attestations* can be made for each metadata structure. Finally, although not explicitly modelled, multiple *credentials* can reference multiple attestations and as such, multiple claims.

The tokens are stored in a blockchain-esque chain structure, referencing the previous token. The chaining of attributes forms the actual identity, as claims may rely on previous claims (e.g., a driver's license may depend on a person's legal name). This chain is stored by each client individually, hence, it is not known by all clients. Chaining also aids in preventing the withholdment of previous attributes by malicious actors as well as making it more difficult for one to use stolen credentials as they require the previous tokens for the chain's hash to be correct with respect to the attestations. The first token, comparable to a genesis-block in blockchain structures such as [64], contains the hash of the public key of the Subject. Any subsequent attribute, thus, generates a new token, occupying a place as a shackle in the chain. As such, it is impossible for a client to attempt to hide the existence of attestations—apart from when the entire identity is destroyed—or attempt to change an attribute, as otherwise, the attestations of other Authorities become invalid (as the signatures provided by Authorities will not match the hash of the changed token and chain).

Next, we discuss our architecture showcasing the lifecycle of these credentials.

A. Attestation Signing

The attestation procedure is visible in Figure 5. It consists of two phases: the Claim-phase (I) and the Attestation-phase (II) which do not necessarily require subsequent execution. As multiple Authorities can attest to a single claim, the claim-phase only has to occur once per claim, after which the Attestation-phase can be performed indefinitely with Authorities.

1) Claim-phase: The Claim-phase is initiated by a Subject through a request. In this request, a subject makes information such as its public key, the proof format and the attribute name apparent. The public key belongs to a single-use key pair, this aids in privacy as claims are not directly linked to the same key. The Authority may respond by creating a Zero-Knowledge Proof [63] over the value and public key belonging to the requested claim. As may become apparent from this description two modus operandi are possible. Firstly, a client may self-create this claim, following the natural description of

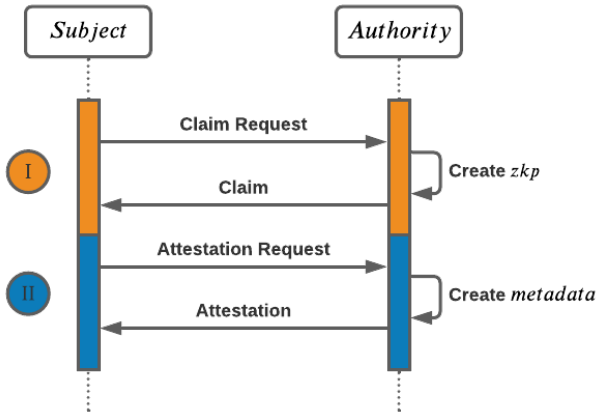


Fig. 5: Attestation flow

a claim. However, a client may not know the associated value, hence, the second modus operandi delegates the creation of the claim to an Authority.

2) Attestation-phase: After possessing a claim, a Subject requests an attestation for said claim, creating a VC and subsequently an attribute. When a Subject requests an attestation from an Authority, it discloses the prior attestations and the Authority may request prior tokens, allowing the Authority to verify previous attributes. An Authority does not have to request all tokens, however, may do so until it has built enough confidence to provide an attestation. Furthermore, the Subject creates a metadata structure for properties of the attestation, including information such as the signature date and the hash of the underlying claim value. The attestation is made through a signature over the metadata (which in turn, hence, references the underlying claim). However, as a hash would allow for trivial preimage attacks for attributes with a limited message space (e.g. an *age* attribute), we propose the usage of salts [65, 66].

B. Presentation Flow

For presentation, a Verifier requests an attribute with a specific name. A Subject may subsequently decide to respond to such a request and to disclose the corresponding attribute. Next, similarly to the attestation flow, an Authority may request the tokens of previous claims until it has gained enough confidence. Note here that the attribute request is not necessarily required, as a client can disclosure an attribute directly. However, the specification of an attribute name aids in selective disclosure whilst additionally allowing the Authority to determine whether a specific attribute presentation is solicited. After an attribute has been disclosed and, thus, presented, the Authority may verify its validity.

C. Verification Flow

We propose two types of verification: interactive verification relying on ZKPs and non-interactive verification dependent on signatures and non-interactive ZKPs.

Verification is two-fold in the sense that the attestations are verified as well as the underlying attribute value. Hence, in both verification methods, the list of attestors must contain an Authority that is trusted by the Verifier and its attestation must still be valid through the revocation mechanism described in section V. However, as not each ZKP is non-interactive, the signatures introduce an alternative for verification in case there is no connection between the Subject and the Verifier.

The former variant is presented in Figure 6. For *active verification*, a Verifier requests the underlying claim of an attribute from the Subject (procured through prior presentation). The Subject may consent by sending the requested claim. Next, the Verifier may send challenges to verify the underlying ZKP. Note that for this to happen, the Authority must either be aware of the value belonging to the attribute (in case of an exact value proof) or must know the namespace of possible values (in case of a range proof). Sharing of the plaintext value can be done during the presentation. This should be performed using encryption in order to preserve confidentiality and integrity. Furthermore, the Authority verifies the presented attestations.

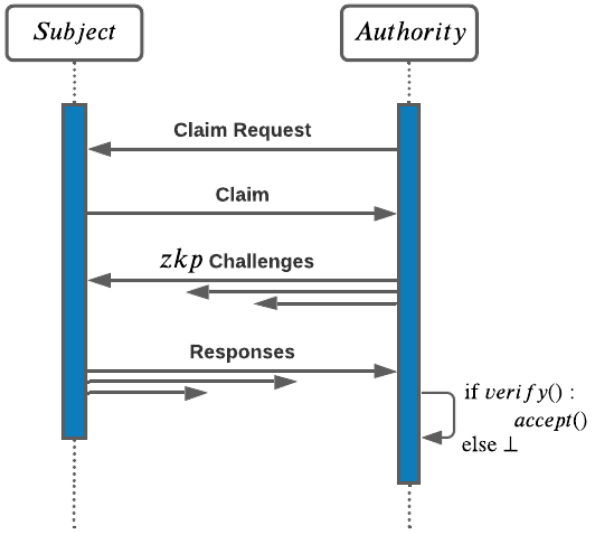


Fig. 6: Interactive verification

The second variant does not require any connectivity between the Subject and Verifier, apart from the presentation itself. However, a presentation does not necessarily require any form of digital communication (e.g. it can be performed through QR-codes), allowing full offline verification through signatures or non-interactive ZKPs. It is, however, to note that this offline verification, thus, does not rely on any additional token requests and, as such, all relevant tokens must either be made directly apparent to the Verifier during the presentation of the attribute or the Verifier must make its decision based solely on the presented attribute.

VII. Algorithms & Simulation

In order to realise the proposed revocation mechanism discussed in section V, each client in the network runs three algorithms. A gossiping client runs algorithm 1, which enables the periodic advertisement of revocations. Firstly, a subset of peers is generated using the node selection function (line 2), next the advertisement is gossiped to each of these clients (lines 3-4). Finally, the gossiping client awaits the start of the next gossip interval (line 5). An advertisement consists of pairs of Authority public keys and the latest versions of revocations they published.

A node receiving the advertisement runs algorithm 2. In this procedure, the node verifies whether any TA is present in the advertisement (lines 2-3). Then it verifies, using the function `FindMissingVersion` whether it has any missing or unknown revocation versions belonging to the Authority (lines 4-5). I.e., `FindMissingVersion` determines on a advertisement containing the revocation version v_i part of revocations

by Authority a_i whether $\exists(v_j, a_i) \in \mathcal{ARL}$ such that $(\forall(v_k, a_i) \in \mathcal{ARL}$ it holds that $v_j \geq v_k \wedge v_j < v_i) \vee (v_{j+1} \notin \mathcal{ARL} \wedge v_{j+1} < v_i)$. If this is the case, an update is requested from the gossiping client for the respective Authority and lowest missing version (line 6). The advertising node verifies whether he advertised to the node recently and sends the revocations.

Following the reception of requested revocations, a node executes algorithm 3. This procedure verifies the relevance of the revocations (line 1) and then their validity (line 2). This validity check is performed by verifying the attached signature over the revocations and their version using the public key of the TA. Finally, the revocations are stored in the ARL (line 3).

A. Simulation

The analysis of the mechanism is two-fold. Firstly, we discuss a simulation showcasing scalability amongst a relatively high number of clients (up to 10,000) using the aforementioned algorithms. Secondly, we showcase analysis through the deployment on smartphones in section IX, portraying usability on handheld devices. The simulation was performed on a system with an Intel i7-6700HQ CPU clocked at 2.60 GHz and 16 GB of RAM.

Algorithm 1: Revocation Advertisement Gossip

input :
- Set of Clients in the network $\mathcal{C} = \{c_0, \ldots, c_i\}$
- Set of known Authority-Version pairs $\mathcal{A} = \{(a_0, v_j), \ldots, (a_j, v_k)\}$
- Gossip interval t_g
- Gossip amount n_g

output: Revocation advertisements

1 **while** True **do**
2 $\mathcal{C}_g \leftarrow$ `SelectPeers` $(\mathcal{C}, n_g)$;
3 **foreach** $c_i \in \mathcal{C}_g$ **do**
4 `GossipRevocations`$(c_i, \mathcal{A})$;
5 `Wait`(t_g);

Algorithm 2: Revocation Update Request Procedure

input : Set of Authority-Version pairs $\mathcal{A} = \{(a_0, v_j), \ldots, (a_j, v_k)\}$

output: Revocation update request

1 *On reception of* $\mathcal{A}$ *by* Client c_i;
2 **for** Authority a_i, Version v_j **in** $\mathcal{A}$ **do**
3 **if** $a_i \in \mathcal{TAS}$ **then**
  `// Returns the lowest missing version or null.`
4 $v_{local} \leftarrow$ `FindMissingVersion`(a_i);
5 **if** $v_{local} < v_j$ **then**
6 `RequestUpdate`(c_i, a_i, v_j);

Algorithm 3: Revocation Reception

input : Set of revocations $R = \{r_0, \ldots, r_n\}$ of version v_i with signature s_i revoked by Authority a_i

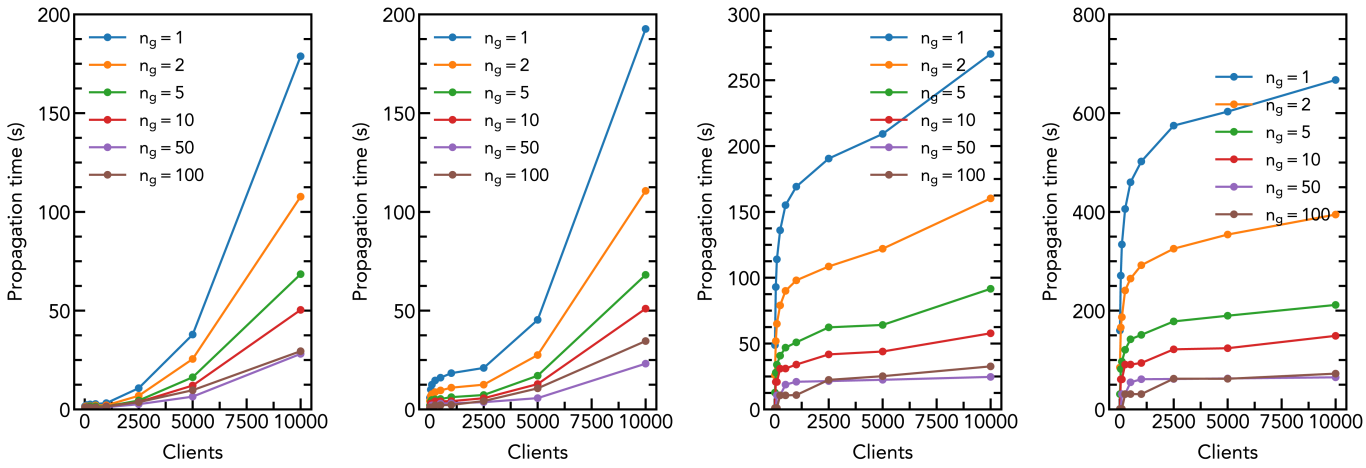
output: $R \subseteq \mathcal{ARL}$

1 **if** Authority a_i **in** $\mathcal{TAS}$ **then**
2 **if** `Verify`$(a_i, s_i, v_i|R)$ **then**
3 $\mathcal{ARL} \leftarrow \mathcal{ARL} \cup R$;
4 **else**
5 $\perp$
6 **else**
7 $\perp$

The simulation has been performed through the mimicking of gossip of 1 million revocations between clients released by a single client serving as an Authority. Each client runs the three algorithms and acknowledges the gossiping Authority. The measurements for the simulation were gathered by timing the duration until full propagation of the revocations across the simulated network. The simulated network allowed for communication between all clients. As the simulation is performed on a single machine, network usage was of no impact. As such, arbitrary delays between 20-50 ms are introduced in order to simulate the impact of network latency, based on the global average reported by [67]. Furthermore, an arbitrary delay between 2500-3000 ms is added to simulate the reception of 1 million SHA3-256 hashes of 32 bytes each, based on the global average network speed of around 100 Mbps [67]. The revocations were released on $t = 0$ by the Authority. Each simulation was repeated 10 times.

B. Simulation Results

The averages of the timings are visible in Figure 7. As expected, increasing the gossip interval t_g leads to higher propagation times. The high-interval timings (Figure 7c and Figure 7d) portray the expected logarithmic increase of the propagation time with respect to the number of clients. However, we note that this logarithmic increase is less prominent in a higher number of clients. For instance, after 5000 clients in Figure 7c, a more quadratic increase starts to appear, contrary to expectations. This is more prominent in Figure 7a and Figure 7b. This behaviour can be explained by hardware limitations on the workstation limiting the number of messages between clients, as we noted high CPU usage. As visible in Figure 7a, before 2500 clients a logarithmic increase is visible, after which the timings start to increase quadratically. This same behaviour is present in the propagation timings of Figure 7a, although, this pattern is less discernible. However, as mentioned, the increase of the gossip interval still leads to an increase in propagation time, as such, the simulations can still aid in drawing a conclusion on scalability. Furthermore, it can be seen in each simulation that the increase of n_g leads to lower propagation timings. This is expected as this allows for higher throughput



(a) $t_g = 100\ ms$ (b) $t_g = 1000\ ms$ (c) $t_g = 10000\ ms$ (d) $t_g = 30000\ ms$

Fig. 7: Simulated propagation times

of the number of revocations. However, this requires similarly more resources from individual clients.

Due to the aforementioned limitations imposed on the simulation, it is difficult to draw a conclusion. It can, however, be noted that a lower gossip interval leads to higher stress on the client. Hence, we deem the simulations with higher gossip intervals more realistic as they impose less demanding system requirements, making deployment on e.g. smartphones more viable. The simulations portray that revocations are able to gossip relatively quickly; e.g. with $n_g \geq 5$, all but one simulation results in a propagation time lower than 200 seconds. However, we do note that the network topology is not realistic enough to simulate a globally deployed setting. We draw the conclusion that the simulations portray that the revocation mechanism achieves usable propagation timings in an unpartitioned network, however, further experiments are required to show scalability in a more wide variety of network topologies to draw more decisive conclusions.

VIII. Implementation & Field Trial

Sections V & VII presented a Self-Sovereign Identity architecture based on the prior works by [14, 15] with the novel fully distributed revocation algorithm and offline verification capabilities. Based on this architecture, two implementations have been made using the IPv8 protocol stack[4]. The selection of IPv8 stems from firstly its academic background, proving its viability through various publications [16, 17] and is used by Tribler [68, 69]. Secondly, IPv8 allows for direct client-to-client communication, hence, enabling a fully distributed infrastructure at the core of the solution. Thirdly, IPv8 does not require (expensive) Proof-of-Work algorithms utilised by blockchain structures such as [64] and [70].

Three semantic layers have been implemented on top of the Kotlin implementation of IPv8. These layers facilitate the claim logic, attestation logic and revocation logic, respectively. Per the authors choice two ZKPs claim types have been implemented: firstly, a ZKP proof allowing arbitrary data and the verification of exact values. The implementation is based on the algorithm proposed by [71], allowing verifiable computation through 2-DNF formulae over bits. Secondly, the range ZKP proposed by [72], allowing the encoding of integer values laying in a specific range. The commitment scheme proposed by [73] has been implemented in order to realise this range proof, based on the work by [14]. Both of these proofs are interactive. However, as shown by [45], the schema introduced by [72] can be made non-interactive. The code for the reference implementation of these semantic layers is available on the IPv8 repository[5].

Secondly, a mobile client has been implemented in the form of an Android application. This client uses the implementation of the three semantic layers and showcases the usability on smartphones. The application supports all discussed functionalities. In addition, clients can create multi-party communication channels in order to force visibility with one another. This is performed through a sequence of alphanumerical characters generated by the client, which, when shared, allows clients possessing this sequence to find one another. The application enables offline verification through the presentation of attributes through QR-codes. As verifiable claims can comprise any form of data, the client additionally supports attestations to pictures; opening up the possibility for digitally attested to passport photographs.

The application was validated using a minor real-life trial backed by the Dutch National Office for Identity Data (RvIG). Figure 9 portrays the usage of the application for the ZKP verification of the age of majority. Further trials were cancelled due to the COVID-19 pandemic. The implementation can be found on the Trustchain `superapp` repository[6].

[4]For the official (Python) documentation of IPv8, see: https://py-ipv8.readthedocs.io/en/latest/

[5]For the Kotlin IPv8 repository, see: https://github.com/Tribler/kotlin-ipv8

[6]For the Android application, see: https://github.com/Tribler/trustchain-superapp

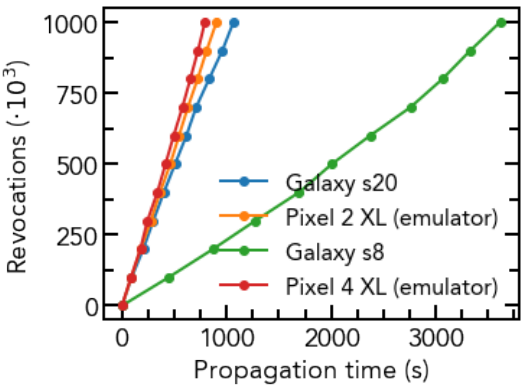


Fig. 8: Propagation timings on smartphones

IX. Performance Analysis

The analysis of the implementation was performed in a test setup measuring the time required to gossip revocations between an Authority and regular clients running on smartphones. For revocations, we generated a dataset of 1 million revoked 32 byte SHA3-256 hashes, a format used by the implementation. Revocations were split up into sets of 1000 in order to minimise the impact of a single packet loss. In order to prevent network congestion, the gossiping client was restricted to 10 UDP packets per second. For the default parameters, the gossip-interval t_g was set to $100ms$ in order to maximise throughput of gossip. The number of selected peers m_p was set to 5 as the IPv8 uses 20 simultaneous connections per default[7] and the simulation portrayed that this number poses a good trade-off between propagation time and the overhead caused by contacting a large number of clients. However, due to the network size of the test setup, this is of no impact.

A. Revocation Amount

Figure 8 showcases the revocation scaling in a system of 1 client gossiping revocations and 4 clients receiving revocations. As visible, the propagation time scales linearly with respect to the number of revocations. As visible 1 million revocations take up to 3750 seconds or just over 1 hour (although considerably less on more modern smartphones). As this can be deemed more than two years worth of revocations [11] in the UK, we deem this scalability usable as the propagation is expected to grow logarithmic with respect to the number of clients.

Compared to the simulations discussed in section VII, the performance is worse. We note that this can be explained mostly due to communication overhead caused by UDP packet splitting. The tremendous amount of packets led to many packet drops, in turn leading to the loss of specific revocation versions. As the reference implementation naively provides the gossiping client with a lower bound of missing versions, the additional network traffic of already gossiped versions causes more packet losses. This snowballing effect worsens the performance of the algorithm. As such, the investigation of other network protocols or more sophisticated handling of packet loss can prove to significantly improve performance. However, the achieved performance can be deemed usable.

[7]For the default parameter, see: https://github.com/Tribler/kotlin-ipv8/blob/master/ipv8/src/main/java/nl/tudelft/ipv8/Community.kt

X. Conclusion

This paper addresses revocation in Self-Sovereign Identity systems. We deem revocation to be the last remaining open issue for SSI to become a feasible contender for the next generation of identity management. We proposed the first fully distributed revocation mechanism requiring no interactivity with revoking Authorities, whilst adhering to the principles of the SSI concept. Revocations are propagated through the network using a gossip-based protocol, in which the acknowledgement of revocations is up to the discretion of Verifiers. The revocation mechanism is part of a fully distributed SSI schema, enabling offline verification. Privacy is aided through the usage of Zero-Knowledge Proofs and communication with selected peers. Our small scale trial shows that fully distributed SSI is feasible on modern handheld devices and that this is a promising direction to further explore. We conclude that our proposed architecture is a valid candidate to facilitate the digital identity needs of the European Union.

Fig. 9: Real-life trial

References

[1] U. Von der Leyen, "State of the union address by president von der leyen at the european parliament plenary," Sept. 2020. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/SPEECH_20_1655

[2] European Commission, "Eu digital covid certificate," June 2021. [Online]. Available: https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/eu-digital-covid-certificate_en

[3] Siftery, "Top social login tools compared," Jan. 2017. [Online]. Available: https://medium.com/@siftery/top-social-login-tools-compared-b350eae26118

[4] O. Tene and J. Polonetsky, "Big data for all: Privacy and user control in the age of analytics," *Nw. J. Tech. & Intell. Prop.*, vol. 11, p. xxvii, 2012.

[5] R. Rogers, "Deplatforming: Following extreme internet celebrities to telegram and alternative social media,"

European Journal of Communication, vol. 35, no. 3, pp. 213–229, 2020.

[6] K. Cameron, "The laws of identity," *Microsoft Corp*, vol. 5, pp. 8–11, 2005.

[7] C. Allen, "The Path to Self-Sovereign Identity," July 2016. [Online]. Available: https://www.coindesk.com/path-self-sovereign-identity

[8] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," pp. 80–86, Nov. 2018.

[9] J. Johnson, "Annual number of data breaches and exposed records in the United States from 2005 to 2020," Mar. 2021. [Online]. Available: https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

[10] M. Hill and D. Swinhoe, "The 15 biggest data breaches of the 21st century," July 2021. [Online]. Available: https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html

[11] H. HM Passport Office, Border Force and The Rt Hon Caroline Nokes MP, "Report your lost or stolen passport," June 2018. [Online]. Available: https://www.gov.uk/government/news/report-your-lost-or-stolen-passport

[12] P. Zhang and G. Rui, "China floods: 'digital dark age' after disaster wreaks havoc on internet and electricity," July 2021. [Online]. Available: https://www.scmp.com/news/people-culture/environment/article/3142544/china-floods-digital-dark-age-after-disaster-wreaks

[13] D. Khovratovich and J. Law, "Sovrin: digital identities in the blockchain era," The Sovrin Foundation, Tech. Rep., 2017. [Online]. Available: https://sovrin.org/library/sovrin-digital-identities-in-the-blockchain-era/

[14] Q. Stokkink and J. Pouwelse, "Deployment of a blockchain-based self-sovereign identity," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1336–1342.

[15] Q. Stokkink, D. Epema, and J. Pouwelse, "A Truly Self-Sovereign Identity System," *arXiv preprint arXiv:2007.00415*, 2020.

[16] G. Halkes and J. Pouwelse, "Udp nat and firewall puncturing in the wild," in *International Conference on Research in Networking*. Springer, 2011, pp. 1–12.

[17] N. Zeilemaker, B. Schoon, and J. Pouwelse, "Dispersy bundle synchronization," *TU Delft, Parallel and Distributed Systems*, 2013.

[18] J. Xu, K. Xue, H. Tian, J. Hong, D. S. Wei, and P. Hong, "An identity management and authentication scheme based on redactable blockchain for mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6688–6698, 2020.

[19] A. Abraham, S. More, C. Rabensteiner, and F. Horandner, "Revocable and offline-verifiable self-sovereign identities," *Proceedings - 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020*, pp. 1020–1027, Dec. 2020.

[20] N. Lasla, M. Younis, W. Znaidi, and D. Ben Arbia, "Efficient Distributed Admission and Revocation Using Blockchain for Cooperative ITS," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2018 - Proceedings*, vol. 2018-January. Institute of Electrical and Electronics Engineers Inc., Mar. 2018, pp. 1–5.

[21] B. C. Popescu, B. Crispo, and A. S. Tanenbaum, "A certificate revocation scheme for a large-scale highly replicated distributed system," in *Proceedings - IEEE Symposium on Computers and Communications*, 2003, pp. 225–231.

[22] C. Y. Liau, S. Bressan, and K.-L. Tan, "Efficient Certificate Revocation : A P2P Approach," *HICSS'05*, 2005.

[23] J. J. Haas, Y. C. Hu, and K. P. Laberteaux, "Efficient certificate revocation list organization and distribution," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 595–604, Mar. 2011.

[24] K. P. Laberteaux, J. J. Haas, and Y.-C. Hu, "Security certificate revocation list distribution for vanet," in *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, 2008, pp. 88–89.

[25] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002, pp. 41–47.

[26] G. Alpár, F. van den Broek, B. Hampiholi, B. Jacobs, W. Lueks, and S. Ringers, "Irma: practical, decentralized and privacy-friendly identity management using smartphones," *HotPETs 2017*, 2017.

[27] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," *The Sovrin Foundation*, 2016.

[28] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, "Uport: A platform for self-sovereign identity," Oct. 2016. [Online]. Available: https://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf

[29] M. Sporny, D. Longley, and D. Chadwick, "Core data model," in *Verifiable Credentials Data Model 1.0*. W3C, Nov. 2019. [Online]. Available: https://www.w3.org/TR/vc-data-model/

[30] Sovrin, "Stewards." [Online]. Available: https://sovrin.org/stewards/

[31] Privacy by Design Foundation, "Irma in detail." [Online]. Available: https://privacybydesign.foundation/irma-explanation/

[32] uPort, "uPort Developer Portal." [Online]. Available: https://developer.uport.me/

[33] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—a systematic review," *PloS one*, vol. 11, no. 10, p. e0163477, 2016.

[34] L. Hughes, Y. K. Dwivedi, S. K. Misra, N. P. Rana, V. Raghavan, and V. Akella, "Blockchain research, practice and policy: Applications, benefits, limitations,

emerging research themes and research agenda," *International Journal of Information Management*, vol. 49, pp. 114–129, 2019.

[35] A. Biryukov and S. Tikhomirov, "Deanonymization and linkability of cryptocurrency transactions based on network analysis," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 172–184.

[36] R. Xie, "Why china had to ban cryptocurrency but the us did not: A comparative analysis of regulations on crypto-markets between the us and china," *Wash. U. Global Stud. L. Rev.*, vol. 18, p. 457, 2019.

[37] M. Sporny, D. Longley, and D. Chadwick, "Basic concepts," in *Verifiable Credentials Data Model 1.0*. W3C, Nov. 2019. [Online]. Available: https://www.w3.org/TR/vc-data-model/

[38] IRMA, "Revocation." [Online]. Available: https://irma.app/docs/revocation/

[39] D. Hardman, "HIPE 0011-cred-revocation," Feb. 2018. [Online]. Available: https://github.com/hyperledger/indy-hipe/blob/4fd9db58/text/0011-cred-revocation/README.md

[40] ——, "What if I lose my phone?" 2019. [Online]. Available: https://sovrin.org/wp-content/uploads/2019/03/What-if-someone-steals-my-phone-110319.pdf

[41] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Annual international cryptology conference*. Springer, 2002, pp. 61–76.

[42] J. Camenisch, M. Kohlweiss, and C. Soriente, "An accumulator based on bilinear maps and efficient revocation for anonymous credentials," in *International workshop on public key cryptography*. Springer, 2009, pp. 481–500.

[43] N. Fazio and A. Nicolosi, "Cryptographic accumulators: Definitions, constructions and applications," *Paper written for course at New York University: www. cs. nyu. edu/nicolosi/papers/accumulators. pdf*, 2002.

[44] Veramo, "Revocation-registry: Ethereum revocation registry contract," 2019. [Online]. Available: https://github.com/uport-project/revocation-registry

[45] T. Koens, C. Ramaekers, and C. Van Wijk, "Efficient Zero-Knowledge Range Proofs in Ethereum," ING, Tech. Rep., 2018. [Online]. Available: https://www.ingwb.com/media/2122048/zero-knowledge-range-proof-whitepaper.pdf

[46] P. Praitheeshan, L. Pan, J. Yu, J. Liu, and R. Doss, "Security analysis methods on ethereum smart contract vulnerabilities: a survey," *arXiv preprint arXiv:1908.08605*, 2019.

[47] IETF, "Public-key infrastructure (x.509) (pkix)." [Online]. Available: https://datatracker.ietf.org/wg/pkix/

[48] C. Allen, A. Brock, V. Buterin, J. Callas, D. Dorje, C. Lundkvist, P. Kravchenko, J. Nelson, D. Reed, M. Sabadello, G. Slepak, N. Thorp, and H. T. Wood, "Decentralized public key infrastructure. a white paper from rebooting the web of trust," 2015.

[49] S. Garfinkel, *PGP: pretty good privacy*. " O'Reilly Media, Inc.", 1995.

[50] P. Zimmermann, "Why I Wrote PGP," 1999. [Online]. Available: https://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html

[51] A. Whitten and J. D. Tygar, "Why johnny can't encrypt: A usability evaluation of pgp 5.0." in *USENIX security symposium*, vol. 348, 1999, pp. 169–184.

[52] C. Fromknecht, D. Velicanu, and S. Yakoubov, "A decentralized public key infrastructure with identity retention." *IACR Cryptol. ePrint Arch.*, vol. 2014, p. 803, 2014.

[53] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[54] D. Chaum and E. Van Heyst, "Group signatures," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1991, pp. 257–265.

[55] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, "Cuckoo filter: Practically better than bloom," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 2014, pp. 75–88.

[56] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, and J.-P. Hubaux, "Certificate revocation in vehicular networks," *Laboratory for computer Communications and Applications (LCA) School of Computer and Communication Sciences, EPFL, Switzerland*, pp. 1–10, 2006.

[57] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of Misbehaving and Faulty Nodes in Vehicular Networks," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 25, no. 8, 2007.

[58] M. Kwiatkowska, G. Norman, and D. Parker, "Analysis of a gossip protocol in prism," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 3, pp. 17–22, 2008.

[59] A. De1mers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, 1987, pp. 1–12.

[60] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 299–314, 2002.

[61] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 261–269.

[62] M. Sporny, D. Longley, and D. Chadwick, "Verifiable credentials data model 1.0," Nov. 2019. [Online]. Available: https://www.w3.org/TR/vc-data-model/

[63] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems," *Journal of the ACM (JACM)*, vol. 38, no. 3, pp. 690–728, 1991.

[64] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[65] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell, "Stronger password authentication using browser extensions." in *USENIX Security Symposium*.

Baltimore, MD, USA, 2005, pp. 17–32.

[66] D. Arias, "Adding salt to hashing: A better way to store passwords," Feb. 2021. [Online]. Available: https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/

[67] Ookla, "Speedtest global index," July 2021. [Online]. Available: https://www.speedtest.net/global-index

[68] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. Epema, M. Reinders, M. R. Van Steen, and H. J. Sips, "Tribler: a social-based peer-to-peer system," *Concurrency and computation: Practice and experience*, vol. 20, no. 2, pp. 127–138, 2008.

[69] N. Zeilemaker, M. Capotă, A. Bakker, and J. Pouwelse, "Tribler: P2p media search and sharing," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 739–742.

[70] V. Buterin, "A next generation smart contract & decentralized application platform," *Ethereum Foundation*, 2013.

[71] D. Boneh, E. J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Lecture Notes in Computer Science*, vol. 3378. Springer Verlag, 2005, pp. 325–341.

[72] K. Peng and F. Bao, "An efficient range proof scheme," in *2010 IEEE Second International Conference on Social Computing*. IEEE, 2010, pp. 826–833.

[73] F. Boudot, "Efficient proofs that a committed number lies in an interval," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2000, pp. 431–444.

II

Supplementary Material

The second part of this thesis comprises supplementary material serving as an accompaniment to the thesis article of Part I. This material is composed of four chapters: 1 background information, 2 extended related work, 3 implementation details, and 4 extended analysis.

Supplementary Material to “Distributed Attestation Revocation in Self-Sovereign Identity”

1

Background Information

As Self-Sovereign Identity is a relatively new scientific field with its origin outside of academia, this section aims to provide the reader with an understanding of the concept. Firstly, the terms *identity* and *digital identity* are discussed as well as the history of Digital Identity Management Systems. Next, difficulties in the current ecosystem are laid out, the different definitions of SSI, and our theoretical framework encapsulating them is introduced. Finally, the opportunities and challenges of SSI are discussed.

1.1. Identity

Identity has a broad spectrum of definitions. The terminology itself stems from the Latin word for *sameness*, namely *identitās* (Merriam Webster, n.d.). Philosophy draws the distinction between *qualitative* and *numerical* identity (Noonan & Curtis, 2018). Qualitatively, identity is defined as entities sharing certain characteristics. Whilst numerically, we speak of total qualitative identity, thus requiring a set of characteristics that an entity only shares with itself. These characteristics are referred to as attributes (Camp, 2004). The notion of the numerical identity of a person through time is referred to as the *personal identity* (Olson, 2021). The foundations of this law can be traced back to Aristotle's *Law of Identity*, broadly stating that everything is equal to itself (Aristotle, 350 B.C.E./1925).

The requirements for the technical sense of identity are fulfilled most by the definition of the *numerical* variant, as it can be said that the goal of digital identity is to uniquely identify entities. Hence, *personal identity* may prove to fall short in such specification, as digital identity does not solely consider persons. Namely, ISO, 2019, p. 8 defines identity as "any set of attributes that describe a particular entity". Therefore, it can be stated that identity is the set of characteristics uniquely describing an entity. Hence, we make no distinction between human identity and the identity of software-based entities (e.g. Artificial Intelligence or IoT devices)

When such a characterisation is transformed to the digital domain, we speak of *digital identity*. The goals of digital identity are *identification* and *authentication* (Bertino,

2006). Where identification can be seen as the authorisation of one's identity (Camp, 2004) allowing the unique identification of a user in a system (IBM, 2021). Authentication is the act of proving one's identity. This can be achieved by three means:

- Something you know (e.g. a password).
- Something you have (e.g. a smartcard or key).
- Something you are (e.g. biometrics: fingerprint, face, etc.).

Often, measures are combined, referred to as *multi-factor authentication*.

1.2. Digital Identity

The Internet was not created with a method for knowing with whom or what you are communication with (Cameron, 2005). Even the conceptual OSI model (Zimmermann, 1980) does not contain a layer specifically designed for identity. As a consequence, there is no *digital identity*. The current digital ecosystem comprises one's digital presence through fragments of pseudo-identities. These pseudo-identities ultimately belong to a single entity and, thus, all attempt to be a digital identity. Of course, one is able to be identified digitally through these shards. However, these pseudo-identities lack the knowledge to fully uniquely identify an entity. We refer to this phenomenon as the *Sharding of Identity*. Each of these pseudo-identities often attempts to authenticate the same data. For instance, name, age, and a means of communication (e.g. e-mail). As such, they all can be labelled as being derivatives of one's actual identity: the true digital identity. One that is uniformly true and does not require indefinite copies for each new encounter.

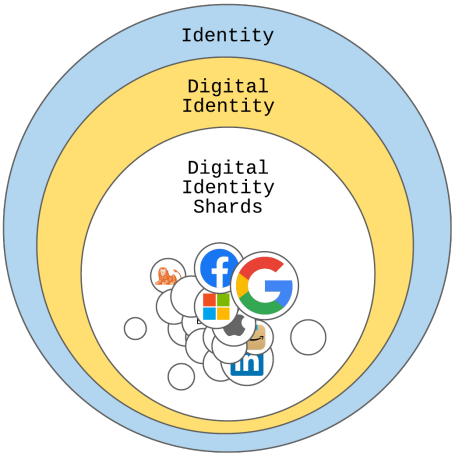


Figure 1.1: Identity groups

The relationship between these groups is visible in Figure 1.1. As visible, one's digital identity is a subset of one's physical identity, indicating that the digital identity is

invariably linked to the entity's physical identity. Furthermore, that the group of identity shards is a subset of that which the digital identity is. We note the overlap between identity shards, which is caused by a non-empty union of the attributes comprised by said shards. For instance, the vast majority of services require a registration per name. As such, most digital identity shards will have at least an overlap on this attribute. As may become apparent from this description, these digital pseudo-identities fall under qualitative identity as most of them share attributes with other shards. This follows naturally from the fact that each of them attempts to identify the same entity.

1.3. The Evolutions of DIMS

Allen, 2016 describes the four phases of digital identity. These phases group four types of Digital Identity Management Systems (DIMS'). With the fourth phase being SSI, this section discusses the first three as SSI is discussed more thoroughly in section 1.5. Hence, the following three prior evolutions[1] of digital identities exist:

Evolution One: Centralised Identity

With the onset of the Internet, centralised authorities such as IANA[2] and ICANN[3] became the issuers and authenticators of digital identities. For instance, the IANA determined the validity of IP addresses (IANA, n.d.), whilst the IANA managed the registration of domain names (ICANN, 2017). Next, in order to generate trust through certificates, Certificate Authorities were created, which were able to also delegate some power through hierarchies. Finally, as mentioned by Cameron, 2005, the distributed nature of the internet led to online services implementing their own digital identity management systems, which for the user often led to username and password combinations. All of the aforementioned organisations present in the Internet ecosystem are inherently centralised authorities, with the capabilities of revoking these identities. This comes with the consequence of users not owning any of their digital identities, as they are all either assigned to them or are managed by others. For instance, the registration of a domain name is performed on an annual bases (ICANN, 2017), allowing one to never fully own a domain name.

Evolution Two: Federated Identity

The second generation attempted to overcome the hierarchies, by imagining a *federated identity* (Chadwick, 2009; Pfitzmann & Waidner, 2003). An example of this is Microsoft's Passport initiative (PressPass, 1999), allowing identities across different domains. However, this initiative soon proved to be far from optimal, as it is comprised of a single authority. This was improved upon by allowing each site to remain an authority (Allen, 2016). However, users were not in control of their credentials (Vossaert et al., 2013).

[1] Although the chronological ordering of the phases is correct, we argue that the term *phase* is not correct for these specifications as phases indicate non-concurrent existence. For instance, the eight phases of the Moon do not exist simultaneously. Therefore, we propose the usage of the term *evolution*. As evolution indicates gradual development, whilst allowing simultaneous existence with prior iterations. Note that evolution does not necessarily indicate improvement (Hall et al., 2008), which is also not insinuated by the term *phases*.

[2] For IANA, see: https://www.iana.org/

[3] For ICANN, see: https://www.icann.org/

Hence, there was a need for a new evolution catering to the user aspect of digital identities, as opposed to the identity management aspect.

Evolution Three: User-Centric Identity

Currently, identity management systems are in the third generation, the *User-Centric Identity Management* (Jøsang & Pope, 2005; Recordon & Reed, 2006b). This generation attempts to put the user at the centre of their identity. Open-sourced examples of these include OpenID[4], OAuth[5] and FIDO[6]. These systems focus on user-centricity through consent and interoperability, allowing users to select their own provider.

Unfortunately, these efforts have mostly failed due to the register still being the owner of the identity. However, more problematic are the introduction of initiatives such as Facebook Connect (Morin, 2008) (contemporary known as Facebook Login[7]) or Google Identity[8]. Whilst these initiatives do allow selective sharing of identity information and regard user consent, they still store identities in a centralised fashion and are managed by a single commercial authority. The global adoption of these digital identities provided by big tech has led to an oligopoly, as portrayed by the market shares reported by Siftery, 2017. Where a regular oligopoly results in a price-wise disadvantage for consumers (Stigler, 1964), this technical oligopoly leads to asymmetrical control held by identity providers.

1.4. Challenges in the Current Ecosystem

The current ecosystem of digital identities suffers from several drawbacks and limitations, both from the perspective of identity providers and that of users.

1.4.1. Problems for Identity Providers

For identity providers, identification measures can prove to be a double-edged sword: whilst it allows them to manage their users' digital identities, allowing them e.g. to gather user statistics to improve their services, it can also prove to be a burden. Firstly identity providers must adhere to specific data compliance legislation such as the GDRP (The European Parliament and Council, 2016) or the PCI DSS (PCI Security Standards Council, 2004). Additionally, companies strive for international standards such as ISO/IEC 27001 (ISO, 2013). The leakage of Personal Identifiable Information (PII) cannot only lead to liability in accordance with said legislation (e.g., the GDPR has the possibility to fine companies in the millions), but also has side effects for the users. For instance, in case passwords are compromised, other services utilised by the user may be at peril or the leaked PII can be used for spear-phishing attacks and identity theft. Moreover, such losses can have a tremendous impact on the reputation of an organisation (Gatzlaff & McCullough, 2010).

[4] For *OpenID*, see https://openid.net/connect/
[5] For *OAuth*, see https://oauth.net/
[6] For *FIDO*, see https://fidoalliance.org/
[7] For *Facebook Login*, see: https://developers.facebook.com/docs/facebook-login/
[8] For *Google Identity*, see: https://developers.google.com/identity

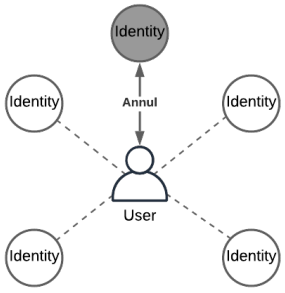
1.4.2. Problems for Users

On the other end, users must keep track of all their identities, often managing a multitude of credentials. On average employees of small businesses manage 85 passwords (LastPass, 2019). With the statistic that the use of brute-forced or stolen credentials is responsible for over 80% of the vulnerabilities utilised in breaches (Verizon, 2020), credentials continue to be a weakness in online identification measures. Moreover, users' information is stored in numerous locations, increasing the attack surface and the chances of their PII being stolen. For instance, Thales, 2020 reported that in 2020, 49% of US companies reported a digital breach to some degree.

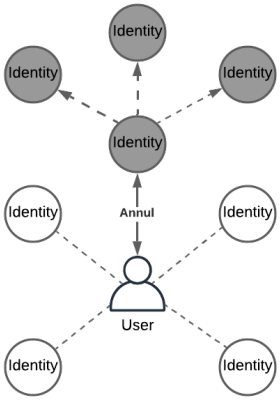
Furthermore, the oligopoly poses additional threats to users. The main issues regarding this oligopoly are (I) a disproportional balance of power, (II) privacy issues and (III) information asymmetries.

The Balance of Power

The disproportional balance of power is caused by the connection with other services. In a central identity, i.e. the service provider is also the identity provider, the user and the service provider hold relatively the same amount of power. More specifically, the user has the ability to terminate their usage of the service and, thus, losing a single digital identity shard. Similarly, the service provider has the ability to discontinue providing service to the user, revoking, in turn, a single digital identity shard and, thus, revoking access to a single service. This generates a balance of power within their relationship, as both of their abilities to annul the digital identity lead to a single loss. Figure 1.2a portrays this one-to-one annulment relationship. It is to note that this lays more delicately, as often the service provider generates value, shifting this balance in the favour of the provider. However, these details are not relevant for revocation as this is not an issue caused by DIMS'.



(a) Centralised identity



(b) Federated or user-centric identity

Figure 1.2: Balance of Power

When the identity provider manages a federated or user-centric identity, this balance

shifts. As a user desiring to annul such an identity will cascadingly annul his access to any connected service. Hence, they are often not able to discontinue any arrangement with them without affecting their arrangement with other service providers. On the other hand, as the identity provider has the ability to revoke one's digital identity, users may face loss of access to any services connected to said identity. For instance, in case a user is deemed to have breached a term of use. This has been visualised in Figure 1.2b, which portrays the imbalance of annulment power. Furthermore, due to the vastness of the platforms owned by big tech, issues such as deplatforming have emerged, in which high profiled controversial actors are banned from platforms in an effort to limit their reach (Rogers, 2020). This aids in further portraying the effects of the unbalance.

Privacy Issues

IBM, 2019 shows that 84% of the people believe that they have lost all control regarding the usage of their data by companies. Furthermore, trackers such as Google have been shown to be present in over 80% of websites (Karaj et al., 2018; The Duck, 2021). This aids in portraying the privacy issues experienced by users. The digital identities managed by big tech can further impact privacy, as they enable the gathering of more information on their users through other services. Any connected service has the potential to serve as a funnel for additional user data. The identity providers are essentially commercial parties, profiting from data received through managing these identities. This breach of privacy often comes hand in hand with the free to use service offered by digital identity service providers.

Information Asymmetries

The privacy concerns can lead to market mechanisms such as information asymmetries, due to the extra opportunities for data farming (Kshetri, 2014). As the identity providers possess large amounts of information on their users, any economic transaction made with them results in them possessing more knowledge than the other party. These concerns portray a need for a different approach to identification, breaking the oligopoly and creating the ability to generate trust over the Internet. Self-Sovereign Identity has the capabilities to fulfil this role.

1.5. Self-Sovereign Identity

There is no clear onset of Self-Sovereign Identity. Allen, 2016; Preukschat and Reed, 2021 refer to the work of *"What is 'Sovereign Source Authority'?"* (Loffreto, 2012) to be the first literature on the topic[9].

In their work, Loffreto describes the concept of *Sovereign Source Authority* (SSA). With SSA, Loffreto calls for an overhaul of the current national administrative identities. They refer to the current system as lacking the ability to provide a real identity, as current identities can be seen as a registration process for participation in society. SSA can be seen as a need for what Loffreto refers to as *human identity*. This falls in line with the Identity Groups as discussed in section 1.2.

[9] Allen, 2016 misattributes this work to Moxie Marlinspike, the co-founder of the messaging service Signal. However, presumably, this was performed on purpose (Sheldrake, 2016).

Principle	Description
Human Life	An SSI originates from an individual human life.
Human Identity	The human identity is the source authority of an SSI.
Attestations	An SSI has no personal control or authority until it is attested to by others.
Unpragmatic	SSI is not to be pragmatically defined as it is a function of time and place.

Table 1.1: The principles by Loffreto (2016)

Loffreto's main argument for alternative identity systems is comprised of societal participation being a choice, hence one must be able to have a valid identity without participation. Loffreto, 2012, para. 2 states that *"Within any Society, Individuals have an established Right to an 'identity'"*. The term *Sovereign Source Authority* itself did not gain much traction. However, it did lead to the coining of the term *Self-Sovereign Identity*. Whilst no key literature has been identified for coining the term itself, it can be said that SSI has gained traction due to Christopher Allen. Allen has often been erroneously credited for the invention of the term *Self-Sovereign Identity*, however, has explicitly credited Loffreto. Four years later, in 2016, Loffreto made another essay explicitly discussing SSI. Loffreto, 2016 describes four properties of SSI, which have been summarised in Table 1.1. As becomes apparent from this description, Loffreto does not consider SSI to be a digital technology, but more a digital concretisation of the human life, capable of authenticating the human identity.

Later in the same year, Allen, 2016 released their essay "The Path to Self-Sovereign Identity", which, undeniably has been a major influence on the field, with all major publications referencing said work, e.g. Baars, 2016; Ferdous et al., 2019; Mühle et al., 2018; Stokkink and Pouwelse, 2018; Tobin and Reed, 2016. They describe ten principles to which SSI is to adhere to. However, often uncredited literature is the work "Laws of Identity" (Cameron, 2005), where *Laws* is used in the scientific sense. In their work, published more than a decade prior to any literature directly referencing SSI, Cameron, 2005, p. 3 calls for a need for "a unifying identity metasystem". Furthermore, the concepts of digital *subjects* and *claims* are introduced, making way for claim-based identities. This work describes a large number of fundamentals of SSI, however, is often disregarded in the literature on SSI. Although, it is a highly influencing article in DIMS in general, with laws being implemented in systems such as OpenID 2.0 (Recordon & Reed, 2006a). Hence, we make the case that SSI was created in 2005, at the very least the foundations. These laws explain the shortcomings and successes of digital identity systems and, as such, are applicable to SSI.

1.5.1. The Laws of Identity

As mentioned Cameron, 2005 describes the seven laws of identity, which DIMS are to adhere to. Whilst not directly calling for sovereignty over digital identity, the majority of principles described can be identified in contemporary notions of SSI. The following laws are posed:

1. **User control and consent**: a DIMS must only reveal personal information given prior consent by the user. Through this law, trust can be built between the system

and the user.

2. **Minimal disclosure for a constrained use**: the solution which discloses the least amount of and best limits the use of PII, is the most stable long-term solution. This law minimises risk, as it is assumed that a breach is always possible.

3. **Justifiable parties**: disclosure of data with third parties must always be justifiable in a given identity relationship. Through this law, the user is aware of any third parties with whom is interacted whilst sharing information.

4. **Directed identity**: a DIMS must support omnidirectional identifiers, which can be said to be public, and unidirectional identifiers, which can be said to be private, enabling identification whilst facilitating privacy.

5. **Pluralism of operators and technologies**: a DIMS must support multiple identity technologies run by multiple identity providers. This law enables technologically agnosticism, disallowing vendor lock-in and encouraging the use of open standards.

6. **Human integration**: a DIMS must incorporate the user as a component of the system, offering protection against identity attacks. This law attempts to bridge the discontinuity between the actual (human) users and machines with which they communicate.

7. **Consistent experience across context**: a DIMS must allow for a separation of domains, whilst enabling consistent experiences across them. This law thus enables interoperability across different operators and technologies.

Whilst not coining a specific term for such a system, we do identify key aspects relevant to SSI which were later—in an adapted form—reiterated in the conceptualisation by Allen, 2016.

1.5.2. The Path to Self-Sovereign Identity

Allen, 2016 is undeniably the most commonly referenced literature with respect to SSI. In their work, the following set of *principles* are posed:

1. **Existence**: users must have an independent existence. I.e., a sovereign identity does not solely exist digitally. As a result, it can be interpreted as requiring to be tied to a physical entity.

2. **Control**: users must have control over their identities. This entails full authority over the user's own identity: the ability to share, update, and even hide.

3. **Access**: users must have access to their own data. Similarly to the above principle, users must be able to access all of their data.

4. **Transparency**: all involved systems and algorithms must be transparent. This entails open standards and open-source software.

5. **Persistence**: identities must be long-lived. Identities should, thus, exist until destroyed by the user.

6. **Portability**: information and services regarding identity must be transportable. I.e., identities must not be held by a single third party, as they may not support it live-long.

7. **Interoperability**: identities must be as widely usable as possible. This ensures that digital identities can be globally deployed. This property is aided by the *Transparency* principle, as open standards allow for more seamless integration with other systems.

8. **Consent**: users must agree to the use of their digital identity. This principle strengthens the *Control* principle, as the sharing of identity data may only occur with the consent of the user.

9. **Minimalisation**: disclose of identity data must be minimised. I.e., the minimal amount of information must be disclosed when sharing identity data. This principle focuses on privacy and prevents misuse of data.

10. **Protection**: the rights of users must be protected. The rights of users must take precedence over the identity network itself.

The above set of principles is often adhered to as a set of requirements (Mühle et al., 2018). These principles portray that digital identities must be tied to the human, which is the most important entity in the system. Furthermore, human control is key to the design. We note a large overlap with the work of Cameron, 2005. The laws “User control and consent"; “Minimal disclosure for a constrained use"; “Pluralism of operators and technologies"; “Human integration"; and “Consistent experience across context" can be directly identified. In addition to these ten principles, Stokkink and Pouwelse, 2018 add the principle of *Provability*: claims must be provable, as otherwise they can be deemed worthless. Tobin and Reed, 2016 builds upon these ten principles by subdividing them into three categories:

- **Security**: aims to keep the digital identity information secure. This consists of *Protection*, *Persistence*, and *Minimisation*

- **Controllability**: focuses on the user-centric foundation of SSI. This consists of *Existence*, *Persistence*, *Control*, and *Consent*.

- **Portability**: this requirement results in the user not being tied to a single provider and being able to use their identity without bounds. This consist of *Interoperability*, *Transparency*, and *Access*.

The additional principle defined by Stokkink and Pouwelse, 2018 can be categorised into *Security*, as the provability of claims aids in generating trust.

1.5.3. Critique of the Term

Sovereignty is defined as "[a] supreme authority within a territory" (Philpott, 2020). In terms of *Self-Sovereign Identity*, this would translate to *supreme authority over one's identity*. This term is prone to misinterpretation. As *supreme authority* insinuates that one has the full power of some territory. However, the extent to which this power reaches is open for interpretation. For instance, Good ID, 2021 defines *Self-Sovereignty* as "a feature of an ID or identity system, whereby, individual users maintain control over when, to whom, and how they assert their identity". There exists a discrepancy between this definition and the definition created using the definition by Philpott, 2020. We identify the same discrepancy in literature. For instance, the works set out by Loffreto, 2012, 2016 portray a philosophical nature of SSI, not necessarily indicating the usage of DIMS'. DIMS' are merely an implementation that, allows a realisation of SSI. Furthermore, proposed solutions such as Ferdous et al., 2019; Khovratovich and Law, 2017; Lundkvist et al., 2016; Zhou et al., 2019 do not necessarily adhere to this description. However, the case can be made that Stokkink et al., 2020; Stokkink and Pouwelse, 2018 as well as our proposed scheme, allow for the human identity as the origin of source authority, as described by Loffreto, 2016. It can be noted that SSI and DIMS' are undeniably intertwined, having led to misinterpretations of the term itself. Concerns for the usage of the term have been raised (Cameron, 2018; Ruff, 2018). Common misconceptualisations due to the term itself, are the following (Ruff, 2018):

- **Self-sovereign means self-attested**
  The term sovereignty implies total dominion and, as such, could lead to self-attestation. However, even in the descriptions proposed by Loffreto, 2016, claims require attestations. We do believe that verifiable claims allow for self-attestations, as in certain instances verifiability through others is simply not required. However, it is not the case that a self-attested nature is a given.

- **SSI attempts to reduce government's power over an identity owner**
  This claim we deem invalid due to a multitude of reasons. Firstly, SSI, as is generally true for any form of technology, is not an entity that can act, hence it is inherently neutral. The realisation and usage of SSI could impact a government's power in the identity domain due to shifts in ownership. Loffreto, 2012, 2016 do propose SSI as an alternative to the centralised governmental identities, as they deem the centralised registration unnatural. Secondly, the case can be made that the traditional governmental identity can evolve into SSI. With active plans from the European Commission to introduce a European Digital Identity, wide-spread SSI may even be introduced by the government (European Commission, 2021b). Moreover, SSI can prove to not delegate any power from governments as they can simply become an attestor to a digital identity, making them intrinsically an authority. As governments can be considered a commonly accepted authority, the network will most probably acknowledge—and even require—the government for verification of a digital identity. SSI can even prove to aid governments by reducing the need for maintaining identities.

- **SSI gives absolute control over identity**
  This misconceptualisation is most likely caused by the ambiguous nature of the

term *Self-Sovereignty*. As we established that *self-sovereignty* does not lead to *self-attested*, the dependency on attestations directly deteriorates the level of control one has over claims. As some claims simply require authorities to attest (e.g. a driver's license), the lack of such an attestation may lead to a weak claim. Hence, whilst one does have the power to self-attest, one's self-sovereign identity will still be dependent on others. For instance, it is a possibility that digital identity will only become valid in case it is attested to by a governmental institution, as otherwise there is no neutral party in which one can build trust for a claim. Hence, there is no true sovereignty over what attributes one has, but merely, sovereignty over what happens with said attributes.

The above critiques and misconceptualisations sketch the ambiguous nature of the *sovereignty* side of SSI. It leads to a need for a more defining term. Ruff, 2018 proposes the use of *decentralised identity* as an intermediary term. However, the major shortcoming of this term is that it does not convey the level of control that a user has. As, in order to be classified as *decentralised*, a system must simply consist of multiple parts which collaborate in order to achieve some goal. Hence, the selection of *decentralised identity* is not strict enough in order to explicitly convey the users' rights. Therefore, we propose the usage of the term *Self-Governed Identity* (SGI) in order to specify and distinguish what literature most commonly refers to as SSI from the more anarchic SSI discussed by Loffreto, 2012, 2016.

To govern can be defined as "to conduct the policy, actions, and affairs of (a state, organization, or people) with authority" (The Oxford Dictionary, n.d.). When placing this definition in the context of *identity*, one would be able to conduct the policy, actions, and affairs of one's identity. This constraints the power of the principles behind SSI, as *self-governed* does not imply total dominion over one's identity as sovereignty does. As a consequence *self-governed* implies that one has full control of one's identity, whilst not necessarily defining what the identity is. This flows naturally from the instances in which identity is to be assigned to one. For instance, in a society the government possesses the power to delegate identities, hence, SSI will most likely have to adhere to this structure. However, this does not mean that SSI itself must force this behaviour, as self-attestations have valid use-cases as validity is ultimately determined by a verifier.

The term *Self-Governed Identity* can prove to encapsulate these unavoidable unbalances of power. SSI can be seen as a digital alternative as opposed to a digital revolution, as it is unavoidable that certain authorities continue to exist in the digital domain in order to safeguard the identity. However, we believe that the most important nature of SSI and, subsequently, the more lenient proposition of SGI is to place data back in the hands of citizens and to provide the digital domain with valid identities and verifiable information without the need for a central authority. However, in the remainder of this document, the term SSI is used as opposed to SGI in order to be in line with the majority of the literature.

1.5.4. The Pyramid of Sovereignty

The previous sections portray a crisis in terms of both definition and the naming of the *Self-Sovereign Identity* concept. We believe that this is mostly caused by the unacademic

origin of SSI. As such, we propose a new set of principles based on the commonly cited works of Allen, 2016; Cameron, 2005, however, also taking into account the literature that sketched the beliefs of SSI (Loffreto, 2012, 2016). We propose the pyramid of sovereignty as presented in Figure 1.3.

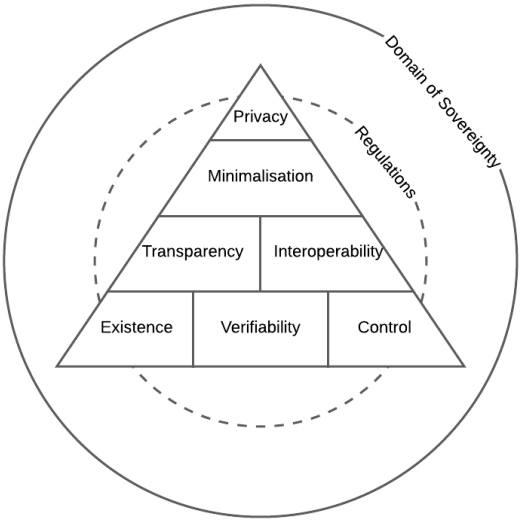


Figure 1.3: The Pyramid of Sovereignty

The main pyramid consists of seven principles, having overlap with Allen, 2016 and Cameron, 2005. The cornerstones of the framework are *existence* and *control*. Where most literature requires a link with a human identity (Loffreto, 2016; Speelman, 2020), we state that SSI must be linked to an entity in order to exist. We argue that this is a necessity for the long-liveness of SSI. Especially for the ongoing fourth industrial revolution (Moore, 2019), in which SSI can prove to fulfil a prominent role in the communication with IoT (Sovrin, 2019). *Control* enables the user-centric nature of SSI, allowing complete access, consent, and usage of the data stored by an identity for the user. Allen, 2016 splits this up in *control*, *access* and *consent*. However, we argue that control implies the requirement of consent, as in full control no action is to be performed without knowledge of the one in control. Similarly, we argue that control implies access and choice of storage location. Hence, we deem the term *control* sufficient for enabling the user-centric nature. Furthermore, the bottom layer of the pyramid is reinforced by *verifiability*: a property not explicitly mentioned in most literature, apart from Stokkink et al., 2020. However, we deem verifiability to be one of the main foundations of SSI. As without verifiability, information holds little value.

The second layer consists of *transparency* and *interoperability*. Where transparency strives for the usage of open standards and implementations, of which the very least the details of used algorithms and protocols are openly defined. This aids in making SSI

more accessible and ensures that the principles are adhered to due to publicly accessible implementations. *Interoperability* ensures that a user is not locked in a specific implementation of SSI, allowing communication with other services and possibly other systems. This aids in both ensuring users' rights as well as the adoption of SSI through usage with existing identity solutions.

The third layer is comprised of *minimalisation*: this principle ensures that no more information is shared than is required. This also entails that no information is shared with parties that do not explicitly require it. This falls in line with the comparable law posed by Cameron, 2005.

The fourth layer consists of *privacy*. The combination of all previous layers allows one to achieve a certain degree of privacy. Of course, no full privacy is achievable when sharing personal data. However, the system must attempt to guarantee a certain level of privacy. Which is especially reinforced by the *control*, *transparency* and *minimalisation* principles.

Finally, the Pyramid is contained by two shells. The inner shell represents regulations imposed upon the system by governments. For digital signatures (and therefore also attestations) to be legally valid, they must adhere to legislation (European Commission, 2014; Netherlands Enterprise Agency, n.d.; PwC, 2020). This may deteriorate the strength of some of the principles. For instance, privacy may be deteriorated by derogations such as (European Commission, 2020). This is visualised by the intersection with the inner shell. The outer shell represents the *Domain of Sovereignty*, in which the further the pyramid nears the bounds of the domain, the higher the degree of sovereignty. As mentioned previously, Loffreto, 2012, 2016 describes a more anarchic nature of SSI than most other literature envisions SSI to be. The outer bounds of this domain represent this level of sovereignty. As is visible, the proposed framework is restricted in its levels of sovereignty.

1.6. Opportunities & Challenges

The previously discussed notions portray an alternative to the current ecosystem of identity systems with potentially far-reaching implications. As such, we deem it necessary to discuss the implications of such a system. This section firstly describes privacy and usage implications and then opportunities for economic inclusion and certificates.

The End of Privacy

A major case for SSI is the introduction of legally valid digital identities, a concept that is beginning to be properly defined by governments (European Commission, 2021a). In case governments back such a system, the possibility for legally bound digital identities opens up. It can be said that the current ecosystem of managing digital identities can be quite cumbersome for online services, as such they may opt to require such a system. A major benefit of this is the possibility of eliminating bots and spam accounts. Without a digital identity attested to by a government, a platform may choose to deny service to

such users. After all, in case SSI is adopted by the government there is no reason that a human has no access to such an identity. In case each service requires authentication through SSI, we can speak of an end to digital privacy as one able to be uniquely identified to a legal extend. Furthermore, this makes tracking across contexts effortless. This opens up the possibility for more data farming and targeted advertisements and for attackers more personalised spear-phishing attempts.

As such, a certain degree of anonymisation should be implemented in order to prevent such a scenario. For instance, services should not require your full legal name apart from when they are obliged by legislation to gather such information (e.g. an e-commerce platform or a financial service provider). Furthermore, the usage of Zero-Knowledge Proofs (Goldreich et al., 1991) can aid privacy.

Economic Inclusion

An often overlooked opportunity for SSI is economic inclusion: residents in countries devoid of proper (central) identity infrastructure, are excluded from essential services enabled through identification systems. World Bank Group, 2016 defines identification to be required for the following:

- Inclusion and access to essential services: e.g., healthcare, education, and financial services.
- Effective and efficient administration of public services, policy decisions and governance.
- Accurate measure of development progress in areas.

Hence, without any form of valid identification measures, these residents are devoid of essential services and are less likely to be able to improve their living conditions or receive aid. Globally there exist an estimated 1 billion people without valid proof of identity (World Bank Group, 2021). However, the usage of SSI on e.g. smartphones could allow them to generate a digital identity. With the statistic that almost 50% of the world population possesses a smartphone (Turner, 2021), SSI may aid them in overcoming these issues.

Alternative for CAs

As discussed prior, the Internet uses certificates created by Certificate Authorities for delegating trust to online services. A system that suffers from single points of failure and MITM attacks (Allen, 2016). SSI can prove to overcome this current system through verifiable claims. As services can authenticate their user through SSI, likewise users can verify service providers. Currently, this verification is performed through the aforementioned certificates, requiring centralised infrastructure for managing revocations and trusted authorities. SSI can replace this through services presenting their identity via verifiable claims, achieving bidirectional authentication.

2

Extended Related Work

In this chapter, additional related work is discussed. Where the article focused on related work in distributed revocation, this chapter discusses related work in Self-Sovereign Identity. Commercially available SSI solutions are discussed, as well as proposed SSI systems in academia.

Sovrin

Sovrin[1] (Khovratovich & Law, 2017; Reed et al., 2016; Sovrin, 2018; Tobin & Reed, 2016) is an SSI solution created by The Sovrin Foundation. They use a public permissioned blockchain, the Sovrin Network, consisting of *members* (users) and *stewards* (verification nodes). The foundation itself manages the identity network. The network comprises two layers of nodes. The outer layer consists of *observer nodes* managing read-only copies of the blockchain. The inner layer consists of *validator nodes*, managing read and write copies. No private data is stored on the actual blockchain, only references using the DID[2] standard. Sovrin uses the blockchain identity framework *Hyperledger Indy* [3] and incorporated their designs into the framework. Therefore, we omit the discussion on Hyperledger Indy. Sovrin has no support for offline verification. We note that the usage of validator nodes introduces authorities in the network, going against the principles of Self-Sovereign Identity. Moreover, as discussed in the main article, the usage of blockchain introduces privacy and security issues (Hughes et al., 2019; Yli-Huumo et al., 2016) and deanonymisation (Biryukov & Tikhomirov, 2019). Also, their use may be limited by legislation (Xie, 2019).

Serto

Veramo[4] (Lundkvist et al., 2016; uPort, n.d.), formally known as uPort, is an SSI solution built on the Ethereum blockchain (Koens et al., 2018). Serto is compatible with

[1] For Sovrin, see: https://sovrin.org/
[2] For *DID*, see https://www.w3.org/TR/did-core/
[3] For Hyperledger Indy, see: https://www.hyperledger.org/use/hyperledger-indy
[4] For Serto, see: https://www.serto.id/

the DID standard from W3C. All uPort identities are stored on a single shared Ethereum smart contract. The on-chain contract stores references to the actual identity information stored off-chain. The usage of the Ethereum blockchain requires synchronisation in order to guarantee certainty on stored identity information. Furthermore, the single smart contract may introduce a large security risk (Praitheeshan et al., 2019).

IRMA

IRMA[5] (IRMA, n.d.; Privacy by Design Foundation, n.d.) is an identity solution backed by the Dutch government. The network is managed by the Privacy by Design Foundation[6]. Claims are stored on smartphones, however, the use of cryptographic keys requires communication with key servers, managed by the foundation. Keys are split up and are partially stored on said servers. Due to governmental backing, IRMA possesses some legally valid credentials. However, the Foundation manages its infrastructure and requires the aforementioned key splitting. This possibly impacts privacy and affects the *Control* principle discussed in our model in chapter 1, as data may not be portable to other systems. Furthermore, their key servers can be considered central authorities.

Decentralized Identifiers

Sovrin and Veramo utilise W3C's Decentralized Identifiers (DIDs) standard (W3C, 2021). DIDs link a subject to a document, that specifies identity information and specifics such as cryptographic types and verification methods. The format uses the URI specification to create a standardised format for decentralised identities. Their verifiable claims structure is based on the RDF standard [7], lacking the bit-serialisation required for signatures (Halpin, 2020). Furthermore, the DID standard does not mandate, from a technical perspective, the use of decentralised storage, hence, allowing centralised databases for storing identities to remain (Halpin, 2020).

Mühle et al., 2018

Mühle et al., 2018 describe an overview of SSI. They state that ISS differentiates itself from traditional identity management systems by being a user-centric model as opposed to service provider-centric. They describe two architectures for SSI: the *Identifier Registry Model* and the *Claim Registry Model*. Wherein the former model the pairing of identifiers and public keys of users are stored on-chain and claims off-chain. In the latter model, in addition to serving as a registry for identifiers and public keys, the claims themselves are also stored on-chain. They deem identification; authentication; verifiable claims; and attribute storage to be the core components of SSI. They deem that Blockchains are a requirement for SSI schemes. However, our architecture showcases that this is not a necessity. We note that blockchains suffer from the aforementioned issues.

[5] For IRMA, see: https://irma.app/
[6] For Privacy by Design Foundation, see: https://privacybydesign.foundation/
[7] For the RDF standard, see: https://www.w3.org/RDF/

Der et al., 2017

Der et al., 2017 describe the opportunities and challenges for a digital revolution caused by SSI, moving the requirements of privacy and trustworthiness to the user, requiring them to provide evidence. The authority define *digital identity* as a temporal reflection of a regular identity, containing specific characteristics of identity, with varying levels of detail. A digital identity can be held by any type of entity and functions to use a particular service. In addition, a *secure digital identity* adheres to the requirements of *privacy* and *trustworthiness*. Where privacy leads to only authorised access to the identity and trustworthiness to the correctness of the attributes contained in the digital identity.

The authority note three opportunities for SSI. Firstly, SSI can counteract the oligopoly present in the management of current digital identities. Secondly, it can provide help to people living in crisis areas, as identities may no longer require ties to local government. Finally, SSI may help companies to adhere to the GDPR as privacy can be more easily implemented.

The challenges for SSI are also explained. It is stated that current digital identity services allow for a certain level of comfort by trading in a certain level of control over identity. Based on that assumption, the case is made that one of the core challenges of SSI is that the additional required administrative efforts of SSI must be sufficiently comfortable. However, no solution is proposed.

Stokkink and Pouwelse, 2018

Stokkink and Pouwelse, 2018 present a blockchain-based digital identity solution. It is stated to be an academically pure model for SSI. They state that the first half of the problem regarding the creation of such a model, is the need for Self-Sovereign Identity: identity holders must be identity owners. The second half of the problem is the need for legally valid signatures. They propose an identity system based on Trustchain (Otte et al., 2020). The use of this blockchain, together with the use of Zero-Knowledge Proofs and the chaining of claims, allows for the satisfaction of the principles proposed by Allen, 2016. Their claims use ZKPs and incorporate validity terms and allow for interchangeable signature algorithms. A reference implementation shows sub-second claim-verification performance. However, revocation in their protocol is yet to be fully solved, as it may still require an active check with the authority of an attestation.

Othman and Callahan, 2018

Othman and Callahan, 2018 describe their Horcrux protocol, a decentralised biometric credential storage option via blockchain using DIDs. The authors mention that the drawback of current biometric-based authentication systems is that they introduce a single point of failure for securing digital identities. This is caused by requiring a central authority for storing templates of biometric samples. The Horcrux protocol combines the SSI ecosystem with the Biometric Open Protocol Standard (BOPS)[8]. This is performed by dividing biometric templates into shares, which are then stored distributed. The actual shares are stored off-chain, but resolvers to the DIDs are stored on-chain. Their solution requires interaction with centralised verifiers for access to the SSI system, hence,

[8] For BOPS, see: https://standards.ieee.org/standard/2410-2019.html

introducing centralised authorities. We note that the reliance on central authorities goes against the principles of SSI, as discussed.

Zhou et al., 2019

Zhou et al., 2019 present EverSSDI, an SSI framework based on Ethereum smart contracts. The smart contracts store the fingerprints of claims. The design uses so-called *Ever-Service* servers to generate unique IDs for subjects. These servers also aid in the login procedure. The reliance on network operators may result in privacy issues in case they are compromised and furthermore limits portability for transferring credentials to other systems. Again, the usage of Ethereum smart contracts and a blockchain suffer from the aforementioned limitations.

Belchior et al., 2020

Belchior et al., 2020 propose their Self-Sovereign Identity Based Access Control (SSIBAC) model: an SSI access control scheme using a blockchain. The design works by creating a verifiable presentation (VP) from a verifiable claim. This VP is sent to a verifier, which confirms that the client holds the VC by verifying whether it satisfies a specific predicate. This approach achieves a throughput of 0.9 seconds per access control request. The drawback to the scheme is that the verification nodes are single points of failure, which is acknowledged by the authors. This introduction of their verifiers further introduce inequalities in the network, possibly leading to privacy issues.

3

Implementation Details

As mentioned previously, a referrence implementation of the architecture has been built on top of the IPv8 protocol. Furthermore, this reference implementation was used to create a demo Android application. In this section, we further discuss implementation details omitted in the main article.

3.1. Semantic Layers

The architecture has been developed on top of the Kotlin implementation of IPv8[1]. Kotlin version 1.4.21 has been used for the development environment. The implementation consists of three semantic layers, over-arched by logic facilitating communication between layers.

3.1.1. The claim layer

The first layer is the *claim layer*. This layer handles all interactions regarding the Zero-Knowledge Proof (ZKP) claims:

- Requesting claims
- Generating claims
- Verifying claims
- Storing claims

All claims are achieved through Zero-Knowledge Proofs. The selection of ZKP is abstracted through modularisation, enabling any client to propose a new proof type. As mentioned, the range proof by Peng and Bao, 2010 and the exact proof based on Boneh et al., 2005; Stokkink et al., 2020 have been implemented. Prior to a request, a client generates a new key-pair for the selected proof type. A request is made to an Authority

[1] For Kotlin-IPv8, see: https://github.com/Tribler/kotlin-ipv8

for generating a claim through an IPv8 payload. This request contains at least the public key, attribute name, and proof type. Additional metadata can be sent along (e.g., the application uses this for the proposal of values). After which, the sending client stores a reference indicating that it has made a request to the Authority, which times out automatically. On reception, the layer awaits explicit consent from the Authority and expects the Authority to assign a value for the requested attribute before the Zero-Knowledge Proof is created. Depending on the used key size, the layer splits up the IPv8 payloads in separate UDP packets. This is performed using a SHA-1 hash and packet numbers, allowing the receiving client to reconstruct the data and note any missing packets. Alongside the claim, the plaintext value is sent along, encrypted using RSA. We note that SHA-1 is not a secure hashing algorithm (Wang et al., 2005), however, it is used for compatibility with the Python implementation of IPv8[2]. On reception, the layer verifies the claim using the aforementioned stored reference, validating the used key, attribute name, and proof type. The claim, together with its hash, key, proof type, and plaintext value, is then stored in an SQLite database. However, similarly to the ZKPs, storage is modularised, allowing for different storage structures.

For verification, a payload containing the SHA-1 hash of the claim is presented to the Subject. After which, a reference to the request is stored. The receiving layer then awaits explicit consent from the user, before the corresponding ZKP claim is sent to the Verifier. Again, a reference for the request is stored. The requesting layer then awaits the claim and if deemed solicited the layer will send ZKP challenges, dependent on the used proof type. The Subject responds to the challenge, after which the Verifier can make a decision, as discussed in the main article.

3.1.2. Attestation layer

The attestation layer handles the procedures regarding attestations, creating attributes and credentials. As mentioned, our architecture differentiates between data structures as visible in Figure 3.1 and manages the following interactions:

- Attribute attestation
- Attribute presentation
- Attribute verification
- Attribute storage

Figure 3.1: Data structures

[2]For Py-IPv8, see: https://github.com/Tribler/py-ipv8

As the claim layer handles claims, the attestation layer manages the remainder of the data structures. Hence, in the implementation, the claim layer serves as a foundation to the attestation layer.

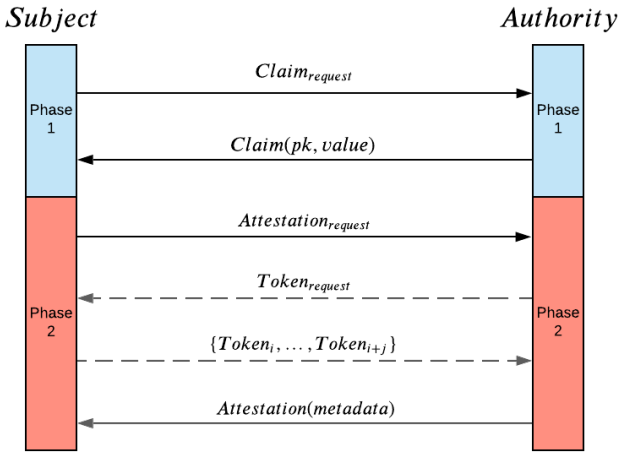


Figure 3.2: Full attestation procedure

Figure 3.2 portrays a more detailed schematic overview of the attestation procedure. After a claim has been generated in phase 2, a Subject requests an attestation for the claim from an Authority. For this, the attestation layer generates a metadata structure containing the attribute name, proof type, timestamp of the request, and the hashed plaintext value (using SHA3-256). Together with a subset of the previous tokens, the set of prior attestations and their corresponding authorities, this metadata structure is sent using an IPv8 payload. The Subject again creates a reference to the request. The receiving Authority then attempts to reconstruct the token chain and requests any missing tokens from the Subject as an absence of tokens can be calculated through the pointers within them as visualised in Figure 3.3. The Authority can build trust in the presented attribute through prior attestations. The attestation layer then verifies whether at least the prior mentioned metadata information is present and whether the timestamp is recent. If all information is deemed valid, the layer generates a signature over the hash (SHA3-256) of the metadata object and returns this attestation through a payload. This attestation, together with all other data structures are again stored in an SQLite database. Alternatively, the Subject can self-attest to a claim using the attestation layer by performing (relatively) the same steps as the Authority would.

The presentation procedure is similar to attesting to an attribute. However, the Subject additionally provides the plaintext value, encrypted using RSA, for verification. Verification in this layer is dependent on the procedure in the claim layer and on offline verification which is discussed in the overarching logic section. As we discussed the presentation and verification in the main article, it is omitted here.

Figure 3.3: Token chain

3.1.3. Revocation layer

The revocation layer relies on gossip in the IPv8 network. Per default, revocation advertisements are gossiped in an interval of 10 seconds to 5 neighbouring clients using pseudo-random subsets. The revocations are SHA3-256 hashes of metadata, hence allowing an Authority to revoke their attestation over a specific metadata instance. A revoking Authority revokes hashes in sets of arbitrary sizes and assigns a unique numerical incremental version (unique with respect to prior versions of the Authority, hence not globally). A signature is created using its IPv8 key over the version and the hashes. I.e. $sign(pk, v_i|r_0|\ldots|r_n)$, where v_i is the version number, r_x depicts the individual revocations and pk the Authority's public key. Based on 32 bytes per SHA3-256 hash and the average loss of 400 thousand identification documents (HM Passport Office & The Rt Hon Caroline Nokes MP, 2018). Ten years of revocations would take up merely 128 megabytes of storage, a requirement that is easily satisfied.

An advertising client sends a payload containing the hashes of the public keys of the Authorities that it acknowledges and their latest known versions. When advertising, the client stores references that allow the gossiped-to clients to request updates.

When receiving an advertisement, the layer verifies whether any versions are missing or are unknown and whether the key hash belongs to a Trusted Authority, handled by the overarching logic. After which, a request is made for each Trusted Authority containing an unknown version, by making the lower bound of the known versions apparent. Furthermore, a reference to allow the reception of this update is stored, providing the sender with a window of time in which information is accepted. Hence, further advertisements for the same value by the same gossiping client are ignored until said window passed. This counteracts DoS attempts by other clients, which could for instance occur through continuous advertisements of revocations, without actual transmissions. After receiving the request, the gossiping client sends the revocations. The revocations are sent using UDP packet splitting. The receiving client verifies the signature through the public key of the Authority and stores the revocations in the SQLite database. Furthermore, they are held in memory in the Bloom filter. The reasoning for only accepting revocations from Trusted Authorities is the prevention of spam and possible DoS attacks as a consequence of said spam (see section 4.2). The usage of hashes is privacy-preserving to a certain extent. The propagation of revoked hashes does not directly leak any information on the attestation itself. Only after a client is presented with the attestation, it is able to link the revocation to the attribute.

3.1.4. Over-arching logic

The overarching logic handles the connection with other peers, trusted authorities, and offline verification.

Communication & Identities

The overarching logic allows for the management of multiple identities. This logic stores unique IPv8 keys in separate containers allowing switching of identities on the same device. This is performed modularised, allowing for different storage types. As IPv8 connects with 30 other clients per default, the full network is not visible at all times. The over-arching logic allows for the visibility of specific clients. This is performed by a localisation token. This is created by a client and is comprised of alphanumerical characters. When shared, these tokens transform to unique community identifiers in IPv8, enabling the forming of a community with those aware of the token.

Trusted Authorities

Authorities are trusted by storing their public key and the hash of the said key. By storing their hash, the revocation layer is able to reduce overhead by only having to propagate public key hashes in advertisements. Furthermore, this strengthens privacy as the public key must be encountered before the hash is known.

Offline Verification

Offline verification requires the function of all the aforementioned layers. In addition to the presentation procedure mentioned in the attestation layer, a Subject presents the Verifier with their signature over a nonce in order to prevent replay attacks. Verification is then dependent on the following criteria:

- Whether the metadata signature is correct.
- Whether the hashed value is correct.
- Whether the list of attestors contains an Authority that is trusted.
- Whether the attestation is not revoked.
- Whether the challenge has not timed out.

The assumption for this verification is that revocations are received prior and presentation occur physically, enabling offline verification.

3.2. Mobile Application

The reference implementation has been used to create a demo application for smartphones. The resulting Android application enables validation of the usage of SSI (and our novel revocation mechanism) on handheld devices. This implementation has been validated in the aforementioned trial. The application requires Android API level 22, enabling distribution on 98% of the Android devices (StatCounter, 2021).

Figure 3.4 displays the main screen of the application. The QR-code display the public key of the client. When scanned using the action or the scanner button, the public key can be registered as a Trusted Authority (Figure 3.6a) or the client can be looked up in the network to request an attribute (Figure 3.6c). This QR-code also contains the aforementioned localisation token, enabling the visibility of specific clients. After adding a

client as an Authority, it is stored in the Trusted Authority Storage (see Figure 3.6b). In this screen, requests can also be directly made to clients visible in the network.

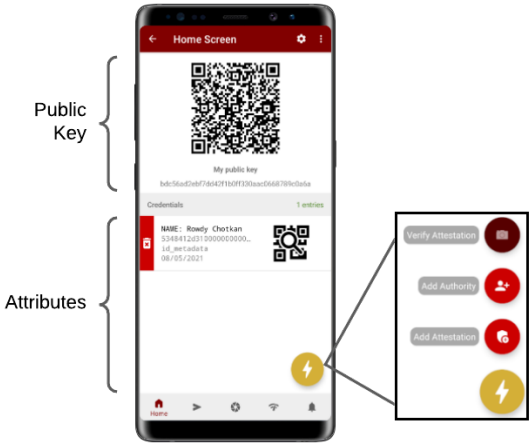


Figure 3.4: SSI Application home screen

Attributes are presented as visible in Figure 3.7a. Note the timeout bar indicating the window of validity for the challenge as discussed previously. A Verifier may scan the QR-code using the built-in scanner, after which they are presented with the options visible in Figure 3.7b. Where the former option is for offline verification and the latter option is for active ZKP verification. Offline verification directly leads to either success (Figure 3.7c) or failure (Figure 3.7d). Active verification requires further interaction: firstly active consent by the Subject (Figure 3.8a) after which the aforementioned ZKP challenges are sent (Figure 3.8b). Finally, revocation of attestations can be performed using the sent attestations screen visible in Figure 3.8c.

The signing of arbitrary data opens up the ability for larger credentials such as ID photographs. Figure 3.9a portrays an image being used as the claim value. This is performed by encoding the image data to the Base64 format. Subsequently, the Subject can display the image credential as visible Figure 3.9b. However, as a QR-code can hold up to 4296 alphanumerical characters (Keyence, 2019), an image results in too much information. As such, the QR-code is split up into two images (see Figure 3.9c). The first QR-code contains the attribute information and the second QR-code the encoded image value (allowing the Verifier to reconstruct the image).

3.2.1. Web services

For use-case evaluation, a reference implementation of an online service utilising the SSI implementation for authentication has been created. Figure 3.5 portrays the interface, in which a web service serves a QR-code to the user. This QR-code contains a credential name, the aforementioned nonce as well as the public key of the webserver. After scan-

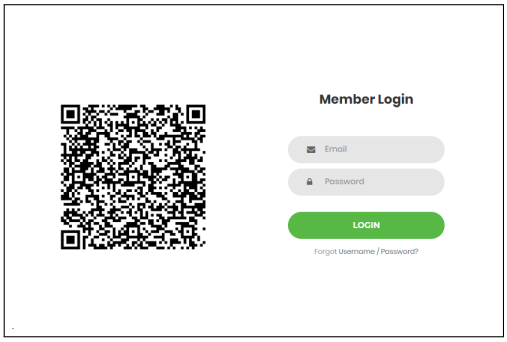
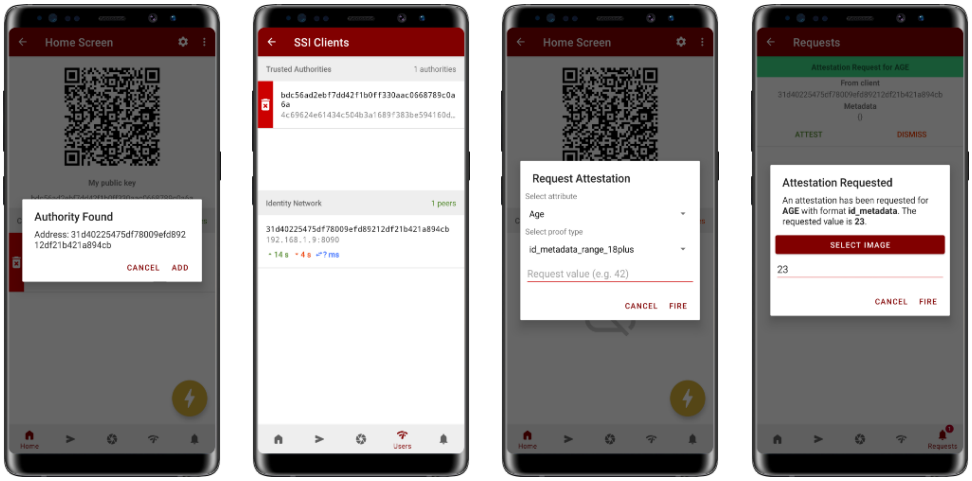


Figure 3.5: Web-service login example

ning the QR-code using the application, the Subject is requested to verify the credential as portrayed prior in Figure 3.8a. After which the web service is able to verify e.g. one's name and provide access to their profile. This procedure is secure against replay attacks due to the nonce. Furthermore, this authentication can be performed bidirectionally as discussed in section 1.6.



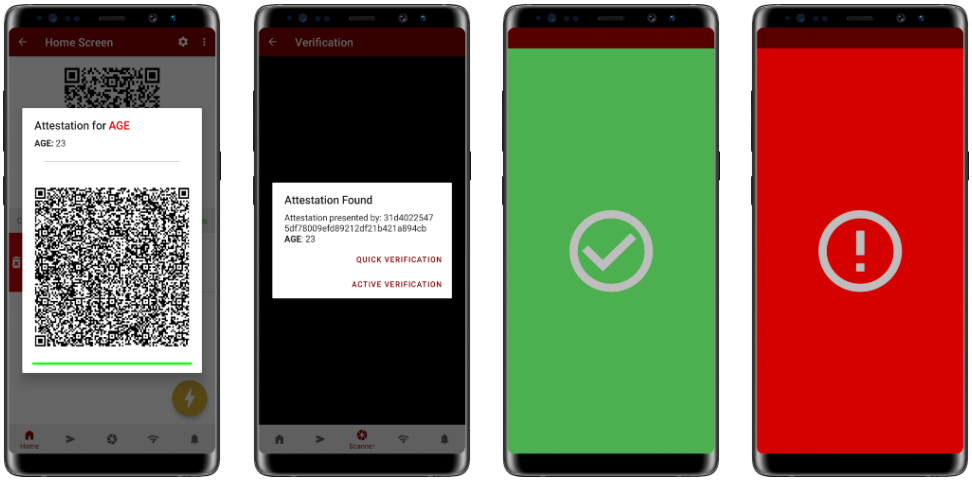
(a) Adding Trusted Authority

(b) Authority & network screen

(c) Attribute requesting

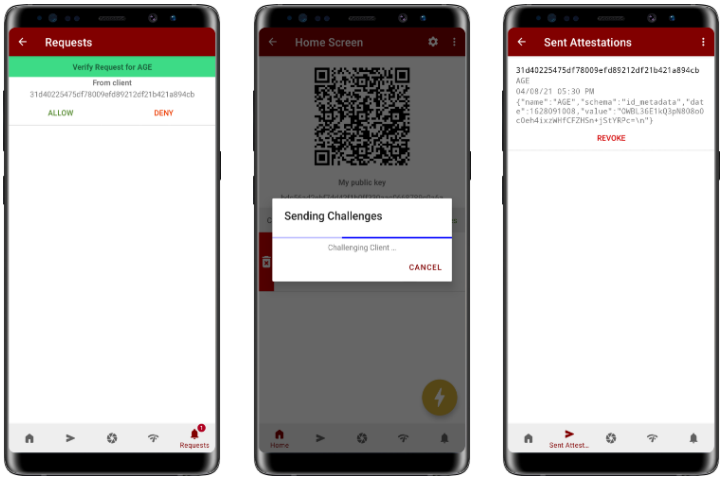
(d) Attribute signing

Figure 3.6: Attributes & network



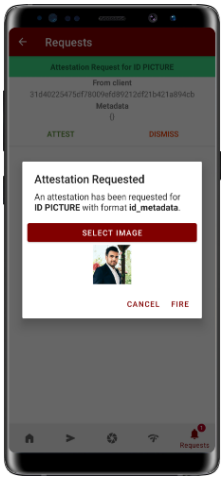
(a) Attribute presentation (b) Verification options (c) Verification passing (d) Verification failure

Figure 3.7: Presentation & verification

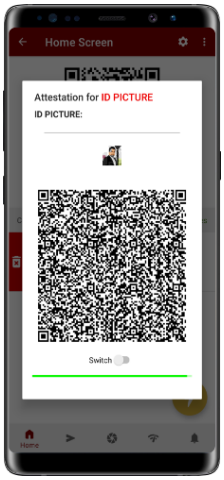


(a) Verification request (b) ZKP challenges (c) Revocation

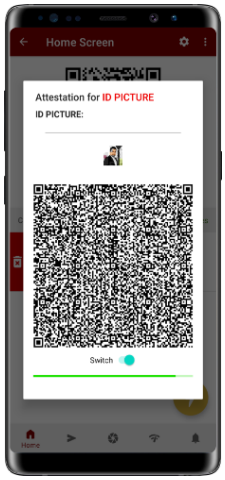
Figure 3.8: Active verification & revocation



(a) Image credential creation



(b) Image credential presentation I



(c) Image credential presentation II

Figure 3.9: Image credential

4

Extended Analysis

This sections discusses further analysis performed through emulation of IPv8 clients. Furthermore, privacy and security considerations are discussed as well as future work.

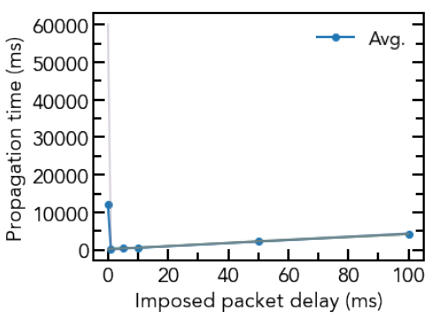
4.1. Emulation

Further analysis has been performed using the IPv8 implementation of the architecture. The workstation used to gather results was equipped with an Intel i7-6700HQ CPU clocked at 2.60 GHz, possessed 16 GB of RAM and allowed for the emulation of up to 11 simultaneous IPv8 clients. For revocations, we again generated datasets of 32 bytes SHA3-256 hashes. Revocations were split up into sets of 1000 in order to minimise the impact of a single packet loss. As IPv8 uses UDP, network congestion introduced additional constraints when using multiple clients. In the performed measures this was counteracted through manual delays. As such, some presented measure contains an additional adjusted result. For the default parameters, the gossip-interval t_g was set to $50ms$ in order to maximise the throughput of gossip. The number of selected peers m_p was set to 5 as this showcased a good trade-off between throughput and required system performance during the simulations. Additionally, this number is of little impact due to the low gossip interval. The number of revocations used is 1000 (unless specified differently), as this number leads to low impact of a single loss and the number of revocations scale linearly, as will be shown. Finally, each measure was repeated 10 times after 5 warm-up iterations in order to minimise system impact.

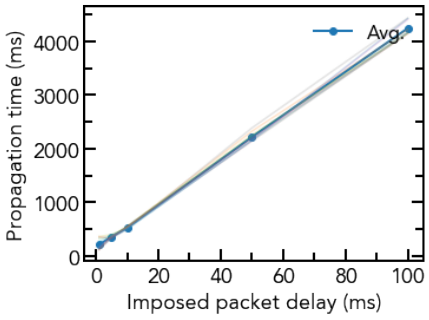
4.1.1. The Impact of UDP loss

As mentioned, manual UDP delays were introduced to counteract network congestion due to the number of packets and clients emulated on a single machine. Figure 4.1a portrays the impact of the imposed packet delay on the propagation time between two clients. As visible, the measures differ greatly due to packet loss leading to the receiving client ignoring further advertisements from the gossiping client until the current request timed out. Figure 4.1b portrays that the remained of the imposed delays affected the time

linearly, as expected. This effect is also measured for multiple clients in figure Figure 4.2. Hence, we conclude that the introduction of manual delays aids in preventing the effects of a single packet loss on the propagation time. Therefore, the remainder of the evaluations use similar delays as a countermeasure against network congestion in the IPv8 clients.

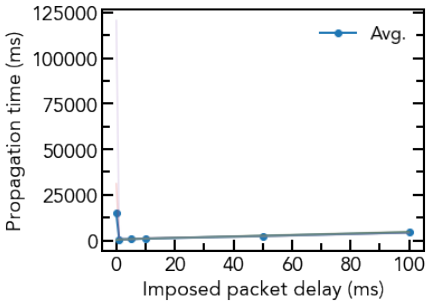


(a) All measures

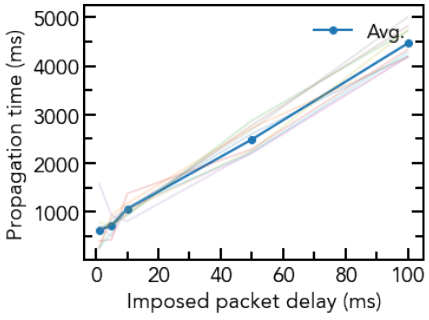


(b) First measure omitted

Figure 4.1: The impact of UDP delay on propagation time for 1 client



(a) All measures



(b) First measure omitted

Figure 4.2: The impact of UDP delay on propagation time for 10 clients

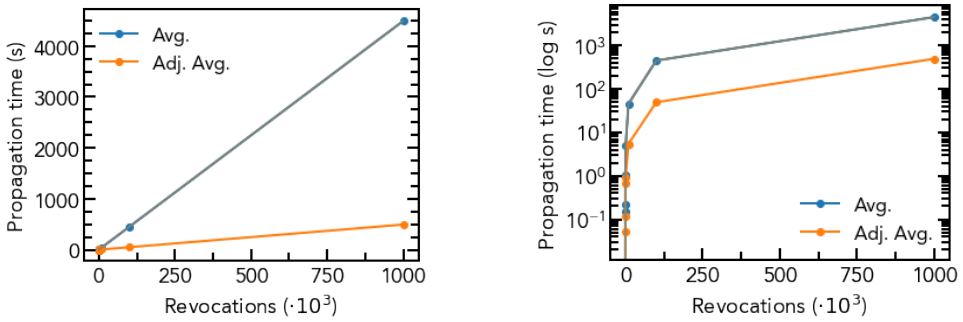
4.1.2. Revocation Amount

Figure 4.3a showcases the revocation scaling in a system of 1 gossiping and 10 receiving clients. As visible, the propagation time scales linearly with respect to the number of revocations. In this setup, up to 1 million revocations were used, with increments of factor 10. The adjusted rate showcases the performance adjusted for the manual delays. As visible 1 million revocations take roughly 500 seconds or 8 minutes. As this can be deemed more than two years worth of revocations, based on the statistic of annually lost identification documents (HM Passport Office & The Rt Hon Caroline Nokes MP, 2018), we deem this scalability usable. The corresponding Figure 4.3b showcases the results in logarithmic scale, further substantiating the linear scaling.

Furthermore Figure 4.4a portrays this scaling for a single client. As expected, on a single client the reception scales likewise linearly with the number of revocations. See also Figure 4.4b, portraying the results in logarithmic scale. A single client takes roughly 10 seconds to update 100 thousand revocations.

As IPv8 also supports the TFTP protocol, the measures for a single client were repeated using this protocol. Figure 4.5 portrays the results. Although the system appears to scale better on a lower number of revocations, the impact of packet loss made the protocol fail consistently across a higher number of revocations. Hence, no proper conclusion can be drawn on the usage of this protocol.

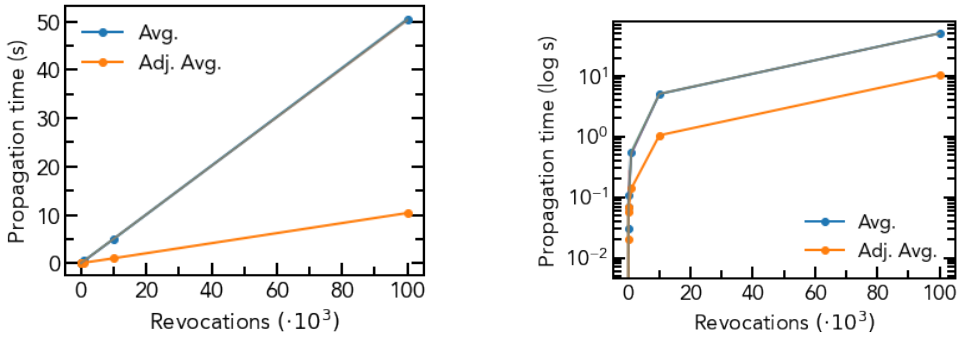
Compared to the simulation discussed prior, the performance is worse. We note that this can be explained mostly due to communication overhead caused by UDP packet splitting. Due to the limited sample size and network topology, further experiments are required to properly analyse the performance of our algorithm. However, these results portray that the scaling of revocations yields usable timings in a limited test environment.



(a) Normal scale

(b) Logarithmic scale

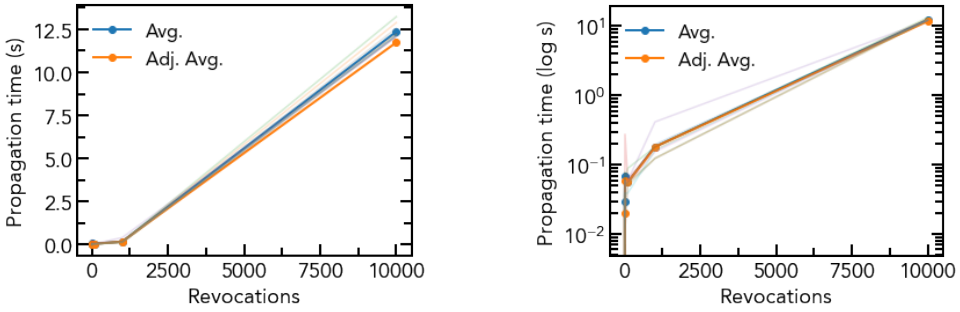
Figure 4.3: The impact of the number of revocations (10 clients)



(a) Normal scale

(b) Logarithmic scale

Figure 4.4: The impact of the number of revocations (1 client)



(a) Normal scale

(b) Logarithmic scale

Figure 4.5: The impact of the number of revocations using TFTP (1 client)

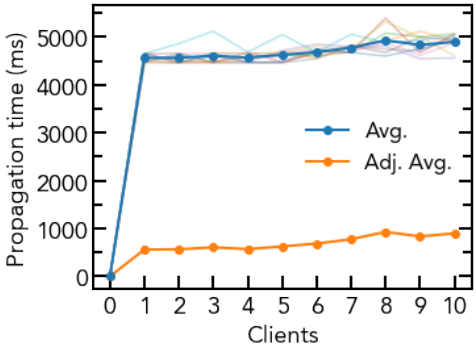


Figure 4.6: Scaling of Clients

4.1.3. Client Scaling

Figure 4.6 portrays the effect of the number of clients on the propagation time. As visible, the number of clients does not appear to have a large impact. It can be seen that the propagation time roughly doubles with the increase of 10 clients. However, due to the limited sample size, it is difficult to draw a conclusion on the scalability.

4.1.4. Bloom filter

Revocation validation is performed using the Attestation Revocation List in conjunction with a Bloom filter (Bloom, 1970). Based on the expected 400 thousand lost identification documents per year, as presented by HM Passport Office and The Rt Hon Caroline Nokes MP, 2018, the following memory and time considerations can be made. The storage for 400.000 hashes of 32 bytes each, results in a space usage of at least 12.21 megabytes. Where a Bloom filter of the same size ($n = 400,000$), with a probability of false positives of 1 in 17.2 million ($p = 5.8 \cdot 10^{-8}$) and 24 hash functions ($k = 24$), can achieve storage using merely 1.65 MB ($m = 13,872,594$). Whilst both such space requirements are easily satisfied by modern handheld devices, e.g. the average smartphone possesses over 4GB

of RAM (GSMArena, 2018), the run-time benefits do introduce a noteworthy improvement.

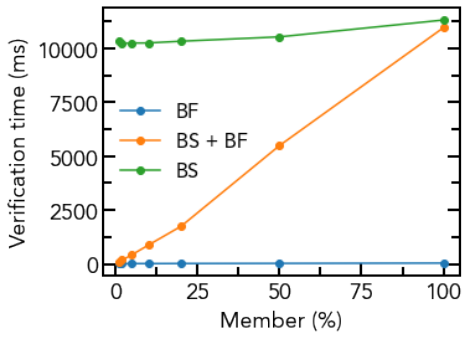


Figure 4.7: Verification time per 100 transactions

Parameter	Value
n	400,000
p	$5.8 \cdot 10^{-8}$
m	13,872,594 (1.65 MiB)
k	24

Table 4.1: Bloom filter parameters

Figure 4.7 showcases the analysis of verification time, where *BF* and *BS* stand for *Bloom filter* and *binary search*, respectively. The Bloom filter uses the aforementioned parameters summarised in Figure 4.1 and the 400,000 revoked SHA3-256 hashes were stored in an SQLite database. SQLite uses binary search for entry lookup (SQLite, n.d.). The x-axis varies the membership percentage, i.e., the percentage of hashes in the test set that are revoked. At each measure, a test set was generated of size 100, containing a certain percentage of revoked hashes. As expected, the verification solely using the Bloom filter (BF) is not impacted by this variation. The binary search verification (BS) remains similarly unaffected. The variation that only performs binary search on a possible match in the Bloom filter (BS + BF), is impacted the most. As becomes apparent, the benefits from the Bloom filter decrease with the increase of the membership percentage. Hence, the speed-up is most prominent with a lower membership percentage. In terms of attestation verification, a Bloom filter is thus most beneficent in case the vast majority of the encountered attestations are non-revoked.

We draw the conclusion based on the reported statistic of 400 thousand identification documents lost annually in the UK for its 50 million passport holders. (HM Passport Office & The Rt Hon Caroline Nokes MP, 2018), leading to an annual loss of 0.8% of all passports. Note that this estimation does not include the number of different identification documents held by a resident (e.g. driving license or identification card) and that the properties of physical identification measures do not directly translate to a digital variant. Thus, as a consequence, this actual number most likely differs greatly. However, this showcases that it is expected to encounter far more valid credentials than revoked ones, especially with the assumption that the majority of the network is honest. Hence, based on this comparison, we conclude that the speed-up benefit provided by the usage of Bloom filters is significant. Therefore, the hybrid solution overcoming their probabilistic nature during verification can be deemed the optimal candidate.

4.2. Privacy & Security

This section analyses the security and privacy of our architecture. We focus on different types of known network attacks as well as scenarios discussing possible weaknesses in our proposed architecture.

Eclipse Attacks

In *Eclipse attacks* (Castro et al., 2002; Sit & Morris, 2002) malicious actors attempt to isolate a specific node. In our algorithm, the successful execution of an Eclipse attack could lead to a client not being informed of the latest revocations. This, in turn, could lead to the verification of a revoked credential passing. The susceptibility of this attack relies on the implementation and network topology. In the reference implementation, IPv8 connects to a subset of the network for optimisation purposes, resulting in vulnerability of the Eclipse attack. However, this attack can be prevented through the use of credentials. The Eclipse attack can be mitigated by requesting identities of neighbouring clients, requiring those to be properly attested to by trusted Authorities. This introduces a reputation system, however, the requirement of attestations for credentials complicates the execution of Sybil attacks (Douceur, 2002). Depending on the overhead caused by the credential verification, the requirement of credentials may fully prevent both Sybil and Eclipse attacks. We note that this is a direction to further investigate.

Denial-of-Service Attacks

Denial-of-Service (DoS) attacks are an attempt to make a certain node inaccessible from the network. In our algorithm, DoS attacks are prevented by the usage of advertisements for gossip. The advertisements reduce overhead by firstly sharing metadata about known information, hence the revocations are not directly gossiped. Furthermore, the indirect refusal of revocations created by unacknowledged Authorities prevents DoS attempts through the spam of false revocations.

However, it is to note that this measure does counteract the propagation of revocations made by Authorities that are unacknowledged by the majority of the network. However, in the alternative that all revocations are stored by all clients, the number of locally stored revocations would grow to unmanageable amounts. We note that clients can forcibly create connections with Authorities, ensuring propagation. However, this goes against the distributed nature of our proposed algorithm.

Malicious actors are able to continuously advertise non-existing revocations from Authorities that are trusted. However, the use of signatures ensures that clients can verify the validity of revocations. As such, malicious actors that do distribute false revocations are detectable and may be ignored by an honest node. Furthermore, malicious actors may never send corresponding revocations after having sent a revocation advertisement. This can be performed in an attempt to DoS a node. This is counteracted in the implementation by not depending on a single advertisement for specific revocations. Hence, in this regard, malicious actors do not hinder the reception of revocations by a node.

Privacy

As discussed, the revocations are the hashes belonging to a *metadata* structure of a credential. In turn, an attestation is made by a signature over the hash of the metadata. This design choice impacts privacy. Each revocation is able to uniquely identify a credential. However, they only allow identification after the said credential has been encountered. Given the principles of SSI we assume that credentials are only shared with justifiable parties and, as such, the privacy of revocations is guaranteed to the extent that privacy is upheld when sharing credentials. However, we note that collusion can lead to the revoked credential being identified in case a Verifier shares this knowledge with the network.

Censorship

One may raise the argument that the revocation of metadata introduces censorship by Authorities. Authorities, indeed, possess the ability to revoke credentials to which they never attested. Note that this is possible because attestations point to a metadata structure and not vice versa, meaning that the absence of an attestation cannot be proven. However, this ability cannot be used for censorship as the validity of a credential lays in its attestations. Therefore, the lack of attestation by an Authority inherently leads to the assumption by a Verifier that said Authority does not vouch for the credential. Hence, an Authority attempting to impose censorship by revoking credentials to which it did not attest to leads to no difference in the outcome of verification. Furthermore, as the outcome of verification is decided by a Verifier, a revocation is merely a factor in said outcome as opposed to a binary truth.

Malicious Actors

Malicious actors may attempt to hide a revocation by withholding a specific attestation. The withholdment of attestations is of no impact on the outcome of the verification of credentials. Consider a Subject holding a credential that is revoked by an Authority. In other words, a single attestation is no longer valid. When a Verifier is presented with said credential the Subject has two choices: either he remains honest and presents the credential and all of its corresponding attestations or he hides the revoked attestation. Because the verifiability of a credential lays within its attestations, the withhold of an attestation that is revoked by the Authority leads to verification failure in either way. Similarly, to how the previously discussed censorship attack is of no consequence, the withholdment of attestation leads to failure as its indirectly assumed that the Authority (which revoked its attestation) does not vouch for the credential.

4.3. Future work

Depending on the number of revocations required in a deployed SSI system, the storage requirement may grow too large to be manageable by devices. As such, an alternative is the usage of probabilistic data structures such as Bloom filters (Bloom, 1970), Cuckoo filters (Fan et al., 2014) or cryptographic accumulators (Ozcelik et al., 2021). Where our proposed solution already uses Bloom filters for speed improvements, further usage of probabilistic data structures can prove to overcome the storage requirement. We opted

against the usage of these structures for their false positives of revocation during verification. As discussed, cryptographic accumulators have been used to facilitate revocation mechanisms, however, have been reported to provide overhead (IRMA, n.d.). However, the addition of such a space-efficient data structure to our gossip-based solution can prove to make the mechanism more accessible in terms of system requirements. As such, this interesting topic to further investigate.

Furthermore, we note that the results of the analyses of the implementations indicate that the reliance on UDP is a limiting factor for the throughput of revocations. As such, the investigation into the use of other protocols may yield improved performance. Apart from that, the investigation into more sophisticated retransmissions of lost revocations versions or packets can already improve the protocol. Moreover, as the analysis with smartphones and the emulation of IPv8 clients was performed with few devices, analysis with more clients may yield further insights into the actual usability and scalability of the protocol. Finally, we note that research into the impact of malicious nodes may be valuable, as well as simulations in more realistic network settings.

Bibliography

Allen, C. (2016). The Path to Self-Sovereign Identity. https://www.coindesk.com/path-self-sovereign-identity

Aristotle. (350 B.C.E./1925). *Metaphysics* (W. Ross, Ed.; T. I. C. Archive, Trans.) [Original work published 350 B.C.E.]. http://classics.mit.edu/Aristotle/metaphysics.4.iv.html

Baars, D. (2016). Towards self-sovereign identity using blockchain technology.

Belchior, R., Putz, B., Pernul, G., Correia, M., Vasconcelos, A., & Guerreiro, S. (2020). *SSIBAC: Self-Sovereign Identity Based Access Control* (tech. rep.). https://vonx.io/

Bertino, E. (2006). Establishing and protecting digital identity in federation systems. *Article in Journal of Computer Security*. https://doi.org/10.1145/1102486.1102489

Biryukov, A., & Tikhomirov, S. (2019). Deanonymization and linkability of cryptocurrency transactions based on network analysis. *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, 172–184.

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, *13*(7), 422–426.

Boneh, D., Goh, E. J., & Nissim, K. (2005). Evaluating 2-DNF formulas on ciphertexts. *Lecture Notes in Computer Science*, *3378*, 325–341. https://doi.org/10.1007/978-3-540-30576-7_18

Cameron, K. (2005). The laws of identity. *Microsoft Corp*, *5*, 8–11.

Cameron, K. (2018). Let's find a more accurate term than 'Self-Sovereign Identity'. https://www.identityblog.com/?p=1693

Camp, L. J. (2004). Digital Identity. https://doi.org/10.1109/MTAS.2004.1337889

Castro, M., Druschel, P., Ganesh, A., Rowstron, A., & Wallach, D. S. (2002). Secure routing for structured peer-to-peer overlay networks. *ACM SIGOPS Operating Systems Review*, *36*(SI), 299–314.

Chadwick, D. W. (2009). Federated identity management. *Foundations of security analysis and design v* (pp. 96–120). Springer.

Der, U., Jähnichen, S., & Sürmeli, J. (2017). Self-sovereign identity - opportunities and challenges for the digital revolution. *arXiv preprint arXiv:1712.01767*.

Douceur, J. R. (2002). The sybil attack. *International workshop on peer-to-peer systems*, 251–260.

European Commission. (2014). Regulation (eu) no 910/2014 of the european parliament and of the council of 23 july 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing directive 1999/93/ec. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%5C%3AOJ.L_.2014.257.01.0073.01.ENG

European Commission. (2020). Proposal for a regulation of the european parliament and of the council on a temporary derogation from certain provisions of directive 2002/58/ec of the european parliament and of the council as regards the use

of technologies by number-independent interpersonal communications service providers for the processing of personal and other data for the purpose of combatting child sexual abuse online. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%5C%3A52020PC0568

European Commission. (2021a). Proposal for a regulation of the european parliament and of the council amending regulation (eu) no 910/2014 as regards establishing a framework for a european digital identity. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2021:281:FIN

European Commission. (2021b). Regulation of the european parliament and of the council amending regulation (eu) no 910/2014 as regards establishing a framework for a european digital identity.

Fan, B., Andersen, D. G., Kaminsky, M., & Mitzenmacher, M. D. (2014). Cuckoo filter: Practically better than bloom. *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 75–88.

Ferdous, M. S., Chowdhury, F., & Alassafi, M. O. (2019). In search of self-sovereign identity leveraging blockchain technology. *IEEE Access*, *7*, 103059–103079.

Gatzlaff, K. M., & McCullough, K. A. (2010). The effect of data breaches on shareholder wealth. *Risk Management and Insurance Review*, *13*(1), 61–83.

Goldreich, O., Micali, S., & Wigderson, A. (1991). Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM (JACM)*, *38*(3), 690–728.

Good ID. (2021). Glossary. https://www.good-id.org/en/glossary/self-sovereignty/

GSMArena. (2018). Counterclockwise: RAM capacity through the years. https://www.gsmarena.com/counterclockwise_ram_capacity_through_the_years-news-30756.php

Hall, B. K., Benedikt, H., & Strickberger, M. W. (2008). Strickberger's evolution. *Strickberger's evolution* (4th ed., pp. 4–4). Jones & Bartlett Learning.

Halpin, H. (2020). Vision: A critique of immunity passports and w3c decentralized identifiers. *International Conference on Research in Security Standardisation*, 148–168.

HM Passport Office, H., Border Force, & The Rt Hon Caroline Nokes MP. (2018). Report your lost or stolen passport. https://www.gov.uk/government/news/report-your-lost-or-stolen-passport

Hughes, L., Dwivedi, Y. K., Misra, S. K., Rana, N. P., Raghavan, V., & Akella, V. (2019). Blockchain research, practice and policy: Applications, benefits, limitations, emerging research themes and research agenda. *International Journal of Information Management*, *49*, 114–129.

IANA. (n.d.). IANA — Number Resources. https://www.iana.org/numbers

IBM. (2019). Consumer Attitudes Towards Data Privacy. https://newsroom.ibm.com/Survey-Consumer-Attitudes-Towards-Data-Privacy

IBM. (2021). Identification and authentication - IBM Documentation. https://www.ibm.com/docs/en/ibm-mq/9.1?topic=mechanisms-identification-authentication

ICANN. (2017). https://www.icann.org/resources/pages/register-domain-name-2017-06-20-en

ISO. (2019). *IT Security and Privacy — A framework for identity management — Part 1: Terminology and concepts* (Standard). International Organization for Standardization.

IRMA. (n.d.). Revocation. https://irma.app/docs/revocation/

ISO. (2013). *ISO/IEC 27001 INFORMATION SECURITY MANAGEMENT* (tech. rep.). International Organization for Standardization.

Jøsang, A., & Pope, S. (2005). User Centric Identity Management. *AusCERT Conference 2005*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.1563&rep=rep1&type=pdf

Karaj, A., Macbeth, S., Berson, R., & Pujol, J. M. (2018). Whotracks. me: Shedding light on the opaque world of online tracking. *arXiv preprint arXiv:1804.08959*.

Keyence. (2019). *Basic practice of 2d codes* (Vol. 1). Keyence Corporation of America. https://www.keyence.com/ss/products/auto_id/barcode_lecture/basic_2d/qr/

Khovratovich, D., & Law, J. (2017). *Sovrin: digital identities in the blockchain era* (tech. rep.). The Sovrin Foundation. https://sovrin.org/library/sovrin-digital-identities-in-the-blockchain-era/

Koens, T., Ramaekers, C., & Van Wijk, C. (2018). *Efficient Zero-Knowledge Range Proofs in Ethereum* (tech. rep.). ING. https://www.ingwb.com/media/2122048/zero-knowledge-range-proof-whitepaper.pdf

Kshetri, N. (2014). Big data's impact on privacy, security and consumer welfare. *Telecommunications Policy*, *38*(11), 1134–1145.

LastPass. (2019). *THE 3RD ANNUAL GLOBAL PASSWORD SECURITY REPORT* (tech. rep.). LastPass. https://lp.logmeininc.com/rs/677-XNU-203/images/LastPass_State-of-the-Password-Report.pdf

Loffreto, D. (2012). What is "Sovereign Source Authority"? https://www.moxytongue.com/2012/02/what-is-sovereign-source-authority.html

Loffreto, D. (2016). Self-Sovereign Identity. https://www.moxytongue.com/2016/02/self-sovereign-identity.html

Lundkvist, C., Heck, R., Torstensson, J., Mitton, Z., & Sena, M. (2016). Uport: A platform for self-sovereign identity. https://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf

Merriam Webster. (n.d.). Identity. Retrieved July 11, 2021, from https://www.merriam-webster.com/dictionary/identity

Moore, M. (2019). What is Industry 4.0? Everything you need to know. https://www.techradar.com/news/what-is-industry-40-everything-you-need-to-know

Morin, D. (2008). Announcing Facebook Connect. https://developers.facebook.com/blog/post/2008/05/09/announcing-facebook-connect/

Mühle, A., Grüner, A., Gayvoronskaya, T., & Meinel, C. (2018). A survey on essential components of a self-sovereign identity. https://doi.org/10.1016/j.cosrev.2018.10.002

Netherlands Enterprise Agency. (n.d.). Using an electronic signature. https://business.gov.nl/regulation/electronic-signature/

Noonan, H., & Curtis, B. (2018). Identity. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Summer 2018). Metaphysics Research Lab, Stanford University.

Olson, E. T. (2021). Personal Identity. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Spring 2021). Metaphysics Research Lab, Stanford University.

Othman, A., & Callahan, J. (2018). The Horcrux Protocol: A Method for Decentralized Biometric-based Self-sovereign Identity. *Proceedings of the International Joint Conference on Neural Networks, 2018-July*. https://doi.org/10.1109/IJCNN.2018.8489316

Otte, P., de Vos, M., & Pouwelse, J. (2020). TrustChain: A Sybil-resistant scalable blockchain. *Future Generation Computer Systems, 107*, 770–780. https://doi.org/10.1016/j.future.2017.08.048

Ozcelik, I., Medury, S., Broaddus, J., & Skjellum, A. (2021). An overview of cryptographic accumulators. *arXiv preprint arXiv:2103.04330*.

PCI Security Standards Council. (2004). Payment Card Industry Data Security Standard (PCI DSS).

Peng, K., & Bao, F. (2010). An efficient range proof scheme. *2010 IEEE Second International Conference on Social Computing*, 826–833.

Pfitzmann, B., & Waidner, M. (2003). Federated identity-management protocols. *International Workshop on Security Protocols*, 153–174.

Philpott, D. (2020). Sovereignty. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Fall 2020). Metaphysics Research Lab, Stanford University.

Praitheeshan, P., Pan, L., Yu, J., Liu, J., & Doss, R. (2019). Security analysis methods on ethereum smart contract vulnerabilities: A survey. *arXiv preprint arXiv:1908.08605*.

PressPass. (1999). Microsoft Passport: Streamlining Commerce and Communication on the Web. https://web.archive.org/web/20071214080959/https://www.microsoft.com/presspass/features/1999/10-11passport.mspx

Preukschat, A., & Reed, D. (2021). *Self-sovereign identity: Decentralized digital identity and verifiable credentials*. Manning Publications Co. LLC.

Privacy by Design Foundation. (n.d.). Irma in detail. https://privacybydesign.foundation/irma-explanation/

PwC. (2020). Using electronic signatures in the netherlands during the covid-19 crisis. https://www.pwc.nl/nl/actueel-publicaties/assets/pdfs/using-electronic-signatures-in-the-Netherlands-during-the-covid-19-crisis.pdf

Recordon, D., & Reed, D. (2006a). Openid 2.0: A platform for user-centric identity management. *Proceedings of the Second ACM Workshop on Digital Identity Management*, 11–16. https://doi.org/10.1145/1179529.1179532

Recordon, D., & Reed, D. (2006b). Openid 2.0: A platform for user-centric identity management. *Proceedings of the second ACM workshop on Digital identity management*, 11–16.

Reed, D., Law, J., & Hardman, D. (2016). *The Technical Foundations of Sovrin A White Paper from the Sovrin Foundation* (tech. rep.).

Rogers, R. (2020). Deplatforming: Following extreme internet celebrities to telegram and alternative social media. *European Journal of Communication, 35*(3), 213–229.

Ruff, T. (2018). 7 Myths of Self-Sovereign Identity. https://medium.com/evernym/7-myths-of-self-sovereign-identity-67aea7416b1

Sheldrake, P. (2016). [On the misattribution in Allen (2016)]. https://sheldrake.medium.com/see-also-http-www-lifewithalacrity-com-2016-04-the-path-to-self-soverereign-identity-html-44c6b53cd737

Siftery. (2017). Top social login tools compared. https://medium.com/@siftery/top-social-login-tools-compared-b350eae26118

Sit, E., & Morris, R. (2002). Security considerations for peer-to-peer distributed hash tables. *International Workshop on Peer-to-Peer Systems*, 261–269.

Sovrin. (2018). *Sovrin™: A Protocol and Token for Self-Sovereign Identity and Decentralized Trust* (tech. rep.).

Sovrin. (2019). Sovrin SSI & IoT Working Group Charter (Version 1).

Speelman, T. (2020). *Self-Sovereign Identity: Proving Power over Legal Entities* (Doctoral dissertation). TU Delft. http://resolver.tudelft.nl/uuid:aab1f3ff-da54-47f7-8998-847cb78322c8

SQLite. (n.d.). https://www.sqlite.org/queryplanner.html

StatCounter. (2021). Mobile & tablet android version market share worldwide. Retrieved August 5, 2021, from https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide

Stigler, G. J. (1964). A theory of oligopoly. *Journal of Political Economy*, *72*(1), 44–61. http://www.jstor.org/stable/1828791

Stokkink, Q., Epema, D., & Pouwelse, J. (2020). A Truly Self-Sovereign Identity System. *arXiv preprint arXiv:2007.00415*.

Stokkink, Q., & Pouwelse, J. (2018). Deployment of a blockchain-based self-sovereign identity. *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 1336–1342.

Thales. (2020). *2020 Thales Data Threat Report* (tech. rep.). Thales. https://cpl.thalesgroup.com/sites/default/files/content/research_reports_white_papers/field_document/2020-04/2020-data-threat-report.pdf

The Duck, D. (2021). Duckduckgo tracker radar exposes hidden tracking. https://spreadprivacy.com/duckduckgo-tracker-radar/

The European Parliament and Council. (2016). Regulation (EU) 2016/679 of the european parliament and of the council. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679%5C#d1e6226-1-1

The Oxford Dictionary. (n.d.). Govern.

Tobin, A., & Reed, D. (2016). The inevitable rise of self-sovereign identity. *The Sovrin Foundation*.

Turner, A. (2021). How many people have smartphones worldwide (jun 2021). https://www.bankmycell.com/blog/how-many-phones-are-in-the-world

uPort. (n.d.). uPort Developer Portal. Retrieved July 26, 2021, from https://developer.uport.me/

Verizon. (2020). *2020 Data Breach Investigations Report* (tech. rep.). Verizon. https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf

Vossaert, J., Lapon, J., De Decker, B., & Naessens, V. (2013). User-centric identity management using trusted modules. *Mathematical and Computer Modelling*, *57*(7-8), 1592–1605.

W3C. (2021). Decentralized identifiers (dids) v1.0. Retrieved August 6, 2021, from https://www.w3.org/TR/did-core/

Wang, X., Yin, Y. L., & Yu, H. (2005). Finding collisions in the full sha-1. *Annual international cryptology conference*, 17–36.

World Bank Group. (2016). *Identification for Development Strategic Framework* (tech. rep.). World Bank Group.

World Bank Group. (2021). ID4D Data: Global Identification Challenge by the Numbers. https://id4d.worldbank.org/global-dataset

Xie, R. (2019). Why china had to ban cryptocurrency but the us did not: A comparative analysis of regulations on crypto-markets between the us and china. *Wash. U. Global Stud. L. Rev.*, *18*, 457.

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology?—a systematic review. *PloS one*, *11*(10), e0163477.

Zhou, T., Li, X., & Zhao, H. (2019). EverSSDI: Blockchain-based framework for verification, authorisation and recovery of self-sovereign identity using smart contracts. *International Journal of Computer Applications in Technology*, *60*(3), 281–295. https://doi.org/10.1504/IJCAT.2019.100300

Zimmermann, H. (1980). Osi reference model-the iso model of architecture for open systems interconnection. *IEEE Transactions on communications*, *28*(4), 425–432.