A-SIT Technology
E-Government Innovation Center

10 September 2020 Study · Digital Identity · E-Government · 19 min read

SSI Offline-Verifiable Attributes

Archived. This content is no longer updated. It remains available for reference.

Version 2020-09-10
Source https://technology2.a-sit.at/en/publications/ssi-offline-verifiable-attributes/

In a world with an increasing number of online services, digital identities become more and more important. Even in eGovernment, where public services are offered electronically play are digital identities vital. In order to have a digital ID as the full counterpart of a physical ID, the digital ID would have to support offline authentication. In particular, the authenticity and validity of the authentication data have to be able to be verified in a fully offline setting, where neither the prover nor the verifier has an established internet connection. This work proposes a first concept aiming offline verifiable self-sovereign identities.

1. Introduction

Digital identities play a more and more important role in daily life. Dozens of online services require digital identities from their users in order for the service providers (SP) to perform identification and authentication as well as authorization for services and resources.

The digitalization of processes pushes the development and enhancement of online services and this not only in the private sector but also in the public sector like in eGovernment. Additionally, goes the trend to the so-called “mobile-only” solutions where only a mobile device is used in order to use those online services, whereas in the past, for some cases, a personal computer was required.

In the next step of digitalization is to provide eGovernment applications for the mobile device, a digital identification (ID), representing the counterpart of a physical ID like, for instance, an id card, passport, or a driving license. To achieve a digital ID, additional requirements have to be met in order to proof the authenticity as well as the validity of this ID. These requirements are not really new, but in order to have a full counterpart of a physical ID, the digital ID has to be also valid and verifiable in an offline scenario. The offline verifiability of such a digital ID is mandatory because it cannot be sure that both of the parties, the verifier and the prover, have an Internet connection. For instance, during a police check when a police officer stops a car and tries to verify the digital driver’s license of the driver. In this case, this must also work when one party or even both do not have a valid internet connection.

This problem of offline verifiable digital IDs is a topic that was not fully solved yet, and this fact hinders the development of a physical ID. This work addresses this problem and proposes the first concept based on self-sovereign identities (SSI), which can be used to achieve offline IDs. The results of this work are published in a research paper [Abraham2020].

This report is structured as follows. Section 2. Background gives the reader the necessary background information on the main building blocks such as SSI, decentralized identifier (DID) as well as the byzantine fault tolerance protocol as a consensus protocol. Section 3. Concept defines the concept addressing the problem of offline verifiable SSIs. In Section 4. is the evaluation presented.

2. Background

2.1. Self-Sovereign Identity

The self-sovereign identity (SSI) model presents a new identity management (IdM) concept, which grants the owners of digital identities complete control over their data. When considering the evolution of identity models, SSI can be seen as the next model in the evolution after the user-centric model with the advantage of not having to trust a central authority, as depicted in Figure 1. SSIs ensure the security and privacy of a user's identity data, full portability of the data, no central authorities, and data integrity. Consequently, only the owners of the data can alter their identity data.

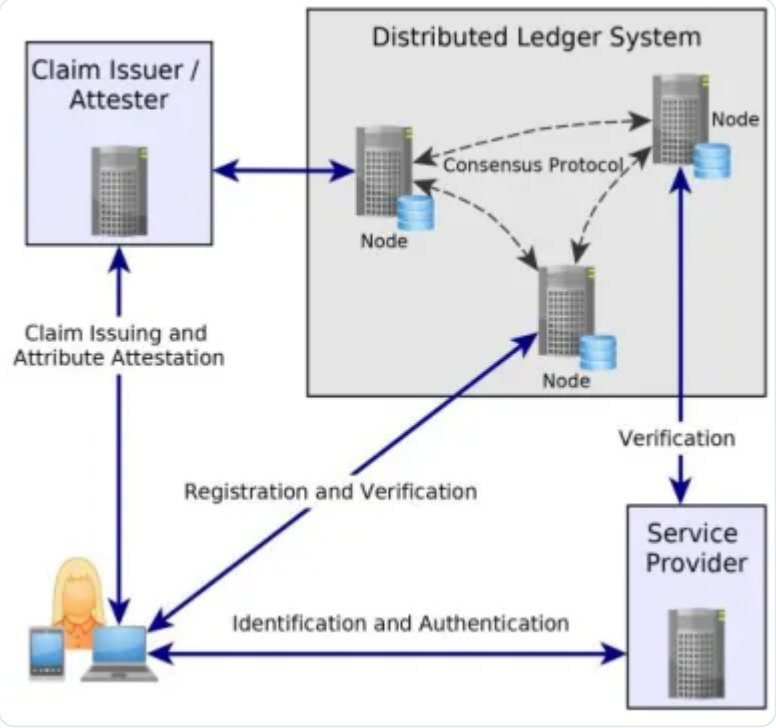


Figure 1: Architecture of an SSI System

2.1.1. Conceptual Requirements

SSI is a relatively new concept which does not provide a strict architectural definition but rather conceptional requirements. According to the Sovrin Foundation [Sovrin] an SSI system satisfies four major requirements: governance to ensure the system is trusted by all stakeholders, performance at internet scale, accessibility, and privacy. Allen [Allen] defined the 10 principles of SSI consisting of existence (the user must exist independently), control (the users must be in control over their identity), access (the users must be able to access their own data), transparency (the utilized system together with its algorithms must be transparent), persistence (identities must persist), portability (identity service and information must be transportable), interoperability (identities should be usable as wide as possible), consent (explicit user consent must be required), minimization (the disclosed data must be minimized), and protection (the users' rights have to be protected). Mühle et al.

[Muehle] have presented the architecture of such an SSI system and surveyed essential components of SSI and identified identification, authentication, verifiable claims, and attribute storage as these components.

2.1.2. Technical Concept

One of the core building blocks of an SSI system is the distributed ledger (DL), which serves as a decentralized public key infrastructure (DPKI) and provides properties such as immutability and transparency. Besides the DL, an SSI system requires identifiers that do not depend on an issuing party, such as a decentralized identifier (DID). Users create such identifiers and register them at the DL. Then, trusted claim issuers (or attesters) attest attributes of a user. As users in such an SSI system should be in full control, the users' identity data, DIDs, and private key material are stored in the users' domain, whereas the public information of a user is stored on the ledger, including public keys and revocation information. When performing authentication at a service provider (SP), this SP can then verify the users' claims ownership as well as attestations.

2.2. Decentralized Identifier (DID)

2.2.1. Decentralized Identifiers

Decentralized Identifiers (DIDs) [DID] were designed and are used to create self-sovereign digital identities. They are URLs that provide a way for trustworthy interactions with its subject. A DID subject is the identifier that the DID describes and DIDs redirect to DID Documents.

2.2.2. DID Documents

DID Documents contain three major sections: proof purposes, verification methods, and service endpoints. Service endpoints are URIs pointing to a service provided by a DID subject. Verification methods describe cryptographic methods that can be used with proof purposes to prove things such as the integrity of the DID Document or the relationship of an entity to the DID. DID Documents optionally contain public key(s) of the DID subject, and the proof of the public key ownership can be done statically or dynamically. Static

proof of ownership requires that a DID Document is signed with the private key and later verified with the public key, whereas the dynamic proof requires a challengeresponse protocol sent to the responsible service endpoint.

An example of a DID is **did:method:123456789** and this DID resolves to the corresponding DID document stored on the DL. It contains a context property that maps a valid string to an Internationalized Resource Identifier or a JSON Object; an *id* property that identifies the DID subject; an *authentication* property that defines the subject, authentication type, controller of the corresponding private key, the public key and the public key format needed for the authentication; a *service* property that defines the service type and the service endpoint. We refer interested readers to the extensive online documentation [DID].

2.3. Consensus Protocol

Software bugs, administrator mistakes or faulty hardware parts can introduce faults into systems that require high availability. Such faults are called Byzantine faults, and they include service interruption, unexpected behavior, among others. Byzantine fault tolerance protocol (BFT) [Castro] is a replication algorithm for building systems that tolerate Byzantine faults. The BFT protocol replicates services across N nodes that perform arbitrary computation provided they return deterministic results. The maximum number of faulty nodes allowed in a system using BFT is $f = \lfloor (N - 1)/3 \rfloor$ otherwise, the whole system is assumed to be corrupt. In contrast to the whole system, the whole replica in a node is considered compromised if any single process is compromised.

3. Concept

This section describes the concept to achieve offline-verifiable SSI attributes. First, the actors of this concept are introduced. As the second part are the different phases of our concept described. Figure 2 gives an overview of the concept, including the actors, different phases, as well as the main process flows.

In a nutshell, our concept is based on attestations, stating that a certain credential has not been revoked at the time of creating the attestation. We achieve the offline verifiability through the usage of a so-called trust-store containing the public keys of the SSI nodes. When the user is online, she can gather the attestation from the SSI network confirming that a credential is not revoked. This attestation is in case of offline authentication handed over as well. The verifier can then verify the credential as well as its validity, thanks to the trust-store.

3.1. Actors

A user who wants to perform identification and authentication is called **prover** . This prover wants to authenticate towards another party in a scenario where both participants are offline. For example, this could be the case when a citizen shows the digital driver's license to a police officer.

The verifying party, which is the counterpart to the prover, is called the **verifier.** This is the party to which the prover performs identification and authentication, which could be a police officer or service provider.

Devices are used by the prover and the verifier for their interaction, which is typically a smartphone or tablet. To protect the key-material as well as verifiable credentials (VCs), these devices ideally support hardware based key protection as well as credentials. This protection often rely on biometric authentication means such as fingerprint or face recognition. In other cases, the device of the verifier can also be a constrained device responsible for giving access for instance a parking lot, or some kind of vending machine which requires authentication.

A software application called **wallet** is installed on at least the prover's device. This wallet is responsible for maintaining and storing the key material as well as the VCs. These VCs are cryptographically linked to the private key of the prover through DIDs.

The party which is issuing VCs is called the **issuer** . The issuer, in particular, is not only responsible for issuing the VCs but also to revoke the issued VCs. Furthermore, the issuer might be a party that can be recognized like a government agency or a company.

The **SSI system** uses a distributed ledger (DL) to provide a DPKI used by the SSIs. The SSI system consists of a network of semi-trusted nodes, where each node holds a copy of the ledger in a permissioned ledger system. This way, the nodes build a web-of-trust (WoT) based network. The SSI network also maintains a distributed revocation list.

Figure 2: Concept of Offline SSI including its Actors, Phases and main Process-Flows [Abraham2020]

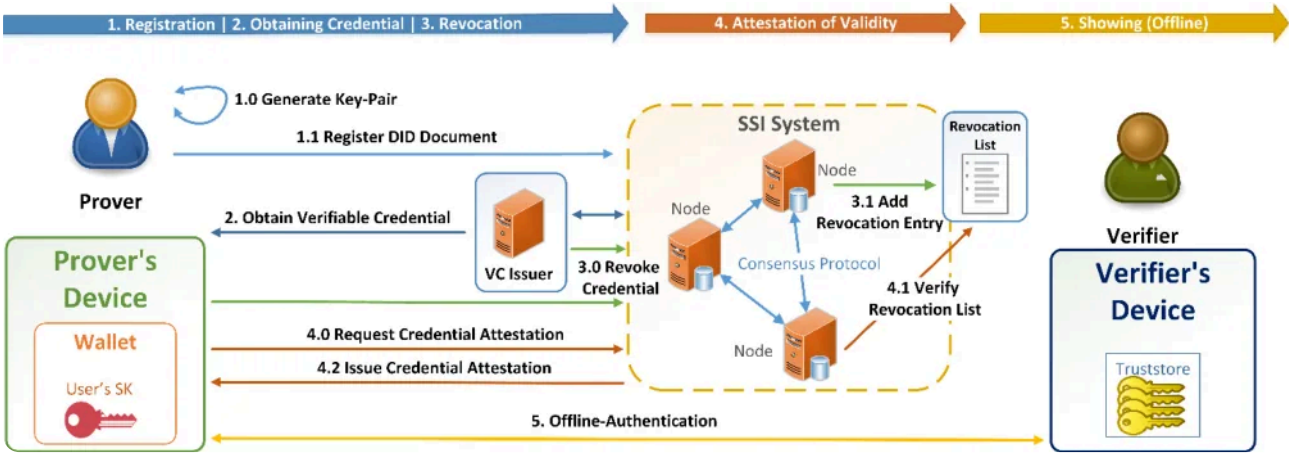
3.2. Process Phases

Our concept consists of 5 different phases [Abraham2020] detailed as follows and depicted in Figure 2:

1. In phase 1, the parties register at the system and perform a setup.
2. Phase 2 deals with the fact of obtaining related VCs.
3. The revocation entry is added in phase 3.
4. Attestations are issued stating that a credential was not revoked at a certain time, described in Phase 4.
5. Phase 5 illustrates the offline authentication flow.

3.2.1. Phase 1: Registration

In the registration phase, the user creates a key pair as well as a related identifier (DID). The DID is directly linked to the key-pair by utilizing a part of the public key or the whole as unique identifier within the DID. The key-pair for the DID is ideally stored within the secure element of the device protected by e.g. biometric authentication. After creating the keys and DID, the user registers her DID document at the SSI system, which requires to prove the ownership of the DID. Finally, the nodes write the DID document, containing the user's public key, to the DL.



3.2.2. Phase 2: Obtaining Credentials

In this phase, the user obtains a VC, containing a certain set of attributes depending on the issuer and type of VC, from a respective issuer. The user has to prove ownership of the DID as well as maybe additional identification and authentication depending on the SP. Finally, the VC is issued to the related user containing attributes as well as the DID of both, the user and the issuer and a signature.

3.2.3. Phase 3: Revocation

This phase covers the revocation of VCs. VCs can be revoked if attributes in the credentials no longer apply to the user or if the key-material or device was compromised or got broken. Thus, there are two parties who are able to perform revocation, the issuer as well as the prover, which is the owner of the credential. In case of a revocation, the revoking party convinces the nodes of the SSI network to add the credential to the revocation list. This approach clearly reflects the SSI methodology, in which the users are in full control over their own identity data.

3.2.4. Phase 4: Attestation of Validity

In phase 4 is the attestation of validity issued to the user. There are two different ways how to do this. First, the SSI network could have an issued credentials list and perform a crown job where, in a predefined time interval, the nodes verify if an issued VC was already revoked by checking if a revocation entry was written to the revocation list on the ledger. In the second way, the user can request attestation in the defined interval.

When the network gets a request for attesting, the requester has to prove the ownership of the DID. The nodes then verify the revocation list on the ledger. If the VC was not revoked, is not in the revocation list, the network issues an attestation. This attestation contains the id of the VC, the subject and the issuer of the VC as well as a timestamp stating the time of attestation. At the end, the attestation is signed with an aggregated signature, a so-called multi-signature, from sufficient nodes of the network. This way, the trust in the attestation is distributed throughout the network.

The validity period of a credential can be predefined by the issuer. The period also depends on the type of credential; like for instance, when having a credential containing the minimal data set (first name, last name, birthdate, and unique identifier), the validity period of attestation can be for example, a day or even longer.

These data are most likely not to change so often; therefore, a longer period can be chosen. In contrast, when having a driver's license credential, a higher attestation frequency might be necessary. Nevertheless, during the authentication, the verifier can decide if the attestation is still new enough or not.

The revocation and attestation process has to deal with the challenge that the network should not learn sensitive information of the user and the related credential. We still have to ensure that only the subject or issuer are able to revoke credentials, while a link between the validity attestation and the presentation of a partial credential needs to remain. Possible approaches would be to employ revocation or to record the hashes of issued credentials as well as their subject and issuer in an additional list at the ledger, which is checked in the revocation process.

3.2.5. Phase 5: Showing

The phase 5 (showing phase) consists of two main cases, first the fully offline showing in which both parties are without an Internet connection. Second, the partly offline showing, where one of the parties does not have an Internet connection but the other one has. Detailing both cases further shows the flexibility of our concept.

Page: 6 out of 9 *Phase 5-A Fully Offline Showing* In this phase, users authenticate towards a verifier and present attributes, while both are offline the prover nor the verifier have an internet connection.

The user who wants to prove identity attributes to a verifier sends the credential containing attributes. Additionally, the previously received attestation, stating the validity of the credential. The verifier starts with verifying the ownership of the user's DID as well as the presented credential, by verifying the signature as well as if it was issued by a trustworthy source. Next, the verifier checks the attestation and its included timestamp and signature to figure out when the last time was that the credential was valid and checked for revocation. The verifier can then decide if the attestation fulfils the required freshness or not.

If mutual authentication is required, the previous steps have to be performed vice-versa as well, so the parties reverse their roles. This is necessary to ensure, that the party who requires identification and authentication e.g. a police officer, is really a police officer and the prover is then convinced to hand over sensitive data such as the driver's license.

Phase 5-B Offline Showing with one Online Party If the prover is online, while the verifier is offline, fresh information can be obtained by the prover and forwarded to the verifier. As part of the showing phase (5), the prover could obtain a fresh attestation of validity for the credential that will be presented. A fresh attestation reduces the risk, that the verifier accepts obsolete information. Additionally, this concept may also be applied to the verifier's trust store. The makeup of the SSI network may change over time, as nodes leave or enter the system, which has to be reflected in the verifier's trust store. The online prover could obtain an updates to the verifiers trust store, which are signed by the network, and forward them to the verifier. Such an update mechanism would be useful for verifiers that are never or only rarely able to come online.

4. Evaluation

This section details the PoC implementation details on a high level as well discusses open points.

4.1. Proof-of-Concept Implementation

This section describes the proof-of-concept implementation to underline the feasibility of our concept.

4.1.1. Communication Channel

As our implementation uses mobile phones as devices of prover and verifier, we had to choose a communication technology for the showing/authentication process. We considered Bluetooth, Wifi direct and NFC. The decision was made to use Bluetooth over the other technologies because Bluetooth is available throughout devices and operating systems. Furthermore, we implemented the DIDComm [DIDComm] protocol for the communication between the devices. The connection is established starting with scanning a QR-Code that contains an invitation.

4.1.2. Mobile-Phone Based Identity Wallet

An identity wallet, running on the mobile phone, is responsible for creating and managing cryptographic keys, identifiers and credentials. Our implementation follows the common SSI approach used for communicating with a verifying party. We developed our own agent implementation to interact with the verifier's agent, which are rather libraries and not fully agents.

Page: 7 out of 9

4.1.3. Trust Store

The trust store holds the public keys of the SSI network in form of a VC. This credential contains a list of all nodes that are currently in the SSI network, including their DIDs and public keys, signed by the SSI network. The trust store enables the verification of signatures in an offline seating. In case new nodes joining the network or others leaving, the trust store needs to be updated whenever the party is online. An up-to-date trust store is required for a successful offline authentication.

4.1.4. SSI Network

The SSI network used in this work considers semi-trusted organizations as node operators, following a common approach of implementing SSI networks, e.g. Sovrin [Sovrin]. This network is considered a permissioned ledger system built upon DIDs and verifiable credentials (VCs). The user who performs an offline authentication is following the DIDComm protocol [DIDComm] and the data exchanged during the authentication was extended according to our needs.

4.1.5. Revocation

Our proposed system maintains a revocation list in the distributed ledger to track which credentials have been revoked. The VC might has been revoked by either the issuer or the owner itself. When users obtain a VC, they also receive a revocation token, which establishes a link between the credentials identity as well as the user's and issuer's DID and without revealing any sensitive information. Users and issuers keep this signed revocation token private until they want to revoke the associated credential. In case of revocation, the SSI nodes write an revocation entry to the ledger.

4.2. Discussion

This system is using timestamps, therefore, the questions rises if timestamps are secure? In our opinion they are, because the verifier is considered to be honest since this role is not attacking but rather the prover might be an attacker trying to trick the verifier. Therefore, timestamps are not an issue. The attacker would have to have access to the verifier's device to manipulate the time on the device, which is a very strong assumption. Further, the attestations could also include a field, like valid until or not valid after, stating the validity period of the attestation. This period depends on the importance of the data, for example for the name and birth date might 24 hours or more still be enough.

By utilizing a DL for storing revocation information, issues such as the single point of failure and the central trusted party are directly addressed. Furthermore, by utilizing a trusted SSI network for the revocation check, each node can trigger the revocation check as well as the trigger the issuance of the attestation. This way, trust and power is distributed across the SSI network resulting in a decentralized PKI serving as root of trust.

5. Conclusion

In this work, we presented a new concept to achieve offline verifiable SSIs by utilizing timestamped attestations as well as local trust store. The attestations are issued by an SSI network stating when the time when the validity of the credential was checked. The signed attestation is send to the user, which in case of offline-authentication, hands this attestation together with the VC over to the verifying party such as a SP. The SP can verify the ownership of the VC through the ownership of the corresponding DID. Furthermore, the SP can also verify the attestation by utilizing the local trust store.

Page: 8 out of 9

6. Bibliography

[Abraham2020] A. Abraham, S. More, C. Rabensteiner, F. Hörandner, "Revocable and Offline Verifiable Self-Sovereign Identities, "[in Submission], 2020.

[Allen] Christopher Allen. 2016. The Path to Self-Sovereign-Identity. http://www.
[lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html Online; Accessed 09-09-2020].

[Sovrin] Sovrin Foundation. 2018. Sovrin: A Protocol and Token for Self- Sovereign Identity and decentralized Trust. Sovrin January (2018), 1–41. https://sovrin.org/wpcontent/uploads/Sovrin-Protocol-and-Token-White-Paper.pdf [Muehle] Alexander Mühle, Andreas Grüner, Tatiana Gayvoronskaya, and Christoph Meinel. 2018. A survey on essential components of a self-sovereign identity. Computer Science Review 30 (2018), 80–86.

[Castro] Miguel Castro, Barbara Liskov: Practical byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst. 20(4): 398-461 (2002) [DIDComm] R. West, D. Bluhm, M. Hailstone, S. Curran, S. Curren, and G. Aristy. (2020) Aries rfc 0434: Out-of-band protocols. [Online] Accessed 02 Sept. 2020. [Online]. Available: https://github.com/hyperledger/aries-rfcs/blob/ master/features/0434-outofband/README.md [DID] W3C Working Draft, "Decentralized Identifiers (DIDs) v1.0 - Core Data Model

and Syntaxes," 2019, online, Accessed: 2020-08-27. [Online]. Available: https://www.w3.org/TR/did-core/
[VC] M. Sporny, D. Longley, and D. Chadwick. (2019) Verifiable Credentials Data Model 1.0 - Expressing verifiable information on the Web. [Online] Accessed 08 July 2020.

[Online]. Available: https://www.w3.org/TR/vc-data-model/ Page: 9 out of 9