**Image for the blog background (choose one that fit well):**

* <https://pixabay.com/photos/chain-security-metal-iron-3481377/>
* <https://pixabay.com/illustrations/blockchain-data-record-blocks-3052119/>
* <https://pixabay.com/photos/buffer-passport-travel-borders-1143485/>

**Title:**

Discovery of Self Sovereign Identity (SSI) from a security perspective.

**Abstract:**

Identity management system is a critical part in all information systems. Unavailability of such system has a direct availability impact on all the systems leveraging it to authenticate users. Let’s discovery an innovative way to rethink the identity management in a decentralized way.

**GitHub repository associated that will be publicly released alongside with this blog post:**

<https://github.com/ExcelliumSA/SelfSovereignIdentity-Study>

**SEO rules indicated by Mathilde:**

* Paragraphs with fewer than 300 words.
* Keyword used as much as possible: *SSI*
* Presence of sections.

**Author(s):** Dominique Righetto

**⚠ Notes about the content below:**

* All figure captions refer to the image file that Mathilde must use when she creates the blog post. So, it's normal if the caption does not describe the figure.

# Introduction

This post is based on my understanding and feedback after studying the **Self Sovereign Identity** concepts (called **SSI** from now) via all the documents and videos provided by Damien Bod on its blog post about SSI [1].

In addition, the following version of W3C specifications were used:

|  |  |  |
| --- | --- | --- |
| Specification name | Version | URL |
| Decentralized Identifiers (DIDs) | 1.0 | <https://www.w3.org/TR/2021/PR-did-core-20210803/> |
| Verifiable Credentials Data Model | 1.1 | <https://www.w3.org/TR/2022/REC-vc-data-model-20220303/> |
| Linked Data Cryptographic Suite Registry | Draft Community Group Report 29 December 2020 | <https://w3c-ccg.github.io/ld-cryptosuite-registry/> |
| RSA Signature Suite 2018 | Draft Community Group Report 26 May 2020 | <https://w3c-ccg.github.io/lds-rsa2018/> |

# What is SSI?

To quote Damien: “*Self-sovereign identity is an emerging solution built on blockchain technology for solving digital identities which gives the management of identities to the users and not organizations.*”

The two schemas below provide a high-level view (detailed view is provided in the next section):

Diagram

Description automatically generated

Figure 1: Figure11.png

Shape

Description automatically generated

Figure 1: Figure10.png

# Some terminology

Before to deep dive into involved flows and actors, it is important to define different terms used in SSI (definitions are taken from the corresponding W3C specification).

**Decentralized identifier (DID):**

A globally unique persistent identifier that does not require a centralized registration authority and is often generated and/or registered cryptographically (source [2]).

**DID document:**

A set of data describing the DID subject, including mechanisms, such as cryptographic public keys, that the DID subject or a DID delegate can use to authenticate itself and prove its association with the DID (source [2]).

Example of **DID identifier** and associated **DID document** taken from the specification (source [3]):

Graphical user interface, text, application, email

Description automatically generated

Figure 1: Figure00.png

**Verifiable credential:**

A standard data model and representation format for cryptographically verifiable digital credentials as defined by the W3C “*Verifiable Credentials”* specification. (source [5])

Note about a *verifiable presentation* of a *verifiable credential*: "a *verifiable presentation* expresses data from one or more *verifiable credentials*, and is packaged in such a way that the authorship of the data is verifiable. If *verifiable credentials* are presented directly, they become *verifiable presentations*". (page 74 of source [4])

Example of verifiable credential issued: A national identity card.

**Holder:**

A role an entity might perform by possessing one or more verifiable credentials and generating presentations from them. A holder is usually, but not always, a subject of the verifiable credentials they are holding. Holders store their credentials in credential repositories (wallet) (source [6)].

A wallet is a secure container for credentials and private keys.

**Issuer:**

A role an entity can perform by asserting claims about one or more subjects, creating a verifiable credential from these claims, and transmitting the verifiable credential to a holder (source [6)].

Example of issuer: Government.

**Verifier:**

A role an entity performs by receiving one or more verifiable credentials, optionally inside a verifiable presentation for processing. Other specifications might refer to this concept as a relying party (source [6)]).

Example of verifier: Bank.

**Blockchain:**

A blockchain is a growing list of records, called blocks, that are linked together using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (source [7]).

# Which are the actors involved in an SSI system?

Below is a high-level overview of the flows involved in a SSI systems taking a university diploma as an example (page 28 of source [4]):

Timeline

Description automatically generated

Figure 2: Figure01.png

Here:

* The academic institution represents the **Issuer** of the verifiable credentials.
* The user represents the **Holder** of the verifiable credentials.
* The employer represents the **Verifier** of the verifiable credentials.

Let’s deep dive into the two flows involved:

1. Issuing of a **verifiable credential** by an **Issuer** to a **Holder**.
2. Validation, by a **Verifier**, of a **verifiable presentation** provided by the **Holder**.

**Verifiable credentials issuing flow:**

Graphical user interface, text, application

Description automatically generated

Figure 3: Figure05.png

**Verifiable presentation validation flow:**

Graphical user interface, application

Description automatically generated

Figure 3: Figure06.png

# SSI key aspects

* It is possible to verify a verifiable credential even if the Issuer is not available or do not exist anymore.
* There is no direct communication between an Issuer and a Verifier during the issuing of a verifiable credential or validation of a verifiable presentation.
* Once stored in a blockchain, a DID document cannot be modified or deleted.
* Holder fully controls the level of information provided to a Verifier within the verifiable credentials provided to it.
* Digital signature, using asymmetric RSA/EC key pair, is a critical pillar of SSI and trust/integrity foundation.
* Availability of the blockchain used by the Issuer/Verifier is a critical pillar of SSI.

# Remarks about SSI providers

* At present it is not possible to use any wallet with any SSI solution. Each solution is locked to its own wallet (October 2021 – source [8]).
* An SSI provider (or solution) provide the Issuer and Verifier posture of a verifiable credential and it seemed not to be possible to currently mix two different SSI providers (one for Issuer and one for Verifier) even if it is theoretically possible (source [9]):

Graphical user interface, application, Teams

Description automatically generated

Figure 3: Figure04.png

* An SSI provider (or solution) provide a REST API providing an abstraction over the Issuing and Verification operation of verifiable credentials, the wallet management as well as the blockchain to store DID identifier + DID document.

# Cryptographic algorithms and SSI related W3C specifications

Algorithms supported by the “*Verifiable Credentials Data Model*” [10] are defined in the specification named “*Linked Data Cryptographic Suite Registry*” [11]. It is important to note that this one was into the “Unofficial Draft“ status when this post was written (March 2022). Therefore, there is not currently W3C official standard regarding supported and recommended signature algorithms.

Among the list of mentioned signature algorithms in the document, there is one, named “*RsaSignature2018*” [12]:

Graphical user interface, text, application

Description automatically generated

Figure 3: Figure09.png

This one is using a signature algorithm, named “*RSASSA-PKCS1-v1\_5*” [13], identified as not recommended for future cryptographic operation by the Cryptosense company [14] in 2014:

Graphical user interface, text, application, email

Description automatically generated

Figure 3: Figure07.png

This document [26] (page 2), named “*On the Security of the PKCS#1 v1.5 Signature Scheme*”, also mentioned that “*RSASSA-PKCS1-v1\_5*” algorithm should not be used for new applications:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 3: Figure12.png

However, even if “RsaSignature2018” is present into the “*Linked Data Cryptographic Suite Registry*” specification, the GitHub repository of the algorithm specification explicitly indicate to not use it anymore [15]:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 3: Figure08.png

It shows that the “*Linked Data Cryptographic Suite Registry*” specification needs to be finalized to remove any deprecated signature algorithms to prevent any issue by implementer of an SSI systems.

# Security attention points noticed

In addition to the challenge points identified by Damien Bod (source [16]), the following attentions regarding an SSI system were identified:

* Private identification information disclosure via the data stored in the DID document stored on the blockchain.
* Same thing via the DID identifier itself.
* Security issue of the cryptographic algorithms used for keys and signature.
* Attacks on deserialization processes used on Issuer and Verifier sides when manipulating serialized data coming from the blockchain or from the holder.
* Security issues affecting the quality of the implementation on the credential verification operation performed by the Verifier: signature validity, signature coverage (like unsigned claims [17]), type of credentials received against expected, expiration date, revocation state, issuer expected, alteration of properties without affecting the signature, credential replay attack, etc.:
  + Also include the validation of the integrity of the elements stored outside a verifiable credential like mentioned in the “*Verifiable Credentials*” specification (source [18]):

Graphical user interface, text, application

Description automatically generated

Figure 3: Figure03.png

* Attacks on the wallet itself targeting the holder side: Protection strength of the secret keys like ensuring that access to a credential require entering a secret or a physical action like pushing a button (reference in the “*Verifiable Credentials*” specification [19]).
* Attack to perform unexpected alteration of DID documents in the blockchain (verifiable data registry) because the “*Verifiable Credentials*” specification define the following statement in its Trust Model: "*All entities trust the verifiable data registry to be tamper-evident and to be a correct record of which data is controlled by which entities.*" (source [20]).
* Misusage of verifiable credentials for authorization decision. Indeed, verifiable credentials are intended as a means of reliably identifying subjects and not for authorization purpose (source [21]).
* Specific security issues affecting “Bearer Credentials”: Are not single-use where possible, contain personally identifying information and/or are correlatable (source [22]).
* Secure protocols are not used on Issuer and/or Verifier sides (source [23]).

# SSI playground

In addition to the free tiers offered by some SSI providers (source [1] section “Companies”), the “Hyperledger Indy” open-source project [24] offer a complete SSI playground in the form of a collection of docker images.

Using this playground, it is possible to deep dive into the different elements involved in SSI flows like for example the reception by an Holder of a verifiable credential from an Issuer.

A demonstration video [25] was performed to show an example of usage of this playground.

# Conclusion

Self Sovereign Identity (SSI) is a very interesting approach to significantly enhance the resiliency of any delivered credentials as well as a boost for the dematerialization of official identification document like passports, driver licenses, national identity cards and so on.

However, it misses nowadays a point from an SSI provider perspective in the way in which that they need to strongly follow the W3C specification to allow interoperability between providers. In addition, the W3C specification about “*Linked Data Cryptographic Suite Registry”* must be finalized and released to provide a clean guidance about cryptographic algorithms that must be used to securely handle the keys and signature aspects of SSI flows.

# Authors

* Dominique Righetto

# References

1. <https://damienbod.com/2021/03/29/getting-started-with-self-sovereign-identity-ssi/>
2. <https://w3c.github.io/did-core/#terminology>
3. <https://w3c.github.io/did-core/#a-simple-example>
4. <https://publications.iadb.org/publications/english/document/Self-Sovereign-Identity-The-Future-of-Identity-Self-Sovereignity-Digital-Wallets-and-Blockchain.pdf>
5. <https://www.w3.org/TR/vc-data-model/>
6. <https://www.w3.org/TR/vc-data-model/#terminology>
7. <https://en.wikipedia.org/wiki/Blockchain>
8. <https://damienbod.com/2021/10/11/challenges-to-self-sovereign-identity/>
9. <https://damienbod.com/2021/10/11/challenges-to-self-sovereign-identity/comment-page-1/#comment-139146>
10. <https://www.w3.org/TR/vc-data-model/#data-integrity-proofs>
11. <https://w3c-ccg.github.io/ld-cryptosuite-registry/>
12. <https://w3c-ccg.github.io/ld-cryptosuite-registry/#rsasignature2018>
13. <https://w3c-ccg.github.io/lds-rsa2018/#modification-to-algorithms>
14. <https://cryptosense.com/blog/algorithm-choice-in-pkcs11-part-1-rsa-in-v2-20>
15. <https://github.com/w3c-ccg/lds-rsa2018>
16. <https://damienbod.com/2021/10/11/challenges-to-self-sovereign-identity/>
17. <https://www.w3.org/TR/vc-data-model/#unsigned-claims>
18. <https://www.w3.org/TR/vc-data-model/#content-integrity-protection>
19. <https://www.w3.org/TR/vc-data-model/#device-theft-and-impersonation>
20. <https://www.w3.org/TR/vc-data-model/#trust-model>
21. <https://www.w3.org/TR/vc-data-model/#authorization>
22. <https://www.w3.org/TR/vc-data-model/#bearer-credentials>
23. <https://www.w3.org/TR/vc-data-model/#token-binding>
24. <https://github.com/hyperledger/indy-sdk/blob/master/docs/getting-started/run-getting-started.md>
25. <https://github.com/ExcelliumSA/SelfSovereignIdentity-Study/blob/main/demo-playground.mp4>
26. <https://eprint.iacr.org/2018/855.pdf>