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

his post is based on my understanding and feedback after studying the Self Sovereign Identity concepts (SSI) via all the documents and videos provided by Damien Bod in his blog post about SSI. In addition, the following version of W3C specifications was used:

Specification name Version URL

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

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

User Bob

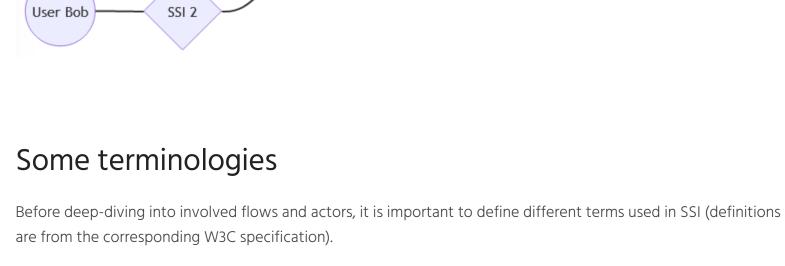
What is SSI?

User Alice

SSI 1 Blockchain

identities which gives the management of identities to the users and not organizations."

```
User Alice
                     SSI 1
                                     Blockchain
```



Scheme

generated and/or registered cryptographically.

Decentralized identifier (DID):

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. Example of DID identifier and associated DID document taken from the specification :

A globally unique persistent identifier that does not require a centralized registration authority and is often

DID document:

A DID is a simple text string consisting of three parts: 1) the did URI scheme identifier, 2) the identifier for the DID method, and 3) the DID method-specific identifier.

DID Method DID Method-Specific Identifier Figure 1 A simple example of a decentralized identifier (DID)

did:example:123456789abcdefghi

The example DID above resolves to a DID document. A DID document contains information associated with the

DID, such as ways to cryptographically authenticate a DID controller **EXAMPLE 1: A simple DID document** "@context": ["https://www.w3.org/ns/did/v1",

"publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"

An example of a verifiable credential issued: A national identity card.

"https://w3id.org/security/suites/ed25519-2020/v1"

"id": "did:example:123456789abcdefghi",

// used to authenticate as did:...fghi

"type": "Ed25519VerificationKey2020",

"id": "did:example:123456789abcdefghi#keys-1",

"controller": "did:example:123456789abcdefghi",

"authentication": [{

Verifiable credential: A standard data model and representation format for cryptographically verifiable digital credentials as defined by the W3C "Verifiable Credentials" specification. (source) 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 source)

credential repositories (wallet) (source). A wallet is a secure container for credentials and private keys.

Example of issuer: Government.

Example of verifier: Bank.

Holder:

Issuer:

Verifier:

A role that 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).

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).

contains a cryptographic hash of the previous block, a timestamp, and transaction data (source).

Who are the actors involved in an SSI system? Below is a high-level overview of the flows involved in an SSI system taking a university diploma as an example (page 28 of source):

Academic

Institution

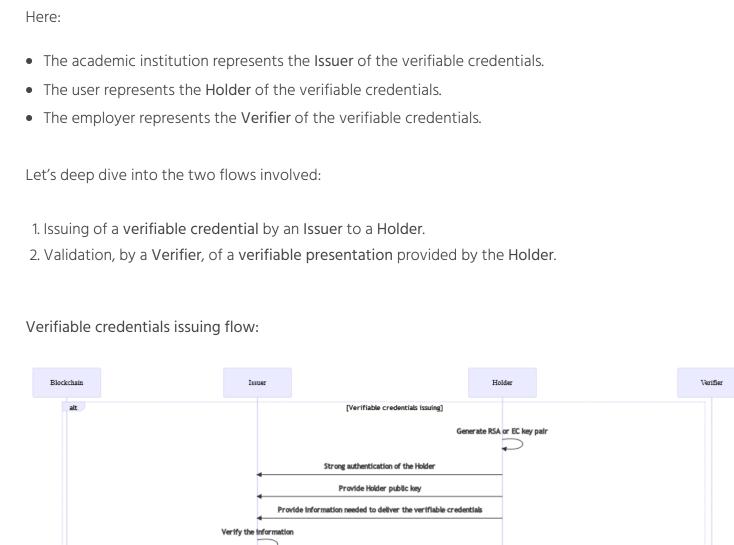
Presents digital diploma

Employer (Verifier)

Trusts and recognizes digital diploma

Issues digital diploma

Registers Verifies digital cryptographicdiploma against the **Blockchain** blockchain proofs

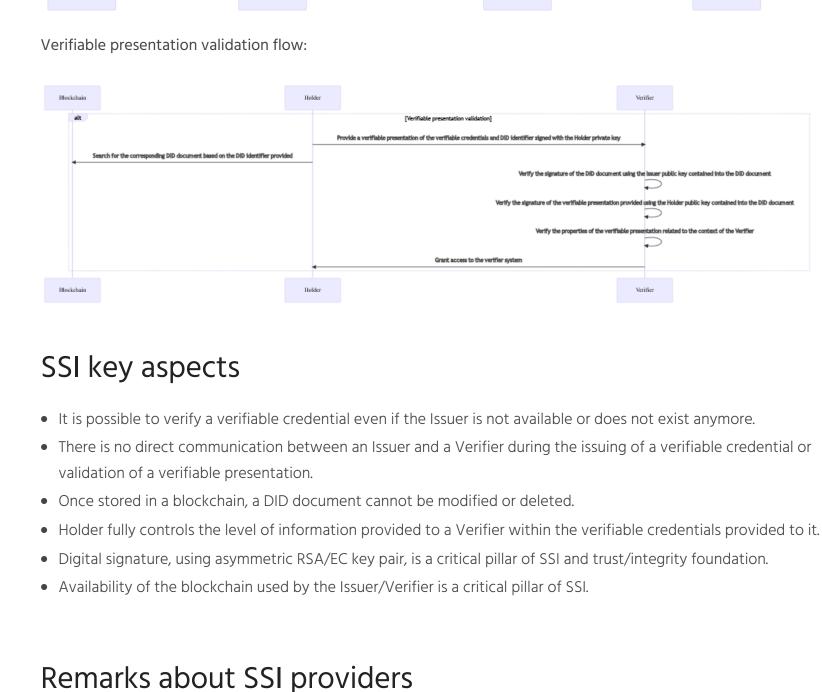


DID document is identified by the DID unique identifier

Return generated unique DID identifier + DID document

A key pair is unique to a DID identifier + DID document

Store generated DID identifier and DID document



• For the moment; it is not possible to use any wallet with any SSI solution. Each solution is locked to its own

• An SSI provider (or solution) provides the Issuer and Verifier posture of a verifiable credential and it seems not to be possible to currently mix 2 different SSI providers (one for Issuer and one for Verifier) even if it is theoretically

Currently it is not possible to use a SSI provider A or language verification API to verify the Verifiable Credentials issued by a SSI provider B? Thank you very much in advance for your feedback damienbod · March 21, 2022 - 19:57 · Reply→

possible (source):

wallet (October 2021 - source).

 $\textbf{righettod} \cdot \textbf{March 21, 2022 - 16:59} \cdot \textbf{\textit{Reply}} {\rightarrow}$

Thank you very much for such great infos and blog posts!!!!

I have a small question to ensure my correct understanding:

using a common protocol then these can work together when the verifier knows and trusts the issuers DID **Greetings Damien** righettod · March 22, 2022 - $08:32 \cdot Reply \rightarrow$

Hi righttod, If the SSI A can understand the VC produced by the SSI B and both can use the same ledger where the DID is persisted and communicate using an agent

So based on your feedback and the information from your blog posts, it is theoretically possible

Yes correct, if the providers use the same standards, then most can work together.

• An SSI provider (or solution) provides a REST API providing an abstraction over the Issuing and Verification

but currently SSI providers your evaluated do not support. It is correct?

damienbod · March 22, 2022 - 11:05 · Reply \rightarrow

recommended signature algorithms.

TABLE OF CONTENTS

Introduction

Signature Suites

Ed25519

RSA

3.

3.1

3.1.1

3.2

3.2.1

3.3

3.3.1 3.3.2

3.4 3.4.1

3.5 3.5.1

3.6.1

3.7

3.7.1

The Registration Process

Ed25519Signature2018

JsonWebSignature2020

JcsEd25519Signature2020

GpgSignature2020

3.6 JCS Ed25519 Signature 2020

BBS+ Signature 2020

BbsBlsSignature2020

EcdsaSecp256k1RecoverySignature2020

RsaSignature2018

→ C w3c-ccg.github.io/ld-cryptosuite-registry/#rsa

document. Cryptographic algorithms and SSI-related W3C specifications

Algorithms supported by the "Verifiable Credentials Data Model" are defined in the specification named "Linked"

post was written (March 2022). Therefore, there is not currently W3C official standard regarding supported and

Among the list of mentioned signature algorithms in the document, there is one, named "RsaSignature2018":

§ 3.2.1 RsaSignature2018

RsaVerificationKey2018

RsaSignature2018

Dave Longley, Manu Sporny

RsaVerificationKey2018

Dave Longley, Manu Sporny

"id": "did:example:123456789abcdefghi#keys-1",

"controller": "did:example:123456789abcdefghi",

Algorithm Choice in PKCS#11 (part 1)

Following my post on the security of the algorithms in the W3C Crypto API (our most viewed blog post by far), I thought I'd repeat the exercise for other cryptographic APIs. Here at Cryptosense we do a lot of work with PKCS#11, widely used in applications that use devices like HSMs and smartcards to provide cryptography. How do the algorithms in PKCS#11 measure up?

One problem with PKCS#11 is it hasn't been updated since 2004 (though this is about to change, we'll look at the <u>proposed changes</u> in a future post). The state of the art in cryptanalysis, however, has certainly advanced, to the extent that many of the cryptographic algorithms, or mechanisms proposed in PKCS#11 are now considered broken. There are a lot more mechanisms in PKCS#11 than in the W3C Crypto API, so we'll treat one section of the standard at a time, starting with RSA mechanisms. Below is the summary

Note

See text

No security proof

"publicKeyPem": "----BEGIN PUBLIC KEY...END PUBLIC KEY----\r\n"

PROVISIONAL

EXAMPLE 2: Example of RsaVerificationKey2018

Specification RSA Signature Suite 2018

"type": "RsaVerificationKey2018",

"expires": "2017-02-08T16:02:20Z",

PROVISIONAL

Specification RSA Signature Suite 2018

§ 3.2 RSA

Identifiers

Status

Authors

Summary

Identifiers

Status

Data Cryptographic Suite Registry". It is important to note that this one was in the "Unofficial Draft" status when this

operation of verifiable credentials, the wallet management as well as the blockchain to store DID identifier + DID

This one is using a signature algorithm, named "RSASSA-PKCS1-V1_5", identified as not recommended for future cryptographic operation by the Cryptosense company in 2014:

Graham Steel June 12, 2014

table

Algorithm/mode

RSASSA-PKCS1-

V1_5

RSAES-PKCS1-v1_5

This document (page 2), named "On the Security of the PKCS#1 v1.5 Signature Scheme", also mentioned that

- RSA in v2.20

```
"RSASSA-PKCS1-v1_5" algorithm should not be used for new applications:
 Security of RSA PKCS#1 v1.5 signatures. Even though RSA PKCS#1 v1.5 is still the most important
 digital signature scheme used in practice, we do not yet have any formal evidence of its security, provided
 by a rigorous reduction-based security proof under any standard complexity assumption. We do not even
 know any security proof under a non-standard but plausible interactive assumption, apart from the trivial
 assumption that the scheme is secure.
    Due to the lack of security proofs for PKCS#1 v1.5 signatures, some standards allow to use PKCS#1
v1.5, but recommend RSA-PSS [BR96] instead. This includes TLS 1.3, X.509v3 (RFC 4055), and PKCS#1
 itself, since version 2.1 (RFC 3447):
       "Although no attacks are known against RSASSA-PKCS#1 v1.5, in the interest of increased
       robustness, RSA-PSS is recommended for eventual adoption in new applications." (RFC 3447)
However, even if "RsaSignature2018" is present in the "Linked Data Cryptographic Suite Registry" specification, the
GitHub repository of the algorithm specification explicitly indicates to not use it anymore:
  a github.com/w3c-ccg/lds-rsa2018
                       ₽ gn-pages ▼ ₽ i branch ♥ o tags
                                                                                    ✓ c1b711c on May 26, 2020 😘 8 commits
                       OR13 Create README.md
                       LICENSE
                       README.md
                       index.html
                                                     Correct section number in reference
                       README.md
                        RsaSignatute2018
                         Although support for this suite exists to varying degrees in software implementations, the specification is out of date,
                         and the suite is not recommended for further use.
                         If you need to support RSA or NIST curves, please consider supporting:
                          • https://github.com/w3c-ccg/lds-jws2020
```

deprecated signature algorithms to prevent any issue by the implementer of SSI systems.

• Security issues of the cryptographic algorithms used for keys and signatures.

mentioned in the "Verifiable Credentials" specification (source):

Security attention points noticed

an SSI system were identified:

• Same thing via the DID identifier itself.

from the blockchain or from the holder.

signature, credential replay attack, etc.:

protected but probably should be:

"credentialSubject": {

"alumniOf": {

"name": [{

"lang": "en"

"@context": [

EXAMPLE 34: Non-content-integrity protected links

"https://www.w3.org/2018/credentials/v1",

"id": "http://example.edu/credentials/58473",

"value": "Example University",

"https://www.w3.org/2018/credentials/examples/v1"

"type": ["VerifiableCredential", "AlumniCredential"],

"id": "did:example:ebfeb1f712ebc6f1c276e12ec21", "image": "https://example.edu/images/58473",

"id": "did:example:c276e12ec21ebfeb1f712ebc6f1",

§ 8.2 Content Integrity Protection This section is non-normative. Verifiable credentials often contain URLs to data that resides outside of the verifiable credential itself. Linked content that exists outside a verifiable credential, such as images, JSON-LD Contexts, and other machinereadable data, are often not protected against tampering because the data resides outside of the protection of

the proof on the verifiable credential. For example, the following highlighted links are not content-integrity

• Private identification information disclosure via the data stored in the DID document stored on the blockchain.

• Attacks on deserialization processes used on Issuer and Verifier sides when manipulating serialized data coming

• Security issues affecting the quality of the implementation of the credential verification operation performed by the Verifier: signature validity, signature coverage (like unsigned claims), type of credentials received against

```
}, {
           "value": "Exemple d'Université",
           "lang": "fr"
          }]
      "proof": { ... }
• Attacks on the wallet itself targeting the holder side: Protection strength of the secret keys like ensuring that
  access to a credential requires entering a secret or a physical action like pushing a button (reference in the
  "Verifiable Credentials" specification).
• Attack to perform unexpected alteration of DID documents in the blockchain (verifiable data registry) because
  the "Verifiable Credentials" specification defines 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).
• Misusage of verifiable credentials for an authorization decision. Indeed, verifiable credentials are intended as a
  means of reliably identifying subjects and not for authorization purposes (source).
• Specific security issues affecting "Bearer Credentials": Are not single-use where possible, contain personally
  identifying information and/or are correlatable (source).
• Secure protocols are not used on Issuer and/or Verifier sides (source).
Real-world use cases
```

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

Dominique Righetto

stored on their mobile in an SSI mobile wallet.

SSI playground

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 documents like passports, driver's licenses, national identity cards and so on.

W3C specification to get interoperability between providers. In addition, the W3C specification for "Linked Data"

Cryptographic Suite Registry" must be finalized and released. It is needed to provide a clear guidance about

cryptographic algorithms for secure handling of the keys and signature aspects of SSI flows.

Did you like this article? Find some more articles in our blog section.

In addition to the free tiers offered by some SSI providers (source section "Companies"), the "Hyperledger Indy"

open-source project offers a complete SSI playground in the form of a collection of docker images.

General

Let's deliver the right solution for your business Services Eyeguard Information Security Governance

≺ Share **Recent Posts** The Necessity of Cyber Crisis Exercises The Art of Password Spraying: A Comprehensive Analysis Common client-side vulnerabilities of web applications Categories Advice Consulting Cyber Blog Post General Newsletter PR The Cyber Blog Times Uncategorized Search Q Search ...

To quote Damien: "Self-sovereign identity is an emerging solution built on blockchain technology for solving digital

It 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 A role that 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). A blockchain is a growing list of records, called blocks, that are linked together using cryptography. Each block

Go to file Code ▼ It shows that the "Linked Data Cryptographic Suite Registry" specification needs to be finalized to remove any In addition to the challenge points identified by Damien Bod (source [16]), the following attention points regarding

expected, expiration date, revocation state, issuer expected, alteration of properties without affecting the • Also, include the validation of the integrity of the elements stored outside a verifiable credential as

The company TYKN implemented SSI in different contexts, for example, in a humanitarian aid one. The objective was to allow Syrian refugees to prove their identity, to Turkish companies, to find a job via a digital work permit

CONTACT US

A demonstration video was performed to show an example of the usage of this playground. Conclusion However, it misses nowadays a point from an SSI provider perspective because of the need to strictly follow the

Intrusion Tests - Red Team Where to meet us **Application Security** Privacy Notice Contact Us Network & Security Infrastructure CERT-XLM Eyetools Training

© 2021 Excellium Services Group 5 rue Goell L-5326 Contern Luxembourg +352 26 20 39 64

Meet Success About Us

Mentions légales/Privacy Notice

Follow us y in

Career Opportunities