Decentralized revocation of Verifiable Credentials

By:

- Andrea Scorza, LTO Network
- Arnold Daniels, LTO Network

Introduction

One of the core concepts of verifiable credentials is that the credential can be verified between the holder and the verifier, independent of the issuer. The way in which revocation is currently being implemented undermines this premise. We this proposal we want to establish a method where participation, cooperation, and trust of the original issuer are not a requirement.

W3C Draft standards

The Verifiable Credentials Data Model [1] does not dictate how revocation should be implemented. Credentials have a `credentialStatus` property, which is an object with an `id` and `type` property. The precise contents of the credential status information are determined by the specific credentialStatus type definition

Credential Status List 2017

The initial draft for a status list is the Credential Status List 2017 [2]. This standard revolved around resolving the credential status at an HTTP endpoint made available by the issuer. This draft was abandoned due to privacy concerns. This method gives the issuer an active role in verification and would make it possible for the issuer to track statistics based on the status requests.

Status List 2021

This concern has been addressed in the Status List 2021 draft [3], subtitled "Privacy-preserving status information for Verifiable Credentials". By generating a bitstring with the revocation status of at least 16,000 entries, the holder enjoys herd privacy. The `statusListIndex` property in the verifiable credentials tells the verifier which bit in the string corresponds with this specific credential.

Similar to the 2017 draft, the status list is made available through an HTTP endpoint. A verifier would request this bitstring rather than a single revocation status, preventing the issuer from tracking the use of a single credential.

Issues with the current draft

By making the status list available through an HTTP endpoint, the privacy issue is only partly solved. Other issues which existed in the approach of the 2017 draft remain unaddressed in the 2021 draft. Additionally, non-privacy-related concerns are introduced.

The main reason for using a bitstring is to protect the privacy of the holder and verifier from a malicious issuer. Since the issuer generates the bitstring and endpoint, there is no guarantee that the bitstring actually reflects thousands or credentials. It may contain dummy values.

The issuer has an active role in verification. If the issuer is no longer able or willing to host the status list, it becomes impossible to properly verify the credential. This leaves the verifier with the choice to possibly accept revoked credentials, or flat out reject all credentials of that issuer.

If the system of the issuer is temporarily unavailable, it could have a ripple effect through the network as it delays verification.

By reducing the status to a boolean, you lose information, like a timestamp when the credential was revoked. There are many use cases where this is important. For instance, if there's a dispute about a construction permit, we'd like to know if that credential was valid during construction. Whether it's valid now is not relevant.

Distributed status list

These issues are not present when the status list is resolved using a decentralized ledger. All revocations are distributed on the ledger, and available to the verifier. When the verifier runs a node, there's no reliance on an external system.

Most ledgers are implemented as append-only. This means that it's always possible to validate at a specific point in time. In contrary to a centralized revocation list, it is hard-to-impossible to backdate a status change.

LTO Network

LTO Network supports a modified version of the 2017 draft [4]. Revocations are registered on the network and presented in the expected format. The credential status id is a DID URL (DID + path), which can be resolved similarly to DIDs.

A concern with this approach is that it can reintroduce issues that the 2021 draft tries to solve. In case the verifier is not running a blockchain node, but relying on an external service to resolve the status list, that external service could track statistics about verifiable credentials.

Sovrin

The solution of Sovrin is similar to that of the 2021 draft [5]. The status of many credentials is combined into a value that can be used to check the credential status. However, it uses Camenisch-Lysyanskaya blind signatures and elliptic curve cryptography to create a tails file instead of a simple bitstring.

Each credential is associated with a number from the tails file, and it calculates a number that contains all the other credentials except its own, this number is called a witness. For a credential to be considered valid the holder must prove that it knows a number and a witness together.

When a credential is issued or revoked, a new tails file is generated. This provides more privacy than a bitstring, as it obscures the reason for the mutation.

The downside of this approach is that issuing a credential requires a transaction on the ledger and thus an internet connection. Also, it relies on somewhat experimental, non-standardized cryptography.

Decentralized generation of the bitstring

An option that has yet to be explored it to decentralize the generation of the status bitstring. Individual status changes would be distributed through the ledger. This bitstring is constructed by the node that's responsible for resolving the status list.

This would combine the approach of LTO with the 2021 draft. The credential status id would be a DID URL that could be resolved. However the result would contain a bitstring rather than a single status.

Considerations

Similar to DIDs, the use of a distributed ledger should have benefits but not be mandatory. It's also important to ensure a new draft doesn't reintroduce issues that have been solved by the current draft.

References

1) https://www.w3.org/TR/vc-data-model/
2) https://w3c-ccg.github.io/vc-csl2017/
3) https://w3c-ccg.github.io/vc-status-list-2021/
4) https://docs.ltonetwork.com/protocol/identities/verifiable-credentials
5) https://www.surf.nl/files/2021-05/technical-exploration-surf-ledger-based-self-sovereign-identity.pdf