Link: Verifiable Credentials Holder Binding

This is a proposed collaborative paper to be completed at RWOT 2022, Den Haag, Netherlands, 26-30 September. Advance reading: see here.

W3C Verifiable Credentials Holder Binding Specification

by Oliver Terbu, Spruce Systems, Inc. (Berlin/New York)

Abstract

The W3C Verifiable Credentials Data Model does not define how to bind the W3C Verifiable Credential to the W3C Verifiable Presentation, so that the Verifier can verify that the holder of the verifiable presentation is the rightful or intended holder of the verifiable credential. Verifying a verifiable presentation does not include verifying the binding between the verifiable credential subject and the verifiable presentation holder. While this can be done by simply matching the credential subject identifier against the holder identifier, there is no normative reference for this approach, and it might not be suitable for a number of reasons outlined in this paper.

For these reasons, this paper proposes a specification for a new top-level property for Holder Binding that has an extension mechanism and a registry for Holder Binding methods. The Holder Binding allows Holders and/or Issuers to indicate how the rightfulness of the presentment can be verified at the time of presentment. If no Holder Binding property was provided, this matches the definition of the W3C Verifiable Credentials Data Model 1.1 specification which is essentially equivalent to no guidance on the Holder Binding is provided. Therefore, introducing this mechanism is fully backward compatible with existing verifiable credentials and verifiable presentations. This specification does not mandate a specific form of holder binding or W3C Verifiable Credential proof type or format. Instead it provides a framework for Issuers, Holders and Verifiers to provide guidance on how Holder Binding can be checked deterministically according to their intentions.

Intro

A W3C Verifiable Presentation (VP) can be created by anyone which can be different from the Subject of the W3C Verifiable Credentials (VCs) in the presentation. The W3C Verifiable Credentials Data Model specification defines a proof property in the VP but it does not define further semantics other than the proof of the VP can be used to verify the VP was not tampered with and to verify the authorship. Authorship means that the VP was generated by the Holder of the VP. It does not ensure that the Holder is the rightful Holder of the

presented VCs which is a very common use case in the Self-Sovereign Identity (SSI) ecosystem.

The Verifier would typically need to perform extra steps to ensure that the Holder is the rightful Holder of the presented VCs. Those steps are trivial in case the VC is bound to a Subject and the Holder of the VP is the same as the Subject. In that case, the Verifier would need to verify that the `holder` property matches the `credentialSubject.id` property of the VC. By verifying the proofs of the VP and the VCs, the Verifier can then ensure that the Holder is the rightful Holder of the VC.

Those steps become more complex in case a different Holder Binding method was used. At least the following Holder Binding methods were observed so far:

- VC-based Holder Binding where the VP contains relationsip-VCs that bind the Subject identifiers of the VCs and the Holder identifier together.
- DID-based Holder Binding which is similar to VC-based Holder Binding but the relationship between the identifiers of the VCs and the Holder are established through properties in the DID Document, e.g., alsoKnownAs.
- Delegation-based Holder Binding where the Subject of the VC delegates the capability to present the VCs to another rightful Holder and typically limits the scope of the presentment (expiration time, not before time, only specific VC types, valid domains etc.).
- Signature IDs (or linked secret)-based Holder Binding for ZKP-based VCs.
- Evidence-based Holder Binding where the Holder Binding is established through an out-of-band agreement.

Because there are many different options, a Verifier usually needs to guess the intended method for Holder Binding which is an issue for interoperability. Having the Verifier to have this knowledge pre-populated would prevent interoperability as well. Guessing the method is prone to side effects which might compromise security.

For these reasons, this specification introduces a new property that allows a Verifier to determine the intended Holder Binding between the VP and the presented VCs based on a new property which removes the necessasity of guessing the Holder Binding method. Since different options are possible, this specification also introduces an extension mechanism. Uniquly identifying the type of Holder Binding will also allow Verifiers to give Holders guidance on what types of Holder Bindings the Verifier can support.

Specification

This specification defines the following `holderBinding` property for the determination of the intended Holder Binding method between the VP and included VCs. The `holderBinding` MAY be included in VCs and/or in VPs. If the `holderBinding` property is included in one VC and in the VP, verifiers MUST ensure the `holderBinding` in the VP is allowed by the `holderBinding` type of

the VC.

`holderBinding` property

If present in the VP or VC, the value of the `holderBinding` property MUST include the following: - `type` property, which expresses the Holder Binding method type. It is expected that the value will provide enough information to determine the Holder Binding method between the VP and included VCs. The precise contents of the Holder Binding information is determined by the specific holderBinding type definition, and varies depending on the Holder Binding method. The Holder Binding information MAY also include information about for which VC in the VP the Holder Binding applies. For example, this can be done by including a reference of the VC such as the id of the VC.

Each Holder Binding method MUST define how Holder Binding for an input VP and one or more input VCs contained in the VP can be deterministically verified. For example, a simple Holder Binding method might define that for a given input VP Holder Binding could be verified based on checking that the `holder` property matches the `credentialSubject.id` property in every `verifiableCredential` array in the VP.

EXAMPLE 1: Usage of the holderBinding property in VP

```
{
   "holder": "did:key:1234:...",
   "holderBinding": {
      "type": "delegationHolderBinding2022",
      "delegation": "https://my.holder-binding.abc/12345"
   },
   ...
   "proof": {
      "verificationMethod": "did:key:1234#key-1",
      "type": "Ed25519Signature2018",
      "jws": "...", ...
   }
}
```

EXAMPLE 2: Usage of multiple holderBinding properties in VP

```
{
   "holder": "did:key:1234:...",
   "holderBinding": [
      {
         "type": "DelegationHolderBinding2022",
         "delegation": "https://my.holder-binding.abc/12345"
      },
      {
         "type": "IdentifierMatchingHolderBinding2022",
```

```
            "someOtherHolderBindingProperties": "..."
        },
        {
            "type": "SomeAgreementBasedHolderBinding2022",
            "moreHolderBindingProperties": "..."
        },
        {
            "type": "SomeBiometricsBasedHolderBinding2022",
            "evenMoreHolderBindingProperties": "..."
        }
    ],
    ...
    "proof": {
        "verificationMethod": "did:key:1234#key-1",
        "type": "Ed25519Signature2018",
        "jws": "...", ...
    }
}
```

TBD: add more examples for VCs and VPs during RWOT.

Registry

This table summarizes the Holder Binding Method specifications currently in development. The table lists the method name, associated specification, authors, stability of the specification, and conformance test suite (if applicable).

TBD