Revocation

Revoking a Credential Suggest Edits

This tutorial demonstrates:

1. How an issuer can create a revocable credential
2. How an issuer can revoke a credential
3. How a verifier can determine whether a given credential is revoked

About

Verite's revocation implementation uses the approach described inW3C Status List 2021 -- a highly compressible, privacy-promoting approach.

In this method, the revocation status of a given verifiable credential is stored in a specified index of a bitstring. If the value is 0, the credential is not revoked; otherwise it's revoked.

Note: Deployments of this approach vary, and this project uses a simple REST endpoint to serve the bitstring.

Then, at verification time, a verifier obtains the entire bitstring and looks up the value at the index specified in the credential to determine whether it's been revoked.

This has the advantage that, even if the issuer is consulted for the bitstring, the verifier doesn't tell the issuer exactly which index it is checking, enabling what's called "herd privacy".

Example of a Revocable Credential

Below is anexample from Status List 2021 demonstrating what a Revocable Credential might look like (in JSON-LD format, for clarity). Note the addition of credentialStatus object with several fields:

JSON { "@context": ["https://www.w3.org/2018/credentials/v1", "https://w3id.org/vc-status-list-2021/v1"], "id": "https://example.com/credentials/23894672394", "type": ["VerifiableCredential"], "issuer": "did:example:12345", "issued": "2021-04-05T14:27:42Z", "credentialStatus": { "id": "https://dmv.example.gov/credentials/status/3#94567", "type": "StatusList2021Entry", "statusPurpose": "revocation", "statusListIndex": "94567", "statusListCredential": "https://example.com/credentials/status/3" }, "credentialSubject": { "id": "did:example:6789", "type": "Person" }, "proof": { ... } } The presence of this object tells verifiers that they should check the credential for its current status, and where/how to do it. Specifically:

- statusListCredential
- is the URI from which the revocation list should be retrieved
- statusListIndex
- indicates which index in the bitstring this credential corresponds to
- The type
- StatusList2021Entry
- informs verifiers the data structure to expect and steps to use to check whether the credential has been revoked

How an Issuer Creates a Revocable Credential

The issuer must decide at creation time whether the credential should be revocable, because the credentialStatus property will needed to be added to the issued verifiable credential.

Suppose the following:

- url
- is the URL where the bitstring revocation list will be accessible
- index
- is the index within the bitstring storing the specific credential's revocation status

Then credentialStatus can be populated as follows:

JavaScript const credentialStatus: { id: {url}#{index}, type: "StatusList2021Entry", statusPurpose: "revocation", statusListIndex: index.toString(), statusListCredential: url } When issuing the credential, the issuer would pass credentialStatus as follows:

JavaScript const encoded = await composeVerifiableCredential(issuer, subject.id, attestation, { credentialStatus }) A complete example follows:

JavaScript // We will create a random did to represent our own identity wallet const subject = randomDidKey(randomBytes)

// Stubbed out credential data const attestation: KYCAMLAttestation = { type: "KYCAMLAttestation", process: "https://verite.id/definitions/processes/kycaml/0.0.1/usa", approvalDate: new Date().toISOString() } const credentialType = "KYCAMLCredential"

/* * *Assume the credential's index within the bitstring will be 0* / const credentialStatus = { id: {revocationListUrl}#0, type: "StatusList2021Entry", statusPurpose: "revocation", statusListIndex: "0", statusListCredential: {revocationListUrl} }

// Generate the signed, encoded credential const encoded = await composeVerifiableCredential(issuer, subject.id, attestation, credentialType, { credentialStatus }) In a real implementation, an issuer may want to maintain multiple revocation status lists. Seerevocation best practices for details.

Revoking a Credential

When an issuer needs to revoke a credential, the bitstring approach requires the credential's index within the bitstring to be flipped to 1. Verite libraries expose a convenience method for issuers, as follows:

JavaScript statusList = await revokeCredential(credential, statusList, signer) Verite similarly exposes unrevokeCredential to undo the revocation. Unrevoking doesn't have to be supported by issuers; some issuers choose to issue a new credential instead.

In a deployment, the revoking and unrevoking operations could be exposed by authenticated API calls that would be accessed by an authorized party, such as a compliance agent.

Note that revoking a credential doesn't change the subject's credential or the revocation list itself since both are immutable. A new, modified revocation list is built, which the party hosting the revocation list will need to persist for future lookups.

Verification

The credential itself will inform a verifier whether the credential's status needs to be checked and how. A revocable Verite credential will have a credentialStatus field of type RevocationList2021Status , which informs the verifier how to proceed.

At verification time, the verifier will fetch the content at the revocationListUrl . This content will be a verifiable credential, of type "StatusList2021Credential" , which contains the bitstring at the credentialSubject.encodedList property (zlib-compressed, base64-encoded). SeeStatusList2021Credential for additional details.

JSON { "@context": ["https://www.w3.org/2018/credentials/v1", "https://w3id.org/vc-status-list-2021/v1"], "id": "https://example.com/credentials/status/3", "type": ["VerifiableCredential", "StatusList2021Credential"], "issuer": "did:example:12345", "issued": "2021-04-05T14:27:40Z", "credentialSubject": { "id": "https://example.com/status/3#list", "type": "StatusList2021", "statusPurpose": "revocation", "encodedList": "H4sIAAAAAAAAA-3BMQEAAADCoPVPbQsvoAAAAAAAAAAAAAAAP4GcwM92tQwAAA" }, "proof": { ... } } The verifier then inspects the credential's index in the bitstring to determine if it's revoked.

Verite exposes a helper function isRevoked :

JavaScript await isRevoked(credential, statusList) Updated5 months ago *Table of Contents * * About * * Example of a Revocable Credential * * How an Issuer Creates a Revocable Credential * * Revoking a Credential * * Verification