Bachelor Thesis

Unlinkability of Verifiable Credentials in a practical approach

April 23, 2024

Joel Robles | TI

Inhaltsverzeichnis

Ziel

Was is das Ziel?

Die analyse, ob eine echt-welt implementation von Verifiable Credentials mit dem BBS Signature Scheme, unlinkability beibehält

Self-sovereign Identity

Self-sovereign Identity (SSI)

- Ist ein konzept wo eine Person (**Holder**) entscheiden kann, wer was über sie wissen darf
- Holders dürfen wählen was sie offenbaren und was nicht, auch bekannt als **selective disclosure**
- Erstes Problem:
  - Holder zeigt eine Staatliche ID
  - Ist eine Menge von Daten oder eine Menge von **attributes**
  - Die person welche verifiziert sieht alle attribute
- Zweites Problem:
  - Holder zeigt attribute einer person die diese verifizieren will, bekannt als **verifier**
  - Holder zeigt die gleichen attribute einem zweiten **verifier**
  - Der kann ge-**linked** werden
- Heutiger stand - Holder haben keine kontrolle über ihre Attribute
- Zukünftiger stand dank SSI - Holder haben volle kontrolle über ihre Attribute

Trust Triangle

- Wie weiss ein verifier das eine Menge von Attributen (**credential**) valid ist?
- Er vertraut dem issuer!
- Beispiel: Schweizer ID hat hologramme

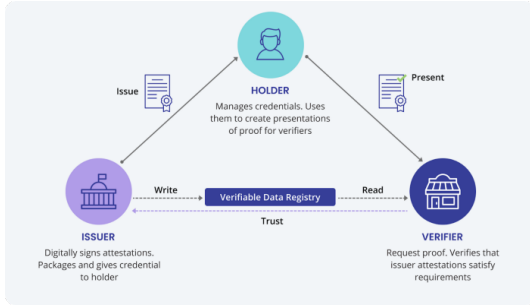

Abbildung: Trust triangle

Verifiable Credentials

Verifiable Credentials (VC)

- Verifiable Credentials sind eine digitale repräsentation von physischen Credentials
- JSON-LD repräsentiert attribute als **key-value pairs**
- Beispiel:
  - Vorname auf einer ID
  - Repräsentiert als {first_name: "John"}
  - first_nameïst der key und "Johnïst der value

Abbildung: Beispiel VC

VCs and BBS

- Warum werden sie **Verifiable** Credentials genannt?
- Der verifier kann ein VC, welches ihm präsentiert wurde (**Verifiable Presentation**), verifizieren, wegen Kryptographischen Signaturen
- Diese zeigen, dass das credential seit der ausstellung nicht verändert wurde
- Wir nutzen das BBS Signature Scheme (**BBS**)
- Diese Schema bietet **selective disclosure** and **unlinkability**
- Aber wie unlinkability? - Der Verifierbraucht die Signatur
- BBS kann **proofs** generieren
- Diese beweisen das der Holder die Signatur kennt, ohne diese zu offenbaren
- Weiter sind die proof unlinkable zwischen jeder generierung

Verifiable Presentations

Verifiable Presentation (VP)

- Ein holder würde gerne ein VC präsentieren
- Dafür werden **Verifiable Presentations** genutzt
- BBS kann nur staments signieren
- Der **RDF** canonicalization Algorithmus, welcher staments aus key-value pairs generiert

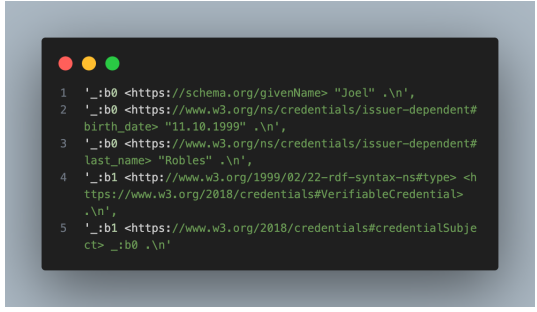

Abbildung: Beispiel canonicalized VP

Sicherheitsüberlegungen von VC/VPs

Shuffling of statements

- Der verifier bekommt ein VP von einem holder, welches gegen den public key vom issuer geprüft werden kann
- Der holder bekommnt eine version des credentials, wo ein attribut gewechselt wurde
- In einem sehr spezifischen fall für der RDF canonicalization Algorithmus zu einem Daten Leck
- Bei jedem gebrauch des Algorithmus, müssen die canonical staments mit einer Hahs function gemischt werden

OpenID Connect for Verifiable Presentations

OIDC4VP

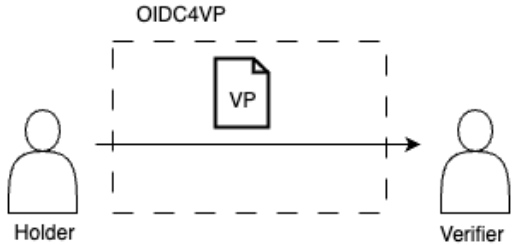

Abbildung: OpenID connect for Verifiable Presentations

OIDC4VP Fluss

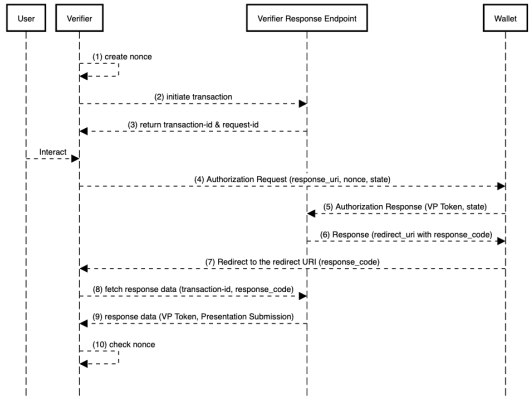

Abbildung: OpenID connect for Verifiable Presentations Fluss

Sicherheitsüberlegungen von OIDC4VP

Replay attack

- Der Holder sendet dem Verifier ein VP
- Ein Man in the Middle speichert die Vorstellung
- Der Man in the Middle kann das gespreicherte VP nutzen um sich als der Holder auszugeben
- Um dieses Problem zu umgehen, wird eine random nummer genutzt (nonce)

Session fixation attack

- Der Holder legt das VP im Endpoint des Verifiers ab
- Falls das System des Verifiers infisziert ist, kann ein Angreifer das VP einsehen
- Der endpoint des Verifiers retourniert eine redirect URL und ein response code
- Dieser code wird an das Terminal weitergeleitet, somit kann das infiszierte System das VP nicht einsehen

Ausblick

Ausblick

- Link Secrets und Blind BBS Signatures für linkability und selective disclosure analysieren
- Implementieren und testen