Funded by the National Science Foundation, Grant #2127548

Verifiable Credentials and Decentralized Identifiers

How do you prove who you are or what you are without giving away the keys to the castle?

Trigger Warning: Contains Blockchain

Funded by the National Science Foundation, Grant #2127548

Today’s Topic(s)

- Verifiable Credentials: Privacy-Preserving Claims and Assertions
- Decentralized Identifiers: VC’s foundation technology
- Digital Wallets

Funded by the National Science Foundation, Grant #2127548

The Central Problem: Privacy

We would like for end users to have control over the information they share, and with fine-grained control.

We'd like to do this in a way that minimizes information leakage.

A user's metadata is nearly as valuable as their information payload.

Funded by the National Science Foundation, Grant #2127548

Very simple example: Login

Here’s a ”simple” one: a user wants to log in to Wordpress by using a Google ID.

Current approach (grossly simplified):

1. User goes to Wordpress, hits the login page
2. Gets sent to Google’s login page
3. If the login is valid, Google calls in to Wordpress
4. Plenty of HTTP redirects hide many of the details

Funded by the National Science Foundation, Grant #2127548

The problems:

Google finds out I’m a Wordpress user

Wordpress finds out I’m a Google user

I have relatively limited control over how much information is shared between Google and Wordpress (unless both companies agree to play nicely)

Funded by the National Science Foundation, Grant #2127548

Enter Verifiable Credentials (‘VCs’)

Basic Idea:

1. A user (called a “subject”) gets a Verifiable Credential (a specific kind of JSON) from an “issuer” (maybe a university or DMV)
2. The user keeps a copy potentially forever (and is known as a “holder”)
3. To make the claim testified to by the VC, the user sends a copy of the VC to whoever wants to verifiy that claim (aka the “verifier”) - (“My name in Inigo Montoya”)

CI Compass

Funded by the National Science Foundation, Grant #2127548

Getting a Credential from an Issuer

Basic Idea:

Our “subject” communicates with some “issuer” of credentials and asks for a “verifiable credential” – a digitally-signed testament of some claim like “My name is Inigo Montoya”.

The issuer generates this credential (a JSON file in fact) and digitally signs it.

A hash of the cred is stored on a publicly-readable, only-writable-with-permission, blockchain (currently, an “OpenEtherium” variant, usually) – otherwise all kinds of attacks become trivial.

Our subject then holds on to this VC for a long time (milliseconds to centuries) in some way – on a thumb drive, at a cloud provider, in a Digital Wallet, or made into a QR Code and cast into bronze. Doesn’t matter.

CI Compass Funded by the National Science Foundation, Grant #2127548

Now we have a credential

Good news: that credential is now good for either

- All eternity, or
- Until the expiration date and time coded into the credential passes and the cred becomes invalid
- There is an issuer-generated hash of it safely stored in a publicly-readable place *but the specific contents are only known to the subject and potentially the issuer.*

Funded by the National Science Foundation, Grant #2127548

How use that credential

Our Wordpress example again:

1. Subject tries to log in to Wordpress, Wordpress asks for a VC.
2. User sends the VC which says “Arizona DMV is satisfied my name is Inigo Montoya”.
3. Wordpress (the “verifier”) hashes the VC and checks that the presented VC has a hash that matches the one the issuer put on the blockchain
4. Profit! (or at least “Access!”)

CI Compass

Funded by the National Science Foundation, Grant #2127548

Is that all? Seriously?

That’s the simplest use case.

Also expected to be the utterly dominant use case – 99+% of the time.

The Devil is, as always, in the details.

CI Compass
Funded by the National Science Foundation, Grant #2127548

Exchanging Minimal Information

If you send more than one piece of information, new knowledge can often be synthesized from it.

There are two people with my First, Middle, and Last names in my state.

Either Age, City, or whether we’ve ever logged into PGA.com is enough to tell us apart.

Funded by the National Science Foundation, Grant #2127548

Special Cases

Revocation: Issuer generates a Revocation VC just like the original, except the type is a revocation and the claimant identifier is changed from the subject’s ID to the revoked VC’s ID.

Disputes (“you spelled my name wrong, UNC!” or “I got phished and someone stole my signing key and made a fake cred somewhere!”) – similar to a revocation, except the subject generates it and saves the hash.

Caution: it’s up to the Verifier to check for this!

Funded by the National Science Foundation, Grant #2127548

Very Special Case: Derived Credentials

Idea: Subject uses one or more VCs to generate a new VC containing a subset of the original information

Use Case: The subject has a VC from a University stating their name, that they’re an undergrad and an English major. Whammy Corporation will let them watch movies for free if they’re an undergrad. The subject is willing to reveal that they’re an undergrad but not that they’re an English major.

Funded by the National Science Foundation, Grant #2127548

Derived Credentials

Easy if we don't need any security: Subject just edits the VC (it's a JSON file, just use emacs).

And… free movies for anyone with a text editor! That's suboptimal.

Solution: Use zero-knowledge proofs to show that the derived credential (the edited one) is in fact a subset of the originals.

Funded by the National Science Foundation, Grant #2127548

Zero-Knowledge Proofs

One party (the "prover") proves to another (the "verifier") that they know something (known only to the prover!) without revealing what that something is.

"Can I prove I know my password without revealing it?"

See: Quisquater, Jean-Jacques; Guillou, Louis C.; Berson, Thomas A. (1990). How to Explain Zero-Knowledge Protocols to Your Children (PDF). Advances in Cryptology – CRYPTO '89: Proceedings. Lecture Notes in Computer Science. Vol. 435. pp. 628–631. doi:10.1007/0-387-34805-0_60. ISBN 978-0-387-97317-3.

Funded by the National Science Foundation, Grant #2127548

ZKPs

Probability of correct proof increases exponentially with increased communications round trips.

If communication is either one-shot or not bi-directional, can be made to be equivalent in security to an algorithm’s underlying hash function.

Sufficiently complicated ZKP Protocol can show subject and issuer(s) agree on selected component(s) of VCs the derived VC stems from.

Funded by the National Science Foundation, Grant #2127548

Example VC’s JSON

```
{ "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/1872",
  "type": ["VerifiableCredential", "AlumniCredential"],
  "issuer": "https://example.edu/issuers/565049",
  "issuanceDate": "2010-01-01T19:23:24Z",
```

Funded by the National Science Foundation, Grant #2127548

Example JSON, cont. (2)

```
"credentialSubject": {
   "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
   "alumniOf": {
     "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
     "name": [{
       "value": "Example University",
       "lang": "en"
     }, {
       "value": "Exemple d'Université",
       "lang": "fr"
     }]  }   },
```

CI Compass — Funded by the National Science Foundation, Grant #2127548

Example JSON, cont. (3)

```
 "proof": {
    "type": "RsaSignature2018",
    "created": "2017-06-18T21:19:10Z",
    "proofPurpose": "assertionMethod",
    "verificationMethod": "https://example.edu/issuers/565049#key-
1",
    "jws":
"eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5XX16dUEMGlv50aqzpqh4Q
ktb3rk-BuQy72IFLOqV0G_zS245 blah blah blah” } }
```

Funded by the National Science Foundation, Grant #2127548

Decentralized Identifier (DID)

A Decentralized Identifier is a standardized way to take a URI and turn it into a JSON blob that identifies a person or other entity.

Roughly analogous to DNS lookups to turn hostnames into IP addresses.

CI Compass

Funded by the National Science Foundation, Grant #2127548

DID URI

did:example:1234567890abcdefghijkl

did:	This URI is a decentralized Identifier
example:	The DID method to use
1234…:	DID method-specific identifier

Funded by the National Science Foundation, Grant #2127548

DID Document

```
{ "@context": [
"https://www.w3.org/ns/did/v1",
"https://w3id.org/security/suites/ed25519-2020/v1"
]
"id": "did:example:123456789abcdefghi",
"authentication": [{ // used to authenticate as did:...fghi
"id": "did:example:123456789abcdefghi#keys-1",
"type": "Ed25519VerificationKey2020",
"controller": "did:example:123456789abcdefghi",
"publicKeyMultibase": "zH3C2AVvL blah blah blah" }] }
```

Funded by the National Science Foundation, Grant #2127548

DID Implementation

W3C Recommendation is reasonable as far as it goes, but leaves most of the heavy lifting to be defined through extensions.

Implementations typically rely on a Distributed Ledger (again, “blockchain”) to share signing keys and expiration dates between entities and resolvers.

CI Compass

Funded by the National Science Foundation, Grant #2127548

Actual Implementations

10 libraries in 7 programming languages (see W3C VC Working Group github)

Microsoft AzureAD

Funded by the National Science Foundation, Grant #2127548

It’s not all unicorns and rainbows

Security is hard. Sorry about that…

Data leakage from re-identification.
What happens when Verifier is Issuer
Interlopers
Correlation Attacks
Full protocol has tons of layers of complexity
Not fully specified enough to write proofs

Funded by the National Science Foundation, Grant #2127548

Thanks. Please Ask Questions