Attesting to Government Issued ID in a Self-Sovereign Identity System

Tim Speelman - 4096533

When putting government issued attributes to a Self-Sovereign Identity system, a problem emerges: the two systems use different identifiers. By definition the identity in an SSI is determined by the user, not by the government. We call this ID_{PK} for Public Key. An SSI can then be modelled as a mapping:

$$SSI : ID_{PK} \rightarrow A \text{ for any attribute } A \tag{1}$$

In contrast, the Dutch government identifies each Dutch citizen with a number *burgerservicenummer* (BSN), which is considered private. Information about citizens is stored in various registries managed by public registrars based on this identifier. A registry of attributes A can be modelled as a mapping:

$$Registry(A) : ID_{BSN} \rightarrow A \text{ for any attribute } A \tag{2}$$

Linking of these identifiers can be done by the same authority that distributes physical passports, the registry of BSNs. Effectively ensuring the following mapping:

$$Passport : ID_{PK} \rightarrow BSN \tag{3}$$

One way for registrars to feed their information into an SSI is to first verify (3), and then make an attestation in the form of (1). Protocol 1 illustrates this.

Protocol 1 Direct Attestaton

```
procedure BOOTSTRAP
    The government attests to the $BSN$ attribute, signing $ID_{PK} \rightarrow ID_{BSN}$.
end procedure
procedure ISSUING
    The registrar verifies $ID_{PK} \rightarrow ID_{BSN}$, issued by the government.
    The registrar now attests to the attribute $A$, signing $ID_{PK} \rightarrow A$.
end procedure
procedure VERIFYING
    The verifier simply checks the presence of an attribute $A$, attested by the registrar.
end procedure
```

Concerns

Sources within the Dutch Chamber of Commerce (Kamer van Koophandel, KVK), the registrar for legal entities, have expressed concerns that the legal framework within which they operate may not allow them to make claims in the form of (1). As the verification of $ID_{PK} \rightarrow ID_{BSN}$ is not information-theoretically secure, the claim $ID_{PK} \rightarrow A$ is not equivalent to the claim $ID_{BSN} \rightarrow A$.

Note that we cannot move the responsibility of verifying $ID_{PK} \rightarrow ID_{BSN}$ to a non-governmental verifier of A because BSN is considered a private attribute, not to be shared outside of governmental context.

Alternative Proposal

We propose to apply an ElGamal based commitment scheme because it is proven to be information-theoretically binding[1]. This allows the registrar to attest to a mathematical equivalent of the $ID_{BSN} \rightarrow A$ relation.

Protocol 2 Commitment Based Attestaton

Given a finite abelian group G of prime order q which is generated by g and h ($h \neq g$) where discrete logarithm of h to the base of g is unknown by any user in the system. Note that g and h need to be generated in a verifiably random manner. Also given random numbers r_1, r_2. We map BSN and A to this group as well.

```
procedure BOOTSTRAP
    The government attests to the commitment $c_1 = (g^{r_1}, BSN * h^{r_1})$
end procedure
procedure ISSUING
    The registrar verifies this commitment by receiving $c_1$, $BSN$ and $r_1$ from the user, thereby making sure the $ID_{PK}$ controls the $ID_{BSN}$.
    The registrar now attests to the commitment $c_2 = (c_1^{r_2}, A * h^{r_2})$.
end procedure
procedure VERIFYING
    The verifier checks that the user has a valid attestation $c_1$, issued by the government.
    The verifier checks that the user has a valid attestation $c_2$, issued by the registrar.
    The verifier receives $(A, r_2)$ and verifies $c_2 = (c_1^{r_2}, A * h^{r_2})$.
end procedure
```

TODO: The second usage of the commitment scheme is non-standard as we replace g by the earlier commitment c_1 so we must check that the same properties still hold.

[1] TODO cite: Cryptography made simple

Challenges

This alternative comes at a price of increased complexity. The bootstrapping, issuing and verifying phase all require additional logic. On a positive note, I believe that this technique could lighten the load for other registries as well, or may even be abstracted to a **generic mapping between identifiers in different systems**.

The other issue is that in the current Wallet application, each attribute is requested and listed separately and in a manner that is transparent to the user. By retrieving these commitments c_1, c_2 as attributes, the interface is cluttered with long numbers whose purpose is hard to explain. Hence, we need to hide these attributes, which may conflict with SSI principles.