PRIVACY : 3.0

CHAP 3:

Anonimity as Unlinkability : Extracting links is hard even if two connected organisations are corrupted and if users are targeted.

Minimal Disclosure : Combine keys into a composite key and each element key can have marked attributes.

↳ Allow mixing Cross-ID Credentials

↳↳ Prove a link existence without revealing it.

~~Key binding~~ :

Anonymous Credential :

↳ UNLINKABILITY => multiple uses of the same credential cannot be linked together, nor linked to the issuance event.

↳ SELECTIVE DISCLOSURE => The user can prove specific attributes or properties without revealing the full credential.

↳ USER CONTROLLED PRIVACY => The credential is stored and used by the user, the issuer cannot track when or where it is shown.

↳ UNFORGEABILITY => Only authorized issuers can create valid credentials, users cannot forge or modify attributes.

↳ MINIMAL DISCLOSURE => zk proofs reveal nothing beyond the truth of the statement being proven.

↳ PSEUDONYM => Several commitments of a single secret key.

↳ COMBINATION => COMMITMENT SCHEME
=> SIGNATURE SCHEME
=> ZERO-KNOWLEDGE PROOFS

Commitment scheme:

Cryptographic equivalent of sealed envelope.

Two phases:
- COMMIT: Sender chooses a message m and a commitment C. m remains secret (C is sent to a receiver)
- OPEN: Sender reveals m and receiver checks it matches C.

Security properties:
- HIDING: Before opening, receiver learns nothing about m.
- BINDING: After committing, the sender cannot change m.

Pedersen commitment: (Based on discrete log.)

$C \equiv (g, q, h)$

Secret $\alpha \in \mathbb{Z}_q \mid h = g^\alpha \% p$

Commit phase: Message $m \in \mathbb{Z}_q$

$r = \text{rand}(\mathbb{Z}_q)$

$C = g^m \cdot h^r \% p$ (C is sent)

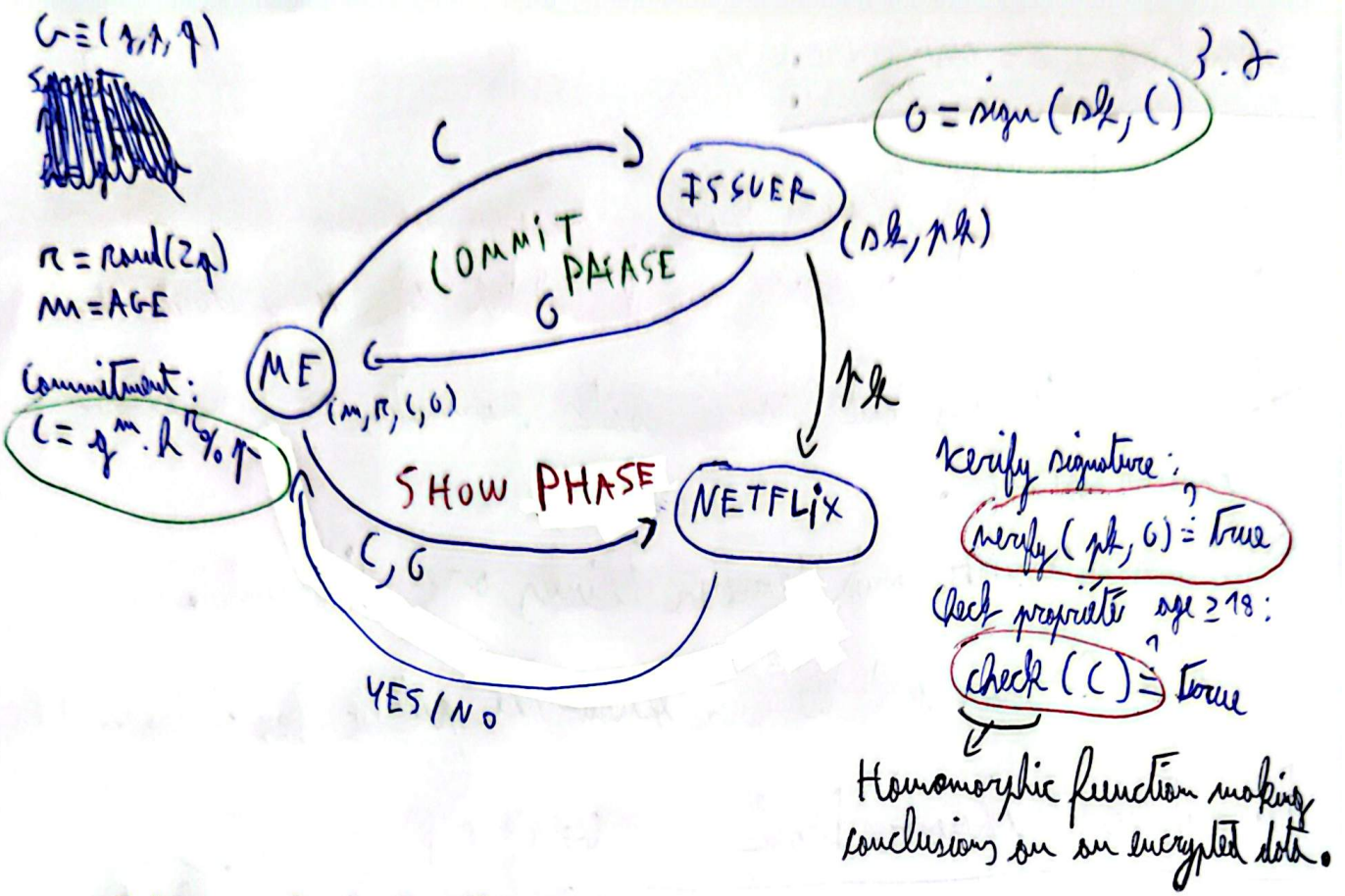
$\Rightarrow$ Alone it proves nothing, and need an issuer to make a signature* of it and provide it to the receiver.

Open phase: Reveal (m, r) (m and r are sent)

Verifier checks $C \stackrel{?}{=} g^m \cdot h^r \% p$

Commitments alone are useless and prove nothing.
But our user

* Issuer signs it with its private key and others check it with its public key.



Multi-Pedersen Commitment:

Same idea but we multiply different properties.

Several messages m_i where each one has its own unique generator g_i from the group $G = (g, r, q)$.

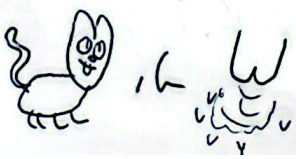
$$\begin{cases} C = \left(\prod_i g_i^{m_i}\right) \cdot h^r \,\%\, q = g_1^{m_1} \cdot \ldots \cdot g_n^{m_n} \cdot h^r \,\%\, q \\ pk = (G, g_1, \ldots, g_n, h) \end{cases}$$

(user)

Pseudonym, more randomization:

New $r' = \text{rand}(Zq)$

$C' = C \cdot h^{r'} = g_1^{m_1} \cdot \ldots \cdot g_n^{m_n} \cdot h^{r+r'}$



$\Rightarrow$ Perfect unlinkability between C and C'.

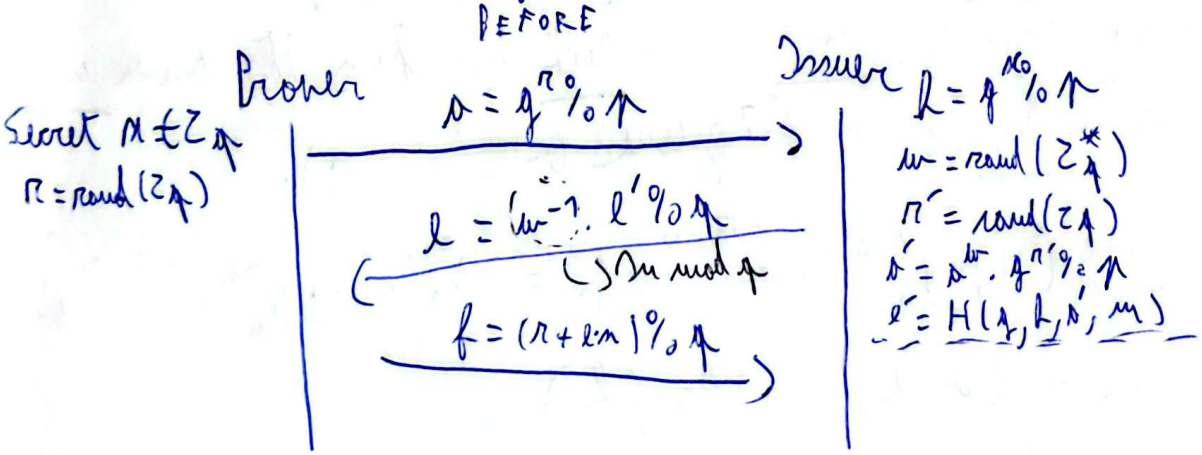
Signature $\sigma = sign_{sk}()$:

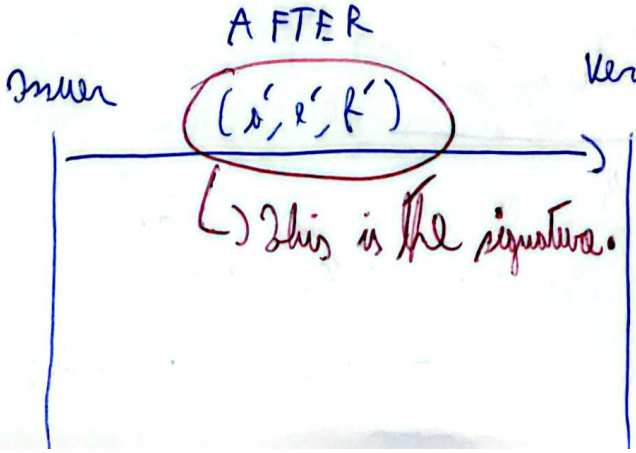
Idea: Divertible Schnorr Proof: Allows someone else than me and more trusted to prove my knowledge of or my secret x.

- Between Prover and Issuer acts normally as a no classic Schnorr proof.
- Between Issuer and Verifier thinks are different.
- Both Issuer and Verifier know the public key of Prover (h).

Divertible Schnorr Proof: $\sigma = (s, t, \tilde{r})$

- Issuer has to be able to prove knowledge of x without
- ... knowing x and without to be able to prove anything else.
- Prover lets the issuer finish Schnorr Transcripts, but in a blinded way.





Divertible Chaum-Pedersen: 3.4

- Same idea than divertible Schnorr.
- Réutilisation of 2 bases g_1, g_2 generating same group G.
- $\Rightarrow h_1, h_2, s_1, s_2, s_1', s_2'$
- $c' = H(s_1', s_2', h_1, h_2, m) \% q$
- We use it to sign a commitment here.
- In a commitment $C = u_1^{m_1} \cdot \ldots \cdot u_l^{m_l} \cdot u_0^{r} \% p$, all u_i are bases of G like g, all m_i are secrets in $\mathbb{Z}_q$ like x (and r too) so by definition C is also a base of G.

Clarification before starting:

- $d = c^x$ ≡ Exponentiation of the commitment:
  1) Prove knowledge of x relative to c ($\log_c(d) = x$)
  2) Re-use or even randomize the commitment later, while still proving knowledge of the same secret x.
- g, h are the classic bases of G, c is another base of G
- Re-randomization $\begin{cases} c' = c \cdot u_0^{r'} \\ d' = (c')^x \end{cases}$ $d' = c \cdot u_0^{r'} \Rightarrow (c')^x = c^x \cdot (u_0^{r'})^x$ ✓ $= d \cdot v_0^{r'} = d'$ so $v_0 = u_0^x$

Instead of computing $(c')^x$ requiring knowing x, precompute $v_0 = u_0^x$ and later just multiply by $v_0^{r'}$ to get d'.

- $\begin{cases} d = c^x \equiv \text{Original target for the proof over base } c \\ v_0 = u_0^x \equiv \text{Helps compute randomized one } d' \text{ efficiently} \\ c', d' \rightarrow \text{Re-randomized} \end{cases}$

→ Goal is blind / divertible signature, not a user proving knowledge of its own secret x.

- d is a kind of public key for x based on commitment c. 3.5
- (c', d') are re-randomized commitments / keys and (s', T') a re-randomized signature (use same signature for each service but unlinkable as it seems different on each service) that the prover will provide to the verifier.
- Prover with signature proves 2 things: key in my commitment c is true and the signature proves knowledge of x that is the secret of a trusted issuer.
- $\Pi = SPK\{(m_1, \dots, m_l, r) : c = u_1^{m_1} \cdot \dots \cdot u_l^{m_l} \cdot u_0^{r}$
  - A lot of Schnorr proof for proving each single m_i (in the model x) with all proofs using the same random r. Proves that user knows each single commitment without revealing it.
- Values for $u_0, u_1, \dots, u_l$ are agreed in advance as convention and are public.