Privacy :

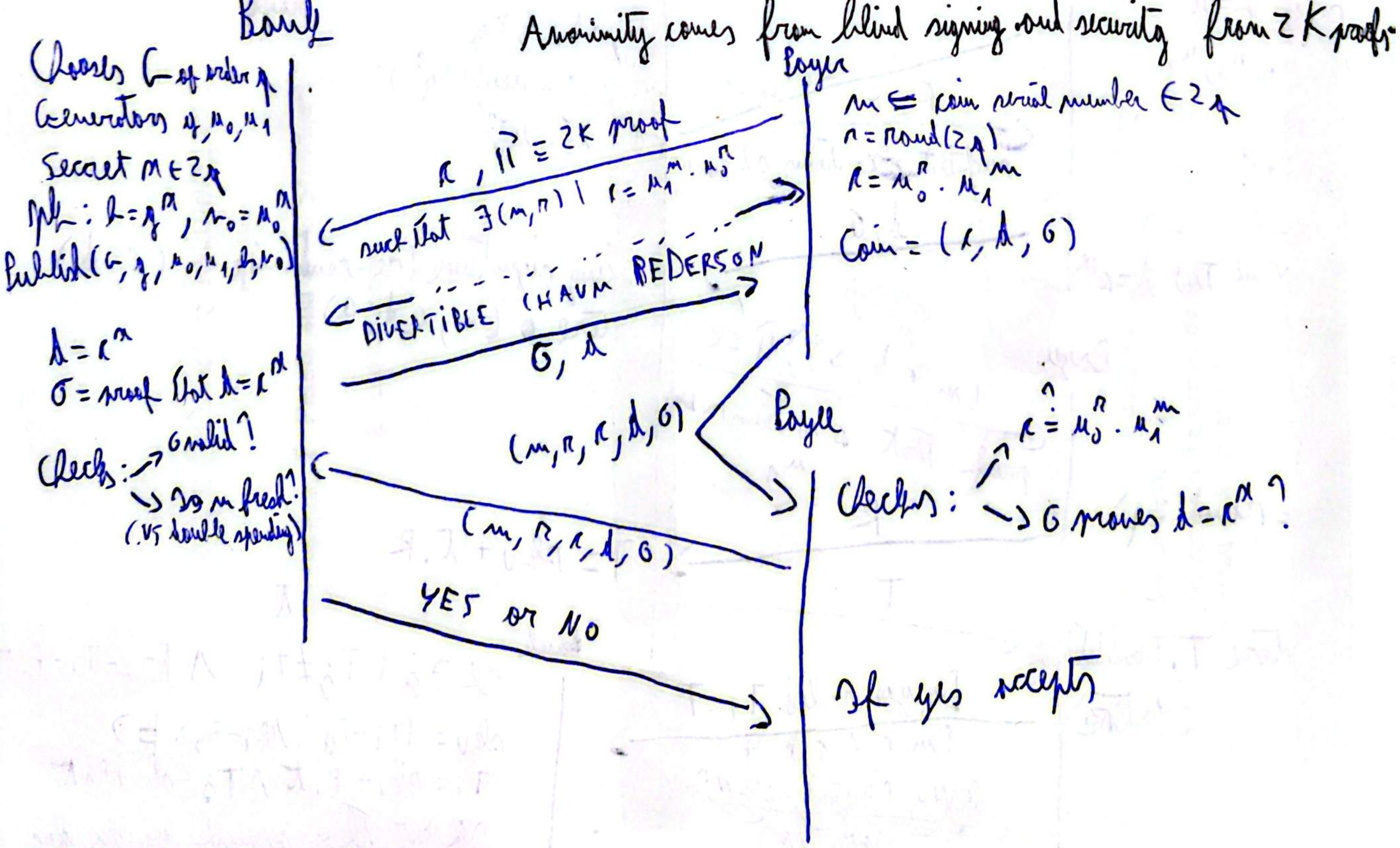
CHAP 4 :

Chaum E-Cash and Anonymous Credential :

* Bank builds a signature sig σ proving the relation $d = c^{x} \% p$. Bank can't link withdraw to future spending.

1) Bank Setup : Bank chooses G of prime order p — Generators g, u_0, u_1 ; Secret $x \in \mathbb{Z}_p$; Publish $h = g^{x_0} \% p$, $v_0 = u_0^{x} \% p$
2) Coin structure : Payer makes a commitment $c = u_0^{r} \cdot u_1^{m} \% p$
3) Withdraw of new coin : m, r, R ; Payer ZK proves to Bank that c is well formed. $r = \text{rand}(\mathbb{Z}_p)$; $m \equiv$ Coin serial number *
   => Divertible Chaum Pedersen
4) Spending : Payer reveals (m, r, c, d, σ), Payee verifies $c = u_1^{m} u_0^{r}$ and σ proves $d = c^{x}$, then forwards to bank.
5) Bank verification : Bank verifies σ, checks if m is fresh against double spending.
6) Anonymity holding : withdraw is blind and divertible Chaum Pedersen can be re-randomized.
7) Key takeaways : $m \neq$ Money amount => it's a coin serial number
   σ isn't a classic signature, it ~~just~~ just proves $d = c^{x}$

Anonimity comes from blind signing and security from ZK proofs.



Chaum offline E-Cash:

1) Hidden user identity: Each payer has a secret identity key sk_u.

2) Coin structure: $c' = u_1^{m} \cdot u_2^{sk_u} \cdot u_3^{t} \cdot u_0^{r+r'}$

m = coin serial number
sk_u = user identity (hidden) Then $c = c' \cdot u_0^{-r'}$
t = one-time randomness
r, r' = random binding values
The bank blindly signs $(c, d = c^{\alpha})$

3) Withdraw: Payer proves ZK to bank c' contains a valid sk_v on u_2 and removes r' (computation of c) to allow the bank to sign c.
Bank blindly signs (c, d) and learns nothing about m or sk_v.

4) Offline spending: Payer and payee make an offline transaction without the need to pass by the bank. ~~Payer to~~ $A = $ Payer, $B = $ Payee
$A \to B: (m, c, d, \sigma)$ proof of c, $B \to A$: R random challenge
Then $A \to B: T = sk_v + R t$ consistency of T and c.

5) Double spending detection:
If twice we got $2 T_i$ so: $T_1 = sk_v + R_1 t$; $T_2 = sk_v + R_2 t \Rightarrow sk_v = \frac{T_1 - T_2}{R_1 - R_2}$: Payee can check equation if it is true!!
sk_v revealed to payee

Bank | **Payer**

Bank: $\alpha, g, h = g^{\alpha}$; u_0, u_1, u_2, u_3

Payer: $m, sk_v \in \mathbb{Z}_p$; $r, r', t = \text{rand}(\mathbb{Z}_p^3)$; $c = c' \cdot u_0^{-r'}$

Payer → Bank: c, Π ZK proof such that c' contains sk_v on u_2

Bank: $d = c^{\alpha}$; $\sigma = (\ldots)$ proof that $d = c^{\alpha}$

Bank → Payer: d, σ

Payer: Bling signature (re-randomize) (c, d); Store (m, c, d, σ)

Payee | **Payer**

Payer → Payee: $(m, c, d, \sigma), \Pi$ ZK proof that c contains m on u_1

Payee: $R = \text{rand}(\mathbb{Z}_p)$

Payee → Payer: R

Payer: $T = sk_v + t \cdot R$

Payer → Payee: T

Payee: Store T, T validated (later)

Payee → Bank: Do you valid $T_i = T$? (m, c, d, σ, R, T)

Bank → Payee: ~~NO DOUBLE SPENDING~~ YES / NO

Bank: If $\exists j \mid T_j \neq T_i \wedge sk_u = T_i - R_i t$
$t = (T_i - T_j)/(R_i - R_j)$
$T_i = sk_v + R_i t \wedge T_j = sk_v + R_j t$
If not, then punish payer for [illegible]