Outline

Chaum-Pedersen ZK

Scribe: Nathan
Proofs on Encrypted Data
Chaum-Pedersen

• Public parameters: $(p, g, h)$
  • $p$: large prime (1024 bit)
  • $g$: generator

• Proof of a given triple being of the following form

$$(u, v, w) = (g^a, g^b, g^{ab})$$

• NP relation $R = \{ (u, v, w) : \exists (a, b) \text{ s.t. } u = g^a, v = g^b, w = g^{ab} \}$
Chaum-Pedersen: first recall Schnorr

Prover

\((u, v, w) = (g^a, g^b, g^{ab})\)

\(r \leftarrow \mathbb{Z}_p\) (\(R = g^r\))

\(e = r + p \cdot b\)

Verifier

\((u, v, w)\)

\(p \leftarrow \mathbb{Z}_p\)

\(R = g^r\)

\(e\)

\(g^r \cdot (v)^p = g^e\)
Chaum-Pedersen: first recall Schnorr

**Prover**

\[(u, v, w) = (g^{a}, g^{b}, g^{ab})\]

\[r \leftarrow Z_{p} \quad (R = g^{r})\]

\[\text{Check: } g^{e} = R.(v)^{p}\]

\[e = r + p.b\]

\[T = R^{a}\]

**Verifier**

\[(u, v, w)\]

\[p \leftarrow Z_{p}\]

\[\text{Check: } h^{e} = T(w)^{p}\]
Chaum-Pedersen

Suppose we have a cheating Prover

Prover

\[(u, v, w) = (g^a, g^b, g^{ab})\]

\[r \leftarrow \mathbb{Z}_p \quad (R = g^r, T = g^{ar})\]

\[e = r + p.b\]

\[(u, v, w) = (g^a, g^b, g^{ab})\]

Verifier

\[(u, v, w)\]

\[p \leftarrow \mathbb{Z}_p\]

\[p = r + p.b\]

Check 1: \[g^e = R.(v)^p\]

\[e = r + bp\]

Check 2: \[u^e = T.(w)^p\]

\[e = e' + bp\]
Chaum-Pedersen

**Prover**

\((u, v, w)\)

\[r \leftarrow \mathbb{Z}_p\quad (R = g^r, \ T = h^r)\]

\[e = r + p.b\]

\((u, v, w) = (h, g^b, w = h^b)\]

**Verifier**

\((u, v, w)\)

\[p \leftarrow \mathbb{Z}_p\]

\[p \leftarrow \mathbb{Z}_p\]

\[e = r + p.b\]

Check 1: \(g^e = R.(v)^p\)

Check 2: \(h^e = T.(w)^p\)

\[
e = r + bp = r' + b'p \]

\[
r - r' = p(b' - b)\]

\[
\frac{r - r'}{b' - b} = p\]

But, \(\Pr[A \text{ guesses } p] = \frac{1}{\exp(n)}\)
Chaum-Pedersen: Soundness

**Prover**

\[(u, v, w) = (g^a, g^b, g^{ab})\]

\[r \leftarrow Z_p, \quad (R = g^r, T = g^{ar})\]

\[e = r + p.b\]

\[p \quad \rightarrow \quad e \quad \rightarrow \quad e' \quad \rightarrow \quad e' = r + p'.b\]

\[(u, v, w) = (g^a, g^b, g^{ab})\]
Chaum-Pedersen: Soundness

**Prover**

\[(u, v, w) = (g^a, g^b, g^{ab})\]

\[r \leftarrow Z_p, \quad (R = g^r, T = g^{ar})\]

\[e = r + p.b\]

\[e' = r + p'.b\]
Chaum-Pedersen: Zero-Knowledge

Simulator

$(u, v, w), \text{guess } p$

$e \leftarrow Z_p, (R = g^e/v^p, T = u^e/w^p)$

Verifier

$(R, T)$

$p$

$e$

Show: \((R, T), p, e)_{\text{Sim}} = (R, T)_{p, e}\)
Honest Proof.

\[
R = g^r \quad T = g^ar \\
P \\
e = r + p \cdot b
\]

Simulated.

\[
P \\
e \leftarrow Z_p \\
R = g^e \cdot g^{bp} = g^{e-bp} \\
T = \frac{u^e}{g^{ap}} = u^{e-bp}
\]
Proofs of Encrypted Data
Encrypting bits

Prover

Verifier

NP language = \{ (g,h,v,w) : \exists \text{ (bit, b) such that bit} \in \{0,1\} \text{ and } v = g^b, w = h^b \cdot g^{\text{bit}} \}
OR Composition

Suppose we have a protocol \((P_0, V_0)\) for \(R_0\), and a protocol \((P_1, V_1)\) for \(R_1\).

Can we combine them to obtain a protocol for \(R_0 \text{ OR } R_1\)?
Suppose we have a protocol \((P_0, V_0)\) for \(R_0\), and a protocol \((P_1, V_1)\) for \(R_1\).

What about running the two in parallel?
Suppose we have a protocol \((P_0, V_0)\) for \(R_0\), and a protocol \((P_1, V_1)\) for \(R_1\). What about letting the prover simulate exactly one of them?
OR Composition

**Prover**

**Verifier**

Suppose we have a protocol \((P_0, V_0)\) for \(R_0\), and a protocol \((P_1, V_1)\) for \(R_1\).
OR Composition

**Prover**

**Verifier**

Suppose we have a protocol \((P_0, V_0)\) for \(R_0\), and a protocol \((P_1, V_1)\) for \(R_1\).
OR Composition

**Prover**

**Verifier**

Suppose we have a protocol \((P_0, V_0)\) for \(R_0\), and a protocol \((P_1, V_1)\) for \(R_1\).