Zero Knowledge Proofs notation

- Alice knows $x^k = X$
- Public key

ZKP is
- Alice can say: "I know my secret key $x$ s.t. $X = g^x$"
- Statement
- Relation/predicate $X = g^x$

If $X = g^x$
- Witness
- Statement
- Relation/predicate $X = g^x$

Where's valid ZKP?

Language $L$ is a set of strings $x \in L \iff \exists w, s.t. L(x, w) = 1$

Ideal Functionality

Verifier

Prover Alice $x, w$ → Ideal functionality checks $L(x, w)$ if $y = x$

Bob → "works as good as the IF"

Interactive Protocol

Verifier

Prover Alice

$P(x, w)$ → "challenge"

Verifier Bob

Response $V(x)$ starts with "just statement"
- Output:
  \[ V(\eta(x)) \]
  \[ \eta(\text{sim}) \]
  means output of \( V \)
- View:
  \[ V(\eta(x)) \]
  means a transcript of all messages received by \( V \)
- Random choices made by \( V \).

- Correctness:
  "verifier accepts if \( \text{prover is honest} \)"
  \[
  \forall x, w, L(x, w) = 1, \quad P_{\eta} \left[ \text{out}_{\eta} \left[ \eta(\text{sim}) \right] = 1 \right] = \frac{1}{2}
  \]

- Soundness:
  "verifier only accepts \( x \in L \)"
  \[
  \forall \eta, x, \quad P_{\eta} \left[ \text{out}_{\eta} \left[ \eta(\text{sim}) \right] = 1 \right] \leq 1 - \delta
  \]

- Extraction (stronger than soundness):
  "verifier only accepts if \( x \in L \)
  and prover "knows" witness \( w \"

- Zero Knowledge:
  "verifier knows no more information after the protocol than before"
  "view of the verifier can be simulated even without interacting with the prover"

\[ \forall x, w, L(x, w), \exists s \approx_{\varepsilon} \text{ simulator} \]
\[ \text{View}_{\eta} \left[ \eta(\text{sim}) \right] \approx \left[ V(\eta(x)) \right] \]
\[ \Downarrow \]
\[ \text{real transcript} \]
\[ \Downarrow \]
\[ \text{simulated transcript} \]

Protocol for \( ZK(\hat{\xi}) \): \( X = g^{x} \hat{\xi} \), \( |G| = p \)

\[ P(X, x) :\]
\[ k \in \mathbb{Z}_{p} \]
\[ K = \alpha^{k} \]
\[ \Sigma \text{ sigma protocol} \]

\[ P(X) \rightarrow V(X) \]
\[ \rightarrow \hat{\xi} \rightarrow \Xi \]

\[ \rightarrow \hat{\xi} \rightarrow \Xi \]
\[ S = xc + k \]

- Correctness: \[ g^S = g^{xc + k} = (g^x)^c g^k = x^c K \]

- Zero knowledge:
  \[
  \text{View } \left[ P(k, x) \rightarrow V(x) \right]
  \]

  \[ (k, s) \text{ where } g^s = x^c K \]

  \[ S(x) = K \in G_1, c \in \mathbb{Z}_p, \quad s = \log_g K + c \log_g x \]

  \[ c \in \mathbb{Z}_p, s \in \mathbb{Z}_p, K = s^i / x^c \]

- Extraction for next time:
  - If \( A \) produces a valid proof for \( x \), with high prob.
  - Extractability: \( A \) can use \( U \) to adapt a witness.

  \[ \text{Crun } A \text{ multiple times} \]

- Adapting validity:
  \[ A(x, z) \]

  \[ c \in C_1 \Rightarrow A \text{ adapts validity} \]

\[ \prod \text{adapt validity} \]

\[ S^u A \]

\[ \text{Adapt validity} \]
"Run A twice, with the different challenges."

We construct an extractor $\mathcal{E}_A(\pi^2, X)$:

1. $z \leftarrow Z_{\ell^2}$ (where $\ell$ bits of randomness are needed).
2. Run $A_1$ until it outputs a message $K$:
   $$\mathcal{A}(\pi^2, X; z) \rightarrow K.$$
3. $c_1 \leftarrow Z_p$ (Send $c_1$ to $A_1$, run until receiving $s_1$).
4. Run $\mathcal{A}(\pi^2, X; z)$ a second time, $A_2$ outputs $K$.
5. $c_2 \leftarrow Z_p$ (Send $c_2$ to $A_1$, receive $s_2$).

If $\text{adv}_{\mathcal{A}}(\pi^2, C) = 1$ with prob $\epsilon$, then

$$g^{s_1} = X^{c_1}K$$
and
$$g^{s_2} = X^{c_2}K$$

$g^{s_1}/g^{s_2} = X^{c_1}K/X^{c_2}K$

$g^{s_1-s_2} = X^{c_1-c_2}$

$g^{s_1-s_2} = (g_{s_1-s_2})^{c_1-c_2}$

Solve for $X = (s_1-s_2)/(c_1-c_2)$.

Commitments:

Create a commitment for $\pi^2$ with a public $\bar{r}$:

$$Z_{\ell^2} \times \{0, 1\}^\ell: C = g^x h^r$$

Commitment Schemes

$A \xleftarrow{\$} Z_p$ (secret $x \in Z_p$)

Bob stores $C$

Bob reopens $\text{Com}(\pi^2, C)$

Pedersen Commitments

$\text{Com}(x, r) = g^x h^r$
\[ 1 \wedge \exists \left( \frac{\ell = g^x}{C} \right) \rightarrow U(L) \]

Why? For any \( x \), and \( C \), exactly one \( r \) exists such that
\[ g^x \cdot r = C \quad \text{and} \quad h^r = C / g^x \]
\[ r = \log_h C / g^x \]

**Discrete log commitment**

\[ g^x = C \quad \text{Hash ("Scissors") (only valid for } x \text{ sampled from large space)} \]

**Could PKE commitment?**

\[ \text{Com}(x, r) = \frac{g^r}{g^{r(x)}} \]

\[ \text{Proof:} \]

Suppose \( A \) breaks "binding."

We have to construct \( A' \) that solves DL-Gr.

\[ A'(1^n, g, X) \rightarrow X = g^x \]

\[ A(1^n, g, X) \rightarrow X_1, X_2, r_1, r_2, C \]

\[ C = g^{X_1 r_1} \quad X_1 \neq X_2 \]

\[ C = g^{X_2 r_2} \quad X_1 \neq X_2 \]

\[ g^{X_1 - X_2} = g^{r_2 - r_1} \]

\[ g^{(X_1 - X_2) / (r_2 - r_1)} = X \]

Output \( (X_1 - X_2) / (r_2 - r_1) \)

**DL-Gr hard**

\[ \Rightarrow \text{Pegerson is binding} \]

Prove "attack on Pedersen \[ \Rightarrow \text{attack on DL-Gr} \]

**Zero Knowledge proof**
\[ K = g^{k_1} h^{k_2} \]

- Correctness:
  \[ g^{s_1} h^{s_2} = g^{xc+k_1} h^{rc+k_2} = (g^x)^c g^{k_1} (h^c)^c h^{k_2} = (g^{x+c})^c (g^{k_1}/k_2) = c^c K \]

- Simulation: \( g^{s_1} h^{s_2} \approx c^c K \)

First step: What is \( \text{Ver}(\text{P}(x), y) = (K, c, c, a) \) and get \( (K, c, s_1, s_2, c_2, s_1, s) \).

Application:
- ZK proofs about commitments and opening

\[ \text{ZK}_E(x, c): c = g^x h^r \text{ and } \text{first } d_r \text{ of } x \text{ is } 0 \]

\[ \text{ZK}_E(\text{doc, sig}): \text{doc is signed by sig from the DMV} \]

Possibility:

\[ \text{hash}(\text{name}) \]

\[ \text{hash}(\text{age, r}) \]

\[ \text{Reveal just the portion you need} \]

OR proofs: \[ \text{ZK}_E(x): x_1 = g^x \]
I know ONE of these public keys.

Run N Pseudorandom keys.

We use the simulator for these.


We have 2 degrees of freedom.

I choose C, S3, S1.

Let prove choose C, S3, S1.

In order to prove that...