Zero Knowledge Proofs

Monday, September 27, 2021 4:40 PM

Zero Knowledge Proofs notation

Alice x, $g^x = X$

private key, public key

Bulletin Board: X ← public key

"no one can compute x from X by DLOG"

ZKP is

Alice can say: "I know my secret key x s.t. $X = g^x$"

"statement"

w/o revealing x

$$\text{ZKP}\{(x) : X = g^x\}$$

- witness
- statement
- relation/predicate $X = g^x$

Alice sends $y = F_k(1)$

Can Bob check y is actual $F_k(1)$ w/o knowing k?

Where's Waldo ZKP

Language $L^{\in NP}$ is a set of strings

$x \in L \iff \exists w$ s.t. $L(x, w) = 1$

statement, witness, predicate

Setup once per predicate

Valid for many statements

Ideal Functionality

Prover Alice $\xrightarrow{x, w}$ Ideal Functionality: Checks $L(x,w)$ if so, out x $\xrightarrow{x}$ Verifier Bob

"learns $x \in L$, nothing else"

"protocol is secure iff it is as good as the IF"

(run the protocol) $\approx$ (interact with IF)

Interactive Protocol

Prover Alice

Verifier Bob

"commit" →

← "challenge"

"response" →

$P(x, w)$

$V(x)$ verifier starts w/ just statement

prover starts

output Accept/Reject

prover w/ wit & stmt ... output accept/reject

- $\text{out}_V[P(x,w) \leftrightarrow V(x)]$

means final output of V

- $\text{View}_V[P(x,w) \leftrightarrow V(x)]$

means a transcript of all
- messages received by V
- random choices made by V.

- Correctness: "verifier accepts if prover is honest"

$$\forall x, w.\ L(x,w)=1,\ \Pr\left[\text{out}_V[P(x,w) \leftrightarrow V(x)] = 1\right] = 1$$

- Soundness: "verifier only accepts $x \in L$"

$$\forall A, x\quad \Pr\left[\begin{array}{l}\text{out}_V[A(x) \leftrightarrow V(x)] = 1 \\ \text{and} \\ x \notin L\end{array}\right] \leq \text{negl}$$

- 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"

$\forall x, w,\ L(x,w),\ \exists S$ ← simulator

$$\text{View}_V[P(x,w) \leftrightarrow V(x)] \approx S(x)$$

⇓ real transcript interacting w/ Prover.

⇓ Simulated transcript

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

$P(\cancel{X}, x)$: "commit" $V(\cancel{X})$

$k \xleftarrow{\$} \mathbb{Z}_p$

$K = g^k$ $\xrightarrow{K}$

$c \xleftarrow{\$} \mathbb{Z}_p$

Σ sigma protocols

$c \in \mathbb{Z}_p$

"challenge"

c

$s = xc + k$ "response" s

$$g^s \stackrel{?}{=} X^c K$$

- Correctness: $g^s = g^{xc+k} = (g^x)^c g^k = X^c K$

- Zero-knowledge:

$\mathrm{View}_V[P(x,w) \leftrightarrow V(x)]$

$|G|$ $\quad|\mathbb{Z}_p|\quad|\mathbb{Z}_p|$

(K, c, s) where $g^s = X^c K$

exactly equal distributions.

$$\Pr\left[\text{transcript} = (K, c, s)\right] = \begin{cases} 1/p^2 & \text{if } g^s = X^c K \\ 0 & \text{otherwise} \end{cases}$$

$S(X)$: ✗ $K \xleftarrow{\$} G,\ c \in \mathbb{Z}_p,\quad s = \log_g K + c \log_g X$?

✓ $c \xleftarrow{\$} \mathbb{Z}_p,\ s \xleftarrow{\$} \mathbb{Z}_p,\ K = g^s / X^c$

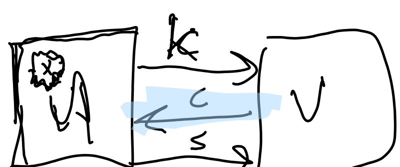
- Extraction for next time.

"If $\mathcal{A}$ produces a valid proof for X, with high prob then we can use $\mathcal{A}$ to adapt a witness"

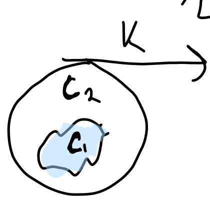
- extractability

(possibly malicious prover)

(run $\mathcal{A}$ multiple times)



V says OK

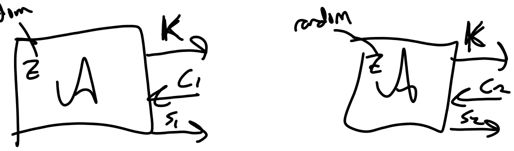


Possible values of c

$\mathcal{A}(1^\lambda, X; z)$ $\quad c \in C_1 \Rightarrow \mathcal{A}$ adapts validly

rand bits.

Success set C_1 must be nonnegligible.



$[\mathcal{A} \leftrightarrow P]$

"

"run $\mathcal{A}$ twice, with two different challenges"

We construct an extractor $\mathcal{E}_{\mathcal{A}}(1^\lambda, X)$:

$z \xleftarrow{\$} \{0,1\}^{q(\lambda)}$ ← however many bits $\mathcal{A}$ needs

run $\mathcal{A}$ until it outputs first message K

$\mathcal{A}(1^\lambda, X; z) \to K$.

(I) $c_1 \xleftarrow{\$} \mathbb{Z}_p$

Send c_1 to $\mathcal{A}$, run until receiving s_1.

run $\mathcal{A}(1^\lambda, X; z)$ a second time, and outputs K.

(II) $c_2 \xleftarrow{\$} \mathbb{Z}_p$

Send c_2 to $\mathcal{A}$, receive s_2.

IF $\text{out}_V[\mathcal{A} \leftrightarrow V] = 1$ with prob $\underline{d}$,

then $g^{s_1} = X^{c_1} K$ w/ prob at least $\underline{d^2}$

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

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

$g^{s_1 - s_2} = X^{c_1 - c_2}$ ← size $|G| = p$, division mod p.

$\left(g^{s_1-s_2}\right)^{1/(c_1-c_2)} \quad \left(X^{c_1-c_2}\right)^{\left(1/(c_1-c_2)\right)}$

Solve for $x = (s_1 - s_2)/(c_1 - c_2)$. ← division mod p

Commitments.

Exercise: Come up with a protocol for

$\underline{ZK\{(x,r) : C = g^x h^r\}}$

Commitment Schemes

"Commitment"

Setup: $G, g, h \xleftarrow{\$} |G|$

secret → $x \in \mathbb{Z}_p$

$\mathcal{A}$ $\quad r \xleftarrow{\$} \mathbb{Z}_p$

$C = \text{Com}(x, r)$ → Bob stores C

x, r → "opening"

$\boxed{\text{Bob recompute } \text{Com}(x,r) \stackrel{?}{=} C}$

$\boxed{\text{Pedersen Commitments} \quad \text{Com}(x,r) = g^x h^r}$

— "hiding"

$\forall x \quad \{ r \xleftarrow{\$} \mathbb{Z}_p \quad C = \ldots \} = \ldots$

$\forall x. \{ \frac{C = g^x h^r}{C} \} = U(G)$

Why? for any x, and C, exactly one r s.t. $g^x h^r = C$ $\quad$ $h^r = C/g^x$

$r = \log_h C/g^x$

- "binding"

$\forall \mathcal{A}$, $\Pr\left[\begin{array}{c} g \xleftarrow{\$} G, h \xleftarrow{\$} G \\ x_1, x_2, r_1, r_2, C \leftarrow \mathcal{A}(1^n, g, h) \\ \hline x_1 \neq x_2, C = Com(x_1, r) \\ C = Com(x_2, r) \end{array} \right] \leq \text{negl}$

Discrete log Commitment

$g^x = C$ $\qquad$ Hash("scissors")

only valid for x sampled from large space

- Binding for Pedersen Commitment reduces to DLOG.

Proof:

Suppose $\mathcal{A}$ breaks "binding"

We have to construct $\mathcal{A}'$ that solves DLOG.

Could PRF a commitment?

$Com(x, r) = F(r, x)$?

Ooooo

"DLOG hard $\Rightarrow$ Pedersen is binding"

prove "attack in pedersen $\Rightarrow$ attack on DLOG"

$\mathcal{A}'(1^n, g, X)$:

// need to find x s.t. $X = g^x$

$\mathcal{A}(1^n, g, X) \rightarrow x_1, x_2, r_1, r_2, C$

$C = g^{x_1} X^{r_1}$ $\quad x_1 \neq x_2$

$C = g^{x_2} X^{r_2}$

$g^{x_1 - x_2} = X^{r_2 - r_1}$

$g^{(x_1 - x_2)/(r_2 - r_1)} = X$

output $(x_1 - x_2)/(r_2 - r_1)$

$ZK\{(x, r) : C = g^x h^r\}$

before g^x $\quad$ now $g^x h^r$

"Zero Knowledge proof"

Prover $\qquad$ Verifier

$k_1 \xleftarrow{\$} \mathbb{Z}_p$ $\qquad K$

$k_2 \xleftarrow{\$} \mathbb{Z}_p$ $\longrightarrow$

$k_2 \leftarrow \mathbb{Z}_p$ ⟶ "of knowing the opening of a commitment"

$K = g^{k_1} h^{k_2}$ $\qquad c \xleftarrow{\$} \mathbb{Z}_p$

$\xleftarrow{\quad c \quad}$

$s_1 = xc + k_1$

$s_2 = rc + k_2 \xrightarrow{\quad s_1, s_2 \quad} \underline{g^{s_1} h^{s_2}} \stackrel{?}{=} C^c K$

- Correctness: $g^{s_1} h^{s_2} = g^{xc+k_1} h^{rc+k_2}$

$$= (g^x)^c g^{k_1} (h^r)^c h^{k_2}$$

$$= (g^x h^r)^c (g^{k_1} h^{k_2}) = C^c K$$

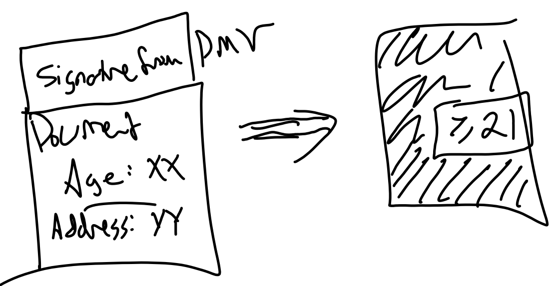
- Simulation ⟹ "construct a simulator"
- Knowledge ⟹ construct an extractor

First step! what is $View_V[P(x,$ … is $(K, c,$ …

run A to get $(K, c_1, s_{11}, s_{21}, c_2, s_{12}, s_{2}$ …

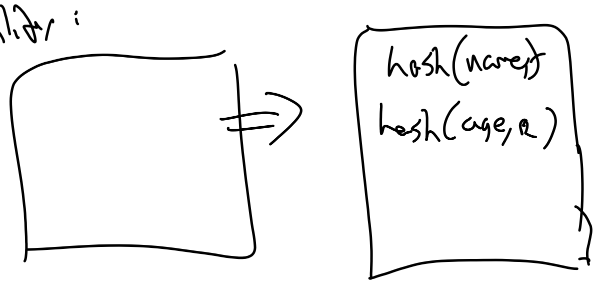
" $ZK\{(x,r) : C = g^x h^r$ and first bit of x is $0\}$ "

Application: of ZK proofs about commitments w/r opening



$ZK\{(doc, sig) :$ doc is signed by sig from the DMV and age $\geq 21\}$

Possibilities:



⟹ $\underline{age, r_2}$ Reveal just the portion you need.

OR proofs: $ZKP\{(x) : X_1 = g^x$ OR

"I know ONE of these public keys' secrets"

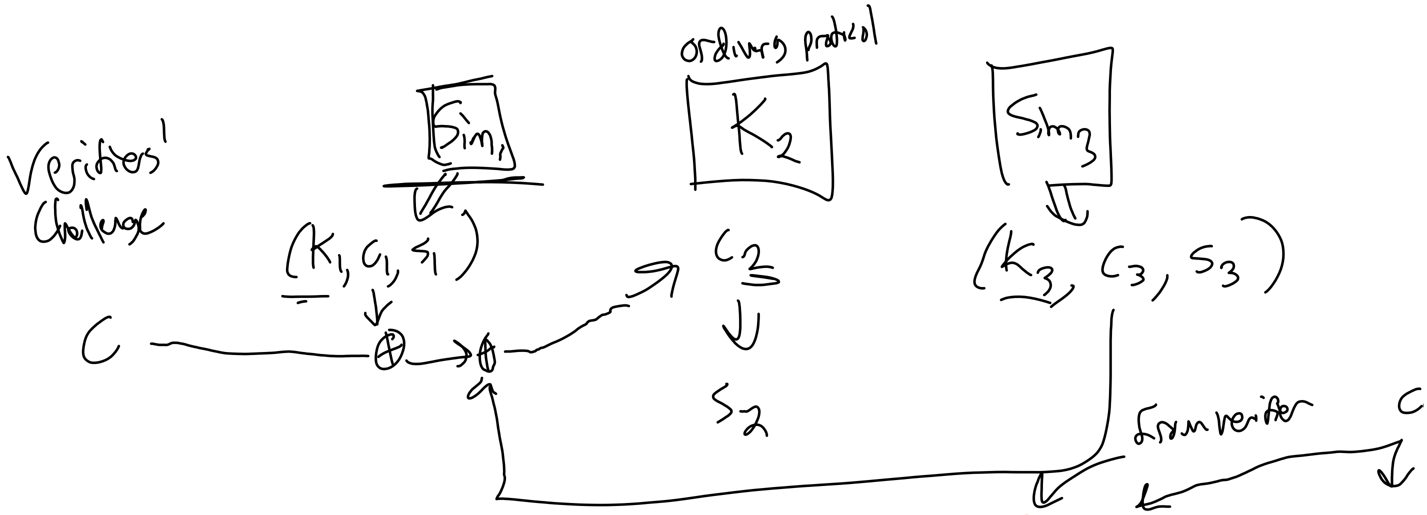
✓ $X_2 = g^x$ OR

$X_3 = g^x$ OR

...

Run N proofs in parallel

Idea: follow the protocol for the one we know, use the simulator for the rest.



Let prover choose c_1, c_2, c_3 s.t. $c = c_1 \oplus c_2 \oplus c_3$

prover has 2 degrees of f[...] choose, but remaining 1[...]

Proof:
- Extraction ⇒ todo
- Simulation ⇒ todo.