Zero knowledge proofs

Tuesday, September 10, 2019 10:42 AM

- Basic definitions for security
- Discrete log assumption.
- Interactive proof

Crypto egg

Sample secret key

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

publickey

$$\rightarrow X := g^x$$

publish this

hard to find x

- $\forall \mathcal{A}, \; \Pr\left[\text{BadEvent}(\mathcal{A})\right] \approx$ negligible

" Computationally feasible adversaries. "

"polynomial time, probabilistic turing machines"

- Turing machines

  "universal" → up to polynomial time equivalence

  "Strong Church-Turing thesis"
  all reasonable computing devices.

- polynomial time

  produces output in $\text{poly}(\underline{\lambda})$ steps

  security parameter

  $$b \leftarrow M(X)$$

  in ordinary complexity: if M is polytime, it means M produces output in $\text{poly}(|X|)$ steps.

  $$b \leftarrow M(1^\lambda, \ldots)$$

  $$\underbrace{111\ldots1}_{\lambda \text{ times}}$$

- probabilistic:

  able to make random coin flips.

  $$b \leftarrow M(1^\lambda, r, \ldots)$$

  polynomial size stream of random bits (usually not included)

  output is a probability distribution

  sample from

- Probability

  D is a discrete probability distribution,

  $$D : \underline{S} \rightarrow [0,1] \in \mathbb{R}$$

  sample space

  $x \leftarrow D$ means sample from D

  $x \xleftarrow{\$} S$ means uniform random sample

  a finite set

Example: $\forall M, \; \Pr\left[\begin{array}{l} b \xleftarrow{\$} \{0,1\} \\ b' \leftarrow \underline{M}(1^\lambda) \end{array} : b' = b\right] = \frac{1}{2}$

p.p.t.

explicit notation $\begin{cases} 0 : & 0.5 \\ 1 : & 0.5 \end{cases}$

' for Pr distribution

- Negligible functions:

  $\text{negl}(\lambda)$ "vanishingly small function"

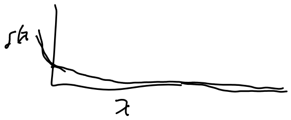
Defn: $f : \mathbb{N} \to \mathbb{R}$ iff

— for any polynomial $p(n)$,

$\llcorner$ $\exists n$ s.t. $\forall n' \geq n$, $f(n') < 1/p(n')$

Examples:

$f(\lambda) = e^{-\lambda}$



(non example):

$f(\lambda) = \frac{1}{x}$

$1/x^3$ no, why not?

Consider $p(x) = 1/x^4$

- Discrete Log Assumption:

Let $\{G_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of groups, and generators $g_\lambda \in G_\lambda$

prime. $|G_\lambda| \geq 2^\lambda$

$$\forall A, \Pr\left[\begin{matrix}\text{draw secret/public} \\ \text{adversary guesses secret}\end{matrix} : \begin{matrix}\text{guess} \\ \text{was right}\end{matrix}\right] \leq \text{negl}(\lambda)$$

$$— \forall A. \Pr\left[\begin{matrix} x \xleftarrow{\$} \mathbb{Z}_{|G_\lambda|} \\ x' \leftarrow A(1^\lambda, g^x)\end{matrix} : x' = x\right] \leq \text{negl}(\lambda)$$

Dlog is thought to hold for:

- schnorr $\overset{\text{sub}}{\text{groups}}$
- some elliptic curves

Interactive Proofs: Ways to specify:

$\textcircled{1}$ — "I know my secret x, such that $g^x = X$"

Informal

Alice x, g^x $\longrightarrow$ Bob X

"Zero Knowledge"

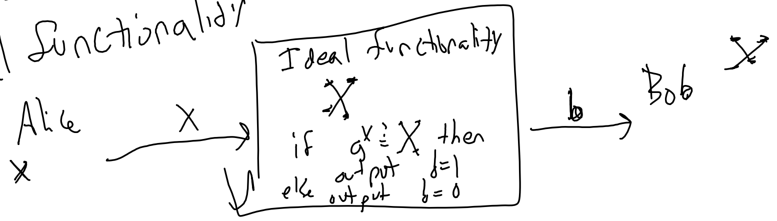
$\longrightarrow$ Security goals:

(Zero Knowledge) Bob should not learn x $\leftarrow$ or any information x

(Zero-knowledge) — Bob shouldn't

(Soundness / Knowledge) — Alice has to actually know x

② Ideal functionality



③ Camenisch-Stadler notation

— $\text{ZKpok}\{ (\underset{\uparrow \text{ witness}}{x}) : \underbrace{g^x = X}_{\text{predicate}} \}$

— Schnorr identification protocol:
$|G| = p$

prover → $P(x)$ verifier → $V(X)$

$k \xleftarrow{\$} \mathbb{Z}_p$
$K := g^k$

"commit" $\xrightarrow{K}$

store $\underline{K}$

$\underline{c} \xleftarrow{\$} \mathbb{Z}_p \setminus \{0\}$

$\underline{s} = \underline{c}x + \underline{k}$

"challenge" $\xleftarrow{c}$

$s := k + c \cdot x$

"response" $\xrightarrow{s}$

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

Properties:

- Correctness (Sanity check)

$$X^c K = g^s$$
$$(g^x)^c K = g^s$$
$$(g^x)^c g^k = g^{cx+k}$$
$$g^{xc+k} = g^{cx+k} \checkmark$$

- View of the Verifier doesn't depend on x at all
  The verifier could have produced this view without interacting w/ the prover at all

→ View is (K, c, s)

→ Vi

We can construct a "simulator" for this view

$$(K, c, s) \leftarrow S(X)$$

$S(X)$:

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

$$c \xleftarrow{\$} \mathbb{Z}_p \setminus \{0\}$$

$$K := (g^s) \cdot (X^c)^{-1} = g^s / X^c$$

This passes the verify check, since

$$K X^c = g^s \quad ✓$$

View "of V" consists of:

→ - any coins flipped in the protocol

→ - any messages received

- any messages sent (redundant)

- Formal statement of Zero-Knowledge / Simulatable property:

$$\exists S, \ \mathrm{View}_V[P(x) \longleftrightarrow V(X)] = S(X)$$

Completing the proof

$$\mathrm{View}_V[P(x) \longleftrightarrow V(X)] = \begin{cases} (K, c, s) \in \mathbb{Z}_p \times \mathbb{Z}_p \setminus \{0\} \times \mathbb{Z}_p : \frac{1}{p \cdot (p-1)} & \text{if } X^c K = g^s \\ 0 & \text{otherwise} \end{cases}$$

(K, c, s)

→ $k \xleftarrow{\$} \mathbb{Z}_p$

→ $K = g^k$

$\{K \in G : \frac{1}{p}$

$c \xleftarrow{\$} \mathbb{Z}_p \setminus \{0\}$

$\{c \in \mathbb{Z}_p : \frac{1}{p-1}$

$\{K, c \in \mathbb{Z}_p \times (\mathbb{Z}_p \setminus \{0\}) : \frac{1}{p(p-1)}$

$$S(X) = \begin{cases} (K, c, s) : \frac{1}{p(p-1)} & \text{if } X^c K = g^s \\ 0 & \text{otherwise} \end{cases}$$

Soundness / Extractability / Knowledge

$\forall \underline{\underline{A}}, X,\quad \Pr[\text{output}_V[A(X) \longleftrightarrow V(X)] = \text{"ok"}] > \text{negl}(\lambda)$

$\exists E_A,$ s.t. $\underbrace{\Pr[x \leftarrow E_A(X) : g^x = X] = 1 - \text{negl}(\lambda)}$

"extractor"

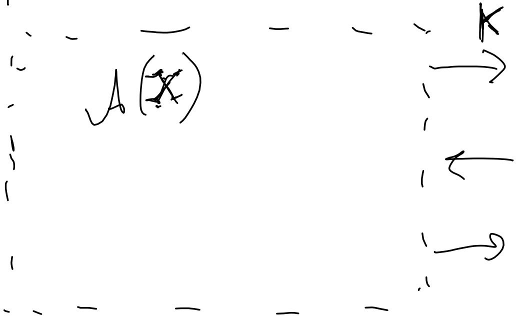
Proof that Schnorr id. protocol is sound:

Supposing we have $A, X,$
such that $\Pr[\text{output}_V[A(X) \longleftrightarrow V(X) = \text{"ok"}]] = P_{\text{accept}}$
and $p > \text{negl}(\lambda)$

Then we can construct E such that

$E(X)$:

Run $A(X$ until it outputs K..



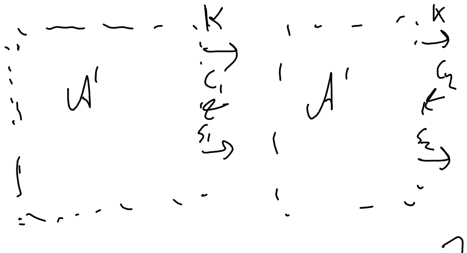
Make a snapshot of the state of A, called A'

Sample $c_1 \xleftarrow{\$} \mathbb{Z}_p \setminus \{0\}$

$c_2 \xleftarrow{\$} \mathbb{Z}_p \setminus \{0\}$

Let $s_1 \leftarrow A'(c_1)$

$s_2 \leftarrow A'(c_2)$



With probability (P_{ACCEPT}^2) ★

★ repeat poly(λ) until true $\quad \frac{\lambda}{(P_{\text{accept}}^2)}$

$c_1 \quad s_1$

$X \; K = g$ Until the (attempts).

$X^{c_2} K = g^{s_2}$ modular division $\mathbb{Z}_p$

prob of failure after all $\frac{\lambda}{p_{accept}^2}$ times is

looking ahead $\rightarrow$ $\boxed{X = (s_1 - s_2) / (c_1 - c_2)}$

$$X^{c_1} K \cdot (X^{c_2} K)^{-1} = g^{s_1 - s_2}$$

$(1 - p_{accept}^2)^{\lambda}$

So success prob is $1 - (1 - p_{acc}^2)^{\lambda}$
$= 1 - \text{negl}(\lambda)$

$$X^{c_1} \cdot (X^{c_2})^{-1}$$

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

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

Extending zkPoK to other languages/predicates

$\rightarrow$ $\text{ZKPoK}\{(a,b) : A = g^a, B = g^b\}$

Can you produce a protocol for it?

1. Run Schnorr id. twice. $\quad K_a, K_b \rightarrow$, $\leftarrow c_a, c_b$, $s_a, s_b \rightarrow$

2. Can we reuse c?

$P(a,b)$ $\qquad\qquad V(A,B)$

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

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

$K_a = g^{k_a} \quad K_b = g^{k_b}$

$\xrightarrow{K_1, K_2}$

$c \xleftarrow{\$} \mathbb{Z}_p \setminus \{0\}$

$\xleftarrow{c}$

$s_a = k_a + c a$

$s_b = k_b + c b$ $\quad \xrightarrow{s_a, \; s_b}$ $\quad$ Check $g^{s_a} \overset{?}{=} A^c K_a$

and $g^{s_b} \overset{?}{=} B^c K_b$

- Correctness/Sanity check

$\forall a,b \; \Pr[\text{Out}_V[P(a,b) \longleftrightarrow V(g^a, g^b)] = \text{"ok"}] = 1$

– Simulation

$$\exists S, \forall a,b, \quad \text{View}[P(a,b) \leftrightarrow V(g^a, g^b)] = S(g^a, g^b)$$

$$(K_a, K_b, c, s_a, s_b)$$

$$\begin{cases} \frac{1}{p} \cdot p \cdot (p-1) & \text{is satisfies check} \\ 0 & \text{otherwise} \end{cases}$$

– Extraction:

$$\forall \mathcal{A},\ A,B,\ \Pr[\text{out}[\mathcal{A}(A,B) \longleftrightarrow V(A,B)] = \text{"ok"}] > \text{negl},$$

then $\exists E_{\mathcal{A}}, \Pr\left[(a,b) \leftarrow E(A,B) : g^a = A, \text{and } g^b = B\right] = 1\ \text{negl}$