Slow-Motion Zero-Knowledge

Identifying With Colliding Commitments

Houda Ferradi   Rémi Géraud   David Naccache

École normale supérieure de Paris

Outline

1. Zero-Knowledge Identification
2. Building Blocks
3. Slow-Motion Zero-Knowledge

Zero-Knowledge Identification

Zero-Knowledge Identification Protocols

Intuitive Goals:

1. Prove your identity
2. Do not reveal anything else ("zero-knowledge")

We use the mathematical framework of Σ-protocols:

$$
\begin{array}{ccc}
 & \xrightarrow{\quad x \quad} & \\
\mathcal{P} & \xleftarrow{\quad c \quad} & \mathcal{V} \\
 & \xrightarrow{\quad y \quad} &
\end{array}
$$

- The prover sends a commitment x to the verifier
- The verifier replies with a challenge c
- The prover gives a response y

Absence of information leakage: Existence of a simulator $\mathcal{S}$.

The output of $\mathcal{S}$ is indistinguishable from the conversation between $\mathcal{P}$ and $\mathcal{V}$.

Imagine that $\mathcal{P}$ is a low-end device with limited resources and power.

$$\begin{array}{ccc} & \xrightarrow{\quad x \quad} & \\ \mathcal{P} & \xleftarrow{\quad c \quad} & \mathcal{V} \\ & \xrightarrow{\quad y \quad} & \end{array}$$

To ensure security, commitments must be collision-resistant [GS94].

$\Rightarrow$ Large commitments, increased communication, resource usage

Our contribution: Breaking the collision-resistance barrier.

Building Blocks

Commitment pre-processing

In DLP-based ZKPs, commitment and challenge are unrelated.

We can thus pre-compute commitments for DLP-based ZKP,

We may delegate this pre-computation to a (powerful) trusted authority.

$\Rightarrow$ Lightning-fast response times!

Mathematically, we can consider an algorithm

$$\{r_i, x_i\} \leftarrow \mathsf{PreComp}(1^k, \mathsf{pp})$$

that generates a list of commitments/responses for $\mathcal{P}$.

Example (GPS pre-computation)

Choose some common seed J and a hash function H, then:

$$\begin{aligned}&\text{for } i = 1 \text{ to } k \text{ do}\\&\qquad r_i \leftarrow H(J, i, s)\\&\qquad x_i \leftarrow g^{r_i} \bmod n\end{aligned}$$

Time-Lock Puzzles

Intuition:

Provably hard problems with tunable hardness.

Example (Rivest et al.)

Compute $f(x) = 2^{2^x} \bmod n$ for composite n.

A good time-lock function should slow-down even computationally powerful adversaries.

Definition

PPT algorithms $\mathcal{T}_G(1^k, t)$ (problem generator) and $\mathcal{T}_V(1^k, a, v)$ (solution verifier) such that:

1. $\mathcal{T}_G$ generates puzzles of hardness t, and B cannot efficently solve any puzzle of hardness $t \geq k^m$ for some constant m depending on B.
2. For any polynomial hardness value, there exists an algorithm that can solve any puzzle of that hardness.

Mathematically,

1. $\forall$ PPT algorithm $B(1^k, q, h)$, $\forall e \in \mathbb{N}$, $\exists m \in \mathbb{N}$ s.t.

$$\sup_{t \geq k^m, |h| \leq k^e} \Pr\left[(q, a) \leftarrow \mathcal{T}_G(1^k, t) \text{ s.t. } \mathcal{T}_V(1^k, a, B(1^k, q, h)) = 1\right]$$

   is negl(k).

2. $\exists m \in \mathcal{N}$ s.t. $\forall d \in \mathcal{N}$, $\exists$ PPT algorithm $C(1^k, t)$ s.t.

$$\min_{t \leq k^d} \Pr\left[(q, a) \leftarrow \mathcal{T}_G(1^k, t), v \leftarrow C(1^k, q) \text{ s.t. } \mathcal{T}_V(1^k, a, v) = 1 \text{ and } |v| \leq k^m\right]$$

   is overwhelming in k.

Slow-Motion Zero-Knowledge

We use the following function

$$f_{\tau,\ell}(x) = \left(\mu(x)^{2^\tau} \bmod \overline{n}\right) \bmod 2^\ell.$$

to build a family of time-lock problems.

- τ controls puzzle hardness
- ℓ is a parameter controlling output size
- $\overline{n}$ is an RSA modulus
- μ is an RSA padding function

There are three ways to find $f_{\tau,\ell}(x)$:

1. Perform τ square operations $\mod 2^\ell$
2. Find the factorization of $\overline{n}$
3. Exhaust all 2^ℓ possible values

Choosing the parameters sizes we can make strategy 2. and 3. intractable

We use the function $f_{\tau,\ell}$ to compress commitments::

$$\text{shorten}_{\tau,\ell} : (r_i, x_i) \mapsto (r_i, f_{\tau,\ell}\,(x_i))$$

We can thus replace PreComp by

$$\text{ShortPreComp}_{\tau,\ell} = \text{shorten}_{\ell,\tau} \circ \text{PreComp}$$

Accordingly, the verifier checks $\{f_{\tau,\ell}(x_i), c_i, r_i\}$ instead of $\{x_i, c_i, r_i\}$.

Step 2: Commitment Shortening

Note 1: No change to $\mathcal{P}$'s response phase

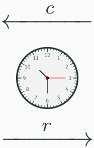
Note 2: Pre-computation fast when factors of $\overline{n}$ are known

Note 3: Commitments are ℓ bits long and colliding...

...But the adversary is greatly slowed-down by the time-lock puzzle!

Step 3: Time Measurement

We measure time between challenge and response :



- Legitimate provers can reply correctly very quickly
- Adversaries must face the time-lock for every try

$\Rightarrow$ Time-constrained soundness despite colliding commitments!

Many applications especially for low-end devices:

- Ultra-fast identification for constrained devices (IoT, CPS, ...)
- Efficient security mechanisms over low-rate networks (LoRa, SigFox...)
- New protocols and approaches

Conclusion

Contributions and Further Research

This paper:

- Introduced Slow Motion Zero Knowledge (SM-ZK) protocols
- Breaking the collision-resistance theoretical barrier

Opens new research directions:

- Fading signatures
- Multi-channel authentication
- Etc.

Thank You for your Attention!

Gǎnxiè nín de guānzhù!