Zero Knowledge Interactive Proofs

Now doing much more than communicating securely:
- Complex interactions: simultaneity, fairness
- Joined by others: auctions, bidding, elections, e-commerce

Proofs

Verifier

Proofs

Easy (efficient)

Verifier

Accept/reject
Example: $n$ is a product of 2 large primes

If $n=pq$, accept
Else reject

After interaction, Bob knows:
1) $n$ is product of 2 primes
2) Also the factors of $n$

Example: $G_1$ is isomorphic to $G_2$

If mapping is good, accept
Else reject

After interaction, Bob knows:
1) $G_1$ is isomorphic to $G_2$
2) Also the correspondence

New Crucial Ingredients

- Interaction
- Randomness

Is there any other way?
I will not give you a correspondence, but I will prove to you that I could if I felt like it.

**HOW?**

I will produce a random graph $H$ for which
1: I can give you a correspondence $\gamma$ from $G_1$ to $H$
2: I can give you a correspondence $\eta$ from $H$ to $G_2$

Hence, I know a correspondance from $G_1$ to $G_2$ directly

Then, you randomly decide if I should demonstrate my ability to do #1 or #2.

**POINT IS:** If I can do both, there exists mapping from $G_1$ to $G_2$ $\phi = \gamma \circ \eta$

---

**REPEAT K INDEPENDENT TIMES.**

An Interactive Proof

Permute the Vertices of $G_1$ at random, call result $H$

- **Graph $H$**
- **Toss coin $b$**

If $b=0$: mapping from $G_1$ to $H$
If $b=1$: mapping from $H$ to $G_2$

**Claims:**
1. Statement true can answer correctly for $b=0$ and $1$
2. Statement false $\text{prob}_{b \in \{0,1\}}(\text{catch a mistake}) = \frac{1}{2^k}$

**Interactive Proofs**

*GMR85*

- **Prover P**
- **Verifier V**

Statement: $T$

Accepts / Rejects

(P,V) is an *interactive proof system* if

**Completeness:** if $T$ is true, then $V$ will accept

**Soundness:** if $T$ is false, then $V$ will reject with overwhelming probability, no matter what prover strategy is
Zero Knowledge Interactive Proofs

Prover P

\[ q_1 \]
\[ a_1 \]
\[ q_2 \]

Verifier V

Accepts / Rejects T

After interactive proof, V "knows":

- T is true
- A view of interaction

P gives Zero-Knowledge to V: when T is true, V can simulate views of what he sees on his own with the same probability distribution.

Prover Gives Zero Knowledge

- "Simulated views" and "real interactions" are computationally indistinguishable

Any Efficient Algorithm

Technical Definition

Define probability space (over coins of V,P)

\[ \text{View}_{V,P}(x) = \{(q_1,a_1,q_2,a_2,\ldots,\text{coins})\} \]

(P,V) is perfect zero-knowledge for L if:

For all \( x \in L \), for all \( V^* \), there exists an \( S \) s.t.

\[ \text{View}_{V^*\rightarrow P}(x) = S(x) \]

(P,V) is statistical zero-knowledge for L if:

For all \( x \in L \), for all \( V^* \), there exists an \( S \) s.t.

\[ \text{View}_{V^*\rightarrow P}(x) \text{ statistically close to } S(x) \]

Can relax to "indistinguishable" in probabilistic polynomial time

Technical Definition

Define random variable

\[ \text{View}_{V\rightarrow P}(x) = \{q_1,a_1,q_2,a_2,\ldots,\text{coins}\} \]

(P,V) is zero-knowledge for L if:

For all \( x \in L \), for all \( V^* \), there exists an \( S \) s.t.

\[ \text{View}_{V^*\rightarrow P}(x) \sim S(x) \text{ expected polynomial time} \]

Can relax to "indistinguishable" in probabilistic polynomial time
Zero Knowledge Interactive Proofs

- Equivalent to the following definition:

**Zero-Knowledge:** when statement is true for all Verifier strategies $V^*$,
Whatever $V^*$ can compute after interaction = Whatever $V^*$ could compute before interaction

\[
\text{Prover } P \quad \begin{array}{c}
q_1 \\
a_1 \\
q_2 \\
\hline
\end{array} \quad \text{Verifier } V
\]

Accepts / Rejects

[GMR85]

Zero Knowledge Arguments

- Completeness:
- Soundness: with respect to PPT Alice
- Zero-Knowledge: when statement true, for all verifier strategy $V^*$
  Whatever $V^*$ can compute after interaction = Whatever $V^*$ could have computed before interaction

\[
\text{Prover } P \quad \begin{array}{c}
q_1 \\
a_1 \\
q_2 \\
\hline
\end{array} \quad \text{Verifier } V
\]

Input: $x$

Actually, Alice proves more: that she actually "knows" the correspondence

Let $R$ be polynomial time testable. Let $(x,s) \in R$

**$P$ Knows Secret $s$:**

Informally, $P$ on public input $x$ knows $s$ if:

- An extractor algorithm $E$ s.t. $E^p(x)$ outputs $s$ quickly.
- $E^p(x)$: $E$ can run $P$ polynomial number of times asking $P$ different questions when $P$ uses same random tape

**ZKPOK:** zero knowledge proof of knowledge

Classical Identification - Password

Public password file

<table>
<thead>
<tr>
<th>Name</th>
<th>$f(\text{pswd})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>shafi</td>
<td>$P_{\text{Shafi}}$</td>
</tr>
<tr>
<td>Alice</td>
<td>$P_{\text{Alice}} = f(\text{summer})$</td>
</tr>
<tr>
<td>Bob</td>
<td>$P_{\text{Bob}}$</td>
</tr>
</tbody>
</table>

Computer 1 checks if $f(\text{pswd}) = P_{\text{Alice}}$

2 erases password from screen.
Problems with Classical Method

For Settings:
- Alice = Smart Card.
- Over the Net

Passwords are no good

Zero Knowledge: ID over the net

More generally,

To identify itself Prover proves in zero-knowledge it knows a proof of the hard theorem.

More examples

- Zero Knowledge Proof that “Alice knows the factors of n”
- Zero Knowledge Proof that “Alice knows the discrete log of $g^x \mod p$”
- Zero Knowledge proof that “Alice knows that square root of $y \mod n$”
Recall, being able to quickly find a root of a random number is equivalent to being able to factor $n$.

- Let $A$ be an algorithm which can compute one root of a random input $x$.
- Pick $r$ at random. Let $x = r^2$, $r_1 = A(x)$.
- With 50% chance $(r, r_1)$ are primitive and you can factor $n$. Repeat until $n$ is factored.

Certainly not ZK

Prove to me that you know a sqrt of $y$  

A root of $y$

I refuse to give you a root of $y$, but I will prove to you that I could if I felt like it.

HOW?

Catch cheating Alice with probability $1/2$  

If $b = 0$: root of $z$  
If $b = 1$: root of $yz$
1: I can give a root of $z$
OR
2: I can give a root of $zy$
Hence, I know a root of $y$
you randomly decide if I should demonstrate my ability to do #1 or #2.

$$\text{ROOT of } y = (\text{ROOT of } Zy) / (\text{ROOT of } Z)$$

**REPEAT** $K$
**INDEPENDENT**
**TIMES.**

If $b=0$: root of $Y$
If $b=1$: root of $zY$

Catch a crook with prob $1 - 1/2^k$

How can we be sure this is zero-knowledge?

Bob can perfectly simulate the probability distribution on conversations with Alice.

**What Made/Makes ZK possible?**

**Randomness**
- The statement to be proven has many possible proofs of which the prover chooses one at random.
- Each such proof is made up of exactly 2 parts: seeing either part on its own gives the verifier no knowledge; seeing both parts imply 100% correctness.
- Verifier chooses at random which of the two parts of the proof he wants the prover to give him. The ability of the prover to provide either part, convinces the verifier

Current research indicates interaction may be not as essential, and may be able to find other ways to force the prover to answer an unpredictable question.
More Examples

For a given $g, p, g^x \mod p$,
Prove in zero knowledge that you “know” $x$.

Examples of NP-assertions

A graph is 3-colorable, a number $n$ is the product of 2 primes,
a number $x$ is a square mod $n$,
a graph has a traveling salesman of cost $C$,
Given encrypted inputs $E(x)$ and program $PROG$, $y = PROG(x)$.

More generally, any statement which Has Efficiently Checkable Proofs

Statement: is 3-colorable

Prover Verifier

Hard Working Easy

Short proof: a 3-coloring of the graph

Short: polynomial size (in the statement)

This example is canonical, represents a class of NP assertions for which there exist short proofs.

Zero Knowledge Fundamental Theorem

Theorem[GMW86]: If one-way permutations exist, then all NP assertions have zero knowledge interactive proofs (NP assertion is any assertion which has a classical “easily checkable proof”).

Building Block:

One Way Permutations $\rightarrow$ Commitments

Think of commitment to $b$ as $c \in E(b)$ (E is PE algo.)
How can you prove something so general?

**Idea:** Show a zero knowledge interactive proof for complete statement for NP.

**Completeness** [Cook-Levin-Karp]: All statements $T$ can be translated into a graph $G$

- $T$ is true $\implies G$ is 3 colorable
- $T$ is false $\implies G$ is not 3 colorable

Show a Zero-knowledge Proof for 3-coloring

Let's talk about commitments

- Privacy
- Binding

How Does ZK Work?

**EX: ZKIP for Graph 3 Coloring**

On common input graph $G$ and private prover input coloring $\pi: G \rightarrow \{0,1,2\}$

- **P:** randomly permute colors, and color all vertices of the graph according to permuted colors.
  - for all $v$, send $E(\text{color of vertex } v)$ to $V$
- **V:** send $P$ a random challenge edge $(v_i,v_j)$
- **P:** reveal to $V$ colors of $v_i$ and $v_j$, by decommiting $E(\text{color of } v_i)$ and $E(\text{color of } v_j)$.
- **If** for all $k$ iterations,
  - color revealed for $v_i$ ≠ color revealed for $v_j$ then $V$ accepts

Ability of $P$ to give colors of all edges $\implies$ Soundness
Fact that $P$ gives color of only one edge $\implies$ Zero Knowledge

General Cryptographic Importance

- Proving correctness of protocols is complex even if users are honest; If users deviate from protocol in arbitrary ways, almost impossible in a case-by-case manner, need tools and framework to prove correctness.
  - Proof of proper behavior is fundamental tool for design of secure protocols
- **Zero Knowledge Proofs** enable automatic conversion of any protocol proven secure against honest-but-curious adversaries to protocol secure against deviating adversaries
Many, Many Applications:
ZK allows...

• Can prove properties about m without ever revealing m, only \( E(m, S) \)
• Can prove relationships between m1 and m2 never revealing either one, only m1, m2.
• Generally: Enables automatic conversion of any protocol proven secure against honest-but-curious adversaries to protocol secure against deviating adversaries

Basic Questions about Zero Knowledge(I)

• Q1: Sequential Compositions
  - Augment the definition to add auxiliary inputs
• Q2: Parallel Compositions?
  - Not always (artificial counter example)
  - Known natural examples cannot be proved using black box simulation
  - A: Weaken definition of ZK to Witness Hiding [FeSh87]

Basic Questions about Zero Knowledge(II)

• Q3: Concurrent Executions? [Fe,DNS]
  - Commitment schemes + timing assumptions constant rounds concurrent ZK arguments for NP [DNS,DS]
• Q4: Resettable Question? [CGMA]
  - Is ZK preserved when verifier can execute the protocol repeatedly resetting the prover to use the same randomness? (e.g. Disconnect the power supply of prover implemented by a Smart Card.)
  - Observation: If prover can be reset all Identification Schemes based on POK-paradigm are insecure.

EX: Reset attack on QR protocol

Let QR=\{(x,n) = x=z^2 \mod n\}
Alice’s public Key: (x,n)

Bob simply runs the Extractor\text{Alice} algorithm

\[
\begin{align*}
(x, n), z & \quad R^2 \mod n \\
B = 0 & \quad \text{Bob}
\end{align*}
\]

\[
\begin{align*}
R \mod n & \\
\text{RESET Alice} & \\
R^2 \mod n & \\
B = 1 & \\
Rz \mod n & \quad \text{Set } z=R/Rz
\end{align*}
\]
Okamoto-Identification Used in Practice

Let $p=2q+1$

$\begin{align*}
&g_1, g_2, X = g_1^{s_1}g_2^{s_2} \mod p \\
&R = g_1^{r_1}g_2^{r_2} \mod p \\
&0 < c < q
\end{align*}$

$Y_1 = r_1 + cs_1$

$Y_2 = r_2 + cs_2$

Check that $g_1^{Y_1}g_2^{Y_2} = RX^c \mod p$,

if so accepts, Else rejects

Claim: prob of cheating successfully is $O(1/p)$

Claim: Fails under rest attack

Resettablity: a real threat?

- Resetting or restoring state of a device is easy for captured smart card.
- Force a crash on device implementing prover so it will resume computation in a previous “computational spot”.
- Smart Card Memory limitations: cannot rewrite too many times. I.e randomness may be stuck

A new notion: Resettable ZK (rZK)

- Completeness, soundness as usual.
- $(P,V)$ is rZK for language $L$ if:
  - Consider the following experiment, w.r.t some $V^*$:
    - Fix the input $x$ (and witness $w$ for $P$).
    - Choose and fix the random input $r$ of $P$.
    - $V^*$ interacts with as many “copies” of $P$ as it wants, in an arbitrary way. (All copies have the same $x,w,r$ and are unaware of each other.)
  - For any $V^*$ there is a simulator $S$ s.t. for any $x$ in $L$ and any witness $w$,

  Super-View $V^* \leftrightarrow P(x) \cong S(x)$.

  \{Coins, messages in all interactions $V^*$ and $P(x,w,r)$\}

Resettable ZK (rZK)

More generally, $V^*$ has access to many “incarnations” $(x,w,r_k)$ of $P$. 

Resettability: a real threat?

- Resetting or restoring state of a device is easy for captured smart card.
- Force a crash on device implementing prover so it will resume computation in a previous “computational spot”.
- Smart Card Memory limitations: cannot rewrite too many times. I.e randomness may be stuck
**Resettable ID (rID)**

P-prover
V*-resetting verifier
R-coins of P

\[
\begin{align*}
P(x, s, r) & \quad P(x, s, r) \quad \ldots \quad P(x, s, r) \\
V^*(x) & \\
\end{align*}
\]

Note: The notion generalizes to the case where 
\(V^*\) has access to many “incarnations” \((x, s, r_i)\) of \(P\).

Salvage Old ID POK Based Protocols

**Claim:** A reset attack on transformed \(\rightarrow\) concurrent attack on original

**Resettable Okamoto-Identification**

\[
\begin{align*}
P & \quad q, g_1, g_2, X = g_1^s g_2^s \quad V \quad \text{concurrent} \\
R & = g_1^r g_2^r \\
Y_1 & = r_1 + cs_1 \\
Y_2 & = r_2 + cs_2 \\
P & \quad q, g_1, g_2, X = g_1^s g_2^s, h \quad V_{\text{reset}} \\
(g_1 h^{\text{SID}}) g_2^d, \text{SID} & \\
R & = g_1^r g_2^r \\
c, d & \\
Y_1 & = r_1 + cs_1 \\
Y_2 & = r_2 + cs_2 \\
\end{align*}
\]

Choose \(h = g_1^{-\text{SID}} g_2^w\) for random \(w\) thus can open commitment at will

Where \(\text{Comm}\) is a perfectly private 
computationally binding 
trapdoor commitment

P replaces 
coins with 
\(\text{Psrf}(\text{prefix})\)