Lecture 14

Zero Knowledge I
Now doing much more than communicating securely:

- **Complex interactions**: games, computations, proofs
- **Complex Adversaries**: Alice or Bob, adaptively chosen
- **Complex Properties**: correctness, simultaneity, fairness
- **Joined by others**: auctions, bidding, elections, e-commerce
Classical Proofs

\[ \sqrt{a^2 + b^2} \]

Prime-Number Thm

<table>
<thead>
<tr>
<th>Statements</th>
<th>Reasons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Assume: ( \angle B \equiv \angle CED )</td>
<td>1. Assumption</td>
</tr>
<tr>
<td>2. ( \angle A \equiv \angle D )</td>
<td>2. Given</td>
</tr>
<tr>
<td>3. ( \angle ABC ) and ( \angle DCE ) are right ( \angle s )</td>
<td>3. Defs. of ( \angle s )</td>
</tr>
<tr>
<td>4. ( \angle ABC \equiv \angle DCE )</td>
<td>4. RAT</td>
</tr>
<tr>
<td>5. ( \angle B \equiv \angle C )</td>
<td>5. Given</td>
</tr>
<tr>
<td>6. ( \angle ABC \equiv \angle DCE )</td>
<td>6. ASA (1, 5, 4)</td>
</tr>
<tr>
<td>7. ( AB \equiv CD )</td>
<td>7. CPCTC</td>
</tr>
<tr>
<td>8. ( AB ) is not ( \parallel ) to ( CD )</td>
<td>8. Given</td>
</tr>
</tbody>
</table>

But statement 7 contradicts statement 8. Consequently, the assumption must be false.
Proofs

Prover

Claim

Verifier

proof

accept/reject
Efficiently Verifiable Proofs (NP)

Prover
Works Hard

Claim

Verifier
Polynomial Time

proof

accept/reject
Efficiently Verifiable Proofs (NP)

NP = decision problems D for which there is a short and polynomial time verifiable proofs (witness) of $x \in D$
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: \( y \) is a quadratic residue mod \( N \) (i.e. \( y = x^2 \mod N \))

After interaction, Bob knows:

1) \( y \) is a quadratic residue mod \( N \)
2) Square root of \( y \)

If \( y = x^2 \mod N \), Accept
Else reject
Example: $G_0$ is isomorphic to $G_1$

Isomorphism $\phi$

If isomorphism is good, accept
Else reject
After interaction, Bob knows:

1) $G_0$ is isomorphic to $G_1$

2) Also the isomorphism

Is there any other way?
Zero Knowledge Proofs

Main Idea:
Prove that
I could prove it
If I felt like it
Two New Ingredients

Interactive and Probabilistic Proofs

Non-trivial interaction: rather than “reading” proof, verifier engages in an non-trivial interaction with the prover.

Randomness: verifier is randomized (tosses coins as a primitive operation), and can err with some small probability
I will not give you an isomorphism, but I will prove to you that I could provide one.

HOW?
I will produce a random graph $H$ for which

1: I can give you an isomorphism $\gamma_0$ from $G_0$ to $H$

OR

2: I can give you an isomorphism $\gamma_1$ from $G_1$ to $H$

Hence, there is an isomorphism $\sigma$ from $G_0$ to $G_1$ directly

YOU randomly choose if I should demonstrate my ability to do #1 or #2.

**Proof:**

$H = \gamma_0(G_0)$,

$H = \gamma_1(G_1)$,

Thus

$G_1 = \gamma_1^{-1}(\gamma_0(G_0))$

Set $\sigma = \gamma_1^{-1}\gamma_0$

**POINT IS:** If I can do both, there exists an isomorphism from $G_0$ to $G_1$
An Interactive Proof

Choose random $\gamma_0$ permutation of vertices of $G_0$. Set $H = \gamma_0(G_0)$

REPEAT K INDEPENDENT TIMES.

If $b=0$: send $\gamma_0$
If $b=1$: send $\gamma_0 \sigma^{-1}$ (where $\sigma(G_0)=G_1$)

Claims:
(1) Statement true can answer correctly for $b=0$ and $1$
(2) Statement false prob$_b$(catch a mistake) = $1 - 1/2^k$
(3) Zero Knowledge (to be defined)
Interactive Proofs [GMR85]

Statement: $T$

Prover $P$

Verifier $V$

Probabilistic Polynomial time algorithm

Accepts / Rejects

(P, V) is an interactive proof system for $T$ if

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

Soundness: if $T$ is false, then regardless of prover $P$’s strategy, $V$ will reject with overwhelming probability
Interactive Proofs for Language Membership [GMR85]

Statement: $x \in L$

Verifier V

Probabilistic Polynomial time algorithm

Accepts / Rejects

$(P, V)$ is an interactive proof system for $L$ if

Completeness: if $x \in L$, then $\text{Prob}[(P, V)[x] = \text{accept}] = 1$

Soundness: if $x \notin L$, then $\forall P^*$

$\text{Prob}[(P^*, V)[x] = \text{accept}] = \text{neg} (|x|)$
Remarks: Interactive Proofs

• P and V are a pair of interactive Algorithms, each having private inputs and private coins as well as a common public input.

• V additionally must run in polynomial time

• \((P,V)\) satisfy completeness \(c(x)\) & soundness \(s(x)\) if 
  \[x \in L, \text{ Prob}((P,V)[x]=\text{accepts}) > c(x)\]
  \[x \notin L, \forall P^*, \text{ Prob}[(P^*,V)[x]=\text{accepts}] < s(x)\]

• Suffice to require: \(c(x)=2/3\) and \(s(x)=1/3\)
Class IP

Prover P

Verifier V

Statement: $x \in L$

Accepts / Rejects

Probabilistic Polynomial time algorithm

$IP = \{L \text{ s.t. there exists (P,V) interactive proof system for } L \text{ with completeness } c(x) = 2/3 \text{ and soundness } s(x) = 1/3\}$

Is IP greater than NP?
Zero Knowledge Interactive Proofs

After interactive proof, V “knows”:

- T is true (or \( x \in L \))
- A \textit{view} of interaction (=transcript + coins V tossed)

\textbf{P} gives Zero-Knowledge to \textit{V}: when \( T \) is true, the \textit{view} gives \textit{V} nothing he couldn’t have obtained on his own without interacting
How Do we Capture Getting “Nothing Extra” (when T is true)

If: the verifier’s view can be efficiently simulated so that `simulated views’ and `real views’ are indistinguishable by an observer
Perfect Zero Knowledge
(when T is true)

If: the verifier’s view can be efficiently simulated so that `Simulated views’ = `real views’

The observer
Any Algorithm
Formal Definition: Perfect Zero-Knowledge

For a given P and V on input x, define probability space $\text{View}_{(P, V)}(x) = \{(q_1, a_1, q_2, a_2, \ldots, \text{coins of } V)\}$ (over coins of V and P)

$(P, V)$ is **honest** verifier perfect zero-knowledge for L if:

$\exists \text{SIM} \text{ a polynomial time randomized algorithm s.t. } \forall x \in L, \text{ View}_{(P, V)}(x) = \text{SIM}(x)$

Will allow SIM

Expected polynomial time
Choose random $\gamma_0$ permutation of vertices of $G_0$. Set $H=\gamma_0(G_0)$.

If $b=0$: send $\gamma_0$
If $b=1$: send $\gamma_0 \sigma^{-1}$ (where $\sigma(G_0)=G_1$)

View of Bob =
{(H, b, random isomorphism from $G_b$ to H)}
Zero Knowledge

**SIMULATOR M:**
- toss coin to
- If coin=head: choose random $\gamma_0$
  set $H = \gamma_0 \cdot G_0$
- If coin=tail choose random $\gamma_1$
  set $H = \gamma_1 \cdot G_1$

View of Bob=
{($H$, coin, random isomorphism of $G_b$ to $H$)}
What if $V$ is not honest: Perfect Zero-Knowledge (Final def)

For a given $P$ and $V$ on input $x$, define probability space $\text{View}_{(P,V)}(x) = \{(q_1,a_1,q_2,a_2,\ldots,\text{coins})\}$ (over coins of $V$ and $P$)

$(P,V)$ is **honest verifier perfect zero-knowledge** for $L$ if:

$\exists \text{SIM} \, \text{an expected polynomial time randomized algorithm s.t. } \forall x \in L, \text{View}_{(P,V)}(x) = \text{SIM}(x)$

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

$\forall \text{PPT } V^* \exists \text{SIM} \, \text{an expected polynomial time randomized algorithm s.t. } \forall x \in L, \text{View}_{(P,V^*)}(x) = \text{SIM}(x)$
Prover Gives Perfect Zero Knowledge

- If: we can efficiently simulate the view of any verifier s.t. `Simulated views’ = `real verifier” for any poly time verifier.
Zero Knowledge Proof that $G_1$ isomorphic to $G_2$

**SIMULATOR SIM:**
1. toss coin

2. If coin = head:
   - choose random $\gamma_0$
   - set $H = \gamma_0(G_0)$
If coin = tail
   - choose random $\gamma_1$
   - set $H = \gamma_1(G_{21})$

3. Feed $H$ to $V^*$ =

4. If $V^*$ outputs $\text{coin} = \text{coin}$
   - output $(H, \text{coin}, \gamma_{\text{coin}})$
Else abort and try again

**Claim:**

\[
\text{prob[coin = coin]} = \frac{1}{2},
\]

Expected [number of repetitions of SIM] = 2.
For k repetitions, SIM expected trials = $2k$
Claim: \( y = x^2 \mod N \) is solvable

Repeat 100 times

Choose \(1 < r < n\) at random

Flip a \(b\) to choose an equation

\[ z = [r^2 \mod n] \]
\[ zy = [(rx)^2 \mod n] \]

- If I gave you solutions to both, that is \(r\) and \(rx\), you would be convinced that the claim is true but also know \(x\)
- Instead, I will give you a solution to only one equation, either \(r\) or \(rx\) but you can choose which!

Gives a solution to the equation requested

Accepts claim only if gets correct solution

\[ 1 - \left( \frac{1}{2} \right)^{100} \]
Zero Knowledge Proof that $Y = x^2 \mod N$

1. Toss coin

2. If coin = head:
   - Choose random $r$
   - Set $z = r^2 \mod n$

   If coin = tail
   - Choose random $r$
   - Set $z = (ry^{-1})^2 \mod n$

3. Feed $z$ to $V^* = \_\_\_$

4. If $V^*(z)$ outputs coin $\neq$ coin, abort and goto 1
   - Else for coin = head, output $(H, \text{coin}, r)$
   - For coin = tail, output $(H, \text{coin}, r)$
Zero Knowledge Proof that $Y = x^2 \mod N$

SIMULATOR SIM:
1. toss coin
2. If coin=head:
   choose random r
   set $z = r^2 \mod n$
3. Feed $z$ to $V^*$
4. If $V^*(z)$ outputs coin ≠ coin abort and goto 1
   else for coin=head
   output(H, coin, r)
   for coin=tail,
   output(H, coin, r)

Claim:
prob[coin=coin] = ½,
Expected [number of repetitions of M] = 2.
For k repetitions, M expected trials = 2k
SIM: Expected Polynomial Time

• Analysis can be confusing
• Instead can change def to allow
  – SIM(x) to output ⊥ with probability at most 1/2 and require
  – View (x) = SIM(x) to be conditioned on the event that M(x) does not output ⊥
  – 1/2 can be relaxed to neg(x)
What Made it 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
Recall, being able to quickly find a root of 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 $r$ and $r_1$ are different and you can factor n. Repeat until n is factored.
Actually, Alice seems to have proved more: that she actually “knows” the isomorphism (square root)

Let $V$ be polynomial time relation. Let $(x,w) \in V$
$V$ defines Language $L_V = \{x|\exists w \text{ s.t. } V(x, w) = 1\}$.

We say that $(P,V)$ is a **proof of knowledge** for $L_V$
[or that $P$ on $x$ knows $w$] if:
$\exists$ an extractor algorithm $E$ s.t. for all $x$
$E^P(x)$ outputs $w \text{ in expected polynomial time}$

**ZKPOK:** zero knowledge proof of knowledge

This is called the **rewinding technique**
ZKPOK that Prover knows an isomorphism from $G_1$ to $G_2$

Extractor Algorithm:

1) On input $H$
   set coin=head
   Store $\gamma_0$

2) **Rewind** and 2$^{nd}$ time
   set coin=tail
   Store $\gamma_1$

3) Output $\gamma_1^{-1}(\gamma_0)$
Let $V$ be polynomial time relation. Let $(x,w) \in V$ $V$ defines Language $L_V = \{x|\exists x \text{ s.t. } R(x, w) = 1\}$.

We say that $(P, V)$ is a **proof of knowledge** for $L_R$ [or that $P$ on $x$ knows $w$] if:

$\exists$ an extractor algorithm $E$ s.t. for all $x$ and for all $P'$,

If $\text{Prob}[(P', V)[x] = \text{accepts}] = \alpha$, Then $E^P (x)$ outputs $w$ in *expected polynomial time* $(|x|, 1/\alpha)$.
Why did we disturb the classical notion of proof?

- Preventing Identity Theft
- Proving Properties of secrets
- Can verify statements not verifiable efficiently with classical NP proofs
- Secure Protocols
Classical Passwords: Identity Theft

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

Passwords are no good
Zero Knowledge: Preventing Identity Theft

To identify itself, prover proves that he knows a proof of the theorem.

**PROVER**

<table>
<thead>
<tr>
<th>Smart Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard Theorem: I know a Square root of $y \mod N$</td>
</tr>
</tbody>
</table>

**VERIFIER**

ATM/Main Frame
More generally,

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

Let G be a cyclic group of prime order q, Let both prover and verifier know y in G and
Claim: (P,V) is ZKPOX for the discrete log of y

1. Choose r At random In \( \mathbb{Z}_q \)
2. Choose c At random in \{0,1\}
3. Let \( z = r + cs \mod q \)
4. Accept iff \( g^z = Ry^c \mod p \),