CSCI 1515 Applied Cryptography

This Lecture:

- Zero-Knowledge Proofs for All NP
- Succinct Non-Interactive Arguments (SNARGs)
Zero-Knowledge Proof for Graph 3-Coloring (All NP)

NP language \( L = \{ G : G \text{ has 3-coloring} \} \)

NP relation \( R_L = \{ (G, 3\text{COL}) \} \)

\[ G = (V, E) \]

If \( G \in L \), \( \Pr[\hat{P}^* \text{ is caught}] \geq \frac{1}{1/\infl} \)

\( \Pr[\hat{P}^* \text{ convinces } V] \leq (1 - \frac{1}{1/\infl})^\lambda \cdot 1/\infl \)

\( (1 - \frac{1}{N})^N \approx \frac{1}{e} \) \( (\frac{1}{e})^\lambda \)

How to amplify soundness?
**Commitment Scheme**

<table>
<thead>
<tr>
<th>Sender</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m \in {0, 1} )</td>
<td>( r \in {0, 1} )</td>
</tr>
</tbody>
</table>

**Commit:**

\[
C := \text{Com}(m; r)
\]

**Open:**

\[
(m, r) \quad \rightarrow \quad C
\]

**Verify:**

\[
C = \text{Com}(m; r)
\]

---

**Example:** Pedersen Commitment

Cyclic group \( G \) of order \( q \) with generator \( g \), \( h \in G \)

\[
r \in \mathbb{Z}_q
\]

\[
\text{Com}(m; r) = g^m \cdot h^r = C
\]
Commitment Scheme

- **Hiding:** \( \text{Com}(0; r) \approx \text{Com}(1; s) \)
  - Perfectly hiding: \( \text{Com}(0; r) \equiv \text{Com}(1; s) \)
  - Computationally hiding: \( \text{Com}(0; r) \not\approx \text{Com}(1; s) \)

- **Binding:** Hard to find \( r, s \) s.t. \( \text{Com}(0; r) = \text{Com}(1; s) \)
  - Perfectly binding: \( \forall r, s, \text{Com}(0; r) \neq \text{Com}(1; s) \)
  - Computationally binding: Any PPT sender cannot find \( r, s \) s.t. \( \text{Com}(0; r) = \text{Com}(1; s) \)

**What does Pedersen commitment scheme satisfy?**

\[
\begin{align*}
  r &\in \mathbb{Z}_q \quad \text{random in } G \text{ (OTP)} \\
  \text{Com}(m; r) &= g^m \cdot h^r = c \\
  &\equiv \text{perfectly hiding} \\
  &\equiv \text{computationally binding}
\end{align*}
\]

**Can a commitment scheme be both perfectly hiding & perfectly binding?** NO!
Zero-Knowledge Proof for Graph 3-Coloring

Input: \( G = (V, E) \)

Witness: \( \phi: V \rightarrow \{0, 1, 2\} \)

Given a computationally hiding, perfectly binding commitment scheme.

<table>
<thead>
<tr>
<th>Prover</th>
<th>Verifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomly sample ( \pi: {0, 1, 2} \rightarrow {0, 1, 2} )</td>
<td>( (u, v) ) randomly picked an edge ( (u, v) \in E )</td>
</tr>
<tr>
<td>( \forall v \in V, u \in {0, 1, 2}, \ \text{CV} := \text{Com}(\pi(\phi(v)); r_v) )</td>
<td>( {\text{CV}}_{v \in V} )</td>
</tr>
</tbody>
</table>

Open commitments \( C_u \) & \( C_v \)

\[
\begin{align*}
\alpha &= \pi(\phi(u)), \ r_u \\
\beta &= \pi(\phi(v)), \ r_v
\end{align*}
\]

Verify:
\[
\begin{align*}
C_u &= \text{Com}(\alpha; r_u) \\
C_v &= \text{Com}(\beta; r_v) \\
\alpha, \beta &\in \{0, 1, 2\}, \ \alpha \neq \beta
\end{align*}
\]
Circuit Satisfiability (NP Complete)

NP language $L_c = \{ x \in \{0,1\}^n : \exists w \in \{0,1\}^m \text{ st. } C(x,w) = 1 \}$

NP relation $R_c = \{ (x, w) : C(x, w) = 1 \}$

Example: pre-image of hash function

$C(x, w) = H(w) - x + 1$

$H(w) = x$
ZKP for Circuit Satisfiability

```
\begin{align*}
\text{Com}(1) & \quad \text{OR} \\
\text{Com}(0) & \quad \text{OR} \\
\text{Com}(0) & \\
\text{Com}(1) = c_1 & \\
\text{Com}(0) = c_2 & \\
\text{Com}(1) = c_3 & \\
\end{align*}
```
Proof Systems for Circuit Satisfiability

NP relation $R_{lc} = \exists (x, w) : C(x, w) = 1$

<table>
<thead>
<tr>
<th></th>
<th>NP</th>
<th>$\Sigma$-Protocol</th>
<th>(Fiat-Shamir) NIZK</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(x, w)$</td>
<td>$\xrightarrow{w} V(x)$</td>
<td>$P(x, w) \leftrightarrow V(x)$</td>
<td>$P(x, w) \xrightarrow{\Pi} V(x)$</td>
</tr>
<tr>
<td>Zero-Knowledge</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Non-Interactive</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Communication</td>
<td>$O(lw_l)$</td>
<td>$O(lc_l \cdot \lambda)$</td>
<td>$O(lc_l \cdot \lambda)$</td>
</tr>
<tr>
<td>V's computation</td>
<td>$O(lc_l)$</td>
<td>$O(lc_l \cdot \lambda)$</td>
<td>$O(lc_l \cdot \lambda)$</td>
</tr>
</tbody>
</table>

Can we have communication complexity & verifier's computational complexity sublinear in $lc_l$ & $lw_l$?
Succinct Non-Interactive Argument (SNARG)

Def: A non-interactive proof/argument system is succinct if
- The proof $\pi$ is of length $|\pi| = \text{poly}(\lambda, \log |C|)$
- The verifier runs in time $\text{poly}(\lambda, |x|, \log |C|)$

- **SNARK**: Succinct Non-Interactive Argument of Knowledge
- **zk-SNARG/zk-SNARK**: SNARG/SNARK + Zero-Knowledge

Why Succinct Proofs?

Is it possible?
Verifiable Computation

Server

X

compute f

y

Client

y = f(x)
Anonymous Transactions on Blockchain

Alice's Account A $\xrightarrow{2\text{BTC}}$ Bob's Account B

$\text{VKA (public)}$

$\text{SKA (private)}$

$\text{VKB (public)}$

$\text{SKB (private)}$

Transaction: $\text{VKA, VKB, 2BTC}$, $6 = \text{Sign}_{SKA}$

Anonymous Transaction:

$\text{Com}(\text{VKA, VKB, 2BTC}, 6 = \text{Sign}_{SKA})$

NIZK: valid transaction
Probabilistically Checkable Proof (PCP)

**Prover**

\((x,w)\)

\(\pi \in \{0,1\}^m\)

\(0 \ 1 \ 1 \ 0 \ 1 \ \cdots \ 1\)

\(\uparrow \ \uparrow \ \uparrow \ \uparrow \)

**Verifier**

\((x)\)

PCP Theorem (Informal):

Every NP language has a PCP where the Verifier reads only a constant number of bits of the proof.
First Attempt

Prover

\((x,w)\)

\[\text{Com}(01101 \cdots 1)\]

Verifier

\((x)\)

\(i,j,k\)

Open \(\text{Com}(ti), \text{Com}(tj), \text{Com}(tk)\)
Merkle Tree

Why (computationally) binding?

Can we make it hiding?