This Lecture:

· Sigma Protocol and Examples (Continued)
· Proving AND/OR Statements
· Non-Interactive Zero-Knowledge (NI\(\text{ZK}\)) Proof
· Fiat-Shamir Heuristic
Zero-Knowledge Proof of Knowledge

- **Completeness:** \( \forall (x,w) \in R_L, \Pr [ P(x,w) \leftrightarrow V(x) \text{ outputs } 1 ] = 1 \).

- **Soundness:** \( \forall x \in L, \forall P^*, \Pr [ P^*(x) \leftrightarrow V(x) \text{ outputs } 1 ] \approx 0 \).

- **Proof of Knowledge:** \( \exists \text{ PPT } E \text{ s.t. } \forall P^*, \forall x, \Pr [ E^{P^*}(x) \text{ outputs } w \text{ s.t. } (x,w) \in R_L ] \approx \Pr [ P^* \leftrightarrow V(x) \text{ outputs } 1 ] \).

- **Honest-Verifier Zero-Knowledge:** \( \exists \text{ PPT } S \text{ s.t. } \forall (x,w) \in R_L, \text{ View}_V [ P(x,w) \leftrightarrow V(x) ] \approx S(x) \).

- **Zero-Knowledge:** \( \forall \text{ PPT } V^*, \exists \text{ PPT } S \text{ s.t. } \forall (x,w) \in R_L, \text{ Output}_{V^*} [ P(x,w) \leftrightarrow V^*(x) ] \approx S(x) \).
Example 1: Schnorr's Identification Protocol

Input: Cyclic group $G$ of order $q$, generator $g$, $h = g^a$

Witness: $a$

$R = \{(h = g^a, a)\}$

**Prover**

$r \in \mathbb{Z}_q$

$A = g^r$

$S = 6 \cdot a + r \pmod{q}$

**Verifier**

$6 \in \mathbb{Z}_q$

Verify:

$g^S \overset{?}{=} h^6 \cdot A$

Completeness?

$g^S = g^{6 \cdot a + r}$

$h^6 \cdot A = (g^a)^6 \cdot g^r = g^{6 \cdot a + r}$

$\Rightarrow$ Verifier always outputs 1
Proof of Knowledge?

Extract a s.t. \( h = g^a \)?

Prover

\[ A \rightarrow 6 \rightarrow S \]

\[ 6 \leftarrow Z_q \]

Rewind \( \sigma' \)

\[ g^s \overset{?}{=} h^s \cdot A \]

\[ \sigma \Rightarrow s \text{ s.t. } g^s = h^s \cdot A \]

\[ \sigma' \Rightarrow s' \text{ s.t. } g^{s'} = h^{s'} \cdot A \]

\[ g^{s-s'} = h^{s-s'} \]

\[ g^{(s-s')(s-s')^{-1}} = h \]

\[ a = (s-s')(s-s')^{-1} \pmod{q} \]
Honest Verifier Zero Knowledge (HVZK)

\[ \forall (x, w) \in R_L, \quad \text{View}_V[\mathcal{P}(x, w) \leftrightarrow V(x)] \approx S(x) \]
### Example 2: Chaum-Pedersen Protocol for DH Tuple

**Input:** Cyclic group $G$ of order $q$, generator $g$, $h$, $u$, $v$

**Witness:** $b$

**Statement:** $\exists b \in \mathbb{Z}_q$ s.t. $u = g^b$ ∧ $v = h^b$

<table>
<thead>
<tr>
<th><strong>Prover</strong></th>
<th><strong>Verifier</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$r \leftarrow \mathbb{Z}_q$</td>
<td>$a = g^r$, $b = h^r$</td>
</tr>
<tr>
<td>$s = 6 \cdot b + r \pmod{q}$</td>
<td>$c \leftarrow \mathbb{Z}_q$</td>
</tr>
<tr>
<td>$g^s = g^6 \cdot b + r$</td>
<td>Verify: $g^s = u^6 \cdot A$</td>
</tr>
</tbody>
</table>

**Completeness check?**

- $g^s = g^6 \cdot b + r$
- $h^s = h^6 \cdot b + r$

$u^6 \cdot A = (g^b)^6 \cdot g^5 = g^6 \cdot b + r$

$\Rightarrow$ Verifier always outputs 1.
Proof of Knowledge?

Extract \( b \) s.t. \( u = g^b \land v = h^b \)?

\[
\begin{align*}
\text{Prover}^* & \quad \text{Extractor} \\
A, B & \quad 6 \\
S & \quad 6 \notin \mathbb{Z}_q \\
\end{align*}
\]

Rewind \( s' \)

Verify:
\[
\begin{align*}
g^s & = u^6 \cdot A \\
h^s & = v^6 \cdot B
\end{align*}
\]

\( 6 \Rightarrow s \) s.t. \( g^s = u^6 \cdot A, \ h^s = v^6 \cdot B \)

\( 6' \Rightarrow s' \) s.t. \( g^{s'} = u^{6'} \cdot A, \ h^{s'} = v^{6'} \cdot B \)

\[
\begin{align*}
g^{s-s'} & = u^{6-6'} \\
h^{s-s'} & = v^{6-6'}
\end{align*}
\]

\[
\begin{align*}
g(s-s')(6-s')^{-1} & = u, \ h(s-s')(6-s')^{-1} = v \\
b = (s-s')(6-s')^{-1} \pmod{q}
\end{align*}
\]
Honest Verifier Zero Knowledge?

\( \forall (x,w) \in R_L, \quad \text{View}_V \left[ P(x,w) \leftrightarrow V(x) \right] \approx S(x) \)
Example 3: Okamoto's Protocol for Representation

**Input:** Cyclic group $G$ of order $q$, generator $g, h$, $u = g^a h^b$

**Witness:** $(a, b)$

$R = \{(u = g^a h^b, (a, b))\}$

<table>
<thead>
<tr>
<th>Prover</th>
<th>Verifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1 \leftarrow Z_q$</td>
<td>$A = g^{r_2} h$</td>
</tr>
<tr>
<td>$r_2 \leftarrow Z_q$</td>
<td>$6 \leftarrow Z_q$</td>
</tr>
<tr>
<td>$s_1 = 6a + r_2$</td>
<td>Verify: $g^{s_1} h^{s_2} = u^6 A$</td>
</tr>
<tr>
<td>$s_2 = 6b + r_2$</td>
<td></td>
</tr>
</tbody>
</table>

Completeness? $g^{s_1} h^{s_2} = g^{6a + r_1} h^{6b + r_2}$

$u^6 A = (g^a h^b)^6 \cdot g^{r_1} h^{r_2} = g^{6a + r_1} h^{6b + r_2}$

$\Rightarrow$ Verifier always outputs 1
Proof of Knowledge?

Extract \((a,b)\) s.t. \(u = g^ah^b\) ?

\[ a = (s_1 - s'_i)(6 - s')^{-1} \cdot (s_2 - s'_2)(6 - s')^{-1} \pmod{q} \]
Honest Verifier Zero Knowledge?

\[ \forall (x,w) \in R_L, \quad \text{View}_V[\text{P}(x,w) \leftrightarrow V(x)] \approx S(x) \]
Example 4: Arbitrary Linear Equations

Input: Cyclic group $G$ of order $q$, generator $g$, $h$, $u$, $v$
Witness: $(a, b, c)$

$$u = g^a h^b$$
$$h = u^a v^b q^c$$

\[
\begin{align*}
\text{Prover} & : & r_1, r_2, r_3 \in \mathbb{Z}_q \\
& & A = g^{r_1} h^{r_2} \\
& & B = u^{r_1} v^{r_2} q^{r_3}
\end{align*}
\]

\[
\begin{align*}
\text{Verifier} & : & 6 \in \mathbb{Z}_q \\
& & g^{s_1} h^{s_2} = u^6 A \\
& & u^{s_1} v^{s_2} q^{s_3} = h^6 B
\end{align*}
\]

Completeness?

PoK? $6 \& 6'$

HVZK? $\{1, 2, 3\}$
Proving AND/OR Statements?

Statements: \( X_1, X_2 \)

Witnesses: \( W_1, W_2 \)

AND: \( R_{\text{AND}} = \{ (X_1, X_2), (W_1, W_2) : (X_1, W_1) \in R_{L_1} \text{ AND } (X_2, W_2) \in R_{L_2} \} \)

OR: \( R_{\text{OR}} = \{ (X_1, X_2), (W_1, W_2) : (X_1, W_1) \in R_{L_1} \text{ OR } (X_2, W_2) \in R_{L_2} \} \)
Proving OR Statement

\[ R_{OR} = \{ (x_1, x_2), (w_1, w_2) : (x_1, w_1) \in R_{L_1} \text{ OR } (x_2, w_2) \in R_{L_2} \} \]

Say Prover only has \( w_1 \), how to generate response?

Completeness?  
\( (A_1, 6_1, S_1) \) is valid following completeness & HVEK of \( R_{L_2} \).  
\( (A_2, 6_2, S_2) \) is valid following completeness of \( R_{L_1} \).
Proof of Knowledge?

Extract \((W_1, W_2)\) s.t. \((X_1, W_1) \in R_{L_1}\) OR \((X_2, W_2) \in R_{L_2}\)?

**Prover**

\[
\begin{array}{cccc}
A_1 & A_2 \\
6 & 6' \\
S_1 & S_2
\end{array}
\]

**Extractor**

\[
\begin{array}{cccc}
A_1 & A_2 \\
6 & 6' \\
S_1 & S_2
\end{array}
\]

Verify:
\[
(A_1, S_1, S_1) \\
(A_2, S_2, S_2) \\
6 = 6_1 + 6_2
\]

\[
\begin{array}{cccc}
6 \Rightarrow 6_1 & 6_2 & \text{s.t.} & (A_1, 6_1, S_1) \\
& S_1 & S_2 & (A_2, 6_2, S_2) \\
& 6 = 6_1 + 6_2 & \Rightarrow & 6_1 \neq 6_1' \text{ OR } 6_2 \neq 6_2'
\end{array}
\]

\[
\begin{array}{cccc}
6' \Rightarrow 6_1' & 6_2' & \text{s.t.} & (A_1, 6_1', S_1') \\
& S_1' & S_2' & (A_2, 6_2', S_2') \\
& 6' = 6_1' + 6_2' & \Rightarrow & 6_1 \neq 6_1'
\end{array}
\]

Say \(6_1 \neq 6_1'\)
Honest Verifier Zero Knowledge?

\[ \forall (x, w) \in R_L, \quad \text{view}_V [P(x, w) \leftrightarrow V(x)] \approx S(x) \]
Sigma Protocols $\Sigma$

**Prover**

Input: $(x, w)$

---

**Verifier**

Input: $x$

---

"Commitment"  
6  
response  
Verify

$\Delta$
Non-Interactive Zero-Knowledge (NIZK) Proof

Prover

Input: \((X, W)\)

\[\Pi\]

Verifier

Input: \(X\)

Verify

\[\cdot \text{Completeness: } \forall (X, W) \in \mathcal{R}_L, \quad \Pr[P(x, w) \rightarrow V(x) \text{ outputs } 1] = 1.\]

\[\cdot \text{Soundness: } \forall x \in L, \forall P^*, \quad \Pr[P^*(x) \rightarrow V(x) \text{ outputs } 1] \approx 0.\]

\[\cdot \text{Zero-Knowledge: } \forall \text{PPT } V^*, \exists \text{PPT } S \text{ s.t. } \forall (X, W) \in \mathcal{R}_L, \quad \mathbb{E}[P(x, w) \rightarrow V^*(x)] \approx S(x)\]

Is it possible? NOT in the "plain" model

Is \((g, h, u, v)\) a DH tuple?

If \((g, h, u, v)\) is a DH tuple, then \(S(\text{tuple})\) outputs a valid proof.

If \((g, h, u, v)\) is not a DH tuple, then \(S(\text{tuple})\) cannot output a valid proof.
Model 1: Common Random String / Common Reference String (CRS)

\[ \text{Prover} \]
\[ \text{Input: } (X, W) \]

\[ \text{Verifier} \]
\[ \text{Input: } x \]

\[ S(x) \text{ generates both } (6, \pi) \]

- **Zero-Knowledge:** \( \forall \text{APPT } V^*, \exists \text{APPT } S \text{ s.t. } \forall (x, w) \in \mathbb{R}_L, \]
  \[ \text{Output}_{V^*} \left[ 6 \leftarrow \text{Gen}(1^\lambda), P(x, w, 6) \rightarrow V^*(x, 6) \right] \approx S(x) \]

  Alternatively: \( (6 \leftarrow \text{Gen}(1^\lambda), P(x, w, 6)) \approx S(x) \)
Model 2: Random Oracle Model

Prover
Input: \((X, W)\)

Verifier
Input: \(x\)

\(H\)

\(S\)

\(\Pi\)

Verify

\(S\) controls input/output behavior of RO
Fiat-Shamir Heuristic

Sigma Protocol $\Rightarrow$ NIZK in the RO model

**Prover**

Input: $(X, W)$

$\sigma = H(x \| m_1)$

**Verifier**

Input: $x$

$6 \in D$

Verify

$\sigma = H(x \| m_2)$
Fiat-Shamir Heuristic

Public-Coin HVZK $\Rightarrow$ NIZK in the RO model

<table>
<thead>
<tr>
<th>Prover</th>
<th>Verifier</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $(X, W)$</td>
<td><strong>Input:</strong> $x$</td>
</tr>
<tr>
<td>$m_1$</td>
<td>$6_1 \leftarrow D_2$</td>
</tr>
<tr>
<td>$6_1$</td>
<td>$m_2$</td>
</tr>
<tr>
<td>$6_2$</td>
<td>$6_2 \leftarrow D_2$</td>
</tr>
<tr>
<td>$m_3$</td>
<td><strong>Verify</strong></td>
</tr>
</tbody>
</table>

$6_1 := H(x \ || \ m_1)$

$6_2 := H(x \ || \ m_1 \ || \ m_2)$
Fiat–Shamir Heuristic

Schnorr's Identification Protocol \Rightarrow Schnorr's Signature in the RO model

Cyclic group $G$ of order $g$, generator $g$

Public verification key $\text{vk} = g^a$; Secret signing key $\text{sk} = a$

\[
\begin{array}{c|c}
\text{Prover} & \text{Verifier} \\
\hline
r \leftarrow \mathbb{Z}_q & A_1 = g^r \\
\hline
6 \leftarrow A_1 & 6 \leftarrow \mathbb{Z}_q \\
\hline
S := 6 \cdot a + r \pmod{q} & \text{Verify:} \\
\end{array}
\]

\[g^S \overset{?}{=} \text{vk}^6 \cdot A\]

To sign a message $m$: $6 := H(m || A || \text{vk})$
Anonymous Online Voting

Voter 1 → $\text{Enc}(V_1) \quad V_1 \in \{0, 1\}$

Voter 2 → $\text{Enc}(V_2) \quad V_2 \in \{0, 1\}$

⋮

Voter $n$ → $\text{Enc}(V_n) \quad V_n \in \{0, 1\}$

$\downarrow$

$\text{Enc}(\Sigma V_i)$

$\downarrow$

Decrypt to $\Sigma V_i$