Lecture 5: Σ-protocols from DLOG

Zero-knowledge proofs

263-4665-00L

Lecturer: Jonathan Bootle

Announcements

- Exercise sheet 5 posted on Moodle
- Graded, 10% of final grade
- Submit through Moodle on or before 23:59, 20/10/2023
- Please email if you think you’ve found a typo or mistake

- 20/10/2023 exercise session used for optional/starred exercises

Last time

- Composition methods for Σ-protocols ✓
- Σ-protocols from MPC in the Head ✓
- The Fiat-Shamir Transformation ✓
- Making Σ-protocols zero-knowledge against malicious verifiers

Agenda

- **Making Σ-protocols zero-knowledge against malicious verifiers**

Sigma protocols from DLOG

- Intro level: Schnorr and homomorphisms
- Medium level: multiplicative relations
- Advanced level: low-degree circuit proofs

Similar techniques when we construct arguments with short proof sizes

Coin flip protocol

Forces honest V by generating challenges together

$\mathcal{C}$ a group

$pp \leftarrow_\$ \mathrm{Setup}(1^\lambda)$

$P(pp)$		$V(pp)$
$c \leftarrow_\$ \mathcal{C}$ $(C, d) \leftarrow_\$ \mathrm{Commit}(pp, c)$	$\xrightarrow{C}$	
	$\xleftarrow{c'}$	$c' \leftarrow_\$ \mathcal{C}$
	$\xrightarrow{c, d}$	Check $\mathrm{Verify}(pp, C, d, c) = 1$ Output $c'' := c + c'$

Compiling Σ-protocols to fully ZK protocols

Variant where (P, V) have opposite roles in coin flip protocol

First assume $\mathcal{C}$ is polynomially bounded

$$pp \leftarrow_{\$} \text{Setup}(1^\lambda)$$

$P^{ZK}(pp, x, w; \rho, c)$		$V^{ZK}(pp, x; c')$
$a \leftarrow_{\$} P_1(x, w; \rho)$		
$c \leftarrow_{\$} \mathcal{C}$		
$(C, d) \leftarrow_{\$} Commit(pp, c)$	$\xrightarrow{a,\, C}$	
	$\xleftarrow{c'}$	$c' \leftarrow_{\$} \mathcal{C}$
$c'' := c + c'$		$c'' := c + c'$
$z \leftarrow_{\$} P_2(x, w, a, c''; \rho)$	$\xrightarrow{z,\, c,\, d}$	Output $V(x, a, c'', z)$ $\wedge \text{Verify}(pp, C, d, c) == 1$

Theorem:

Against malicious verifiers

(P^{ZK}, V^{ZK}) has completeness, k-special soundness and zero-knowledge.

Completeness and special soundness analysis

Completeness:

- Follows from the completeness of the Σ-protocol and correctness of the commitment scheme.

Special Soundness:

accepting k-tree for (P^{ZK}, V^{ZK})

$$
\begin{array}{ccc}
 & a, C & \\
\swarrow^{c_1'} & \cdots & \searrow^{c_k'} \\
z_1 c_1, d_1 & & z_k, c_k, d_k
\end{array}
\quad \text{distinct}
$$

$\forall i \in [k]$

$c_i'' = c_i + c_i'$

$\mathrm{Verify}(pp, C, d_i, c_i) = 1$

$V(x, a, c_i'', z_i) = 1$

$c := c_1 = \cdots = c_k$

Or we break binding

accepting k-tree for (P, V)

$$
\begin{array}{ccc}
 & a, C & \\
\swarrow^{c_1''} & \cdots & \searrow^{c_k''} \\
z_1 & & z_k
\end{array}
\quad \text{distinct} \xrightarrow{E} (x, w) \in \mathcal{R}
$$

$c_i'' = c + c_i'$ values remain distinct

Zero-knowledge analysis

$S^{ZK}(pp, x)$

1. $c'', c \leftarrow_\$ \mathcal{C}$.
2. $(a, z) \leftarrow_\$ S(x, c'')$.
3. $(C, d) \leftarrow_\$ \mathrm{Commit}(pp, c)$.
4. $c' \leftarrow_\$ V^{ZK,*}(pp, x, C, a)$
5. If $c + c' \neq c''$ go back to 1.
6. Output $\big((a, C), c', (z, c, d)\big)$.

$V^{ZK,*}(pp, x)$
???

a, c, z from Σ-protocol

What is the verifier's view?

c' sampled by verifier

- $\big((a, C), c', (z, c, d)\big)$
- $V(x, a, c + c', z) = 1$
- $\mathrm{Verify}(pp, C, d, c) = 1$.

Commit, decommit distributions from Commit

- c' is independent of c or we could break hiding.
- Hence $c + c'$ is uniformly random.

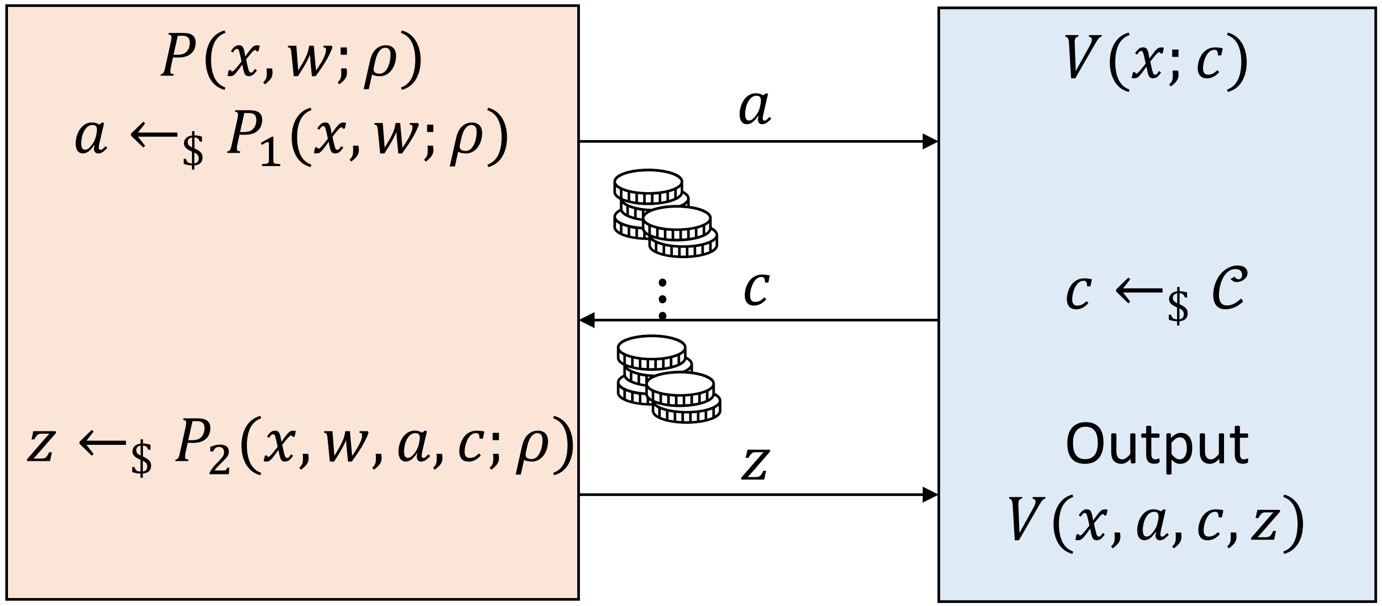
We could have made this ZK without commits using guessing strategy

Why is the simulator valid? (efficient, indistinguishable)

- $\mathcal{C}$ is polynomially bounded and we clear Step 5 in $|\mathcal{C}|$ tries
- SHVZK of the Σ-protocol means (a, z) distributions indistinguishable

Compiling Σ-protocols to fully ZK protocols

Large $\mathcal{C}$ e.g. $\{0,1\}^\lambda$ for simplicity



λ times
1 for each bit

Compiling Σ-protocols to fully ZK protocols

$$pp \leftarrow_\$ \text{Setup}(1^\lambda)$$

$P^{ZK}(pp, x, w; \rho, c_1)$		$V^{ZK}(pp, x; c_2)$
$a \leftarrow_\$ P_1(x, w; \rho)$	$\xrightarrow{a}$	
For $i \in [\lambda]$: $c_i \leftarrow_\$ \{0,1\}$		
$(C_i, d_i) \leftarrow_\$ \text{Commit}(pp, c_i)$	$\xrightarrow{C_1}$	
	$\xleftarrow{c_1'}$	For $i \in [\lambda]$: $c_i' \leftarrow_\$ \{0,1\}$
$c_i'' \coloneqq c_i \oplus c_i'$	$\xrightarrow{c_1, d_1}$	$c_i'' \coloneqq c_i \oplus c_i'$
	$\vdots$	
$c'' \coloneqq (c_1'', \dots, c_\lambda'')$		Output 1 iff $V(x, a, c'', z) = 1$
$z \leftarrow_\$ P_2(x, w, a, c''; \rho)$	$\xrightarrow{z}$	$\wedge\ \text{Verify}(pp, C_i, d_i, c_i) == 1\ \forall i$

Theorem:

Against malicious verifiers

(P^{ZK}, V^{ZK}) has completeness, knowledge-soundness and ZK.

Completeness and knowledge soundness sketch

Completeness:

- Follows from the completeness of the base Σ-protocol and correctness of the commitment scheme.

Knowledge Soundness:

- The theorem that shows special soundness $\Rightarrow$ knowledge soundness has an extractor that gathers trees of accepting transcripts
- Use the same extractor but rewind the whole challenge selection procedure as one block
- Then apply the special soundness extractor

Zero-knowledge sketch

a, c'', z from Σ-protocol

What is the verifier's view?

- $\left(a, (C_i, c_i', d_i, c_i)_{i=1}^{\lambda}, z\right)$
- $V(x, a, c'', z) = 1$
- $\text{Verify}(pp, C_i, d_i, c_i) = 1, i \in [\lambda]$.
- c_i's are independent of c_is or we could break hiding.
- Hence c$'' := (c_1 \oplus c_1', \dots, c_\lambda \oplus c_\lambda')$ is uniformly random.

c_i' sampled by verifier

Why is the simulator valid? (efficient, indistinguishable)

- For each i, we clear Step 6i in 2 tries.
- SHVZK of the Σ-protocol means (a, z) distributions indistinguishable

$S^{ZK}(pp, x)$

1. $c'' \leftarrow_\$ \{0,1\}^\lambda$.
2. $(a, z) \leftarrow_\$ S(x, c'')$.

For $i = 1, \dots, \lambda$:

3i. $c_i \leftarrow_\$ \{0,1\}$.

4i. $(C_i, d_i) \leftarrow_\$ \text{Commit}(pp, c_i)$.

5i. $c_i' \leftarrow_\$ V^{ZK,*}(pp, x, \dots, C_i,)$

6i. If $c_i \oplus c_i' \neq c_i''$ go back to 3i.

7. Output$\left(a, (C_i, c_i', d_i, c_i)_{i=1}^{\lambda}, z\right)$.

$V^{ZK,*}(pp, x)$
???

Hard to guess and simulate bitwise without the commitments

Agenda

- Making Σ-protocols zero-knowledge against malicious verifiers ✓

Sigma protocols from DLOG

- **Intro level: Schnorr and homomorphisms**

- Medium level: multiplicative relations

- Advanced level: low-degree circuit proofs

Similar techniques when we construct arguments with short proof sizes

Schnorr protocol

Changed notation, used capitals for group elements

- $\mathcal{R}_{DLOG} \coloneqq \{((\mathbb{G}, G, A, p), a) : G, A \in \mathbb{G},\ a \in \mathbb{Z}_p, A = a \cdot G\}$.

Trivial language

$P(\mathbb{G}, G, A, p, a)$		$V(\mathbb{G}, G, A, p)$
$b \leftarrow_\$ \mathbb{Z}_p$ $B \coloneqq b \cdot G$	$\xrightarrow{B}$	
	$\xleftarrow{x}$	$x \leftarrow_\$ \mathbb{Z}_p$
$z \coloneqq xa + b$	$\xrightarrow{z}$	Output $(z \cdot G == x \cdot A + B)$

$p \approx 2^\lambda$

Guessing strategy for full ZK will not work

Success probability $1/p \approx 0$

Idea: (P, V) randomize the instance
P solves it, just like GI

Completeness and 2-soundness analysis

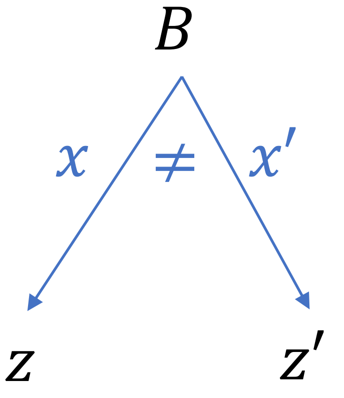
Completeness:

- $z = xa + b$ so $z \cdot G = (xa + b) \cdot G = xa \cdot G + b \cdot G = x \cdot A + B$
- This is exactly the verifier's check.

2-special soundness:

- Consider a 2-tree of accepting transcripts.
- Subtracting, $(z' - z) \cdot G = (x' - x) \cdot A$.
- Dividing, $A = \frac{z' - z}{x' - x} \cdot G$, so $a = \frac{z' - z}{x' - x}$ is a witness.
- The extractor returns $a \in \mathbb{Z}_p$.
- Clearly, a can be computed efficiently.

Used $x \neq x'$



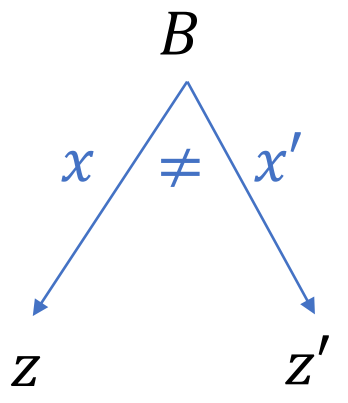
$z \cdot G = x \cdot A + B$

$z' \cdot G = x' \cdot A + B$

Linear algebra view of 2-soundness analysis

Completeness:

- $z = xa + b$ so $z \cdot G = (xa + b) \cdot G = xa \cdot G + b \cdot G = x \cdot H + H'$
- This is exactly the verifier's check.



2-special soundness:

- Consider a 2-tree of accepting transcripts.
- $Q(x, x') := \begin{pmatrix} x & 1 \\ x' & 1 \end{pmatrix}$ is invertible. $Q^{-1} = \frac{1}{x - x'} \begin{pmatrix} 1 & -1 \\ -x' & x \end{pmatrix}$
- Inverting, $\begin{pmatrix} A \\ B \end{pmatrix} = \frac{1}{x - x'} \begin{pmatrix} 1 & -1 \\ -x' & x \end{pmatrix} \begin{pmatrix} z \\ z' \end{pmatrix} \cdot G$

$$\begin{pmatrix} z \\ z' \end{pmatrix} \cdot G = \begin{pmatrix} x & 1 \\ x' & 1 \end{pmatrix} \cdot \begin{pmatrix} A \\ B \end{pmatrix}$$

- The extractor returns $a := \frac{z' - z}{x' - x} \in \mathbb{Z}_p$.

SHVZK analysis

Note: H does not hide x but SHVZK means the protocol doesn't make it easier to compute x

What is the verifier's view?

- (B, x, z) with $z \cdot G = x \cdot A + B$.
- $b \leftarrow_\$ \mathbb{Z}_p$ so $z = xa + b$ is uniform in $\mathbb{Z}_p$.
- $B = z \cdot G - x \cdot A$ is uniquely determined.

$S(\mathbb{G}, G, A, p, x)$
1. $z \leftarrow_\$ \mathbb{Z}_p$.
2. $B \coloneqq z \cdot G - x \cdot A$.
3. Output (B, x, z).

Why is the simulator valid? (efficient, indistinguishable)

- Clearly, the simulator is efficient.
- z and B have identical distributions to the real protocol.
- We simply simulated in reverse order.

Same DLOG protocol

Non-trivial language

- $\mathcal{R}_{=DLOG} := \left\{ \left((\mathbb{G}, G, A, U, V, p\,), a \right) : \begin{array}{c} G, A, U, V \in \mathbb{G},\ \ a \in \mathbb{Z}_p, \\ A = a \cdot G,\ \ V = a \cdot U \end{array} \right\}.$

$P(\mathbb{G}, G, A, U, V, p, a)$		$V(\mathbb{G}, G, A, U, V, p)$
$b \leftarrow_\$ \mathbb{Z}_p$ $B := b \cdot G$ $W := b \cdot U$	$\xrightarrow{B, W}$	
	$\xleftarrow{x}$	$x \leftarrow_\$ \mathbb{Z}_p$
$z := xa + b$	$\xrightarrow{z}$	Output 1 iff $z \cdot G == x \cdot A + B$ $z \cdot U == x \cdot V + W$

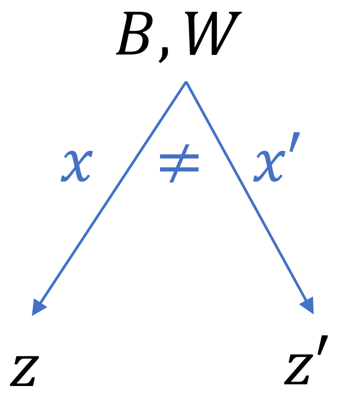
Idea: run two Schnorr protocols with the same witness and randomness

Different from AND composition

Completeness and 2-soundness analysis

Completeness:

- $z = xa + b$ so $z \cdot G = (xa + b) \cdot G = xa \cdot G + b \cdot G = x \cdot A + B$
- Similarly, $z \cdot U = (xa + b) \cdot U = xa \cdot U + b \cdot U = x \cdot V + W$
- These are exactly the verifier's checks.



2-special soundness:

- Consider a 2-tree of accepting transcripts.
- $Q(x, x') := \begin{pmatrix} x & 1 \\ x' & 1 \end{pmatrix}$ is invertible.
- Inverting, $\begin{pmatrix} A & V \\ B & W \end{pmatrix} = Q^{-1} \begin{pmatrix} z \\ z' \end{pmatrix} \cdot (G, U) := \begin{pmatrix} a \\ b \end{pmatrix} \cdot (G, U)$
- The extractor returns $a \in \mathbb{Z}_p$.

$\Rightarrow a$ a witness

$$\begin{pmatrix} z \\ z' \end{pmatrix} \cdot (G, U) = \begin{pmatrix} x & 1 \\ x' & 1 \end{pmatrix} \cdot \begin{pmatrix} A & V \\ B & W \end{pmatrix}$$

SHVZK analysis

$S(\mathbb{G}, G, A, U, V, p, x)$
1. $z \leftarrow_\$ \mathbb{Z}_p$.
2. $B \coloneqq z \cdot G - x \cdot A$.
3. $W \coloneqq z \cdot U - x \cdot V$.
4. Output (B, W, x, z).

What is the verifier's view?

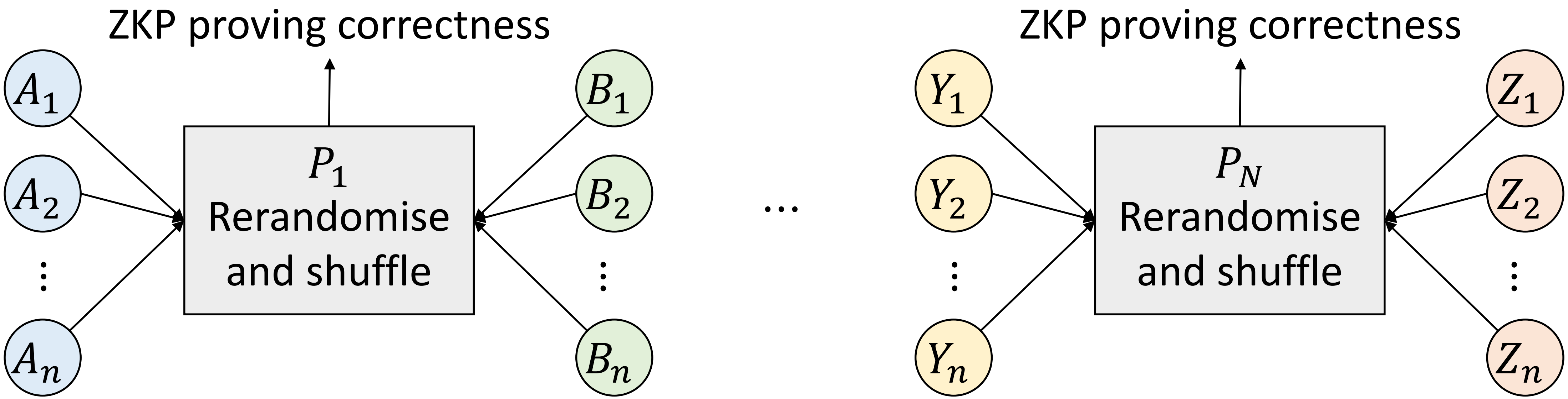
- (B, W, x, z) with $z \cdot G = x \cdot A + B$ and $z \cdot U = x \cdot V + W$.
- $b \leftarrow_\$ \mathbb{Z}_p$ so $z = xa + b$ is uniform in $\mathbb{Z}_p$.
- $B = z \cdot G - x \cdot A$ and $W = z \cdot U - x \cdot V$ are uniquely determined.

Why is the simulator valid? (efficient, indistinguishable)

- Clearly, the simulator is efficient.
- z and B, W have identical distributions to the real protocol.
- We simply simulated in reverse order.

Application: mix-networks

- Encryption mix networks shuffle encrypted messages



Ciphertexts

$\exists$ honest player $\Rightarrow Z_i$ properly rerandomized and shuffled

Proof guarantees that P_i rerandomized and shuffled correctly (but maybe not very randomly)

Proving correct rerandomisation

Rerandomising Elgamal ciphertexts:

- $(C_1, C_2) = (M + r \cdot H, r \cdot G)$.

Add an encryption of $0_{\mathbb{G}}$

- $(C_1, C_2) + (r' \cdot H, r' \cdot G)$

$= (M + (r + r') \cdot H, (r + r') \cdot G) \coloneqq (C_1', C_2')$.

- $(C_1', C_2') \sim_{rerand} (C_1, C_2) \Leftrightarrow (C_1' - C_1, C_2' - C_2)$ have same DLOG.

Shuffling and rerandomising two ciphertexts

We want to prove

$$\begin{tikzcd}
(C_1, C_2) \arrow[r] \arrow[dr] & (C_1', C_2') \\
(D_1, D_2) \arrow[r] \arrow[ur] & (D_1', D_2')
\end{tikzcd}$$

$(C_1, C_2) \sim_{rerand} (C_1', C_2')$ AND $(D_1, D_2) \sim_{rerand} (D_1', D_2')$

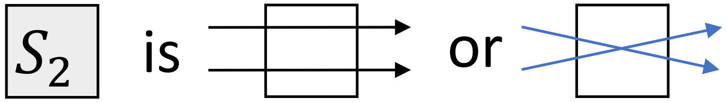
OR

$(C_1, C_2) \sim_{rerand} (D_1', D_2')$ AND $(D_1, D_2) \sim_{rerand} (C_1', C_2')$

We can use the protocol for $\mathcal{R}_{=DLOG}$ with AND, OR composition.

Extending to many ciphertexts

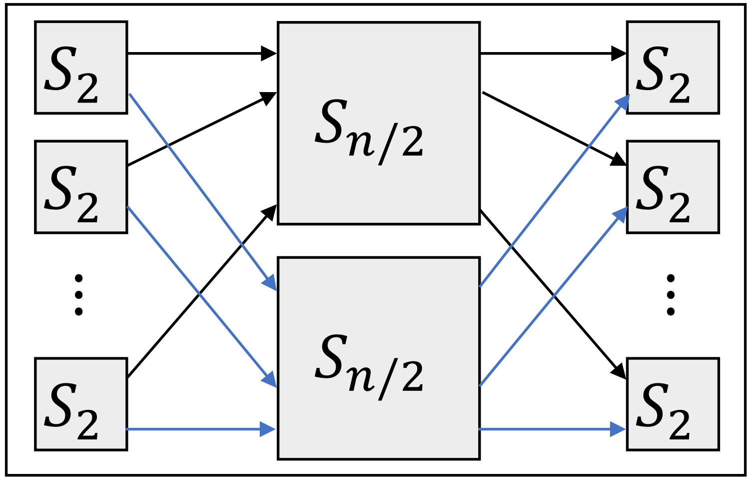
Use a *permutation network*.



Informal Theorem:

For any permutation $\sigma \in \Sigma_n$, there is an efficient algorithm computing settings for the S_2 boxes to produce σ.

S_n



Give a proof for every box, AND compose

$\log n$ layers, $O(n \log n)$ boxes

Proof size, prover, verifier complexity $O(n \log n)$

Pedersen protocol

Trivial language

- $\mathcal{R}_{Ped} := \left\{ \left((\mathbb{G}, G, H, A, p), a, r \right) : \begin{array}{c} G, H, A \in \mathbb{G},\ \ a, r \in \mathbb{Z}_p, \\ A = a \cdot G + r \cdot H \end{array} \right\}.$

$P(\mathbb{G}, G, H, A, p, a, r)$		$V(\mathbb{G}, G, H, A, p)$
$b, s \leftarrow_{\$} \mathbb{Z}$		
$B := b \cdot G + s \cdot H$	$\xrightarrow{B}$	
	$\xleftarrow{x}$	$x \leftarrow_{\$} \mathbb{Z}_p$
$z := xa + b$		
$t := xr + s$	$\xrightarrow{z, t}$	Output 1 iff $z \cdot G + t \cdot H == x \cdot A + B$

Idea: (P, V) randomize the instance

P solves it

Completeness and 2-soundness analysis

Completeness:

- $z = xa + b$ and $t = xr + s$.
- So $z \cdot G + t \cdot H = (xa + b) \cdot G + (xr + s) \cdot H = x \cdot A + B$.
- This is exactly the verifier's check.

2-special soundness:

- Consider a 2-tree of accepting transcripts.
- $Q(x, x')$ is invertible.
- Inverting, $\begin{pmatrix} A \\ B \end{pmatrix} = Q^{-1} \begin{pmatrix} z \\ z' \end{pmatrix} \cdot G + Q^{-1} \begin{pmatrix} t \\ t' \end{pmatrix} \cdot H \coloneqq \begin{pmatrix} a \\ b \end{pmatrix} \cdot G + \begin{pmatrix} r \\ s \end{pmatrix} \cdot H$.
- The extractor returns $a, r \in \mathbb{Z}_p$.

$$\begin{tikzcd} & B \arrow[dl, "x"'] \arrow[dr, "x'"] \arrow[d, phantom, "\neq"] & \\ z, t & & z', t' \end{tikzcd}$$

$$\begin{pmatrix} z \\ z' \end{pmatrix} \cdot G + \begin{pmatrix} t \\ t' \end{pmatrix} \cdot H = \begin{pmatrix} x & 1 \\ x' & 1 \end{pmatrix} \cdot \begin{pmatrix} A \\ B \end{pmatrix}$$

$\Rightarrow$ a a witness

SHVZK analysis

$S(\mathbb{G}, G, H, C, p, x)$
1. $z, t \leftarrow_\$ \mathbb{Z}_p$.
2. $B \coloneqq z \cdot G + t \cdot H - x \cdot A$.
3. Output (B, x, z, t).

What is the verifier's view?

- (B, x, z, t) with $z \cdot G + t \cdot H = x \cdot A + B$.
- $b \leftarrow_\$ \mathbb{Z}_p$ so $z = xa + b$ is uniform in $\mathbb{Z}_p$.
- $s \leftarrow_\$ \mathbb{Z}_p$ so $t = xr + s$ is uniform in $\mathbb{Z}_p$.
- $B = z \cdot G + t \cdot H - x \cdot A$ is uniquely determined.

Why is the simulator valid? (efficient, indistinguishable)

- Clearly, the simulator is efficient.
- z, t and B have identical distributions to the real protocol.

Homomorphisms

Definition:

A function $f : A \to \mathrm{B}$ is a *group homomorphism* if A, B are groups and $\forall a_1, a_2 \in A, f(a_1) + f(a_2) = f(a_1 + a_2)$.

Note: if $f : \mathbb{Z}_p^m \to \mathbb{G}^n$ is a group homomorphism, then $\forall x \in \mathbb{Z}_p, \vec{a} \in \mathbb{Z}_p^m : f(x \cdot \vec{a}) = x \cdot f(\vec{a})$

Definition:

A commitment scheme is *homomorphic* if $\forall pp \in \mathrm{Setup}(1^\lambda)$, for all $m_1, m_2 \in \mathfrak{M}, r_1, r_2 \in \mathfrak{R}$, $\mathrm{Commit}(pp,\cdot,\cdot) : \mathfrak{M} \times \mathfrak{R} \to \mathfrak{C}$ is a group homomorphism i.e.

$$\mathrm{Commit}(pp, m_1, r_1) + \mathrm{Commit}(pp, m_2, r_2) = \mathrm{Commit}(pp, m_1 + m_2, r_1 + r_2)$$

Examples: Pedersen and Elgamal.

Homomorphism preimage protocol

Let $f : \mathbb{Z}_p^m \to \mathbb{G}^n$ be a group homomorphism.

- $\mathcal{R}_{Hom} := \left\{\left(\left(\mathbb{G}, f, \vec{A}, p\right), \vec{a}\right) : \vec{a} \in \mathbb{Z}_p^m, \vec{A} \in Im(f), f(\vec{a}) = \vec{A}\right\}.$

Trivial relation if $Im(f) = \mathbb{G}^n$

$P(\mathbb{G}, f, \vec{A}, p, \vec{a})$		$V(\mathbb{G}, f, \vec{A}, p)$
$\vec{b} \leftarrow_\$ \mathbb{Z}_p^m$ $\vec{B} := f\left(\vec{b}\right)$	$\xrightarrow{\vec{B}}$	
	$\xleftarrow{x}$	$x \leftarrow_\$ \mathbb{Z}_p$
$\vec{z} := x \cdot \vec{a} + \vec{b}$	$\xrightarrow{\vec{z}}$	Output 1 iff $f(\vec{z}) == x \cdot \vec{A} + \vec{B}$

Idea: (P, V) randomize the instance
P solves it

Generalisation of Schnorr protocol

Proof and instantiations: optional exercise

Commitments to 0 and linear relations

- $\mathcal{R}_{Ped} \coloneqq \left\{\left((\mathbb{G}, G, H, A, p\ ), a, r\right) : \begin{matrix} G, H, A \in \mathbb{G},\ \ a, r \in \mathbb{Z}_p, \\ A = a \cdot G + r \cdot H \end{matrix}\right\}.$

- If $a = 0$ then $A = r \cdot H$ so $\left((\mathbb{G}, H, A, p\ ), r\right) \in \mathcal{R}_{DLOG}$.
- Use Schnorr protocol to prove A is a commitment to zero.

- To prove e.g. $a_1 + a_2 = a_3$, run Pedersen proofs on A_1, A_2, A_3 and prove that $A_1 + A_2 - A_3$ is a commitment to zero.
- Include constants using commitments without randomness e.g. $c \cdot G$.

Agenda

- Making Σ-protocols zero-knowledge against malicious verifiers ✓

Sigma protocols from DLOG

- Intro level: Schnorr and homomorphisms ✓
- **Medium level: multiplicative relations**
- Advanced level: low-degree circuit proofs

Similar techniques when we construct arguments with short proof sizes

Multiplication relation

Masked response $z = xa + b$ Mask

Secret

Challenge

- $\mathcal{R}_{Mult} := \left\{ \begin{matrix} \text{Instance } \mathbb{x} \\ (\mathbb{G}, G, H, \{A_i\}_{i\in[3]}, p), \\ \{(a_i, r_i)\}_{i\in[3]} \\ \text{Witness } \mathbb{w} \end{matrix} : \begin{matrix} G, H, A_1, A_2, A_3 \in \mathbb{G}, \\ a_1, a_2, a_3, r_1, r_2, r_3 \in \mathbb{Z}_p, \\ A_i = a_i \cdot G + r_i \cdot H \ \forall i \\ a_1 \cdot a_2 = a_3 \end{matrix} \right\}.$

Technique: computing on masked secrets

- Pedersen protocols for C_1, C_2 give masked z_1, z_2 with a_1, a_2 'inside'.
- Compute $a_3 = a_1 \cdot a_2$ but use z_i instead of a_i.
- $z_1 z_2 = x^2 a_1 a_2 + x(a_1 b_2 + a_2 b_1) + b_1 b_2 := x^2 a_3 + x \cdot m_1 + m_0.$
- P commits to m_1, m_2 before seeing x.
- V checks this equation using the homomorphic commitments.

Multiplication proof

$P(\mathbb{x}, \mathbb{w})$		$V(\mathbb{x})$
$b_1, b_2, s_1, s_2, u_1, u_0 \leftarrow_\$ \mathbb{Z}$ $B_i \coloneqq b_i \cdot G + s_i \cdot H, i \in [2]$ $M_i \coloneqq m_i \cdot G + u_i \cdot H, i \in \{0,1\}$	$\xrightarrow{B_1, B_2, M_1, M_0}$	
	$\xleftarrow{x}$	$x \leftarrow_\$ \mathbb{Z}_p$
$z_i \coloneqq xa_i + b_i$ $t_i \coloneqq xr_i + s_i$ $u_3 \coloneqq x^2 r_3 + xm_1 + m_0$	$\xrightarrow{z_1, z_2, t_1, t_2, u_1, u_2}$	Output 1 iff $z_i \cdot G + t_i \cdot H == x \cdot A_i + B_i, i \in [2]$ $z_1 z_2 \cdot G + u_3 \cdot H == x^2 \cdot A_3 + x \cdot M_1 + M_0$