ساخت توابع چندخطی با استفاده از مبهم‌سازی

پویا فرشیم

(کار مشترک با مارتین آلبرشت، دنیس هوف‌هاینز، انریکو لارایا و کنی پترسون)

دانشگاه کوئینز بلفاست – اکول نرمال سوپریور

جنبه‌های ریاضی علوم کامپیوتری – مبانی رمز

Multilinear Maps from Obfuscation

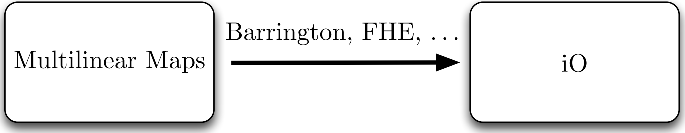
Pooya Farshim

(Joint work with M. Albrecht, D. Hofheinz, E. Larraia and K.G. Paterson)

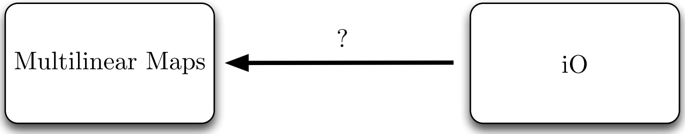
Queen's University Belfast/École Normale Supérieure

Mathematical Aspects of Computer Science – Foundations of Cryptography

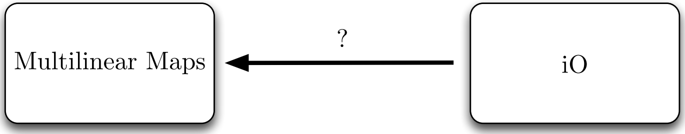
iO Construction



This Raises the Question...



This Raises the Question...

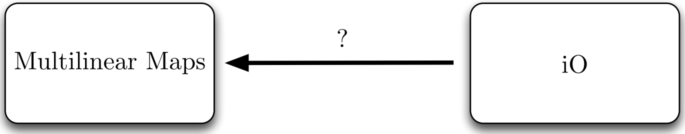


But

$$\mathbf{P} = \mathbf{NP} \implies \text{iO Exists} .$$

(Why?)

This Raises the Question...



But

$$\mathbf{P} = \mathbf{NP} \implies \text{iO Exists} .$$

(Why?) Hence we show

$$\text{iO} + \text{Assumptions} \implies \text{MLM} .$$

Motivation

- Theoretical/conceptual:

  To what extent are MLMs **necessary** for iO?

Motivation

- Theoretical/conceptual:

  To what extent are MLMs **necessary** for iO?

- Applications of MLMs
  - Multiparty key exchange
  - Broadcast Encryption
  - Key-homomorphic PRFs
  - ...

Motivation

- Theoretical/conceptual:

  To what extent are MLMs **necessary** for iO?

- Applications of MLMs
  - Multiparty key exchange
  - Broadcast Encryption
  - Key-homomorphic PRFs
  - ...
- Most notably: The current state of MLMs:

  MLM candidates ([GGH13] and [CLT13]) **broken**, but **not** iO!

Multilinear Groups

κ-linear maps:

$$\mathbf{e} : \mathbb{G}_1 \times \cdots \times \mathbb{G}_\kappa \longrightarrow \mathbb{G}_T$$
$$\mathbf{e}(g_1^{a_1}, \ldots, g_\kappa^{a_\kappa}) \;=\; g_T^{a_1 \cdots a_\kappa}$$

Multilinear Groups

κ-linear maps:

$$\mathbf{e} : \mathbb{G}_1 \times \cdots \times \mathbb{G}_\kappa \longrightarrow \mathbb{G}_T$$
$$\mathbf{e}(g_1^{a_1}, \ldots, g_\kappa^{a_\kappa}) \;=\; g_T^{a_1 \cdots a_\kappa}$$

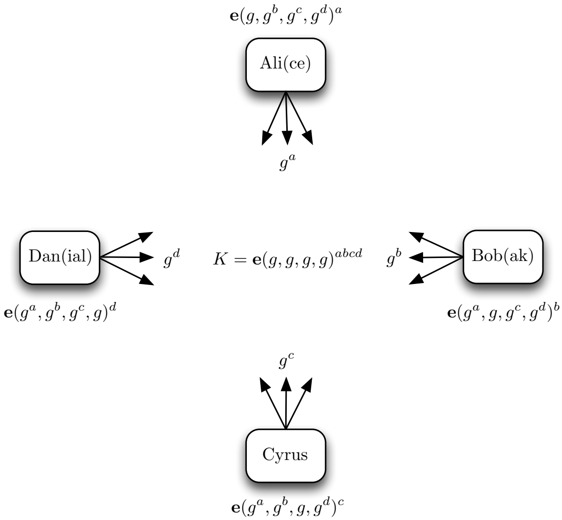
Symmetric setting:

$$\mathbb{G}_i = \mathbb{G}_j = \mathbb{G} \; .$$

This talk:

$$\mathbf{e} : \mathbb{G} \times \cdots \times \mathbb{G} \longrightarrow \mathbb{G}_T$$
$$\mathbf{e}(g^{a_1}, \ldots, g^{a_\kappa}) \;=\; g_T^{a_1 \cdots a_\kappa}$$

NIKE: Non-Interactive Key Exchange



This Talk

$$\boxed{\begin{array}{c}\text{Group}\\ \text{(S)DDH Hard}\end{array}} \xrightarrow{\text{iO, FHE, NIZK}} \boxed{\begin{array}{c}\text{Multilinear Group}\\ \text{MDDH Hard}\end{array}}$$

Some Intractable Problems

The DDH Problem:

$$(g, g^{a_1}, g^{a_2}, g^{a_1 a_2}) \overset{c}{\approx} (g, g^{a_1}, g^{a_2}, g^{r})$$

Some Intractable Problems

The DDH Problem:

$$(g, g^{a_1}, g^{a_2}, g^{a_1 a_2}) \overset{c}{\approx} (g, g^{a_1}, g^{a_2}, g^{r})$$

The q-SDDH Problem:

$$(g, g^{a}, \ldots, g^{a^q}, g^{a^{q+1}}) \overset{c}{\approx} (g, g^{a}, \ldots, g^{a^q}, g^{r})$$

Some Intractable Problems

The DDH Problem:

$$(g, g^{a_1}, g^{a_2}, g^{a_1 a_2}) \overset{c}{\approx} (g, g^{a_1}, g^{a_2}, g^{r})$$

The q-SDDH Problem:

$$(g, g^{a}, \ldots, g^{a^q}, g^{a^{q+1}}) \overset{c}{\approx} (g, g^{a}, \ldots, g^{a^q}, g^{r})$$

The (symmetric) κ-MDDH Problem:

$$(g, g^{a_1}, \ldots, g^{a_\kappa}, g^{a_{\kappa+1}}, g_T^{a_1 \cdots a_{\kappa+1}}) \overset{c}{\approx} (g, g^{a_1}, \ldots, g^{a_\kappa}, g^{a_{\kappa+1}}, g_T^{r})$$

Tool 1: FHE

A fully homomorphic public-key encryption scheme

$$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$$

which is **perfectly** correct wrt. addition in $\mathbb{Z}_p$.

$$\mathsf{Enc}(pk, x_1) \;\text{“}+\text{”}\; \mathsf{Enc}(pk, x_2) = \mathsf{Enc}(pk, x_1 + x_2 \pmod p)\ .$$

Perfect Correctness: Needed later to argue for functional equivalence.

Tool 2: Probabilistic iO (piO)

Similar to iO:

$$\mathsf{piO}(C_0) \overset{c}{\approx} \mathsf{piO}(C_0)$$

but for randomized C_0 and C_1.

Tool 2: Probabilistic iO (piO)

Similar to iO:

$$\mathsf{piO}(C_0) \stackrel{c}{\approx} \mathsf{piO}(C_0)$$

but for randomized C_0 and C_1. For which classes of circuits?

Tool 2: Probabilistic iO (piO)

Similar to iO:

$$\mathsf{piO}(C_0) \overset{c}{\approx} \mathsf{piO}(C_0)$$

but for randomized C_0 and C_1. For which classes of circuits?

X-**IND Samplers**: There is a set X such that

$x \notin X$: Functional equivalence:

$$\forall r \ : \ C_0(x; r) = C_1(x; r)$$

Tool 2: Probabilistic iO (piO)

Similar to iO:

$$\mathrm{piO}(C_0) \stackrel{c}{\approx} \mathrm{piO}(C_0)$$

but for randomized C_0 and C_1. For which classes of circuits?

X-**IND Samplers**: There is a set X such that

$x \notin X$: Functional equivalence:

$$\forall r \; : \; C_0(x;r) = C_1(x;r)$$

$x \in X$: Computational indistinguishability:

$$C_0(x) \stackrel{c}{\approx}_{\varepsilon(\lambda)} C_1(x) \quad \text{and} \quad |X(\lambda)| \cdot \varepsilon(\lambda) \in \mathrm{Negl}$$

Construction: Sub-exp. secure iO + sub-exp. secure puncturable PRF [CLTV14].

Tool 3: Dual-Mode NIZK

NIZK with two CRS modes: “binding” BCRS and “hiding” HCRS

Tool 3: Dual-Mode NIZK

NIZK with two CRS modes: "binding" BCRS and "hiding" HCRS

1. CRS Indistinguishability: $\mathsf{BCRS} \overset{c}{\approx} \mathsf{HCRS}$

Tool 3: Dual-Mode NIZK

NIZK with two CRS modes: "binding" BCRS and "hiding" HCRS

1. CRS Indistinguishability: BCRS $\stackrel{c}{\approx}$ HCRS
2. Perfect Completeness: BCRS and HCRS

Tool 3: Dual-Mode NIZK

NIZK with two CRS modes: "binding" BCRS and "hiding" HCRS

1. CRS Indistinguishability: BCRS $\stackrel{c}{\approx}$ HCRS
2. Perfect Completeness: BCRS and HCRS
3. Perfect Soundness: BCRS

Tool 3: Dual-Mode NIZK

NIZK with two CRS modes: "binding" BCRS and "hiding" HCRS

1. CRS Indistinguishability: BCRS $\overset{c}{\approx}$ HCRS
2. Perfect Completeness: BCRS and HCRS
3. Perfect Soundness: BCRS
4. Perfect WI and ZK: HCRS

Tool 3: Dual-Mode NIZK

NIZK with two CRS modes: "binding" BCRS and "hiding" HCRS

1. CRS Indistinguishability: BCRS $\overset{c}{\approx}$ HCRS
2. Perfect Completeness: BCRS and HCRS
3. Perfect Soundness: BCRS
4. Perfect WI and ZK: HCRS
5. Perfect Extraction: $\mathsf{WExt}(td_{ext}, x, \pi)$ returns w_x under BCRS

Construction: Groth–Sahai Proofs.

The Construction – Intuition

Encode an exponent x as

$$[x] := (g^x, \mathsf{Enc}_1(x), \mathsf{Enc}_2(x), \pi) \ ,$$

analogously to the doubly-encrypt-and-prove Naor-Yung paradigm.

The Construction – Intuition

Encode an exponent x as

$$[x] := (g^x, \mathsf{Enc}_1(x), \mathsf{Enc}_2(x), \pi) \ ,$$

analogously to the doubly-encrypt-and-prove Naor-Yung paradigm.

Group operation:

$$\mathbf{Add}([x_1], [x_2]) := (g^{x_1+x_2}, \mathsf{Enc}_1(x_1 + x_2), \mathsf{Enc}_2(x_1 + x_2), \pi_+) \ .$$

The Construction – Intuition

Encode an exponent x as

$$[x] := (g^x, \mathsf{Enc}_1(x), \mathsf{Enc}_2(x), \pi) \ ,$$

analogously to the doubly-encrypt-and-prove Naor-Yung paradigm.

Group operation:

$$\mathbf{Add}([x_1],[x_2]) := (g^{x_1+x_2}, \mathsf{Enc}_1(x_1+x_2), \mathsf{Enc}_2(x_1+x_2), \pi_+) \ .$$

Multilinear map: Decrypt $[x_i]$ to get x_i and return

$$\mathbf{e}([x_1],\ldots,[x_\kappa]) := g^{x_1 \cdots x_\kappa} \ .$$

Setup

Group parameters:

$$\sigma,$$

Setup

Group parameters:

$$\sigma, \mathsf{TD}, y \notin \mathsf{TD},$$

Setup

Group parameters:

$$\sigma, \mathsf{TD}, y \notin \mathsf{TD}, pk_1, pk_2,$$

Setup

Group parameters:

$$\sigma, \mathsf{TD}, y \notin \mathsf{TD}, pk_1, pk_2, [\mathbf{W}],$$

Setup

Group parameters:

$$\sigma, \mathsf{TD}, y \notin \mathsf{TD}, pk_1, pk_2, [\mathbf{W}], \overline{C}_{\mathsf{Add}}, \overline{C}_{\mathsf{Map}}$$

Setup

Group parameters:

$$\sigma, \mathsf{TD}, y \notin \mathsf{TD}, pk_1, pk_2, [\mathbf{W}], \overline{C}_{\mathsf{Add}}, \overline{C}_{\mathsf{Map}}$$

- TD, a language with hard membership:

$$y \notin \mathsf{TD} \quad \overset{c}{\approx} \quad y \in \mathsf{TD}\ .$$

E.g.,

$$\{(g_1^r, g_2^r) : r \in \mathbb{Z}_p\} \subseteq \mathbb{G} \times \mathbb{G}$$

Setup

Group parameters:

$$\sigma, \mathsf{TD}, y \notin \mathsf{TD}, pk_1, pk_2, [\mathbf{W}], \overline{C}_{\mathsf{Add}}, \overline{C}_{\mathsf{Map}}$$

- TD, a language with hard membership:

$$y \notin \mathsf{TD} \quad \overset{c}{\approx} \quad y \in \mathsf{TD} \ .$$

  E.g.,

$$\{(g_1^r, g_2^r) : r \in \mathbb{Z}_p\} \subseteq \mathbb{G} \times \mathbb{G}$$

- A representation matrix

$$[\mathbf{W}] := \begin{bmatrix} [\mathbf{w}_1] & [\mathbf{w}_2] & \dots & [\mathbf{w}_\kappa] \end{bmatrix}^{\mathrm{t}} =$$

Setup

Group parameters:

$$\sigma, \mathsf{TD}, y \notin \mathsf{TD}, pk_1, pk_2, [\mathbf{W}], \overline{C}_{\mathsf{Add}}, \overline{C}_{\mathsf{Map}}$$

- TD, a language with hard membership:

$$y \notin \mathsf{TD} \quad \overset{c}{\approx} \quad y \in \mathsf{TD} \; .$$

  E.g.,

$$\{(g_1^r, g_2^r) : r \in \mathbb{Z}_p\} \subseteq \mathbb{G} \times \mathbb{G}$$

- A representation matrix

$$[\mathbf{W}] := \begin{bmatrix}[\mathbf{w}_1] & [\mathbf{w}_2] & \dots & [\mathbf{w}_\kappa]\end{bmatrix}^{\mathrm{t}} = \begin{bmatrix} g & g^w \\ g & g^w \\ \vdots & \vdots \\ g & g^w \end{bmatrix}$$

  So $\mathbf{w}_i = (1, w)$ for the **symmetric** setting here.

Group Elements

Encode z as:

$$\big(g^z, \mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{x}_2), \pi\big)$$

where $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{Z}_p^2$ are such that:

$$z = \langle \mathbf{x}_1, \mathbf{w}_1 \rangle = \langle \mathbf{x}_2, \mathbf{w}_2 \rangle$$

Group Elements

Encode z as:

$$\big(g^z, \mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{x}_2), \pi\big)$$

where $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{Z}_p^2$ are such that:

$$z = \langle \mathbf{x}_1, \mathbf{w}_1 \rangle = \langle \mathbf{x}_2, \mathbf{w}_2 \rangle$$

Example: canonical representation $\mathbf{x}_1 = \mathbf{x}_2 = (z, 0)$.

Group Elements

Encode z as:

$$\big(g^z, \mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{x}_2), \pi\big)$$

where $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{Z}_p^2$ are such that:

$$z = \langle \mathbf{x}_1, \mathbf{w}_1 \rangle = \langle \mathbf{x}_2, \mathbf{w}_2 \rangle$$

Example: canonical representation $\mathbf{x}_1 = \mathbf{x}_2 = (z, 0)$.

Proof π for

Encoding is valid **OR** $y \in \mathsf{TD}$

using witness $(\mathbf{x}_1, \mathbf{x}_2, r_1, r_2)$ for the **first** clause.

Algorithm $\mathbf{Add}[sk_1, sk_2, td_{ext}](h_1, h_2)$

Algorithm $\mathbf{Add}[sk_1, sk_2, td_{ext}](h_1, h_2)$

- Parse: $([z_1], \mathbf{c}_{1,1}, \mathbf{c}_{1,2}, \pi_1) \leftarrow h_1$ and $([z_2], \mathbf{c}_{2,1}, \mathbf{c}_{2,2}, \pi_2) \leftarrow h_2$.

Algorithm $\mathbf{Add}[sk_1, sk_2, td_{ext}](h_1, h_2)$

- Parse: $([z_1], \mathbf{c}_{1,1}, \mathbf{c}_{1,2}, \pi_1) \leftarrow h_1$ and $([z_2], \mathbf{c}_{2,1}, \mathbf{c}_{2,2}, \pi_2) \leftarrow h_2$.
- If π_1 or π_2 invalid abort. Else compute:

$$g^z \leftarrow g^{z_1} \cdot g^{z_2}; \qquad \mathbf{c}_1 \leftarrow \mathbf{c}_{1,1} + \mathbf{c}_{2,1}; \qquad \mathbf{c}_2 \leftarrow \mathbf{c}_{1,2} + \mathbf{c}_{2,2} \ .$$

Algorithm $\mathbf{Add}[sk_1, sk_2, td_{ext}](h_1, h_2)$

- Parse: $([z_1], \mathbf{c}_{1,1}, \mathbf{c}_{1,2}, \pi_1) \leftarrow h_1$ and $([z_2], \mathbf{c}_{2,1}, \mathbf{c}_{2,2}, \pi_2) \leftarrow h_2$.
- If π_1 or π_2 invalid abort. Else compute:

$$g^z \leftarrow g^{z_1} \cdot g^{z_2}; \qquad \mathbf{c}_1 \leftarrow \mathbf{c}_{1,1} + \mathbf{c}_{2,1}; \qquad \mathbf{c}_2 \leftarrow \mathbf{c}_{1,2} + \mathbf{c}_{2,2} \ .$$

- Check h_1 and h_2 for consistency using (sk_1, sk_2):

$$g^{z_1} = [z_1] = g^{\langle \mathbf{x}_{1,1}, \mathbf{w}_1 \rangle} \quad \text{and} \quad g^{z_1} = [z_1] = g^{\langle \mathbf{x}_{1,2}, \mathbf{w}_2 \rangle}$$
$$g^{z_2} = [z_2] = g^{\langle \mathbf{x}_{2,1}, \mathbf{w}_1 \rangle} \quad \text{and} \quad g^{z_2} = [z_2] = g^{\langle \mathbf{x}_{2,2}, \mathbf{w}_2 \rangle}$$

Algorithm $\mathbf{Add}[sk_1, sk_2, td_{ext}](h_1, h_2)$

- Parse: $([z_1], \mathbf{c}_{1,1}, \mathbf{c}_{1,2}, \pi_1) \leftarrow h_1$ and $([z_2], \mathbf{c}_{2,1}, \mathbf{c}_{2,2}, \pi_2) \leftarrow h_2$.
- If π_1 or π_2 invalid abort. Else compute:

$$g^z \leftarrow g^{z_1} \cdot g^{z_2}; \qquad \mathbf{c}_1 \leftarrow \mathbf{c}_{1,1} + \mathbf{c}_{2,1}; \qquad \mathbf{c}_2 \leftarrow \mathbf{c}_{1,2} + \mathbf{c}_{2,2} \ .$$

- Check h_1 and h_2 for consistency using (sk_1, sk_2):

$$\begin{aligned} g^{z_1} = [z_1] = g^{\langle \mathbf{x}_{1,1}, \mathbf{w}_1 \rangle} \quad &\text{and} \quad g^{z_1} = [z_1] = g^{\langle \mathbf{x}_{1,2}, \mathbf{w}_2 \rangle} \\ g^{z_2} = [z_2] = g^{\langle \mathbf{x}_{2,1}, \mathbf{w}_1 \rangle} \quad &\text{and} \quad g^{z_2} = [z_2] = g^{\langle \mathbf{x}_{2,2}, \mathbf{w}_2 \rangle} \end{aligned}$$

- If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.

Algorithm $\mathbf{Add}[sk_1, sk_2, td_{ext}](h_1, h_2)$

- Parse: $([z_1], \mathbf{c}_{1,1}, \mathbf{c}_{1,2}, \pi_1) \leftarrow h_1$ and $([z_2], \mathbf{c}_{2,1}, \mathbf{c}_{2,2}, \pi_2) \leftarrow h_2$.
- If π_1 or π_2 invalid abort. Else compute:

$$g^z \leftarrow g^{z_1} \cdot g^{z_2}; \qquad \mathbf{c}_1 \leftarrow \mathbf{c}_{1,1} + \mathbf{c}_{2,1}; \qquad \mathbf{c}_2 \leftarrow \mathbf{c}_{1,2} + \mathbf{c}_{2,2} \ .$$

- Check h_1 and h_2 for consistency using (sk_1, sk_2):

$$\begin{aligned} g^{z_1} = [z_1] = g^{\langle \mathbf{x}_{1,1}, \mathbf{w}_1 \rangle} \quad &\text{and} \quad g^{z_1} = [z_1] = g^{\langle \mathbf{x}_{1,2}, \mathbf{w}_2 \rangle} \\ g^{z_2} = [z_2] = g^{\langle \mathbf{x}_{2,1}, \mathbf{w}_1 \rangle} \quad &\text{and} \quad g^{z_2} = [z_2] = g^{\langle \mathbf{x}_{2,2}, \mathbf{w}_2 \rangle} \end{aligned}$$

- If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.
- Else: use td_{ext} to extract $w'_{i,y}$ from π_i for $y \in \text{TD}$. Generate π.

Algorithm $\mathbf{Add}[sk_1, sk_2, td_{ext}](h_1, h_2)$

- Parse: $([z_1], \mathbf{c}_{1,1}, \mathbf{c}_{1,2}, \pi_1) \leftarrow h_1$ and $([z_2], \mathbf{c}_{2,1}, \mathbf{c}_{2,2}, \pi_2) \leftarrow h_2$.
- If π_1 or π_2 invalid abort. Else compute:

$$g^z \leftarrow g^{z_1} \cdot g^{z_2}; \qquad \mathbf{c}_1 \leftarrow \mathbf{c}_{1,1} + \mathbf{c}_{2,1}; \qquad \mathbf{c}_2 \leftarrow \mathbf{c}_{1,2} + \mathbf{c}_{2,2} \ .$$

- Check h_1 and h_2 for consistency using (sk_1, sk_2):

$$\begin{aligned} g^{z_1} &= [z_1] = g^{\langle \mathbf{x}_{1,1}, \mathbf{w}_1 \rangle} \quad \text{and} \quad g^{z_1} = [z_1] = g^{\langle \mathbf{x}_{1,2}, \mathbf{w}_2 \rangle} \\ g^{z_2} &= [z_2] = g^{\langle \mathbf{x}_{2,1}, \mathbf{w}_1 \rangle} \quad \text{and} \quad g^{z_2} = [z_2] = g^{\langle \mathbf{x}_{2,2}, \mathbf{w}_2 \rangle} \end{aligned}$$

- If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.
- Else: use td_{ext} to extract $w'_{i,y}$ from π_i for $y \in$ TD. Generate π.
- Else (both extractions fail), abort.

Algorithm $\mathbf{Add}[sk_1, sk_2, td_{ext}](h_1, h_2)$

- Parse: $([z_1], \mathbf{c}_{1,1}, \mathbf{c}_{1,2}, \pi_1) \leftarrow h_1$ and $([z_2], \mathbf{c}_{2,1}, \mathbf{c}_{2,2}, \pi_2) \leftarrow h_2$.
- If π_1 or π_2 invalid abort. Else compute:

$$g^z \leftarrow g^{z_1} \cdot g^{z_2}; \qquad \mathbf{c}_1 \leftarrow \mathbf{c}_{1,1} + \mathbf{c}_{2,1}; \qquad \mathbf{c}_2 \leftarrow \mathbf{c}_{1,2} + \mathbf{c}_{2,2} \ .$$

- Check h_1 and h_2 for consistency using (sk_1, sk_2):

$$\begin{aligned} g^{z_1} = [z_1] = g^{\langle \mathbf{x}_{1,1}, \mathbf{w}_1 \rangle} \quad &\text{and} \quad g^{z_1} = [z_1] = g^{\langle \mathbf{x}_{1,2}, \mathbf{w}_2 \rangle} \\ g^{z_2} = [z_2] = g^{\langle \mathbf{x}_{2,1}, \mathbf{w}_1 \rangle} \quad &\text{and} \quad g^{z_2} = [z_2] = g^{\langle \mathbf{x}_{2,2}, \mathbf{w}_2 \rangle} \end{aligned}$$

- If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.
- Else: use td_{ext} to extract $w'_{i,y}$ from π_i for $y \in$ TD. Generate π.
- Else (both extractions fail), abort.

A piO of $\mathbf{Add}[sk_1, sk_2, td_{ext}]$ will be made public.

Algorithm $\mathrm{e}[sk_1](h_1, \ldots, h_\kappa)$

- For $i = 1 \ldots \kappa$:
-     Parse $([z_i], \mathbf{c}_{i,1}, \mathbf{c}_{i,2}, \pi_i) \leftarrow h_i$
-     Check π_i for validity
-     Compute $\mathbf{x}_i \leftarrow \mathsf{Dec}(sk_1, \mathbf{c}_{i,1})$
- $z \leftarrow \langle \mathbf{x}_1, \mathbf{w}_1 \rangle \cdots \langle \mathbf{x}_\kappa, \mathbf{w}_\kappa \rangle \pmod{p}$
- Return g^z

Algorithm $\mathbf{e}[sk_1](h_1, \ldots, h_\kappa)$

- For $i = 1 \ldots \kappa$:
-     Parse $([z_i], \mathbf{c}_{i,1}, \mathbf{c}_{i,2}, \pi_i) \leftarrow h_i$
-     Check π_i for validity
-     Compute $\mathbf{x}_i \leftarrow \mathsf{Dec}(sk_1, \mathbf{c}_{i,1})$
- $z \leftarrow \langle \mathbf{x}_1, \mathbf{w}_1 \rangle \cdots \langle \mathbf{x}_\kappa, \mathbf{w}_\kappa \rangle \pmod{p}$
- Return g^z

Note:

- Map $\mathbf{e}$ is multilinear.

Algorithm $\mathbf{e}[sk_1](h_1, \ldots, h_\kappa)$

- For $i = 1 \ldots \kappa$:
- Parse $([z_i], \mathbf{c}_{i,1}, \mathbf{c}_{i,2}, \pi_i) \leftarrow h_i$
- Check π_i for validity
- Compute $\mathbf{x}_i \leftarrow \mathrm{Dec}(sk_1, \mathbf{c}_{i,1})$
- $z \leftarrow \langle \mathbf{x}_1, \mathbf{w}_1 \rangle \cdots \langle \mathbf{x}_\kappa, \mathbf{w}_\kappa \rangle \pmod{p}$
- Return g^z

Note:

- Map $\mathbf{e}$ is multilinear.
- A piO of $\mathbf{e}[sk_1]$ will be made public.

A Proof Gadget: Forgetting (sk_1, sk_2, td_{ext}) for w_y

Goal (1):

$$\mathbf{Add}[sk_1, sk_2, td_{ext}](\cdot, \cdot) \longrightarrow \mathbf{Add}[w_y](\cdot, \cdot)$$

A Proof Gadget: Forgetting (sk_1, sk_2, td_{ext}) for w_y

Goal (1):

$$\mathbf{Add}[sk_1, sk_2, td_{ext}](\cdot,\cdot) \longrightarrow \mathbf{Add}[w_y](\cdot,\cdot)$$

(1) If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.

(2) Else: use td_{ext} to extract $w'_{i,y}$ from π_i for $y \in \mathrm{TD}$. Generate π.

A Proof Gadget: Forgetting (sk_1, sk_2, td_{ext}) for w_y

Goal (1):

$$\mathbf{Add}[sk_1, sk_2, td_{ext}](\cdot,\cdot) \longrightarrow \mathbf{Add}[w_y](\cdot,\cdot)$$

(1) If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.

(2) Else: use td_{ext} to extract $w'_{i,y}$ from π_i for $y \in$ TD. Generate π.

Proof:

G	σ	y	C_{Add} knows	π-witness	**Remark**
0	binding	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2 or w'_y	

A Proof Gadget: Forgetting (sk_1, sk_2, td_{ext}) for w_y

Goal (1):

$$\mathbf{Add}[sk_1, sk_2, td_{ext}](\cdot,\cdot) \longrightarrow \mathbf{Add}[w_y](\cdot,\cdot)$$

(1) If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.

(2) Else: use td_{ext} to extract $w'_{i,y}$ from π_i for $y \in$ TD. Generate π.

Proof:

G	σ	y	C_{Add} knows	π-witness	**Remark**
0	binding	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2 or w'_y	
1	binding	$\boxed{\in \text{TD}}$	sk_1, sk_2, td_{ext}	sk_1, sk_2 or w'_y	TD indist.

A Proof Gadget: Forgetting (sk_1, sk_2, td_{ext}) for w_y

Goal (1):

$$\mathbf{Add}[sk_1, sk_2, td_{ext}](\cdot,\cdot) \longrightarrow \mathbf{Add}[w_y](\cdot,\cdot)$$

(1) If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.

(2) Else: use td_{ext} to extract $w'_{i,y}$ from π_i for $y \in$ TD. Generate π.

Proof:

G	σ	y	C_{Add} knows	π-witness	**Remark**
0	binding	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2 or w'_y	
1	binding	$\boxed{\in \text{TD}}$	sk_1, sk_2, td_{ext}	sk_1, sk_2 or w'_y	TD indist.
2	binding	$\in$ TD	$\boxed{sk_1, sk_2, w_y}$	sk_1, sk_2 or w_y (2)	piO (unique w_y + prefect soundness)

A Proof Gadget: Forgetting (sk_1, sk_2, td_{ext}) for w_y

Goal (1):

$$\mathbf{Add}[sk_1, sk_2, td_{ext}](\cdot,\cdot) \longrightarrow \mathbf{Add}[w_y](\cdot,\cdot)$$

(1) If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.

(2) Else: use td_{ext} to extract $w'_{i,y}$ from π_i for $y \in$ TD. Generate π.

Proof:

G	σ	y	C_{Add} knows	π-witness	**Remark**
0	binding	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2 or w'_y	
1	binding	$\boxed{\in \text{TD}}$	sk_1, sk_2, td_{ext}	sk_1, sk_2 or w'_y	TD indist.
2	binding	$\in$ TD	$\boxed{sk_1, sk_2, w_y}$	sk_1, sk_2 or w_y (2)	piO (unique w_y + prefect soundness)
3	$\boxed{\text{hiding}}$	$\in$ TD	sk_1, sk_2, w_y	sk_1, sk_2 or w_y (2)	CRS indist.

A Proof Gadget: Forgetting (sk_1, sk_2, td_{ext}) for w_y

Goal (1):

$$\mathbf{Add}[sk_1, sk_2, td_{ext}](\cdot,\cdot) \longrightarrow \mathbf{Add}[w_y](\cdot,\cdot)$$

(1) If OK: proof π using (sk_1, sk_2) that $(g^z, \mathbf{c}_1, \mathbf{c}_2)$ is valid.

(2) Else: use td_{ext} to extract $w'_{i,y}$ from π_i for $y \in$ TD. Generate π.

Proof:

G	σ	y	C_{Add} knows	π-witness	**Remark**
0	binding	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2 or w'_y	
1	binding	$\boxed{\in \text{TD}}$	sk_1, sk_2, td_{ext}	sk_1, sk_2 or w'_y	TD indist.
2	binding	$\in$ TD	$\boxed{sk_1, sk_2, w_y}$	sk_1, sk_2 or w_y (2)	piO (unique w_y + prefect soundness)
3	$\boxed{\text{hiding}}$	$\in$ TD	sk_1, sk_2, w_y	sk_1, sk_2 or w_y (2)	CRS indist.
4	hiding	$\in$ TD	$\boxed{w_y}$	$\boxed{w_y \text{ (1) + (2)}}$	piO + perfect WI

□

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \;\longrightarrow\; \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	iO (soundness)

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	iO (soundness)
5	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_2	$\mathbf{x}_0$	$\mathbf{y}_1$	

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	iO (soundness)
5	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_2	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	iO (soundness)
5	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_2	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
6	hide	$\in$ TD	w_y	w_y	sk_2	$\boxed{\mathbf{x}_1}$	$\mathbf{y}_1$	

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	iO (soundness)
5	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_2	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
6	hide	$\in$ TD	w_y	w_y	sk_2	$\boxed{\mathbf{x}_1}$	$\mathbf{y}_1$	IND-CPA wrt. pk_1

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	iO (soundness)
5	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_2	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
6	hide	$\in$ TD	w_y	w_y	sk_2	$\boxed{\mathbf{x}_1}$	$\mathbf{y}_1$	IND-CPA wrt. pk_1
7	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_2	$\mathbf{x}_1$	$\mathbf{y}_1$	

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	iO (soundness)
5	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_2	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
6	hide	$\in$ TD	w_y	w_y	sk_2	$\boxed{\mathbf{x}_1}$	$\mathbf{y}_1$	IND-CPA wrt. pk_1
7	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_2	$\mathbf{x}_1$	$\mathbf{y}_1$	proof gadget

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	iO (soundness)
5	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_2	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
6	hide	$\in$ TD	w_y	w_y	sk_2	$\boxed{\mathbf{x}_1}$	$\mathbf{y}_1$	IND-CPA wrt. pk_1
7	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_2	$\mathbf{x}_1$	$\mathbf{y}_1$	proof gadget
8	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_1}$	$\mathbf{x}_1$	$\mathbf{y}_1$	

Switching Encodings

Goal (2):

$$\big(\mathsf{Enc}(pk_1, \mathbf{x}_0), \mathsf{Enc}(pk_2, \mathbf{y}_0)\big) \longrightarrow \big(\mathsf{Enc}(pk_1, \mathbf{x}_1), \mathsf{Enc}(pk_2, \mathbf{y}_1)\big)$$

Proof:

G	σ	y	C_{Add} knows	π-witness	C_{Map} knows	$\mathbf{c}_1$ contains	$\mathbf{c}_2$ contains	**Remark**
0	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	
1	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_0$	proof gadget
2	hide	$\in$ TD	w_y	w_y	sk_1	$\mathbf{x}_0$	$\boxed{\mathbf{y}_1}$	IND-CPA wrt. pk_2
3	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_1	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
4	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_2}$	$\mathbf{x}_0$	$\mathbf{y}_1$	iO (soundness)
5	$\boxed{\text{hide}}$	$\boxed{\in \text{TD}}$	$\boxed{w_y}$	$\boxed{w_y}$	sk_2	$\mathbf{x}_0$	$\mathbf{y}_1$	proof gadget
6	hide	$\in$ TD	w_y	w_y	sk_2	$\boxed{\mathbf{x}_1}$	$\mathbf{y}_1$	IND-CPA wrt. pk_1
7	$\boxed{\text{bind}}$	$\boxed{\notin \text{TD}}$	$\boxed{sk_1, sk_2, td_{ext}}$	$\boxed{sk_1, sk_2}$	sk_2	$\mathbf{x}_1$	$\mathbf{y}_1$	proof gadget
8	bind	$\notin$ TD	sk_1, sk_2, td_{ext}	sk_1, sk_2	$\boxed{sk_1}$	$\mathbf{x}_1$	$\mathbf{y}_1$	iO (soundness)

$\square$

Computing $\mathbf{e}$ with Implicit $[\mathbf{W}]$

Recall

$$\mathbf{e}([z_1], \ldots, [z_\kappa]) = g^{\langle \mathbf{x}_1, \mathbf{w}_1 \rangle \cdots \langle \mathbf{x}_\kappa, \mathbf{w}_\kappa \rangle}$$

Computing $\mathbf{e}$ with Implicit $[\mathbf{W}]$

Recall

$$\mathbf{e}([z_1], \dots, [z_\kappa]) = g^{\langle \mathbf{x}_1, \mathbf{w}_1 \rangle \cdots \langle \mathbf{x}_\kappa, \mathbf{w}_\kappa \rangle}$$

This can be computed using $(\mathbf{x}_1, \dots, \mathbf{x}_\kappa)$ and **implicit** values

$$g, g^{w}, g^{w^2}, \dots, g^{w^\kappa} \ .$$

Computing $\mathbf{e}$ with Implicit $[\mathbf{W}]$

Recall

$$\mathbf{e}([z_1], \ldots, [z_\kappa]) = g^{\langle \mathbf{x}_1, \mathbf{w}_1 \rangle \cdots \langle \mathbf{x}_\kappa, \mathbf{w}_\kappa \rangle}$$

This can be computed using $(\mathbf{x}_1, \ldots, \mathbf{x}_\kappa)$ and **implicit** values

$$g, g^{w}, g^{w^2}, \ldots, g^{w^\kappa} \ .$$

How?

Computing $\mathbf{e}$ with Implicit $[\mathbf{W}]$

Recall

$$\mathbf{e}([z_1], \ldots, [z_\kappa]) = g^{\langle \mathbf{x}_1, \mathbf{w}_1 \rangle \cdots \langle \mathbf{x}_\kappa, \mathbf{w}_\kappa \rangle}$$

This can be computed using $(\mathbf{x}_1, \ldots, \mathbf{x}_\kappa)$ and **implicit** values

$$g, g^w, g^{w^2}, \ldots, g^{w^\kappa} \ .$$

How? Recall $\mathbf{w}_i = (1, w)$

$$\prod_{j=1}^{\kappa} \langle \mathbf{x}_i, \mathbf{w}_i \rangle = \prod_{j=1}^{\kappa} (x_{i,0} + x_{i,1} \cdot w) = \sum_{j=1}^{\kappa} \alpha_j w^j$$

Computing $\mathbf{e}$ with Implicit $[\mathbf{W}]$

Recall

$$\mathbf{e}([z_1], \ldots, [z_\kappa]) = g^{\langle \mathbf{x}_1, \mathbf{w}_1 \rangle \cdots \langle \mathbf{x}_\kappa, \mathbf{w}_\kappa \rangle}$$

This can be computed using $(\mathbf{x}_1, \ldots, \mathbf{x}_\kappa)$ and **implicit** values

$$g, g^w, g^{w^2}, \ldots, g^{w^\kappa} \ .$$

How? Recall $\mathbf{w}_i = (1, w)$

$$\prod_{j=1}^{\kappa} \langle \mathbf{x}_i, \mathbf{w}_i \rangle = \prod_{j=1}^{\kappa} (x_{i,0} + x_{i,1} \cdot w) = \sum_{j=1}^{\kappa} \alpha_j w^j$$

All we need are the coefficients α_j. These we can compute iteratively.

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^{r}$$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \stackrel{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^{r}$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \stackrel{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \stackrel{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \stackrel{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa}(a_i + w)$	w	

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^{r}$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa}(a_i + w)$	w	Renaming

□

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa}(a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{\mathrm{r}} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{\mathrm{r}} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	$(\kappa - 1)$-SDDH

□

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \stackrel{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{r} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$\boxed{(a_{\kappa+1} - w, 0)}$	$a_{\kappa+1}(r + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$g^{w^\kappa} \to g^r$	

□

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{r} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$\boxed{(a_{\kappa+1} - w, 0)}$	$a_{\kappa+1}(r + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$g^{w^\kappa} \to g^r$	Encoding switch

□

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{\mathrm{r}} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$\boxed{(a_{\kappa+1} - w, 0)}$	$a_{\kappa+1}(r + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$g^{w^\kappa} \to g^r$	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\sum_{j=1}^{\kappa} \beta_j w^j$	$g^{w^\kappa} \to g^r$	

□

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{r} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$\boxed{(a_{\kappa+1} - w, 0)}$	$a_{\kappa+1}(r + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$g^{w^\kappa} \to g^r$	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\sum_{j=1}^{\kappa} \beta_j w^j$	$g^{w^\kappa} \to g^r$	Renaming

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{r} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$\boxed{(a_{\kappa+1} - w, 0)}$	$a_{\kappa+1}(r + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$g^{w^\kappa} \to g^r$	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\sum_{j=1}^{\kappa} \beta_j w^j$	$g^{w^\kappa} \to g^r$	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\boxed{s + \sum_{j=1}^{\kappa-1} \beta_j w^j}$	$g^{w^\kappa} \to g^r$	

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{r} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$\boxed{(a_{\kappa+1} - w, 0)}$	$a_{\kappa+1}(r + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$g^{w^\kappa} \to g^r$	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\sum_{j=1}^{\kappa} \beta_j w^j$	$g^{w^\kappa} \to g^r$	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\boxed{s + \sum_{j=1}^{\kappa-1} \beta_j w^j}$	$g^{w^\kappa} \to g^r$	$(\kappa - 1)$-SDDH

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{r} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$\boxed{(a_{\kappa+1} - w, 0)}$	$a_{\kappa+1}(r + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$g^{w^\kappa} \to g^r$	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\sum_{j=1}^{\kappa} \beta_j w^j$	$g^{w^\kappa} \to g^r$	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\boxed{s + \sum_{j=1}^{\kappa-1} \beta_j w^j}$	$g^{w^\kappa} \to g^r$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	s	$g^{w^\kappa} \to g^r$	

$\square$

Main Proof: Hardness of Symmetric κ-MDDH

Final Goal:

$$g^{a_1 \cdots a_{\kappa+1}} \overset{c}{\approx} g^r$$

Proof:

$[a_1]$	$[a_2]$	$\cdots$	$[a_\kappa]$	$[a_{\kappa+1}]$	MDDH exponent	C_{Map}	**Remark**
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	MDDH ($b = 1$)
$\boxed{(a_1 - w, 0)}$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1 - w, 0)$	$\boxed{(a_2 - w, 0)}$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_1 - w, 0)$	$(a_2 - w, 0)$	$\cdots$	$\boxed{(a_\kappa - w, 0)}$	$(a_{\kappa+1}, 0)$	$a_1 \cdots a_{\kappa+1}$	w	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \prod_{i=1}^{\kappa} (a_i + w)$	w	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1} \sum_{j=1}^{\kappa} \alpha_j w^j$	$\boxed{[w^j] = g^{w^j}}$	iO
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$a_{\kappa+1}(\boxed{\mathrm{r}} + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$\boxed{g^{w^\kappa} \to g^r}$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$\boxed{(a_{\kappa+1} - w, 0)}$	$a_{\kappa+1}(r + \sum_{j=1}^{\kappa-1} \alpha_j w^j)$	$g^{w^\kappa} \to g^r$	Encoding switch
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\sum_{j=1}^{\kappa} \beta_j w^j$	$g^{w^\kappa} \to g^r$	Renaming
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	$\boxed{s + \sum_{j=1}^{\kappa-1} \beta_j w^j}$	$g^{w^\kappa} \to g^r$	$(\kappa - 1)$-SDDH
$(a_1, 0)$	$(a_2, 0)$	$\cdots$	$(a_\kappa, 0)$	$(a_{\kappa+1}, 0)$	s	$g^{w^\kappa} \to g^r$	Renaming

$\square$

Extensions and Open Problems

See paper (ePrint 2015/780) for:

- **A**symmetric maps down to (plain) DDH.
- Generalization showing hardness of the RANK problem.

Extensions and Open Problems

See paper (ePrint 2015/780) for:

- **A**symmetric maps down to (plain) DDH.
- Generalization showing hardness of the RANK problem.

Avenues to explore:

- Can we get **graded** maps? For all $i + j \leq \kappa$:

$$\mathbf{e} : \mathbb{G}_i \times \mathbb{G}_j \longrightarrow \mathbb{G}_{i+j}$$
$$\mathbf{e}(g_i^{a_i}, g_j^{a_j}) \; = \; g_{i+j}^{a_i a_j}$$

- What are the minimal assumptions needed for (iO $\implies$ MLM)? One-way functions?
- What's the "maximal multilinear hardness" that we prove?

Extensions and Open Problems

See paper (ePrint 2015/780) for:

- **A**symmetric maps down to (plain) DDH.
- Generalization showing hardness of the RANK problem.

Avenues to explore:

- Can we get **graded** maps? For all $i + j \leq \kappa$:

$$\mathbf{e} : \mathbb{G}_i \times \mathbb{G}_j \longrightarrow \mathbb{G}_{i+j}$$
$$\mathbf{e}(g_i^{a_i}, g_j^{a_j}) \;=\; g_{i+j}^{a_i a_j}$$

- What are the minimal assumptions needed for (iO $\implies$ MLM)? One-way functions?
- What's the "maximal multilinear hardness" that we prove?

Thank you.