DMCFE for Inner Products with Strong Security

Ky Nguyen David Pointcheval **Robert Schädlich**

January 24, 2024

DIENS, École normale supérieure, PSL University, CNRS, Inria, Paris, France

Functional Encryption (FE)

$$\begin{aligned}
\mathbf{Setup}(1^\lambda) &\to \mathsf{msk}\\
\mathbf{Enc}(\mathsf{msk}, x) &\to \mathsf{ct}\\
\mathbf{KeyGen}(\mathsf{msk}, f) &\to \mathsf{dk}\\
\mathbf{Dec}(\mathsf{dk}, \mathsf{ct}) &\to f(x)
\end{aligned}$$

$$\begin{aligned} \mathbf{Setup}(1^\lambda) &\to \mathsf{msk} \\ \mathbf{Enc}(\mathsf{msk}, x) &\to \mathsf{ct} \\ \mathbf{KeyGen}(\mathsf{msk}, f) &\to \mathsf{dk} \\ \mathbf{Dec}(\mathsf{dk}, \mathsf{ct}) &\to f(x) \end{aligned}$$

Public-Key vs Secret-Key FE.

- Public-key FE provides unified framework for other encryption primitives, e.g. PKE, IBE, ABE etc.
- Secret-key FE allows stronger function-hiding security

Functional Encryption for Inner Products (IPFE)

$$\begin{aligned} \mathbf{Setup}(1^\lambda) &\to \mathsf{msk} \\ \mathbf{Enc}(\mathsf{msk}, x) &\to \mathsf{ct} \\ \mathbf{KeyGen}(\mathsf{msk}, y) &\to \mathsf{dk} \\ \mathbf{Dec}(\mathsf{dk}, \mathsf{ct}) &\to \langle x, y \rangle \end{aligned}$$

Public-Key vs Secret-Key FE.

- Public-key FE provides unified framework for other encryption primitives, e.g. PKE, IBE, ABE etc.
- Secret-key FE allows stronger function-hiding security

Extension 1: Multiple Encryptors (MCFE)

$$\mathsf{Setup}(1^\lambda) \to (\mathsf{msk}, \mathsf{ek}_1, \ldots, \mathsf{ek}_n)$$

$$\mathsf{Enc}(\mathsf{ek}_i, \quad , \mathbf{x}_i) \to \mathsf{ct}_i$$

$$\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y}) \to \mathsf{dk}$$

$$\mathsf{Dec}(\mathsf{dk}, \{\mathsf{ct}_i\}_{i \in [n]}) \to \sum_{i \in [n]} \langle \mathbf{x}_i, \mathbf{y}_i \rangle$$

Notation. $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$

- multiple clients each encrypting a share of the data
  - no interaction
  - no synchronization
  - possible corruptions

Extension 1: Multiple Encryptors (MCFE)

$$\begin{aligned}
\mathsf{Setup}(1^\lambda) &\to (\mathsf{msk}, \mathsf{ek}_1, \ldots, \mathsf{ek}_n) \\
\mathsf{Enc}(\mathsf{ek}_i, \mathsf{lab}, \mathbf{x}_i) &\to \mathsf{ct}_{\mathsf{lab},i} \\
\mathsf{KeyGen}(\mathsf{msk}, \mathbf{y}) &\to \mathsf{dk} \\
\mathsf{Dec}(\mathsf{dk}, \{\mathsf{ct}_{\mathsf{lab},i}\}_{i\in[n]}) &\to \sum_{i\in[n]} \langle \mathbf{x}_i, \mathbf{y}_i \rangle
\end{aligned}$$

Notation. $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$

- multiple clients each encrypting a share of the data
  - no interaction
  - no synchronization
  - possible corruptions
- labels to reduce data leakage

Extension 2: Multiple Key Generators (DMCFE)

$$\begin{aligned}
\mathbf{Setup}(1^\lambda) &\to (\mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{ek}_1, \ldots, \mathsf{ek}_n) \\
\mathbf{Enc}(\mathsf{ek}_i, \mathsf{lab}, \mathbf{x}_i) &\to \mathsf{ct}_{\mathsf{lab},i} \\
\mathbf{KeyGen}(\mathsf{sk}_i, \mathsf{lab}', \mathbf{y}_i) &\to \mathsf{dk}_{\mathsf{lab}',i} \\
\mathbf{Dec}(\{\mathsf{dk}_{\mathsf{lab}',i}\}_{i\in[n]}, \{\mathsf{ct}_{\mathsf{lab},i}\}_{i\in[n]}) &\to \sum_{i\in[n]} \langle \mathbf{x}_i, \mathbf{y}_i \rangle
\end{aligned}$$

- multiple key generators each providing a decryption key
    - no interaction
    - no synchronization
    - possible corruptions
- labels to reduce data leakage

Function-Hiding Security (Message + Function Privacy)

$$b \xleftarrow{\$} \{0,1\}; (\mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{ek}_1, \ldots, \mathsf{ek}_n) \leftarrow \mathbf{Setup}(1^\lambda)$$
$$b' \leftarrow \mathcal{A}^{\mathsf{QEnc}, \mathsf{QKeyGen}, \mathsf{QCorrupt}}(1^\lambda)$$

$\mathbf{QEnc}(i, \mathsf{lab}, \mathbf{x}_i^{(0)}, \mathbf{x}_i^{(1)})$.

Return $\mathsf{ct}_{\mathsf{lab},i} \leftarrow \mathbf{Enc}(\mathsf{ek}_i, \mathsf{lab}, \mathbf{x}_i^{(b)})$

$\mathbf{QKeyGen}(i, \mathsf{lab}', \mathbf{y}_i^{(0)}, \mathbf{y}_i^{(1)})$.

Return $\mathsf{dk}_{\mathsf{lab}',i} \leftarrow \mathbf{DKeyGen}(\mathsf{sk}_i, \mathsf{lab}', \mathbf{y}_i^{(b)})$

$\mathbf{QCorrupt}(i)$.

Return $(\mathsf{sk}_i, \mathsf{ek}_i)$[1]

[1] This definition follows [CDGPP18]; see [NPP23] for separated corruptions.

Admissibility for DMCFE

Admissibility of $\mathcal{A}$ (Without Corruptions).

For all lab, lab′ and for all queries $\mathsf{QEnc}(i, \mathsf{lab}, \mathsf{x}^{(0)}_{\mathsf{lab},i}, \mathsf{x}^{(1)}_{\mathsf{lab},i})$ and $\mathsf{QKeyGen}(i, \mathsf{lab}', \mathsf{y}^{(0)}_{\mathsf{lab}',i}, \mathsf{y}^{(1)}_{\mathsf{lab}',i})$, it holds

$$\underbrace{\sum_{i\in[n]} \langle \mathsf{x}^{(0)}_{\mathsf{lab},i}, \mathsf{y}^{(0)}_{\mathsf{lab}',i} \rangle}_{\langle \mathsf{x}^{(0)}_{\mathsf{lab},1} \| \cdots \| \mathsf{x}^{(0)}_{\mathsf{lab},n}, \mathsf{y}^{(0)}_{\mathsf{lab}',1} \| \cdots \| \mathsf{y}^{(0)}_{\mathsf{lab}',n} \rangle} = \underbrace{\sum_{i\in[n]} \langle \mathsf{x}^{(1)}_{\mathsf{lab},i}, \mathsf{y}^{(1)}_{\mathsf{lab}',i} \rangle}_{\langle \mathsf{x}^{(1)}_{\mathsf{lab},1} \| \cdots \| \mathsf{x}^{(1)}_{\mathsf{lab},n}, \mathsf{y}^{(1)}_{\mathsf{lab}',1} \| \cdots \| \mathsf{y}^{(1)}_{\mathsf{lab}',n} \rangle} .$$

Admissibility of $\mathcal{A}$ (Without Corruptions).
For all lab, lab′ and for all queries $\mathsf{QEnc}(i, \mathrm{lab}, \mathbf{x}^{(0)}_{\mathrm{lab},i}, \mathbf{x}^{(1)}_{\mathrm{lab},i})$ and $\mathsf{QKeyGen}(i, \mathrm{lab}', \mathbf{y}^{(0)}_{\mathrm{lab}',i}, \mathbf{y}^{(1)}_{\mathrm{lab}',i})$, it holds

$$\underbrace{\sum_{i\in[n]} \langle \mathbf{x}^{(0)}_{\mathrm{lab},i}, \mathbf{y}^{(0)}_{\mathrm{lab}',i} \rangle}_{\langle \mathbf{x}^{(0)}_{\mathrm{lab},1} \| \cdots \| \mathbf{x}^{(0)}_{\mathrm{lab},n}, \mathbf{y}^{(0)}_{\mathrm{lab}',1} \| \cdots \| \mathbf{y}^{(0)}_{\mathrm{lab}',n} \rangle} = \underbrace{\sum_{i\in[n]} \langle \mathbf{x}^{(1)}_{\mathrm{lab},i}, \mathbf{y}^{(1)}_{\mathrm{lab}',i} \rangle}_{\langle \mathbf{x}^{(1)}_{\mathrm{lab},1} \| \cdots \| \mathbf{x}^{(1)}_{\mathrm{lab},n}, \mathbf{y}^{(1)}_{\mathrm{lab}',1} \| \cdots \| \mathbf{y}^{(1)}_{\mathrm{lab}',n} \rangle} .$$

Admissibility of $\mathcal{A}$ (With Corruptions).

1. For all corrupted clients i, $\mathbf{x}^{(0)}_{\mathrm{lab},i} = \mathbf{x}^{(1)}_{\mathrm{lab},i}$ and $\mathbf{y}^{(0)}_{\mathrm{lab}',i} = \mathbf{y}^{(1)}_{\mathrm{lab}',i}$
2. For all lab, lab′ and for all queries $\mathsf{QEnc}(i, \mathrm{lab}, \mathbf{x}^{(0)}_{\mathrm{lab},i}, \mathbf{x}^{(1)}_{\mathrm{lab},i})$ and $\mathsf{QKeyGen}(i, \mathrm{lab}', \mathbf{y}^{(0)}_{\mathrm{lab}',i}, \mathbf{y}^{(1)}_{\mathrm{lab}',i})$, it holds

$$\sum_{i \text{ honest}} \langle \mathbf{x}^{(0)}_{\mathrm{lab},i}, \mathbf{y}^{(0)}_{\mathrm{lab}',i} \rangle = \sum_{i \text{ honest}} \langle \mathbf{x}^{(1)}_{\mathrm{lab},i}, \mathbf{y}^{(1)}_{\mathrm{lab}',i} \rangle .$$

Contributions

[AGT21][2] (Generic from FH-IPFE[3]).

- Selective security, static corruptions
- No repetitions for **QKeyGen** queries

Our Construction 1 (Generic from FH-IPFE).

- Selective security, static corruptions
- Unbounded repetitions for **QKeyGen** queries

Our Construction 2 (Based on DPVS).

- Adaptive security, static corruptions
- Poly-bounded repetitions for **QKeyGen** queries

[2] In fact, this work constructs function-hiding DDFE for inner products.
[3] [Lin17] FH-IPFE exists under the SXDH assumption on pairings.

Selective FH-IP-DMCFE from FH-IPFE

$\mathsf{Setup}(1^\lambda)$: $s_1, \ldots, s_n \xleftarrow{\$} \mathbb{Z}_q$ s.t. $\sum_{i \in [n]} s_i = 0$;
for all $i \in [n]$: $\mathsf{imsk}_i \leftarrow \mathbf{iSetup}(1^\lambda)$,
$\mathsf{ek}_i = (\mathsf{imsk}_i, s_i)$ and $\mathsf{sk}_i = \mathsf{imsk}_i$

$\mathsf{KeyGen}(\mathsf{sk}_i, \mathsf{lab}', \mathbf{y}_i)$:

$\mathsf{Enc}(\mathsf{ek}_i, \mathsf{lab}, \mathbf{x}_i)$:

$\mathsf{Dec}(\{(\mathsf{dk}_i, \mathsf{ct}_i)\}_{i \in [n]})$:

Selective FH-IP-DMCFE from FH-IPFE

$$
\begin{array}{ll}
\mathsf{Setup}(1^\lambda): & s_1,\ldots,s_n \xleftarrow{\$} \mathbb{Z}_q \text{ s.t. } \sum_{i\in[n]} s_i = 0; \\
& \text{for all } i \in [n]\text{: } \mathrm{imsk}_i \leftarrow \mathbf{iSetup}(1^\lambda), \\
& \mathrm{ek}_i = (\mathrm{imsk}_i, s_i) \text{ and } \mathrm{sk}_i = \mathrm{imsk}_i \\
\mathsf{KeyGen}(\mathrm{sk}_i, \mathrm{lab}', \mathbf{y}_i): & [\![\tau']\!]_2 = \mathsf{H}_2(\mathrm{lab}'); \\
& \mathrm{dk}_i \leftarrow \mathbf{iKeyGen}(\mathrm{imsk}_i, [\![(\mathbf{y}_i, \tau', 0)]\!]_2) \\
\mathsf{Enc}(\mathrm{ek}_i, \mathrm{lab}, \mathbf{x}_i): & [\![\tau]\!]_1 = \mathsf{H}_1(\mathrm{lab}); \\
& \mathrm{ct}_i \leftarrow \mathbf{iEnc}(\mathrm{imsk}_i, [\![(\mathbf{x}_i, s_i\tau, 0)]\!]_1) \\
\mathsf{Dec}(\{(\mathrm{dk}_i, \mathrm{ct}_i)\}_{i\in[n]}): &
\end{array}
$$

Selective FH-IP-DMCFE from FH-IPFE

$$
\begin{array}{ll}
\mathsf{Setup}(1^\lambda): & s_1, \ldots, s_n \xleftarrow{\$} \mathbb{Z}_q \text{ s.t. } \sum_{i\in[n]} s_i = 0; \\
& \text{for all } i \in [n]\text{: } \mathsf{imsk}_i \leftarrow \mathbf{iSetup}(1^\lambda), \\
& \mathsf{ek}_i = (\mathsf{imsk}_i, s_i) \text{ and } \mathsf{sk}_i = \mathsf{imsk}_i \\
\mathsf{KeyGen}(\mathsf{sk}_i, \mathsf{lab}', \mathbf{y}_i): & [\![\tau']\!]_2 = \mathsf{H}_2(\mathsf{lab}'); \\
& \mathsf{dk}_i \leftarrow \mathbf{iKeyGen}(\mathsf{imsk}_i, [\![(\mathbf{y}_i, \tau', 0)]\!]_2) \\
\mathsf{Enc}(\mathsf{ek}_i, \mathsf{lab}, \mathbf{x}_i): & [\![\tau]\!]_1 = \mathsf{H}_1(\mathsf{lab}); \\
& \mathsf{ct}_i \leftarrow \mathbf{iEnc}(\mathsf{imsk}_i, [\![(\mathbf{x}_i, s_i\tau, 0)]\!]_1) \\
\mathsf{Dec}(\{(\mathsf{dk}_i, \mathsf{ct}_i)\}_{i\in[n]}): & \text{for all } i \in [n]\text{: } \{[\![z_i]\!]_\mathsf{t} \leftarrow \mathbf{iDec}(\mathsf{dk}_i, \mathsf{ct}_i)\}_{i\in[n]}; \\
& \text{output discrete log of } [\![\textstyle\sum_{i\in[n]} z_i]\!]_\mathsf{t}
\end{array}
$$

Correctness.

$$
\sum_{i\in[n]} z_i = \sum_{i\in[n]} \langle \mathbf{x}_i, \mathbf{y}_i \rangle + s_i\tau\tau' = \sum_{i\in[n]} \langle \mathbf{x}_i, \mathbf{y}_i \rangle + \tau\tau' \sum_{i\in[n]} s_i = \sum_{i\in[n]} \langle \mathbf{x}_i, \mathbf{y}_i \rangle
$$

If we had SIM-Security ...

$$\begin{aligned}
[\![\langle \mathbf{x}^{(j_i,0)}_{\mathsf{lab},i}, \mathbf{y}^{(j'_i,0)}_{\mathsf{lab}',i}\rangle + s_i\tau_{\mathsf{lab}}\tau_{\mathsf{lab}'}]\!]_{\mathsf{t}} &\approx_c \\
&\equiv \\
&= \\
&\approx_c [\![\langle \mathbf{x}^{(j_i,1)}_{\mathsf{lab},i}, \mathbf{y}^{(j'_i,1)}_{\mathsf{lab}',i}\rangle + s_i\tau_{\mathsf{lab}}\tau_{\mathsf{lab}'}]\!]_{\mathsf{t}}
\end{aligned}$$

If we had SIM-Security ...

$$
\begin{aligned}
[\![\langle \mathsf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathsf{y}_{\mathsf{lab}',i}^{(j_i',0)} \rangle + s_i \tau_{\mathsf{lab}} \tau_{\mathsf{lab}'}]\!]_{\mathsf{t}} &\approx_c [\![\langle \mathsf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathsf{y}_{\mathsf{lab}',i}^{(j_i',0)} \rangle + s_{\mathsf{lab},\mathsf{lab}',i}]\!]_{\mathsf{t}} \\
&\equiv \\
&= \\
&\approx_c [\![\langle \mathsf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathsf{y}_{\mathsf{lab}',i}^{(j_i',1)} \rangle + s_i \tau_{\mathsf{lab}} \tau_{\mathsf{lab}'}]\!]_{\mathsf{t}}
\end{aligned}
$$

If we had SIM-Security ...

$$
\begin{aligned}
[\![\langle \mathsf{x}^{(j_i,0)}_{\mathsf{lab},i}, \mathsf{y}^{(j'_i,0)}_{\mathsf{lab}',i}\rangle + s_i \tau_{\mathsf{lab}} \tau_{\mathsf{lab}'}]\!]_{\mathsf{t}} &\approx_c [\![\langle \mathsf{x}^{(j_i,0)}_{\mathsf{lab},i}, \mathsf{y}^{(j'_i,0)}_{\mathsf{lab}',i}\rangle + s_{\mathsf{lab},\mathsf{lab}',i}]\!]_{\mathsf{t}} \\
&\equiv \\
&= \\
&\approx_c [\![\langle \mathsf{x}^{(j_i,1)}_{\mathsf{lab},i}, \mathsf{y}^{(j'_i,1)}_{\mathsf{lab}',i}\rangle + s_i \tau_{\mathsf{lab}} \tau_{\mathsf{lab}'}]\!]_{\mathsf{t}}
\end{aligned}
$$

Admissibility of $\mathcal{A}$. For all j_i, j'_i, it holds that

$$
\sum_i \langle \mathsf{x}^{(j_i,0)}_{\mathsf{lab},i}, \mathsf{y}^{(j'_i,0)}_{\mathsf{lab}',i}\rangle = \sum_i \langle \mathsf{x}^{(j_i,1)}_{\mathsf{lab},i}, \mathsf{y}^{(j'_i,1)}_{\mathsf{lab}',i}\rangle \quad .
$$

If we had SIM-Security ...

$$\llbracket \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,0)} \rangle + s_i \tau_{\mathsf{lab}} \tau_{\mathsf{lab}'} \rrbracket_{\mathsf{t}} \approx_c \llbracket \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,0)} \rangle + s_{\mathsf{lab},\mathsf{lab}',i} \rrbracket_{\mathsf{t}}$$

$$\equiv$$

$$=$$

$$\approx_c \llbracket \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,1)} \rangle + s_i \tau_{\mathsf{lab}} \tau_{\mathsf{lab}'} \rrbracket_{\mathsf{t}}$$

Admissibility of $\mathcal{A}$. For all j_i, j'_i, it holds that

$$\sum_i \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,0)} \rangle = \sum_i \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,1)} \rangle \quad .$$

This implies that

$$\Delta_{\mathsf{lab},\mathsf{lab}',i} := \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,0)} \rangle - \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,1)} \rangle$$

is constant for all j_i, j'_i and $\sum_i \Delta_{\mathsf{lab},\mathsf{lab}',i} = 0$.

If we had SIM-Security ...

$$
\begin{aligned}
[\![\langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j_i',0)}\rangle + s_i\tau_{\mathsf{lab}}\tau_{\mathsf{lab}'}]\!]_{\mathsf{t}} &\approx_c [\![\langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j_i',0)}\rangle + s_{\mathsf{lab},\mathsf{lab}',i}]\!]_{\mathsf{t}} \\
&\equiv [\![\langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j_i',0)}\rangle + (s_{\mathsf{lab},\mathsf{lab}',i} - \Delta_{\mathsf{lab},\mathsf{lab}',i})]\!]_{\mathsf{t}} \\
&= \\
&\approx_c [\![\langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j_i',1)}\rangle + s_i\tau_{\mathsf{lab}}\tau_{\mathsf{lab}'}]\!]_{\mathsf{t}}
\end{aligned}
$$

Admissibility of $\mathcal{A}$. For all j_i, j_i', it holds that

$$
\sum_i \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j_i',0)}\rangle = \sum_i \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j_i',1)}\rangle \quad .
$$

This implies that

$$
\Delta_{\mathsf{lab},\mathsf{lab}',i} := \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j_i',0)}\rangle - \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j_i',1)}\rangle
$$

is constant for all j_i, j_i' and $\sum_i \Delta_{\mathsf{lab},\mathsf{lab}',i} = 0$.

If we had SIM-Security ...

$$\begin{aligned}
[\![\langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,0)}\rangle + s_i\tau_{\mathsf{lab}}\tau_{\mathsf{lab}'}]\!]_{\mathsf{t}} &\approx_c [\![\langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,0)}\rangle + s_{\mathsf{lab},\mathsf{lab}',i}]\!]_{\mathsf{t}} \\
&\equiv [\![\langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,0)}\rangle + (s_{\mathsf{lab},\mathsf{lab}',i} - \Delta_{\mathsf{lab},\mathsf{lab}',i})]\!]_{\mathsf{t}} \\
&= [\![\langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,1)}\rangle + s_{\mathsf{lab},\mathsf{lab}',i}]\!]_{\mathsf{t}} \\
&\approx_c [\![\langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,1)}\rangle + s_i\tau_{\mathsf{lab}}\tau_{\mathsf{lab}'}]\!]_{\mathsf{t}}
\end{aligned}$$

Admissibility of $\mathcal{A}$. For all j_i, j'_i, it holds that

$$\sum_i \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,0)}\rangle = \sum_i \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,1)}\rangle \quad .$$

This implies that

$$\Delta_{\mathsf{lab},\mathsf{lab}',i} := \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,0)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,0)}\rangle - \langle \mathbf{x}_{\mathsf{lab},i}^{(j_i,1)}, \mathbf{y}_{\mathsf{lab}',i}^{(j'_i,1)}\rangle$$

is constant for all j_i, j'_i and $\sum_i \Delta_{\mathsf{lab},\mathsf{lab}',i} = 0$.

… but we have only IND-Security

$$\mathsf{KeyGen}(\mathsf{sk}_i, \mathsf{lab}', \mathbf{y}_{\mathsf{lab}',i}): \quad [\![\tau_{\mathsf{lab}'}]\!]_2 = \mathsf{H}_2(\mathsf{lab}');$$
$$\mathsf{dk}_i \leftarrow \mathsf{iKeyGen}(\mathsf{imsk}_i, [\![(\mathbf{y}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, \mathbf{0})]\!]_2)$$
$$\mathsf{Enc}(\mathsf{ek}_i, \mathsf{lab}, \mathbf{x}_{\mathsf{lab},i}): \quad [\![\tau_{\mathsf{lab}}]\!]_1 = \mathsf{H}_1(\mathsf{lab});$$
$$\mathsf{ct}_i \leftarrow \mathsf{iEnc}(\mathsf{imsk}_i, [\![(\mathbf{x}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, \mathbf{0})]\!]_1)$$

$$([\![(\mathbf{y}^{(0)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, \mathbf{0})]\!]_2, [\![(\mathbf{x}^{(0)}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, \mathbf{0})]\!]_1)$$

$$\vdots$$

$$\approx_c$$

$$\equiv$$

$$\vdots$$

$$\approx_c ([\![(\mathbf{y}^{(1)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, \mathbf{0})]\!]_2, [\![(\mathbf{x}^{(1)}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, \mathbf{0})]\!]_1)$$

… but we have only IND-Security

$$\begin{aligned}
\mathsf{KeyGen}(\mathsf{sk}_i, \mathsf{lab}', \mathbf{y}_{\mathsf{lab}',i}): \quad & [\![\tau_{\mathsf{lab}'}]\!]_2 = \mathsf{H}_2(\mathsf{lab}'); \\
& \mathsf{dk}_i \leftarrow \mathsf{iKeyGen}(\mathsf{imsk}_i, [\![(\mathbf{y}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, \mathbf{0})]\!]_2) \\
\mathsf{Enc}(\mathsf{ek}_i, \mathsf{lab}, \mathbf{x}_{\mathsf{lab},i}): \quad & [\![\tau_{\mathsf{lab}}]\!]_1 = \mathsf{H}_1(\mathsf{lab}); \\
& \mathsf{ct}_i \leftarrow \mathsf{iEnc}(\mathsf{imsk}_i, [\![(\mathbf{x}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, \mathbf{0})]\!]_1)
\end{aligned}$$

$$\begin{aligned}
& ([\![(\mathbf{y}^{(0)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, 0)]\!]_2, [\![(\mathbf{x}^{(0)}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, 0)]\!]_1) \\
& \vdots \\
\approx_c \; & ([\![(\mathbf{y}^{(0)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, s_{\mathsf{lab},\mathsf{lab}',i})]\!]_2, [\![(\mathbf{x}^{(0)}_{\mathsf{lab},i}, 0, 1)]\!]_1) \\
\equiv \; & \\
& \vdots \\
\approx_c \; & ([\![(\mathbf{y}^{(1)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, 0)]\!]_2, [\![(\mathbf{x}^{(1)}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, 0)]\!]_1)
\end{aligned}$$

... but we have only IND-Security

$$\mathsf{KeyGen}(\mathsf{sk}_i, \mathsf{lab}', \mathbf{y}_{\mathsf{lab}',i}): \quad [\![\tau_{\mathsf{lab}'}]\!]_2 = \mathsf{H}_2(\mathsf{lab}');$$
$$\mathsf{dk}_i \leftarrow \mathbf{iKeyGen}(\mathsf{imsk}_i, [\![(\mathbf{y}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, \mathbf{0})]\!]_2)$$
$$\mathsf{Enc}(\mathsf{ek}_i, \mathsf{lab}, \mathbf{x}_{\mathsf{lab},i}): \quad [\![\tau_{\mathsf{lab}}]\!]_1 = \mathsf{H}_1(\mathsf{lab});$$
$$\mathsf{ct}_i \leftarrow \mathbf{iEnc}(\mathsf{imsk}_i, [\![(\mathbf{x}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, \mathbf{0})]\!]_1)$$

$$
\begin{aligned}
&([\![(\mathbf{y}^{(0)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, 0)]\!]_2, [\![(\mathbf{x}^{(0)}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, 0)]\!]_1)\\
&\vdots\\
\approx_c\ &([\![(\mathbf{y}^{(0)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, s_{\mathsf{lab},\mathsf{lab}',i})]\!]_2, [\![(\mathbf{x}^{(0)}_{\mathsf{lab},i}, 0, 1)]\!]_1)\\
\equiv\ &([\![(\mathbf{y}^{(0)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, s_{\mathsf{lab},\mathsf{lab}',i} - \Delta_{\mathsf{lab},\mathsf{lab}',i})]\!]_2, [\![(\mathbf{x}^{(0)}_{\mathsf{lab},i}, 0, 1)]\!]_1)\\
&\vdots\\
\approx_c\ &([\![(\mathbf{y}^{(1)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, 0)]\!]_2, [\![(\mathbf{x}^{(1)}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, 0)]\!]_1)
\end{aligned}
$$

… but we have only IND-Security

$$\begin{aligned}
\mathsf{KeyGen}(\mathsf{sk}_i, \mathsf{lab}', \mathbf{y}_{\mathsf{lab}',i}):\quad & [\![\tau_{\mathsf{lab}'}]\!]_2 = \mathsf{H}_2(\mathsf{lab}');\\
& \mathsf{dk}_i \leftarrow \mathbf{iKeyGen}(\mathsf{imsk}_i, [\![(\mathbf{y}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, \mathbf{0})]\!]_2)\\
\mathsf{Enc}(\mathsf{ek}_i, \mathsf{lab}, \mathbf{x}_{\mathsf{lab},i}):\quad & [\![\tau_{\mathsf{lab}}]\!]_1 = \mathsf{H}_1(\mathsf{lab});\\
& \mathsf{ct}_i \leftarrow \mathbf{iEnc}(\mathsf{imsk}_i, [\![(\mathbf{x}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, \mathbf{0})]\!]_1)
\end{aligned}$$

$$\begin{aligned}
&([\![(\mathbf{y}^{(0)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, 0)]\!]_2, [\![(\mathbf{x}^{(0)}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, 0)]\!]_1)\\
&\quad\vdots\\
\approx_c\ &([\![(\mathbf{y}^{(0)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, s_{\mathsf{lab},\mathsf{lab}',i})]\!]_2, [\![(\mathbf{x}^{(0)}_{\mathsf{lab},i}, 0, 1)]\!]_1)\\
\equiv\ &([\![(\mathbf{y}^{(0)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, s_{\mathsf{lab},\mathsf{lab}',i} - \Delta_{\mathsf{lab},\mathsf{lab}',i})]\!]_2, [\![(\mathbf{x}^{(0)}_{\mathsf{lab},i}, 0, 1)]\!]_1)\\
&\quad\vdots\\
\approx_c\ &([\![(\mathbf{y}^{(1)}_{\mathsf{lab}',i}, \tau_{\mathsf{lab}'}, 0)]\!]_2, [\![(\mathbf{x}^{(1)}_{\mathsf{lab},i}, s_i\tau_{\mathsf{lab}}, 0)]\!]_1)
\end{aligned}$$

Dual Pairing Vector Spaces [OT10,12]

$B \xleftarrow{\$} \mathrm{GL}_N(\mathbb{Z}_q)$ $\qquad$ $B^* = (B^{-1})^\top$

Dual Pairing Vector Spaces [OT10,12]

$$B \xleftarrow{\$} \mathsf{GL}_N(\mathbb{Z}_q) \qquad\qquad B^* = (B^{-1})^\top$$

$$\mathbb{B} = \begin{pmatrix} \text{—}\ \mathbf{b}_1\ \text{—} \\ \vdots \\ \text{—}\ \mathbf{b}_n\ \text{—} \end{pmatrix} := [\![B]\!]_1 \qquad\qquad \mathbb{B}^* = \begin{pmatrix} \text{—}\ \mathbf{b}^*_1\ \text{—} \\ \vdots \\ \text{—}\ \mathbf{b}^*_n\ \text{—} \end{pmatrix} := [\![B^*]\!]_2$$

Dual Pairing Vector Spaces [OT10,12]

$$B \xleftarrow{\$} \mathrm{GL}_N(\mathbb{Z}_q) \qquad B^* = (B^{-1})^\top$$

$$\mathsf{B} = \begin{pmatrix} \text{---}\ \mathbf{b}_1\ \text{---} \\ \vdots \\ \text{---}\ \mathbf{b}_n\ \text{---} \end{pmatrix} := [\![B]\!]_1 \qquad \mathsf{B}^* = \begin{pmatrix} \text{---}\ \mathbf{b}_1^*\ \text{---} \\ \vdots \\ \text{---}\ \mathbf{b}_n^*\ \text{---} \end{pmatrix} := [\![B^*]\!]_2$$

For vectors $(x_1, \ldots, x_N) \in \mathbb{Z}_q^N$ and $(y_1, \ldots, y_N) \in \mathbb{Z}_q^N$, write

$$(x_1, \ldots, x_N)_{\mathsf{B}} := \sum_{i \in [N]} x_i \mathbf{b}_i \in \mathbb{G}_1 \qquad (y_1, \ldots, y_N)_{\mathsf{B}^*} := \sum_{i \in [N]} y_i \mathbf{b}_i^* \in \mathbb{G}_2 \quad .$$

Dual Pairing Vector Spaces [OT10,12]

$$B \xleftarrow{\$} \mathrm{GL}_N(\mathbb{Z}_q) \qquad B^* = (B^{-1})^\top$$

$$\mathbf{B} = \begin{pmatrix} \text{—}\ \mathbf{b}_1\ \text{—} \\ \vdots \\ \text{—}\ \mathbf{b}_n\ \text{—} \end{pmatrix} := [\![B]\!]_1 \qquad \mathbf{B}^* = \begin{pmatrix} \text{—}\ \mathbf{b}_1^*\ \text{—} \\ \vdots \\ \text{—}\ \mathbf{b}_n^*\ \text{—} \end{pmatrix} := [\![B^*]\!]_2$$

For vectors $(x_1, \ldots, x_N) \in \mathbb{Z}_q^N$ and $(y_1, \ldots, y_N) \in \mathbb{Z}_q^N$, write

$$(x_1, \ldots, x_N)_{\mathbf{B}} := \sum_{i \in [N]} x_i \mathbf{b}_i \in \mathbb{G}_1 \qquad (y_1, \ldots, y_N)_{\mathbf{B}^*} := \sum_{i \in [N]} y_i \mathbf{b}_i^* \in \mathbb{G}_2 \quad .$$

Define operation $\times$ which computes inner product in the exponent

$$(x_1, \ldots, x_N)_{\mathbf{B}} \times (y_1, \ldots, y_N)_{\mathbf{B}^*} := [\![x_1 y_1 + \cdots + x_n y_n]\!]_{\mathbf{t}}$$

Basis Changing Matrices

Type 1: Matrix embeds computational problem (e.g. DDH)

Type 2: Matrix does not embed computational problem

Basis Changing Matrices

Type 1: Matrix embeds computational problem (e.g. DDH)

- Computational problem allows to slightly alter the adversary's view by changing only **some** vectors, i.e. more flexibility

Type 2: Matrix does not embed computational problem

- No computational problem, so basis change modifies **all** vectors

Basis Changing Matrices

Type 1: Matrix embeds computational problem (e.g. DDH)

- Computational problem allows to slightly alter the adversary's view by changing only **some** vectors, i.e. more flexibility
- Negligible distinguishing advantage

Type 2: Matrix does not embed computational problem

- No computational problem, so basis change modifies **all** vectors
- Information-theoretic change, i.e. advantage is 0

Basis Changing Matrices

Type 1: Matrix embeds computational problem (e.g. DDH)

- Computational problem allows to slightly alter the adversary's view by changing only **some** vectors, i.e. more flexibility
- Negligible distinguishing advantage

$\rightarrow$ Resemblance to (blackbox) IPFE

Type 2: Matrix does not embed computational problem

- No computational problem, so basis change modifies **all** vectors
- Information-theoretic change, i.e. advantage is 0

$\rightarrow$ Not provided by security definition of IPFE

Formal Basis Changes

Type 2: Matrix does not embed computational problem

- Distinguishing advantage of 0

- Basis change modifies **all** vectors in the same way

Formal Basis Changes

Type 2: Matrix does not embed computational problem

- Distinguishing advantage of **0**
  - Combination with complexity leveraging argument:
    After guessing oracle queries, the advantage is

$$\underbrace{1/\Pr[\text{correct guess}]}_{\text{exponential}} \cdot \underbrace{\mathsf{Adv}[\text{selective game}]}_{0} = 0$$

  - Selective security $\implies$ adaptive security
- Basis change modifies **all** vectors in the same way

Formal Basis Changes

Type 2: Matrix does not embed computational problem

- Distinguishing advantage of **0**
  - Combination with complexity leveraging argument:
    After guessing oracle queries, the advantage is

$$\underbrace{1/\Pr[\text{correct guess}]}_{\text{exponential}} \cdot \underbrace{\mathbf{Adv}[\text{selective game}]}_{0} = 0$$

  - Selective security $\implies$ adaptive security
- Basis change modifies **all** vectors in the same way
  - Move repetitions in distinct (hidden) coordinates
  - Number of repetitions impacts dimension of vectors
  - A-priori bound on number of **QKeyGen** repetitions

Conclusion

Generic Construction from FH-IPFE.

- Selective security, static corruption
- Unbounded repetitions for **QKeyGen** queries

Concrete Construction Based on DPVS (SXDH + pairings).

- Adaptive security, static corruption
- Poly-bounded repetitions for **QKeyGen** queries

Conclusion

Generic Construction from FH-IPFE.

- Selective security, static corruption
- Unbounded repetitions for **QKeyGen** queries

Concrete Construction Based on DPVS (SXDH + pairings).

- Adaptive security, static corruption
- Poly-bounded repetitions for **QKeyGen** queries

Thank you for your attention!