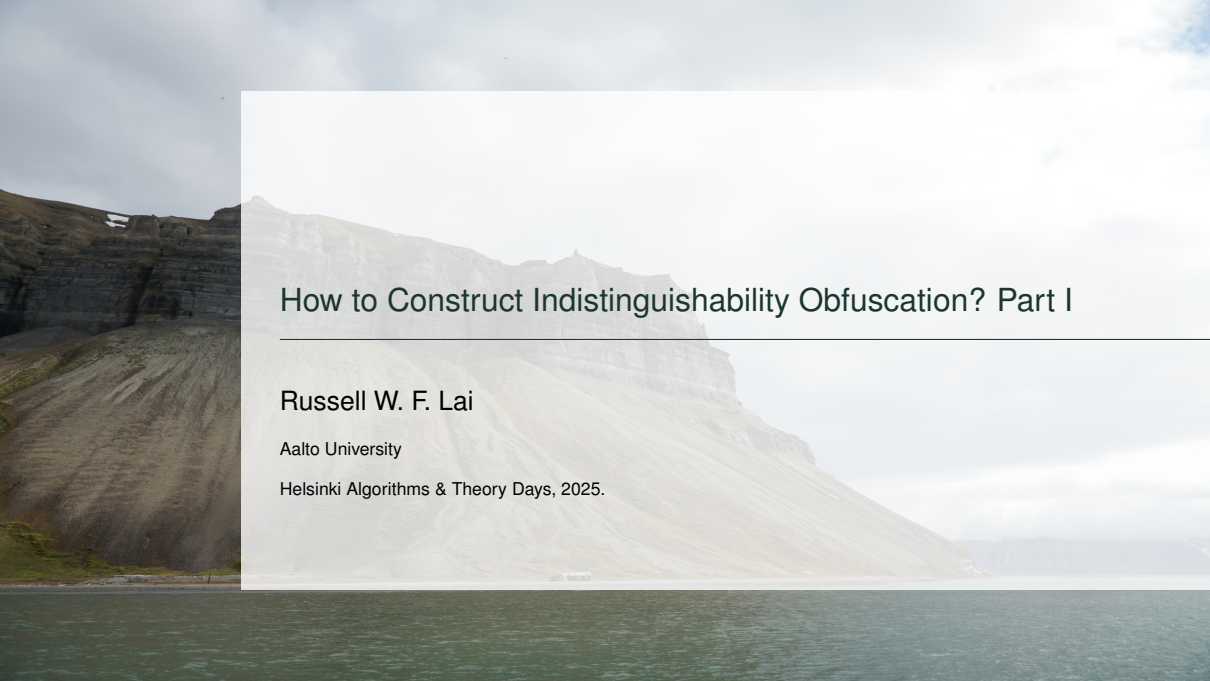




How to Construct Indistinguishability Obfuscation? Part I

Russell W. F. Lai

Aalto University

Helsinki Algorithms & Theory Days, 2025.

Context

† **Indistinguishability obfuscation (iO)** is powerful – implies (almost) all of cryptography, assuming OWF.

† History:

- ‡ 2001: Formal definition and impossibility of virtual black-box obfuscation [BGIRSVY01]
- ‡ 2013: First candidate construction of iO [GGHRSW13], demonstration of usefulness [SW14]
- ‡ 2014-2020: Breaking and fixing, constructions from progressively weaker assumptions, e.g. [BV15; AJ15; LPST16a; LPST16b; LV16; Lin17; LT17; AJLMS19]
- ‡ 2021-2022: First constructions from “well-founded” assumptions [JLS21; JLS22], not post-quantum
- ‡ 2020-2025: Breaking and fixing post-quantum constructions [BDGM20; WW21; GP21; HJL21; DQVWW21; BDGM22; JLLS23; CLW25; HJL25]

† Status quo:

- ‡ All existing constructions require very long chains of transformations
- ‡ No post-quantum construction from “well-founded” assumptions
- ‡ No (remotely) practical construction

† Next talk: Latest attempt on post-quantum iO [CLW25]

Context

† **Indistinguishability obfuscation (iO)** is powerful – implies (almost) all of cryptography, assuming OWF.

† History:

- ‡ 2001: Formal definition and impossibility of virtual black-box obfuscation [BGIRSVY01]
- ‡ 2013: First candidate construction of iO [GGHRSW13], demonstration of usefulness [SW14]
- ‡ 2014-2020: Breaking and fixing, constructions from progressively weaker assumptions, e.g. [BV15; AJ15; LPST16a; LPST16b; LV16; Lin17; LT17; AJLMS19]
- ‡ 2021-2022: First constructions from “well-founded” assumptions [JLS21; JLS22], not post-quantum
- ‡ 2020-2025: Breaking and fixing post-quantum constructions [BDGM20; WW21; GP21; HJL21; DQVWW21; BDGM22; JLLS23; CLW25; HJL25]

† Status quo:

- ‡ All existing constructions require very long chains of transformations
- ‡ No post-quantum construction from “well-founded” assumptions
- ‡ No (remotely) practical construction

† Next talk: Latest attempt on post-quantum iO [CLW25]

Context

† **Indistinguishability obfuscation (iO)** is powerful – implies (almost) all of cryptography, assuming OWF.

† History:

- ‡ 2001: Formal definition and impossibility of virtual black-box obfuscation [BGIRSVY01]
- ‡ 2013: First candidate construction of iO [GGHRSW13], demonstration of usefulness [SW14]
- ‡ 2014-2020: Breaking and fixing, constructions from progressively weaker assumptions, e.g. [BV15; AJ15; LPST16a; LPST16b; LV16; Lin17; LT17; AJLMS19]
- ‡ 2021-2022: First constructions from “well-founded” assumptions [JLS21; JLS22], not post-quantum
- ‡ 2020-2025: Breaking and fixing post-quantum constructions [BDGM20; WW21; GP21; HJL21; DQVWW21; BDGM22; JLLS23; CLW25; HJL25]

† Status quo:

- ‡ All existing constructions require very long chains of transformations
- ‡ No post-quantum construction from “well-founded” assumptions
- ‡ No (remotely) practical construction

† Next talk: Latest attempt on post-quantum iO [CLW25]

Agenda

I will attempt to explain one of the construction chains:

Next Talk $\implies$ XiO

$\xRightarrow{\text{Succinct FE}}$ Sublinear-size FE

$\xRightarrow{\text{Succinct FE}}$ Sublinear-time FE

$\xRightarrow{\text{recursion}}$ iO.

Note: Not a “well-founded assumptions” chain, but a chain with plausible post-quantum constructions.

Computation model

† We assume the (Boolean) circuit model of computation.

† Circuit families (to be obfuscated) are parametrised by input length $n \in \mathbb{N}$.

† A circuit is denoted as

$$\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m.$$

† All circuits considered are $\text{poly}(n)$ size. Therefore $|\text{TruthTable}(\Gamma)| = 2^n \cdot m \leq 2^{\text{poly}(n)}$.

† Cryptographic algorithms are parametrised by a security parameter $\lambda \in \mathbb{N}$. Setting $\lambda = 128$ means achieving 128-bit security.

Indistinguishability Obfuscation (iO)

Definition [BGJRSVY01]

An **Indistinguishability Obfuscation (iO)** is a pair of algorithms (Obf, Eval) such that

$$\tilde{\Gamma} \leftarrow \text{Obf}(1^\lambda, \Gamma), \quad y \leftarrow \text{Eval}(\tilde{\Gamma}, x).$$

† Correctness: For any $\Gamma, x \in \{0, 1\}^n$, $\text{Eval}(\tilde{\Gamma}, x) = \Gamma(x)$.

† Indistinguishability: If $\Gamma_0 \equiv \Gamma_1$ then $\tilde{\Gamma}_0 \approx_c \tilde{\Gamma}_1$.

† Efficiency: $|\tilde{\Gamma}| \leq \text{poly}(|\Gamma|, \lambda)$.

Q: Indistinguishability only for equivalent circuits. Why useful?

A: If $\Gamma_0 \approx_c \Gamma_0^\dagger \equiv \Gamma_1^\dagger \approx_c \Gamma_1$, then $\tilde{\Gamma}_0 \approx_c \tilde{\Gamma}_0^\dagger \approx_c \tilde{\Gamma}_1^\dagger \approx_c \tilde{\Gamma}_1$.

Indistinguishability Obfuscation (iO)

Definition [BGJRSVY01]

An **Indistinguishability Obfuscation (iO)** is a pair of algorithms $(\text{Obf}, \text{Eval})$ such that

$$\tilde{\Gamma} \leftarrow \text{Obf}(1^\lambda, \Gamma), \quad y \leftarrow \text{Eval}(\tilde{\Gamma}, x).$$

- † Correctness: For any $\Gamma, x \in \{0, 1\}^n$, $\text{Eval}(\tilde{\Gamma}, x) = \Gamma(x)$.
- † Indistinguishability: If $\Gamma_0 \equiv \Gamma_1$ then $\tilde{\Gamma}_0 \approx_c \tilde{\Gamma}_1$.
- † Efficiency: $|\tilde{\Gamma}| \leq \text{poly}(|\Gamma|, \lambda)$.

Q: Indistinguishability only for equivalent circuits. Why useful?

A: If $\Gamma_0 \approx_c \Gamma_0^\dagger \equiv \Gamma_1^\dagger \approx_c \Gamma_1$, then $\tilde{\Gamma}_0 \approx_c \tilde{\Gamma}_0^\dagger \approx_c \tilde{\Gamma}_1^\dagger \approx_c \tilde{\Gamma}_1$.

EXponentially-efficient iO (XiO)

† iO is trivial if we drop efficiency requirement:

Construction: $\tilde{\Gamma} = \text{TruthTable}(\Gamma)$.

† What if we require $|\tilde{\Gamma}| \leq |\text{TruthTable}(\Gamma)|^\alpha \cdot \text{poly}(\lambda)$ with constant $\alpha < 1$?

Definition [LPST18a]

An **EX**ponentially-efficient iO (XiO) is a pair of algorithms (Obf, Eval) such that

$$\tilde{\Gamma} \leftarrow \text{Obf}(1^\lambda, \Gamma), \quad y \leftarrow \text{Eval}(\tilde{\Gamma}, x).$$

† Correctness: For any $\Gamma, x \in \{0, 1\}^n$, $\text{Eval}(\text{Obf}(1^\lambda, \Gamma), x) = \Gamma(x)$.

† Indistinguishability: If $\Gamma_0 \equiv \Gamma_1$ then $\text{Obf}(\Gamma_0) \approx_c \text{Obf}(\Gamma_1)$.

† **Non-trivial efficiency:**

$$\ddagger |\tilde{\Gamma}| \leq |\text{TruthTable}(\Gamma)|^\alpha \cdot \text{poly}(\lambda),$$

$$\ddagger \text{Time}(\text{Obf}) \leq |\text{TruthTable}(\Gamma)| \cdot \text{poly}(\lambda).$$

EXponentially-efficient iO (XiO)

† iO is trivial if we drop efficiency requirement:

Construction: $\tilde{\Gamma} = \text{TruthTable}(\Gamma)$.

† What if we require $|\tilde{\Gamma}| \leq |\text{TruthTable}(\Gamma)|^\alpha \cdot \text{poly}(\lambda)$ with constant $\alpha < 1$?

Definition [LPST18a]

An **EX**ponentially-efficient iO (XiO) is a pair of algorithms (Obf, Eval) such that

$$\tilde{\Gamma} \leftarrow \text{Obf}(1^\lambda, \Gamma), \quad y \leftarrow \text{Eval}(\tilde{\Gamma}, x).$$

† Correctness: For any $\Gamma, x \in \{0, 1\}^n$, $\text{Eval}(\text{Obf}(1^\lambda, \Gamma), x) = \Gamma(x)$.

† Indistinguishability: If $\Gamma_0 \equiv \Gamma_1$ then $\text{Obf}(\Gamma_0) \approx_c \text{Obf}(\Gamma_1)$.

† **Non-trivial efficiency:**

$$\ddagger |\tilde{\Gamma}| \leq |\text{TruthTable}(\Gamma)|^\alpha \cdot \text{poly}(\lambda),$$

$$\ddagger \text{Time}(\text{Obf}) \leq |\text{TruthTable}(\Gamma)| \cdot \text{poly}(\lambda).$$

XiO to iO

Theorem [LPST16a]

Assume the existence of XiO and the hardness of Learning with Errors (LWE), then iO exists.

The transformation goes through an intermediate notion called **Functional Encryption (FE)**.

Functional Encryption (FE)

Definition

An FE is a tuple of algorithms (Setup, KGen, Enc, Dec) such that

- † $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda)$: Generate master public and secret keys
- † $\text{sk}_\Gamma \leftarrow \text{KGen}(\text{msk}, \Gamma)$: Generate functional key for circuit $\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m$
- † $\text{ctxt}_x \leftarrow \text{Enc}(\text{mpk}, x)$: Encrypt $x \in \{0, 1\}^n$
- † $y \leftarrow \text{Dec}(\text{sk}_\Gamma, \text{ctxt}_x)$: Decrypt and supposedly obtain $y = \Gamma(x)$

Security: Given $(\text{sk}_\Gamma, \text{ctxt}_x)$, can only learn $(\Gamma, \Gamma(x))$ but nothing else.

The XiO-to-iO transformation is very sensitive to the efficiency of Enc. We say that FE is ...

- † “Sublinear-size” if $|\text{ctxt}_x| \leq |\Gamma|^\alpha \cdot \text{poly}(\lambda)$ and $\text{Time}(\text{Enc}) \leq |\text{TruthTable}(\Gamma)| \cdot \text{poly}(\lambda)$,
- † “Sublinear-time” if $|\text{Time}(\text{Enc})| \leq |\Gamma|^\alpha \cdot \text{poly}(\lambda)$,
- † “Succinct” if $\text{Time}(\text{Enc}) \leq m \cdot \text{poly}(\lambda, \log |\Gamma|)$.

Functional Encryption (FE)

Definition

An FE is a tuple of algorithms (Setup, KGen, Enc, Dec) such that

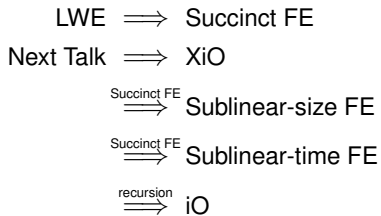
- † $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda)$: Generate master public and secret keys
- † $\text{sk}_\Gamma \leftarrow \text{KGen}(\text{msk}, \Gamma)$: Generate functional key for circuit $\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m$
- † $\text{ctxt}_x \leftarrow \text{Enc}(\text{mpk}, x)$: Encrypt $x \in \{0, 1\}^n$
- † $y \leftarrow \text{Dec}(\text{sk}_\Gamma, \text{ctxt}_x)$: Decrypt and supposedly obtain $y = \Gamma(x)$

Security: Given $(\text{sk}_\Gamma, \text{ctxt}_x)$, can only learn $(\Gamma, \Gamma(x))$ but nothing else.

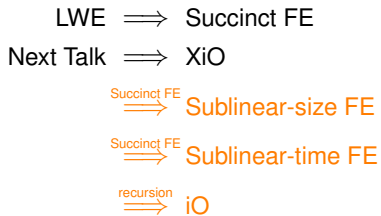
The XiO-to-iO transformation is very sensitive to the efficiency of Enc. We say that FE is ...

- † **“Sublinear-size”** if $|\text{ctxt}_x| \leq |\Gamma|^\alpha \cdot \text{poly}(\lambda)$ and $\text{Time}(\text{Enc}) \leq |\text{TruthTable}(\Gamma)| \cdot \text{poly}(\lambda)$,
- † **“Sublinear-time”** if $|\text{Time}(\text{Enc})| \leq |\Gamma|^\alpha \cdot \text{poly}(\lambda)$,
- † **“Succinct”** if $\text{Time}(\text{Enc}) \leq m \cdot \text{poly}(\lambda, \log |\Gamma|)$.

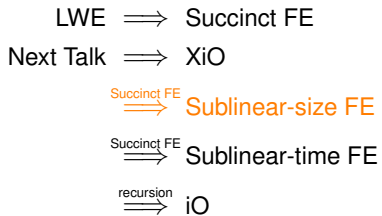
From XiO to iO



From XiO to iO



From XiO to iO



XiO + Succinct FE $\implies$ Sublinear-size FE [LPST16a]

Goal: Construct **Sublinear-size FE** for $\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m$.

- † Setup(1^λ): Run Setup of SuccinctFE.
- † KGen(msk, Γ):
 - ‡ Define $\Gamma^\dagger : (x, i) \mapsto i$ -th bit of $\Gamma(x)$.
 - ‡ Output SuccinctFE secret key $\text{sk}_\Gamma^\dagger$ of $\Gamma^\dagger$.
- † Enc(mpk, x):
 - ‡ Output XiO $\tilde{\Pi}$ of $\Pi : i \mapsto \text{SuccinctFE.Enc}(\text{mpk}, (x, i))$.
- † Dec($\text{sk}_\Gamma^\dagger$, ctxt_x):
 - ‡ Run $\text{ctxt}_{x,i} \leftarrow \tilde{\Pi}(i)$ for all i .
 - ‡ Output $\text{SuccinctFE.Dec}(\text{sk}_\Gamma^\dagger, \text{ctxt}_{x,i})$ for all i . / $\Gamma^\dagger(x, i) = i$ -th bit of $\Gamma(x)$.

XiO + Succinct FE $\implies$ Sublinear-size FE [LPST16a]

Goal: Construct **Sublinear-size FE** for $\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m$.

† $\text{Enc}(\text{mpk}, x)$:

‡ Output XiO $\tilde{\Pi}$ of $\Pi : i \mapsto \text{SuccinctFE}.\text{Enc}(\text{mpk}, (x, i))$.

Proof of sublinear-size ciphertext:

$$\begin{aligned}
 |\text{SuccinctFE}.\text{Enc}(\text{mpk}, (x, i))| &\stackrel{\text{Succinct}}{\leq} 1 \cdot \text{poly}(\lambda, \log |\Gamma|) \\
 |\text{TruthTable}(\Pi)| &= m \cdot |\text{SuccinctFE}.\text{Enc}(\text{mpk}, (x, i))| \\
 &\leq m \cdot \text{poly}(\lambda, \log |\Gamma|) \\
 |\tilde{\Pi}| &\stackrel{\text{XiO}}{\leq} |\text{TruthTable}(\Pi)|^\alpha \cdot \text{poly}(\lambda) \\
 &\leq m^\alpha \cdot \text{poly}(\lambda, \log |\Gamma|) \\
 &\leq |\Gamma|^\alpha \cdot \text{poly}(\lambda), \text{ i.e. sublinear-size ciphertext}
 \end{aligned}$$

XiO + Succinct FE $\implies$ Sublinear-size FE [LPST16a]

Goal: Construct **Sublinear-size FE** for $\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m$.

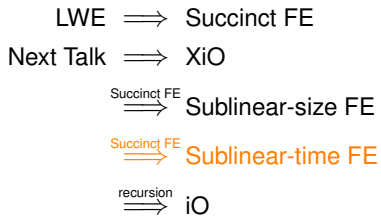
† $\text{Enc}(\text{mpk}, x)$:

‡ Output XiO $\tilde{\Pi}$ of $\Pi : i \mapsto \text{SuccinctFE}.\text{Enc}(\text{mpk}, (x, i))$.

Proof of sublinear-size ciphertext:

$$\begin{aligned}
 |\text{SuccinctFE}.\text{Enc}(\text{mpk}, (x, i))| &\stackrel{\text{Succinct}}{\leq} 1 \cdot \text{poly}(\lambda, \log |\Gamma|) \\
 |\text{TruthTable}(\Pi)| &= m \cdot |\text{SuccinctFE}.\text{Enc}(\text{mpk}, (x, i))| \\
 &\leq m \cdot \text{poly}(\lambda, \log |\Gamma|) \\
 |\tilde{\Pi}| &\stackrel{\text{XiO}}{\leq} |\text{TruthTable}(\Pi)|^\alpha \cdot \text{poly}(\lambda) \\
 &\leq m^\alpha \cdot \text{poly}(\lambda, \log |\Gamma|) \\
 &\leq |\Gamma|^\alpha \cdot \text{poly}(\lambda), \text{ i.e. sublinear-size ciphertext}
 \end{aligned}$$

From XiO to iO



Sublinear-size FE + Succinct FE $\implies$ Sublinear-time FE [LPST16b]

Goal: Construct **Sublinear-time FE** for $\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m$.

- † $\text{Setup}(1^\lambda)$: Run Setup of SuccinctFE and SSizeFE.
- † $\text{KGen}(\text{msk}, \Gamma)$:
 - ‡ Generate SuccinctFE key sk_{Enc} for $x \mapsto \text{SSizeFE.Enc}(\text{mpk}, x)$.
 - ‡ Generate SSizeFE key $\text{sk}_\Gamma^\dagger$ for $x \mapsto \Gamma(x)$.
- † $\text{Enc}(\text{mpk}, x)$: Output $\text{SuccinctFE.Enc}(\text{mpk}, x)$.
- † $\text{Dec}(\text{sk}_\Gamma, \text{ctxt}_x)$:
 - ‡ Use SuccinctFE key sk_{Enc} on ctxt_x to obtain an SSizeFE ciphertext $\text{ctxt}_x^\dagger$.
 - ‡ Use SSizeFE key $\text{sk}_\Gamma^\dagger$ on $\text{ctxt}_x^\dagger$ to obtain $\Gamma(x)$.

Sublinear-size FE + Succinct FE $\implies$ Sublinear-time FE [LPST16b]

Goal: Construct **Sublinear-time FE** for $\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m$ where $n \leq \text{poly}(\lambda)$.

† KGen(msk, Γ):

‡ Generate SuccinctFE key sk_{Enc} for $x \mapsto \text{SSizeFE.Enc}(\text{mpk}, x)$.

‡ Generate SSizeFE key $\text{sk}_{\Gamma}^{\dagger}$ for $x \mapsto \Gamma(x)$.

† Enc(mpk, x): Output SuccinctFE.Enc(mpk, x).

Proof of sublinear-time encryption:

$$|\text{SSizeFE.Enc}(\text{mpk}, x)| \stackrel{\text{SublinSize}}{\leq} |\Gamma|^{\alpha} \cdot \text{poly}(\lambda) \quad \forall x$$

$$\text{Time}(\text{SSizeFE.Enc}) \stackrel{\text{SublinSize}}{\leq} |\text{TruthTable}(\Gamma)| \cdot \text{poly}(\lambda)$$

$$\begin{aligned} \text{Time}(\text{SuccinctFE.Enc}) &\stackrel{\text{Succinct}}{\leq} |\text{SSizeFE.Enc}(\text{mpk}, x)| \cdot \text{poly}(\lambda, \log \text{Time}(\text{SSizeFE.Enc})) \\ &\leq |\Gamma|^{\alpha} \cdot \text{poly}(\lambda, \log |\text{TruthTable}(\Gamma)|) \\ &\leq |\Gamma|^{\alpha} \cdot \text{poly}(\lambda), \text{ i.e. sublinear-time encryption} \end{aligned}$$

Sublinear-size FE + Succinct FE $\implies$ Sublinear-time FE [LPST16b]

Goal: Construct **Sublinear-time FE** for $\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m$ where $n \leq \text{poly}(\lambda)$.

† KGen(msk, Γ):

‡ Generate SuccinctFE key sk_{Enc} for $x \mapsto \text{SSizeFE.Enc}(\text{mpk}, x)$.

‡ Generate SSizeFE key $\text{sk}_{\Gamma}^{\dagger}$ for $x \mapsto \Gamma(x)$.

† Enc(mpk, x): Output SuccinctFE.Enc(mpk, x).

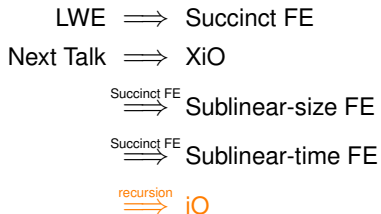
Proof of sublinear-time encryption:

$$|\text{SSizeFE.Enc}(\text{mpk}, x)| \stackrel{\text{SublinSize}}{\leq} |\Gamma|^{\alpha} \cdot \text{poly}(\lambda) \quad \forall x$$

$$\text{Time}(\text{SSizeFE.Enc}) \stackrel{\text{SublinSize}}{\leq} |\text{TruthTable}(\Gamma)| \cdot \text{poly}(\lambda)$$

$$\begin{aligned} \text{Time}(\text{SuccinctFE.Enc}) &\stackrel{\text{Succinct}}{\leq} |\text{SSizeFE.Enc}(\text{mpk}, x)| \cdot \text{poly}(\lambda, \log \text{Time}(\text{SSizeFE.Enc})) \\ &\leq |\Gamma|^{\alpha} \cdot \text{poly}(\lambda, \log |\text{TruthTable}(\Gamma)|) \\ &\leq |\Gamma|^{\alpha} \cdot \text{poly}(\lambda), \text{ i.e. sublinear-time encryption} \end{aligned}$$

From XiO to iO



Sublinear-time FE $\implies$ iO [BV15; AJ15; LPST16b]

Goal: Construct iO for $\Gamma : \{0, 1\}^n \rightarrow \{0, 1\}^m$.

† Obf($1^\lambda, \Gamma$):

‡ Set up $n + 1$ STimeFE key pairs. Generate $\text{ctxt}_\epsilon \leftarrow \text{Enc}^{(0)}(\text{mpk}^{(0)}, \Gamma)$.

‡ For $0 \leq i < n$, generate a function key $\text{sk}_{\text{Enc}}^{(i)}$ for the following function:

$$(\Gamma, x_{[1:i]}) \mapsto (\text{Enc}^{(i+1)}(\text{mpk}^{(i+1)}, (\Gamma, x_{[1:i]} \| 0)), \text{Enc}^{(i+1)}(\text{mpk}^{(i+1)}, (\Gamma, x_{[1:i]} \| 1)))$$

‡ Generate a function key $\text{sk}_U^{(n)}$ for the universal circuit $U : (\Gamma, x) \mapsto \Gamma(x)$.

‡ Output $(\text{ctxt}_\epsilon, \text{sk}_{\text{Enc}}^{(0)}, \dots, \text{sk}_{\text{Enc}}^{(1)}, \text{sk}_U^{(n)})$.

† Eval($\tilde{\Gamma}, x$):

‡ For $0 \leq i < n$, run $\text{ctxt}_{\Gamma, x_{[1:i+1]}} \leftarrow \text{Dec}^{(i)}(\text{sk}_{\text{Enc}}^{(i)}, \text{ctxt}_{\Gamma, x_{[1:i]}})$.

‡ Output $y \leftarrow \text{Dec}^{(n)}(\text{sk}_U^{(n)}, \text{ctxt}_{\Gamma, x})$.

Sublinear-time FE $\implies$ iO [BV15; AJ15; LPST16b]

† Obf($1^\lambda, \Gamma$):

‡ Set up $n + 1$ STimeFE key pairs. Generate $\text{ctxt}_\epsilon \leftarrow \text{Enc}^{(0)}(\text{mpk}^{(0)}, \Gamma)$.

‡ For $0 \leq i < n$, generate a function key $\text{sk}_{\text{Enc}}^{(i)}$ for the following function:

$$(\Gamma, x_{[1:i]}) \mapsto (\text{Enc}^{(i+1)}(\text{mpk}^{(i+1)}, (\Gamma, x_{[1:i]} \| 0)), \text{Enc}^{(i+1)}(\text{mpk}^{(i+1)}, (\Gamma, x_{[1:i]} \| 1)))$$

‡ Generate a function key $\text{sk}_U^{(n)}$ for the universal circuit $U : (\Gamma, x) \mapsto \Gamma(x)$.

Proof of efficiency:

$$\text{Time}(\text{Enc}^{(n)})^{\text{SublinTime}} \leq |U|^{\alpha^\dagger} \cdot \text{poly}(\lambda) \leq |\Gamma|^\alpha \cdot \text{poly}(\lambda)$$

$$\text{Time}(\text{Enc}^{(n-1)})^{\text{SublinTime}} \leq \text{Time}(\text{Enc}^{(n)})^\alpha \cdot \text{poly}(\lambda) = (|\Gamma|^\alpha \cdot \text{poly}(\lambda))^\alpha \cdot \text{poly}(\lambda) = |\Gamma|^{\alpha^2} \cdot \text{poly}(\lambda)^{1+\alpha}$$

$$\text{Time}(\text{Enc}^{(0)})^{\text{SublinTime}} \leq |\Gamma|^{\alpha^{n+1}} \cdot \text{poly}(\lambda)^{1+\alpha+\dots+\alpha^n} \leq |\Gamma|^{\alpha^{n+1}} \cdot \text{poly}(\lambda)^{1/(1-\alpha)} \leq \text{poly}(\lambda)$$

Take Away

LWE $\implies$ Succinct FE
Next Talk $\implies$ XiO
 $\xRightarrow{\text{Succinct FE}}$ Sublinear-size FE
 $\xRightarrow{\text{Succinct FE}}$ Sublinear-time FE
 $\xRightarrow{\text{recursion}}$ iO

or

Size compression $\implies$ time compression $\implies$ iO.

Russell W. F. Lai

Aalto University, Finland

✉ russell.lai@aalto.fi



russell-lai.hk



research.cs.aalto.fi/crypto

Thank You!

References I

- [AJ15] Prabhanjan Ananth and Abhishek Jain. “Indistinguishability Obfuscation from Compact Functional Encryption”. In: *CRYPTO 2015, Part I*. 2015 (pages 2–4, 25, 26).
- [AJLMS19] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. “Indistinguishability Obfuscation Without Multilinear Maps: New Paradigms via Low Degree Weak Pseudorandomness and Security Amplification”. In: *CRYPTO 2019, Part III*. 2019 (pages 2–4).
- [BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. “Candidate iO from Homomorphic Encryption Schemes”. In: *EUROCRYPT 2020, Part I*. 2020 (pages 2–4).
- [BDGM22] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. “Factoring and Pairings Are Not Necessary for IO: Circular-Secure LWE Suffices”. In: *ICALP 2022*. 2022 (pages 2–4).
- [BGIRSVY01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. “On the (Im)possibility of Obfuscating Programs”. In: *CRYPTO 2001*. 2001 (pages 2–4, 7, 8).

References II

- [BV15] Nir Bitansky and Vinod Vaikuntanathan. “Indistinguishability Obfuscation from Functional Encryption”. In: *56th FOCS*. 2015 (pages 2–4, 25, 26).
- [CLW25] Valerio Cini, Russell W. F. Lai, and Ivy K. Y. Woo. “Lattice-Based Obfuscation from NTRU and Equivocal LWE”. In: *Advances in Cryptology – CRYPTO 2025*. 2025 (pages 2–4).
- [DQVWW21] Lalita Devadas, Willy Quach, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. “Succinct LWE Sampling, Random Polynomials, and Obfuscation”. In: *TCC 2021, Part II*. 2021 (pages 2–4).
- [GGHRSW13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. “Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits”. In: *54th FOCS*. 2013 (pages 2–4).
- [GP21] Romain Gay and Rafael Pass. “Indistinguishability obfuscation from circular security”. In: *53rd ACM STOC*. 2021 (pages 2–4).
- [HJL21] Samuel B. Hopkins, Aayush Jain, and Huijia Lin. “Counterexamples to New Circular Security Assumptions Underlying iO”. In: *CRYPTO 2021, Part II*. 2021 (pages 2–4).

References III

- [HJL25] Yao-Ching Hsieh, Aayush Jain, and Huijia Lin. “Lattice-Based Post-quantum iO from Circular Security with Random Opening Assumption”. In: *Advances in Cryptology – CRYPTO 2025*. 2025 (pages 2–4).
- [JLLS23] Aayush Jain, Huijia Lin, Paul Lou, and Amit Sahai. “Polynomial-Time Cryptanalysis of the Subspace Flooding Assumption for Post-quantum $i\mathcal{O}$ ”. In: *EUROCRYPT 2023, Part I*. 2023 (pages 2–4).
- [JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. “Indistinguishability obfuscation from well-founded assumptions”. In: *53rd ACM STOC*. 2021 (pages 2–4).
- [JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. “Indistinguishability Obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0 ”. In: *EUROCRYPT 2022, Part I*. 2022 (pages 2–4).
- [Lin17] Huijia Lin. “Indistinguishability Obfuscation from SXDH on 5-Linear Maps and Locality-5 PRGs”. In: *CRYPTO 2017, Part I*. 2017 (pages 2–4).
- [LPST16a] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. “Indistinguishability Obfuscation with Non-trivial Efficiency”. In: *PKC 2016, Part II*. 2016 (pages 2–4, 9–11, 17–19).

References IV

- [LPST16b] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. “Output-Compressing Randomized Encodings and Applications”. In: *TCC 2016-A, Part I*. 2016 (pages 2–4, 21–23, 25, 26).
- [LT17] Huijia Lin and Stefano Tessaro. “Indistinguishability Obfuscation from Trilinear Maps and Block-Wise Local PRGs”. In: *CRYPTO 2017, Part I*. 2017 (pages 2–4).
- [LV16] Huijia Lin and Vinod Vaikuntanathan. “Indistinguishability Obfuscation from DDH-Like Assumptions on Constant-Degree Graded Encodings”. In: *57th FOCS*. 2016 (pages 2–4).
- [SW14] Amit Sahai and Brent Waters. “How to use indistinguishability obfuscation: deniable encryption, and more”. In: *46th ACM STOC*. 2014 (pages 2–4).
- [WW21] Hoeteck Wee and Daniel Wichs. “Candidate Obfuscation via Oblivious LWE Sampling”. In: *EUROCRYPT 2021, Part III*. 2021 (pages 2–4).