ارتقاء مبهم‌سازی از $\mathbf{NC^1}$ به $\mathbf{P/poly}$ و رمزگذاری تابعی

پویا فرشیم

دانشگاه کوئینز بلفاست – اکول نورمال سوپریور

جنبه‌های ریاضی علوم کامپیوتری – مبانی رمز

Bootstrapping iO from $\mathbf{NC}^1$ to $\mathbf{P/poly}$ and Functional Encryption

Pooya Farshim

Queen's University Belfast/École Normale Supérieure

Mathematical Aspects of Computer Science – Foundations of Cryptography

Breakthrough Result of 2013

Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits

Sanjam Garg
UCLA
sanjamg@cs.ucla.edu

Craig Gentry
IBM Research
craigbgentry@gmail.com

Shai Halevi
IBM Research
shaih@alum.mit.edu

Mariana Raykova
IBM Research
mariana@cs.columbia.edu

Amit Sahai
UCLA
sahai@cs.ucla.edu

Brent Waters
University of Texas at Austin
bwaters@cs.utexas.edu

July 21, 2013

Abstract

In this work, we study *indistinguishability obfuscation* and *functional encryption* for general circuits:

Indistinguishability obfuscation requires that given any two equivalent circuits C_0 and C_1 of similar size, the obfuscations of C_0 and C_1 should be computationally indistinguishable.

In functional encryption, ciphertexts encrypt inputs x and keys are issued for circuits C. Using the key SK_C to decrypt a ciphertext $\mathrm{CT}_x = \mathsf{Enc}(x)$, yields the value $C(x)$ but does not reveal anything else about x. Furthermore, no collusion of secret key holders should be able to learn anything more than the union of what they can each learn individually.

We give constructions for indistinguishability obfuscation and functional encryption that supports all polynomial-size circuits. We accomplish this goal in three steps:

- We describe a candidate construction for indistinguishability obfuscation for $\mathbf{NC}^1$ circuits. The security of this construction is based on a new algebraic hardness assumption. The candidate and assumption use a simplified variant of multilinear maps, which we call *Multilinear Jigsaw Puzzles*.
- We show how to use indistinguishability obfuscation for $\mathbf{NC}^1$ together with Fully Homomorphic Encryption (with decryption in $\mathbf{NC}^1$) to achieve indistinguishability obfuscation for all circuits.
- Finally, we show how to use indistinguishability obfuscation for circuits, public-key encryption, and non-interactive zero knowledge to achieve functional encryption for all circuits. The functional encryption scheme we construct also enjoys succinct ciphertexts, which enables several other applications.

Indistinguishability Obfuscation (iO)

Security: Recall iO security:

$$\mathsf{iO}(C_0) \overset{c}{\approx} \mathsf{iO}(C_1) \ .$$

for **functionally equivalent** C_0 and C_1.

Indistinguishability Obfuscation (iO)

Security: Recall iO security:

$$\mathsf{iO}(C_0) \overset{c}{\approx} \mathsf{iO}(C_1) \ .$$

for **functionally equivalent** C_0 and C_1.

$\underline{\mathrm{IND}^{\mathcal{S},\mathcal{A}}_{\mathsf{Obf}}(1^\lambda)}$:	$\underline{\mathrm{Eq}^{\mathcal{D}}_{\mathcal{S}}(1^\lambda)}$:
$b \leftarrow\$ \{0,1\}$	$(C_0, C_1, st) \leftarrow\$ \mathcal{S}(1^\lambda)$
$(C_0, C_1, st) \leftarrow\$ \mathcal{S}(1^\lambda)$	$x \leftarrow\$ \mathcal{D}(C_0, C_1, st)$
$C \leftarrow\$ \mathsf{Obf}(1^\lambda, C_b)$	return $(C_0(x) \neq C_1(x))$
$b' \leftarrow\$ \mathcal{A}(1^\lambda, st)$	
return $(b = b')$	

Indistinguishability Obfuscation (iO)

Security: Recall iO security:

$$\mathsf{iO}(C_0) \overset{c}{\approx} \mathsf{iO}(C_1) \ .$$

for **functionally equivalent** C_0 and C_1.

$\underline{\mathrm{IND}^{\mathcal{S},\mathcal{A}}_{\mathsf{Obf}}(1^\lambda)}$:	$\underline{\mathrm{Eq}^{\mathcal{D}}_{\mathcal{S}}(1^\lambda)}$:
$b \leftarrow\$ \{0,1\}$	$(C_0, C_1, st) \leftarrow\$ \mathcal{S}(1^\lambda)$
$(C_0, C_1, st) \leftarrow\$ \mathcal{S}(1^\lambda)$	$x \leftarrow\$ \mathcal{D}(C_0, C_1, st)$
$C \leftarrow\$ \mathsf{Obf}(1^\lambda, C_b)$	return $(C_0(x) \neq C_1(x))$
$b' \leftarrow\$ \mathcal{A}(1^\lambda, st)$	
return $(b = b')$	

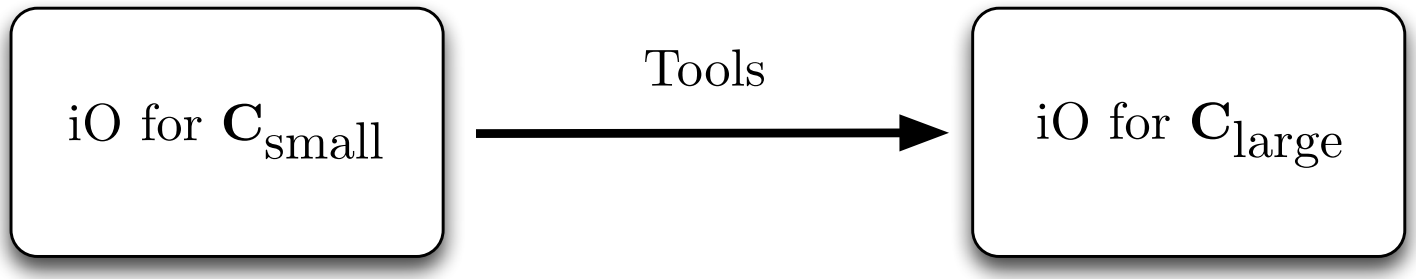
Sampler $\mathcal{S}$ is Eq-secure if for all (even unbounded) $\mathcal{D}$

$$\Pr\left[\mathrm{Eq}^{\mathcal{D}}_{\mathcal{S}}(1^\lambda)\right] \in \mathrm{Negl} \ .$$

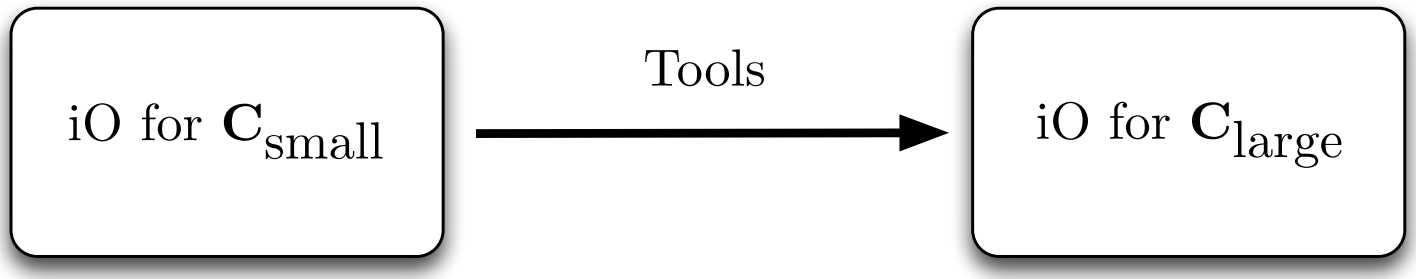
Security for all Eq-secure ppt $\mathcal{S}$ and all ppt $\mathcal{A}$

$$\mathbf{Adv}^{\mathrm{ind}}_{\mathsf{Obf},\mathcal{S},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathrm{IND}^{\mathcal{S},\mathcal{A}}_{\mathsf{Obf}}(1^\lambda)\right] - 1 \in \mathrm{Negl} \ .$$

Bootstrapping



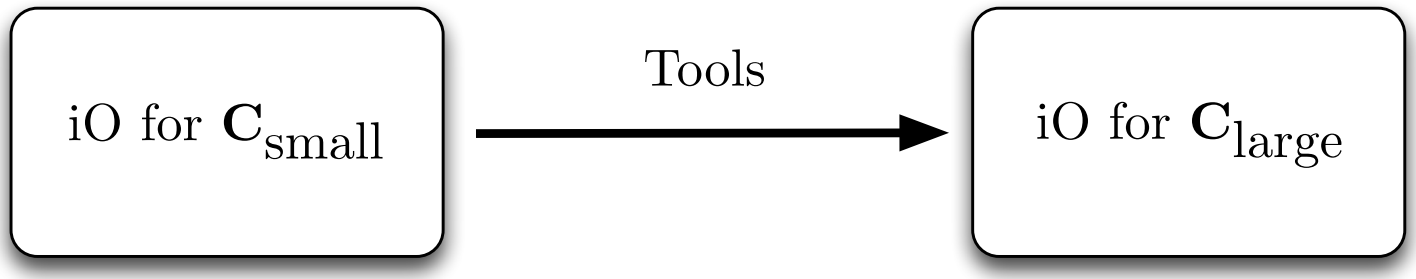
Bootstrapping



For us:

$$\mathbf{C}_{\text{small}} = \mathbf{NC}^1 \quad \text{and} \quad \mathbf{C}_{\text{large}} = \mathbf{P/poly}$$

Bootstrapping



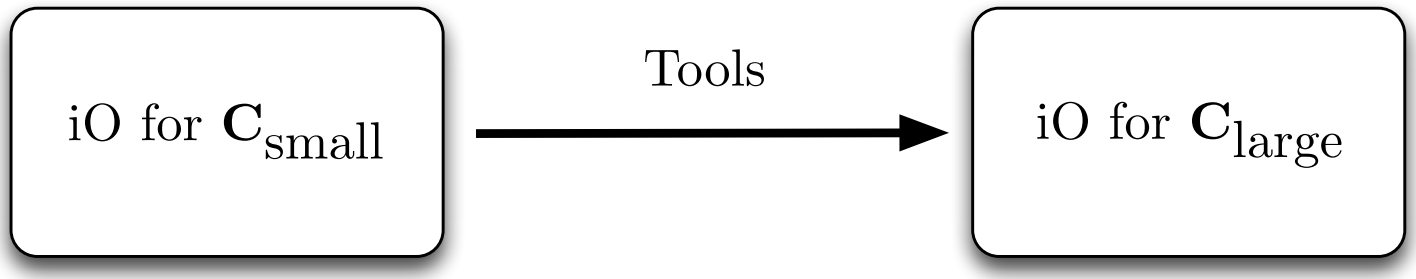
For us:

$$\mathbf{C}_{\text{small}} = \mathbf{NC}^1 \quad \text{and} \quad \mathbf{C}_{\text{large}} = \mathbf{P/poly}$$

Tools:

- Fully homomorphic encryption with $\text{Dec}(\cdot, C) \in \mathbf{C}_{\text{small}}$.
- Low-depth proofs with verification in $\mathbf{C}_{\text{small}}$.

Bootstrapping



For us:

$$\mathbf{C}_{\text{small}} = \mathbf{NC}^1 \quad \text{and} \quad \mathbf{C}_{\text{large}} = \mathbf{P/poly}$$

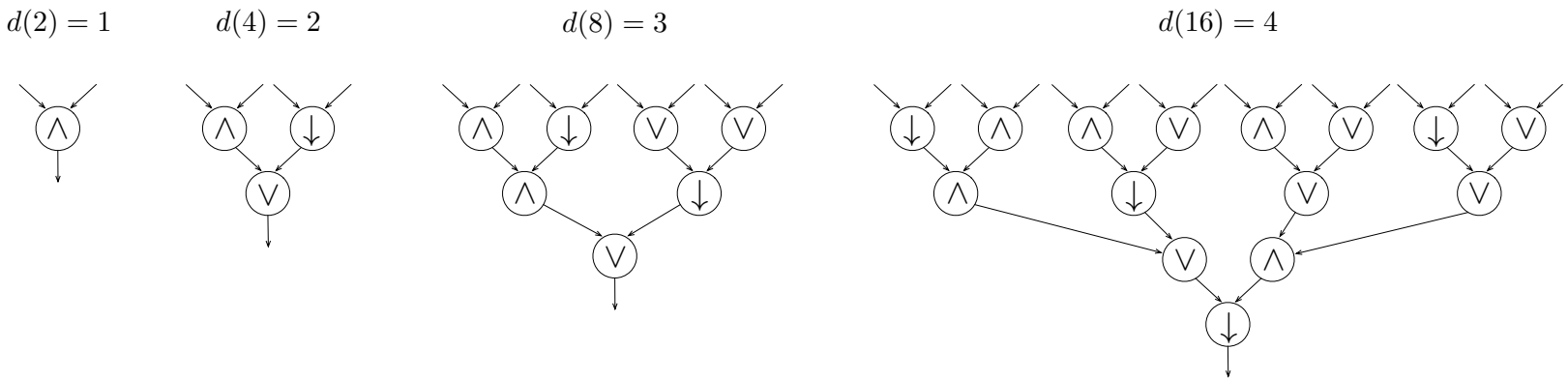
Tools:

- Fully homomorphic encryption with $\text{Dec}(\cdot, C) \in \mathbf{C}_{\text{small}}$.
- Low-depth proofs with verification in $\mathbf{C}_{\text{small}}$.

iO for $\mathbf{NC}^1$: See Mohammad's talk next!

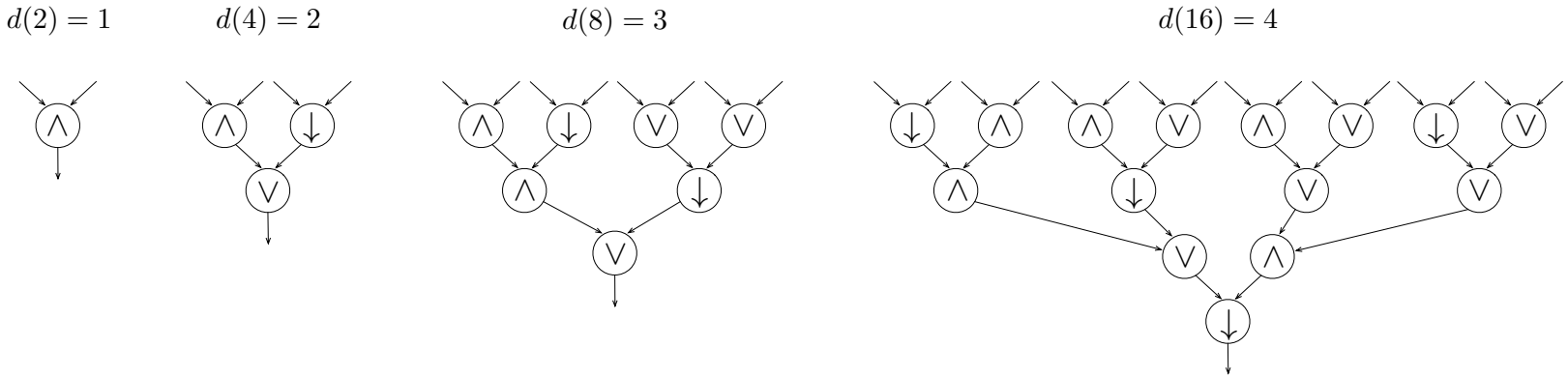
Nick's Class

$\mathbf{NC}^1$ looks something like:



Nick's Class

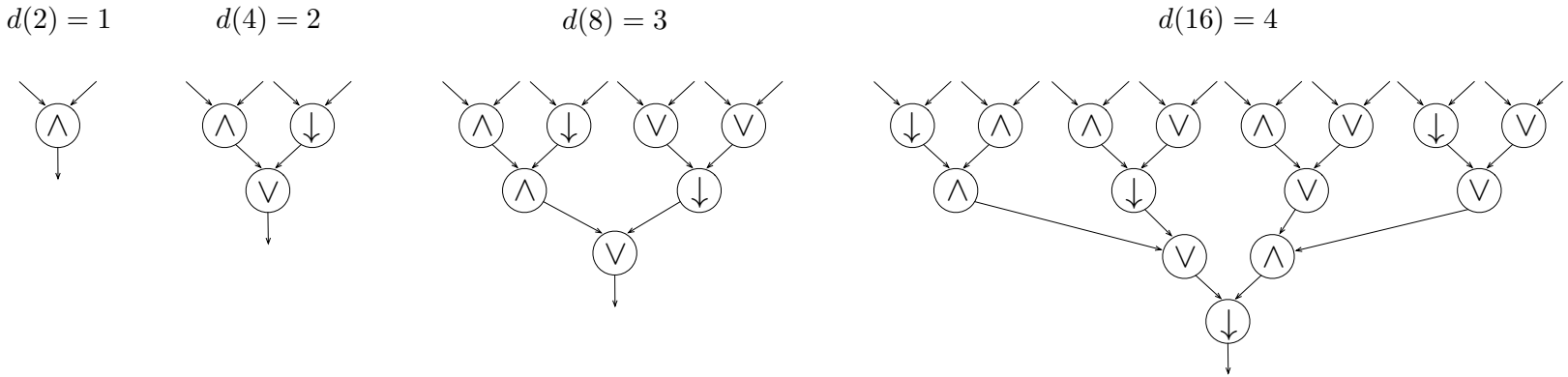
$\mathbf{NC}^1$ looks something like:



- $\mathbf{NC}^{\mathrm{i}}$:= circuit families C_λ with constant fan-in/out gates whose depth grow like $\mathcal{O}(\log^i(\lambda))$ with input size λ.

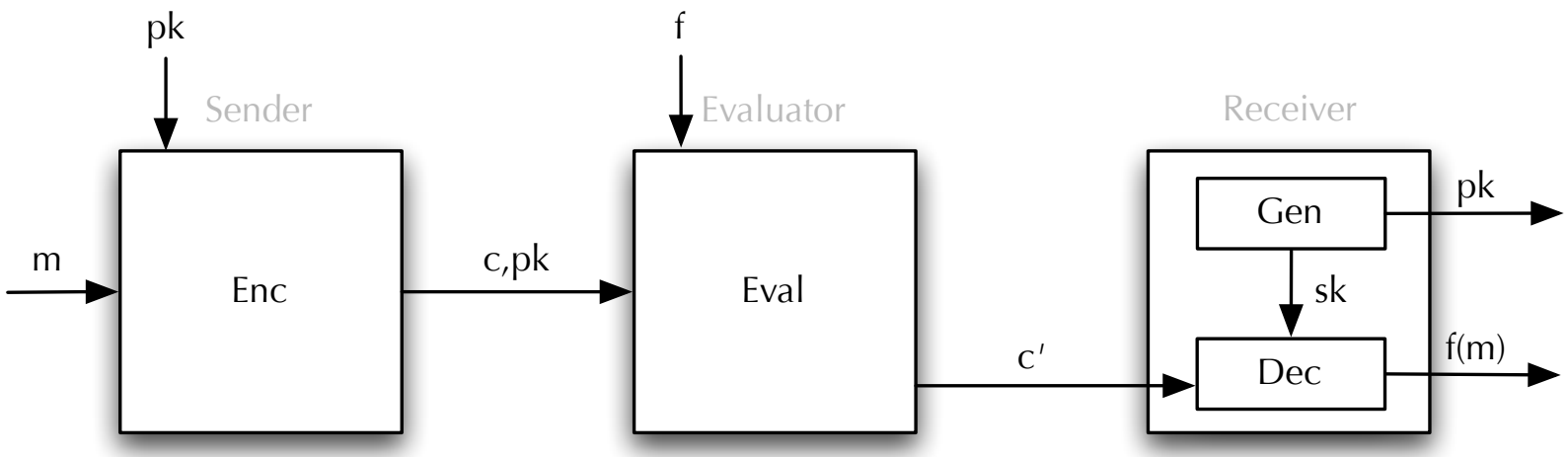
Nick's Class

$\mathbf{NC}^1$ looks something like:

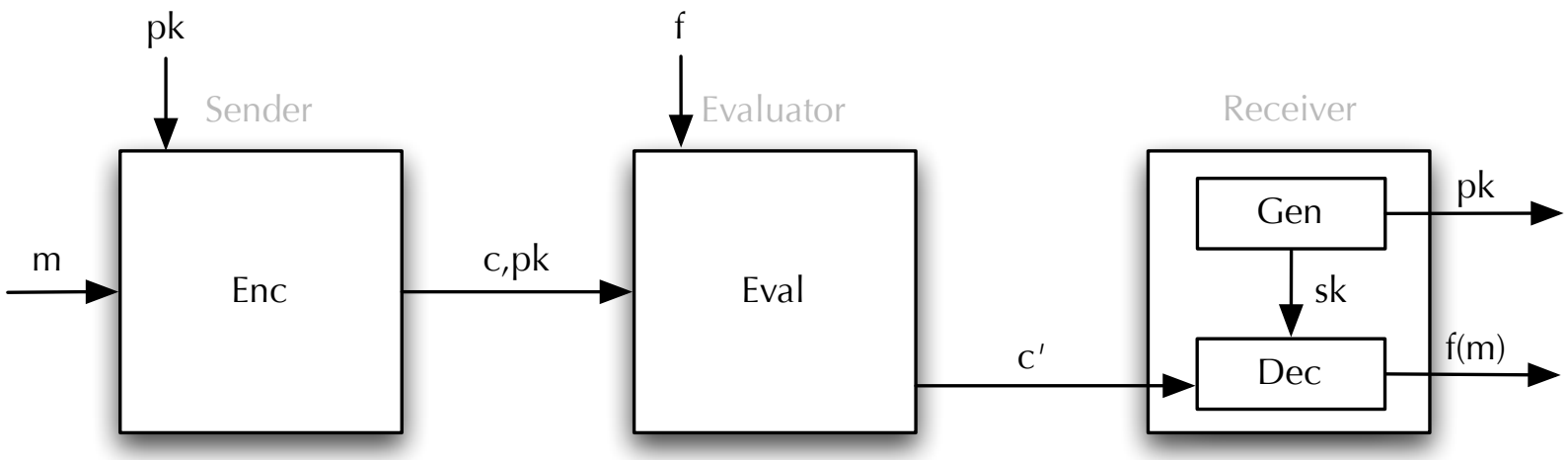


- $\mathbf{NC}^{\mathrm{i}} :=$ circuit families C_λ with constant fan-in/out gates whose depth grow like $\mathcal{O}(\log^i(\lambda))$ with input size λ.
- $\mathbf{P/poly} := \bigcup_{c \in \mathbb{N}} \{$ circuit families whose size grow like $\mathcal{O}(\lambda^c) \}$.

Fully Homomorphic Encryption (FHE)



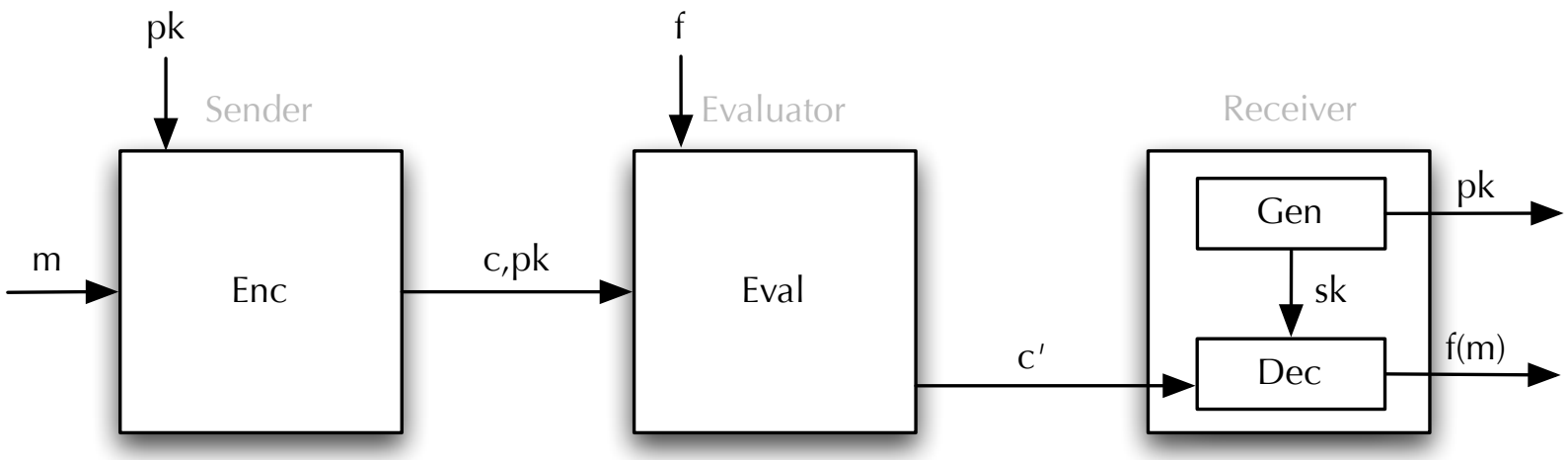
Fully Homomorphic Encryption (FHE)



Scheme is:

Fully homomorphic if it supports all $f \in \mathbf{P/poly}$.

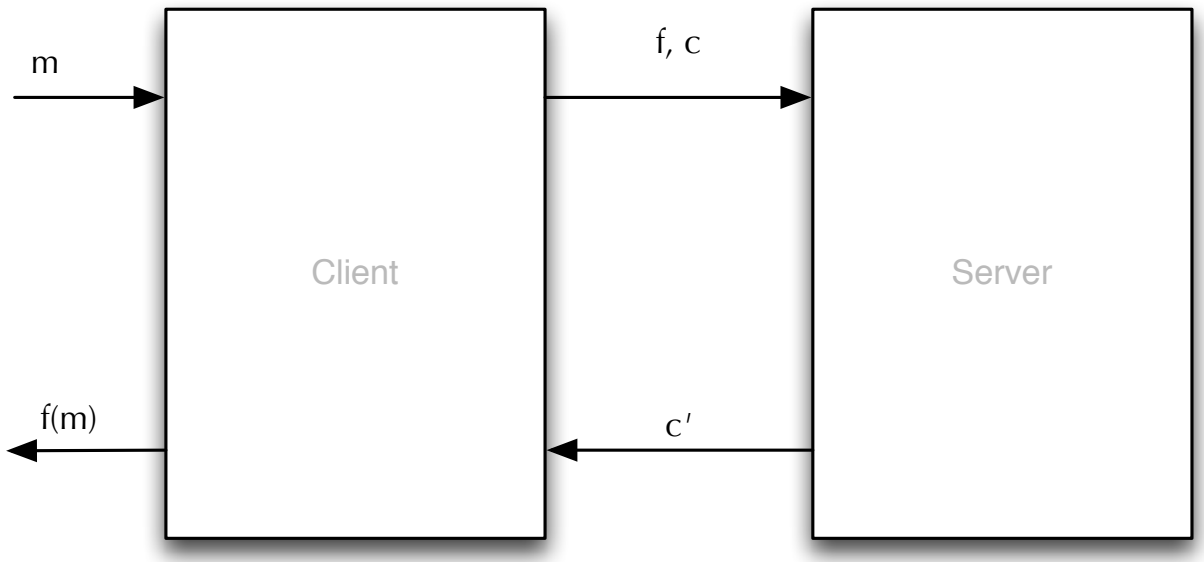
Fully Homomorphic Encryption (FHE)



Scheme is:

Fully homomorphic if it supports all $f \in \mathbf{P/poly}$.
$L(\lambda)$-leveled homomorphic if it supports all $f \in \mathbf{NC}^{L(\cdot)}$.

Outsourcing of Computation



FHE Details

1. $\mathsf{Gen}(1^\lambda)$ outputs key-pair (sk, pk)

FHE Details

1. $\mathsf{Gen}(1^\lambda)$ outputs key-pair (sk, pk)
2. $\mathsf{Enc}(pk, m)$ outputs ciphertext c.

FHE Details

1. $\mathsf{Gen}(1^\lambda)$ outputs key-pair (sk, pk)
2. $\mathsf{Enc}(pk, m)$ outputs ciphertext c.
3. $\mathsf{Eval}(pk, f, c_1, \ldots, c_n)$ outputs ciphertext c.

FHE Details

1. $\mathsf{Gen}(1^\lambda)$ outputs key-pair (sk, pk)
2. $\mathsf{Enc}(pk, m)$ outputs ciphertext c.
3. $\mathsf{Eval}(pk, f, c_1, \ldots, c_n)$ outputs ciphertext c.
4. $\mathsf{Dec}(sk, c)$ outputs a message m.

FHE Details

1. $\mathsf{Gen}(1^\lambda)$ outputs key-pair (sk, pk)
2. $\mathsf{Enc}(pk, m)$ outputs ciphertext c.
3. $\mathsf{Eval}(pk, f, c_1, \ldots, c_n)$ outputs ciphertext c.
4. $\mathsf{Dec}(sk, c)$ outputs a message m.

Correctness: If $c_1, \ldots, c_n$ encrypt $m_1, \ldots, m_n$:

$$\mathsf{Dec}(sk, \mathsf{Eval}(pk, f, c_1, \ldots, c_n)) = f(m_1, \ldots, m_n) \ .$$

FHE Details

1. $\mathsf{Gen}(1^\lambda)$ outputs key-pair (sk, pk)
2. $\mathsf{Enc}(pk, m)$ outputs ciphertext c.
3. $\mathsf{Eval}(pk, f, c_1, \ldots, c_n)$ outputs ciphertext c.
4. $\mathsf{Dec}(sk, c)$ outputs a message m.

Correctness: If $c_1, \ldots, c_n$ encrypt $m_1, \ldots, m_n$:

$$\mathsf{Dec}(sk, \mathsf{Eval}(pk, f, c_1, \ldots, c_n)) = f(m_1, \ldots, m_n) \ .$$

Compactness:

$$|\mathsf{Eval}(pk, f, c_1, \ldots, c_n)| \text{ depends on } \lambda \text{ only.}$$

FHE Details

1. $\mathsf{Gen}(1^\lambda)$ outputs key-pair (sk, pk)
2. $\mathsf{Enc}(pk, m)$ outputs ciphertext c.
3. $\mathsf{Eval}(pk, f, c_1, \ldots, c_n)$ outputs ciphertext c.
4. $\mathsf{Dec}(sk, c)$ outputs a message m.

Correctness: If $c_1, \ldots, c_n$ encrypt $m_1, \ldots, m_n$:

$$\mathsf{Dec}(sk, \mathsf{Eval}(pk, f, c_1, \ldots, c_n)) = f(m_1, \ldots, m_n) \ .$$

Compactness:

$$|\mathsf{Eval}(pk, f, c_1, \ldots, c_n)| \text{ depends on } \lambda \text{ only.}$$

Security:

$$(pk, \mathsf{Enc}(pk, m_0)) \overset{c}{\approx} (pk, \mathsf{Enc}(pk, m_1))$$

The Amplification

Intuition:

$\mathsf{Obf}(F)$: output $c \leftarrow\!\!\$\ \mathsf{Enc}(pk, F)$ and an obfuscation $\overline{\mathsf{Dec}}$ of $\mathsf{Dec}(sk, \cdot)$.

The Amplification

Intuition:

$\mathsf{Obf}(F)$: output $c \leftarrow\!\!\$\ \mathsf{Enc}(pk, F)$ and an obfuscation $\overline{\mathsf{Dec}}$ of $\mathsf{Dec}(sk, \cdot)$.

How can we run $\mathsf{Obf}(F)(x)$?

Evaluate $c' := \mathsf{Eval}(pk, \mathsf{UC}(\cdot, x), c)$ and then return $\overline{\mathsf{Dec}}(c')$

The Amplification

Intuition:

$\mathsf{Obf}(F)$: output $c \leftarrow\!\!\$\ \mathsf{Enc}(pk, F)$ and an obfuscation $\overline{\mathsf{Dec}}$ of $\mathsf{Dec}(sk, \cdot)$.

How can we run $\mathsf{Obf}(F)(x)$?

Evaluate $c' := \mathsf{Eval}(pk, \mathsf{UC}(\cdot, x), c)$ and then return $\overline{\mathsf{Dec}}(c')$

Need to make sure $\overline{\mathsf{Dec}}$ hides sk. We'll use the Naor–Yung paradigm [NY90]:

The Amplification

Intuition:

$\mathsf{Obf}(F)$: output $c \leftarrow\!\!\$\ \mathsf{Enc}(pk, F)$ and an obfuscation $\overline{\mathsf{Dec}}$ of $\mathsf{Dec}(sk, \cdot)$.

How can we run $\mathsf{Obf}(F)(x)$?

Evaluate $c' := \mathsf{Eval}(pk, \mathsf{UC}(\cdot, x), c)$ and then return $\overline{\mathsf{Dec}}(c')$

Need to make sure $\overline{\mathsf{Dec}}$ hides sk. We'll use the Naor–Yung paradigm [NY90]:

Doubly encrypt F and require **consistency** across **two** evaluations.

The Amplification

Intuition:

$\mathsf{Obf}(F)$: output $c \leftarrow\!\!\$\ \mathsf{Enc}(pk, F)$ and an obfuscation $\overline{\mathsf{Dec}}$ of $\mathsf{Dec}(sk, \cdot)$.

How can we run $\mathsf{Obf}(F)(x)$?

Evaluate $c' := \mathsf{Eval}(pk, \mathsf{UC}(\cdot, x), c)$ and then return $\overline{\mathsf{Dec}}(c')$

Need to make sure $\overline{\mathsf{Dec}}$ hides sk. We'll use the Naor–Yung paradigm [NY90]:

Doubly encrypt F and require **consistency** across **two** evaluations.
This allows **switching** of keys in the proof.

Details

- Generate two keys $(sk_i, pk_i) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$ for $i = 1, 2$

Details

- Generate two keys $(sk_i, pk_i) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$ for $i = 1, 2$
- Encrypt $c_i \leftarrow\!\!\$\ \mathsf{Enc}(pk_i, F)$ for $i = 1, 2$

Details

- Generate two keys $(sk_i, pk_i) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$ for $i = 1, 2$
- Encrypt $c_i \leftarrow\!\!\$\ \mathsf{Enc}(pk_i, F)$ for $i = 1, 2$
- Generate $\overline{P}$ as an $\underline{\mathbf{NC}^1 \text{ obfuscation}}$ of:

$$P[sk_1, G_1, G_2](m, E_1, E_2, \pi) := \text{if } \mathsf{V}((m, E_1, E_2), \pi) \text{ ret. } \mathsf{Dec}(sk_1, E_1)$$
$$\text{else return } \perp$$

Details

- Generate two keys $(sk_i, pk_i) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$ for $i = 1, 2$
- Encrypt $c_i \leftarrow\!\!\$\ \mathsf{Enc}(pk_i, F)$ for $i = 1, 2$
- Generate $\overline{P}$ as an $\underline{\mathbf{NC}^1 \text{ obfuscation}}$ of:

$$P[sk_1, G_1, G_2](m, E_1, E_2, \pi) := \text{if } \mathsf{V}((m, E_1, E_2), \pi) \text{ ret. } \mathsf{Dec}(sk_1, E_1)$$
$$\text{else return } \perp$$

- Return $(\overline{P}, pk_1, pk_2, G_1, G_2)$

Details

- Generate two keys $(sk_i, pk_i) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$ for $i = 1, 2$
- Encrypt $c_i \leftarrow\!\!\$\ \mathsf{Enc}(pk_i, F)$ for $i = 1, 2$
- Generate $\overline{P}$ as an $\underline{\mathbf{NC}^1 \text{ obfuscation}}$ of:

$$P[sk_1, G_1, G_2](m, E_1, E_2, \pi) := \text{if } \mathsf{V}((m, E_1, E_2), \pi) \text{ ret. } \mathsf{Dec}(sk_1, E_1)$$
$$\text{else return } \perp$$

- Return $(\overline{P}, pk_1, pk_2, G_1, G_2)$

What's V?

Details

- Generate two keys $(sk_i, pk_i) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$ for $i = 1, 2$
- Encrypt $c_i \leftarrow\!\!\$\ \mathsf{Enc}(pk_i, F)$ for $i = 1, 2$
- Generate $\overline{P}$ as an $\underline{\mathbf{NC}^1 \text{ obfuscation}}$ of:

$$P[sk_1, G_1, G_2](m, E_1, E_2, \pi) := \text{if } \mathsf{V}((m, E_1, E_2), \pi) \text{ ret. } \mathsf{Dec}(sk_1, E_1) \text{ else return } \perp$$

- Return $(\overline{P}, pk_1, pk_2, G_1, G_2)$

What's V? A low-depth proof of consistency that

$$\begin{aligned} \exists\,(G_1, G_2) \ : \quad & E_1 = \mathsf{Eval}(pk_1, \mathsf{UC}(\cdot, m), G_1) \\ \wedge \quad & E_2 = \mathsf{Eval}(pk_2, \mathsf{UC}(\cdot, m), G_2) \end{aligned}$$

Low-Depth Proof System (P, V) for Relation R

Perfect Completeness : For all (x, w) with $R(x, w) = 1$

$$\Pr[\pi \leftarrow_{\$} \mathsf{P}(x, w) : \mathsf{V}(x, \pi) = 1] = 1 \ .$$

Perfect Soundness : For all $x \notin R$ and for all (even unbounded) P^*

$$\Pr[\pi \leftarrow_{\$} \mathsf{P}^*(x) : \mathsf{V}(x, \pi) = 1] = 0 \ .$$

Low-Depth Proof System (P, V) for Relation R

Perfect Completeness : For all (x, w) with $R(x, w) = 1$

$$\Pr[\pi \leftarrow_{\$} \mathsf{P}(x, w) : \mathsf{V}(x, \pi) = 1] = 1 \ .$$

Perfect Soundness : For all $x \notin R$ and for all (even unbounded) P^*

$$\Pr[\pi \leftarrow_{\$} \mathsf{P}^*(x) : \mathsf{V}(x, \pi) = 1] = 0 \ .$$

The proof is **low depth** if

$$\mathsf{V}(\cdot, \cdot) \in \mathbf{NC}^1$$

Low-Depth Proof System (P, V) for Relation R

Perfect Completeness : For all (x, w) with $R(x, w) = 1$

$$\Pr[\pi \leftarrow_{\$} \mathsf{P}(x, w) : \mathsf{V}(x, \pi) = 1] = 1 \ .$$

Perfect Soundness : For all $x \notin R$ and for all (even unbounded) P^*

$$\Pr[\pi \leftarrow_{\$} \mathsf{P}^*(x) : \mathsf{V}(x, \pi) = 1] = 0 \ .$$

The proof is **low depth** if

$$\mathsf{V}(\cdot, \cdot) \in \mathbf{NC}^1$$

Construction:

$\mathsf{P}(x, w) :=$ List of all internal wire values of $R(x, w)$

$\mathsf{V}(x, \pi) :=$ Check if all wire values in π are consistent wrt. input x

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\Big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1},$$

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\Big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2},$$

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\Big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1] \Big) \overset{c}{\approx}$$

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\Big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1] \Big) \overset{c}{\approx} \Big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1] \Big)$$

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\Big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1] \Big) \overset{c}{\approx} \Big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1] \Big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\Big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1] \Big) \overset{c}{\approx} \Big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1] \Big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\Big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1] \Big) \overset{c}{\approx} \Big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1] \Big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	IND-CPA wrt. pk_2

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\Big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1] \Big) \overset{c}{\approx} \Big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1] \Big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	IND-CPA wrt. pk_2
2	C_0	C_1	$\boxed{sk_2}$	

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1]\big) \overset{c}{\approx} \big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1]\big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	IND-CPA wrt. pk_2
2	C_0	C_1	$\boxed{sk_2}$	iO for $\mathbf{NC}^1$ + $\underbrace{P[sk_1] \equiv P[sk_2]}_{\text{Correctness}+\pi+(C_0 \equiv C_1)}$

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1]\big) \overset{c}{\approx} \big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1]\big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	IND-CPA wrt. pk_2
2	C_0	C_1	$\boxed{sk_2}$	iO for $\mathbf{NC}^1$ + $\underbrace{P[sk_1] \equiv P[sk_2]}_{\text{Correctness}+\pi+(C_0 \equiv C_1)}$
3	$\boxed{C_1}$	C_1	sk_2	

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1]\big) \overset{c}{\approx} \big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1]\big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	IND-CPA wrt. pk_2
2	C_0	C_1	$\boxed{sk_2}$	iO for $\mathbf{NC}^1$ + $\underbrace{P[sk_1] \equiv P[sk_2]}_{\text{Correctness}+\pi+(C_0 \equiv C_1)}$
3	$\boxed{C_1}$	C_1	sk_2	IND-CPA wrt. pk_1

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1]\big) \overset{c}{\approx} \big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1]\big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	IND-CPA wrt. pk_2
2	C_0	C_1	$\boxed{sk_2}$	iO for $\mathbf{NC}^1$ + $\underbrace{P[sk_1] \equiv P[sk_2]}_{\text{Correctness}+\pi+(C_0 \equiv C_1)}$
3	$\boxed{C_1}$	C_1	sk_2	IND-CPA wrt. pk_1
4	C_1	C_1	$\boxed{sk_1}$	

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1]\big) \overset{c}{\approx} \big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1]\big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	IND-CPA wrt. pk_2
2	C_0	C_1	$\boxed{sk_2}$	iO for $\mathbf{NC}^1$ + $\underbrace{P[sk_1] \equiv P[sk_2]}_{\text{Correctness}+\pi+(C_0 \equiv C_1)}$
3	$\boxed{C_1}$	C_1	sk_2	IND-CPA wrt. pk_1
4	C_1	C_1	$\boxed{sk_1}$	$\mathbf{NC}^1$ iO + $P[sk_2] \equiv P[sk_1]$

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1]\big) \overset{c}{\approx} \big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1]\big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	IND-CPA wrt. pk_2
2	C_0	C_1	$\boxed{sk_2}$	iO for $\mathbf{NC}^1$ + $\underbrace{P[sk_1] \equiv P[sk_2]}_{\text{Correctness}+\pi+(C_0 \equiv C_1)}$
3	$\boxed{C_1}$	C_1	sk_2	IND-CPA wrt. pk_1
4	C_1	C_1	$\boxed{sk_1}$	$\mathbf{NC}^1$ iO + $P[sk_2] \equiv P[sk_1]$
	C_1	C_1	sk_1	

Proof of Security

Our goal is to prove:

$$\mathsf{Obf}(C_0) \overset{c}{\approx} \mathsf{Obf}(C_1)$$

$$\big(\underbrace{\mathsf{Enc}(pk_1, C_0)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_0)}_{G_2}, P[sk_1]\big) \overset{c}{\approx} \big(\underbrace{\mathsf{Enc}(pk_1, C_1)}_{G_1}, \underbrace{\mathsf{Enc}(pk_2, C_1)}_{G_2}, P[sk_1]\big)$$

Game	G_1 contains	G_2 contains	$P[sk]$ knows	**Remark**
0	C_0	C_0	sk_1	iO Game wrt. C_0
1	C_0	$\boxed{C_1}$	sk_1	IND-CPA wrt. pk_2
2	C_0	C_1	$\boxed{sk_2}$	iO for $\mathbf{NC}^1$ + $\underbrace{P[sk_1] \equiv P[sk_2]}_{\text{Correctness}+\pi+(C_0 \equiv C_1)}$
3	$\boxed{C_1}$	C_1	sk_2	IND-CPA wrt. pk_1
4	C_1	C_1	$\boxed{sk_1}$	$\mathbf{NC}^1$ iO + $P[sk_2] \equiv P[sk_1]$
	C_1	C_1	sk_1	iO Game wrt. C_1

Details: Game $0 \longrightarrow$ Game 1

Given $(\mathcal{S}, \mathcal{A})$ that distinguishes Game 0 from Game 1, consider:

Algo. $\mathcal{B}^{\mathcal{S},\mathcal{A}}(pk_2)$:

$(C_0, C_1, st) \leftarrow\!\!\$\ \mathcal{S}(1^\lambda)$

Return (C_0, C_1) as messages.

Get ciphertext G_2 under pk_2.

$(sk_1, pk_1) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$G_1 \leftarrow\!\!\$\ \mathsf{Enc}(pk_1, C_0)$

Construct circuit $P[sk_1, G_1, G_2]$

$\overline{P} \leftarrow\!\!\$\ \mathsf{Obf}(P[sk_1, G_1, G_2])$

$b' \leftarrow\!\!\$\ \mathcal{A}(\overline{P}, pk_1, pk_2, G_1, G_2, st)$

return b'

Details: Game $0 \longrightarrow$ Game 1

Given $(\mathcal{S}, \mathcal{A})$ that distinguishes Game 0 from Game 1, consider:

Algo. $\mathcal{B}^{\mathcal{S},\mathcal{A}}(pk_2)$:

$(C_0, C_1, st) \leftarrow\!\!\$\ \mathcal{S}(1^\lambda)$

Return (C_0, C_1) as messages.

Get ciphertext G_2 under pk_2.

$(sk_1, pk_1) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$G_1 \leftarrow\!\!\$\ \mathsf{Enc}(pk_1, C_0)$

Construct circuit $P[sk_1, G_1, G_2]$

$\overline{P} \leftarrow\!\!\$\ \mathsf{Obf}(P[sk_1, G_1, G_2])$

$b' \leftarrow\!\!\$\ \mathcal{A}(\overline{P}, pk_1, pk_2, G_1, G_2, st)$

return b'

If IND-CPA bit $= 0$: $\mathcal{B}$ runs $(\mathcal{S}, \mathcal{A})$ according to Game 0.

Details: Game $0 \longrightarrow$ Game 1

Given $(\mathcal{S}, \mathcal{A})$ that distinguishes Game 0 from Game 1, consider:

Algo. $\mathcal{B}^{\mathcal{S},\mathcal{A}}(pk_2)$:

$(C_0, C_1, st) \leftarrow\!\!\$\ \mathcal{S}(1^\lambda)$

Return (C_0, C_1) as messages.

Get ciphertext G_2 under pk_2.

$(sk_1, pk_1) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$G_1 \leftarrow\!\!\$\ \mathsf{Enc}(pk_1, C_0)$

Construct circuit $P[sk_1, G_1, G_2]$

$\overline{P} \leftarrow\!\!\$\ \mathsf{Obf}(P[sk_1, G_1, G_2])$

$b' \leftarrow\!\!\$\ \mathcal{A}(\overline{P}, pk_1, pk_2, G_1, G_2, st)$

return b'

If IND-CPA bit $= 0$: $\mathcal{B}$ runs $(\mathcal{S}, \mathcal{A})$ according to Game 0.
If IND-CPA bit $= 1$: $\mathcal{B}$ runs $(\mathcal{S}, \mathcal{A})$ according to Game 1.

Details: Game $1 \longrightarrow$ Game 2

Given $(\mathcal{S}, \mathcal{A})$ that distinguishes Game 1 from Game 2

$$\underline{\text{Algo. } \overline{\mathcal{S}}^{\mathcal{S}}(pk_2)}:$$
$$(C_0, C_1, st) \leftarrow\!\!\$\ \mathcal{S}(1^\lambda)$$

Details: Game $1 \longrightarrow$ Game 2

Given $(\mathcal{S}, \mathcal{A})$ that distinguishes Game 1 from Game 2

$$\underline{\text{Algo. } \overline{\mathcal{S}}^{\mathcal{S}}(pk_2)}:$$
$$(C_0, C_1, st) \leftarrow_{\$} \mathcal{S}(1^\lambda)$$
$$(sk_1, pk_1) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$$
$$(sk_2, pk_2) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$$

Details: Game $1 \longrightarrow$ Game 2

Given $(\mathcal{S}, \mathcal{A})$ that distinguishes Game 1 from Game 2

$\underline{\text{Algo. } \overline{\mathcal{S}}^{\mathcal{S}}(pk_2)}$:
$(C_0, C_1, st) \leftarrow_{\$} \mathcal{S}(1^\lambda)$
$(sk_1, pk_1) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$
$(sk_2, pk_2) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$
$G_1 \leftarrow_{\$} \mathsf{Enc}(pk_1, C_0)$
$G_2 \leftarrow_{\$} \mathsf{Enc}(pk_2, C_1)$

Details: Game $1 \longrightarrow$ Game 2

Given $(\mathcal{S}, \mathcal{A})$ that distinguishes Game 1 from Game 2

Algo. $\overline{\mathcal{S}}^{\mathcal{S}}(pk_2)$:

$(C_0, C_1, st) \leftarrow\!\!\$\ \mathcal{S}(1^\lambda)$

$(sk_1, pk_1) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$(sk_2, pk_2) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$

$G_1 \leftarrow\!\!\$\ \mathsf{Enc}(pk_1, C_0)$

$G_2 \leftarrow\!\!\$\ \mathsf{Enc}(pk_2, C_1)$

Construct $P_0 := P[sk_1, G_1, G_2]$

Construct $P_1 := P[sk_2, G_1, G_2]$

Return (P_0, P_1) (as two $\mathbf{NC}^1$ circuits).

Details: Game $1 \longrightarrow$ Game 2

Given $(\mathcal{S}, \mathcal{A})$ that distinguishes Game 1 from Game 2

Algo. $\overline{\mathcal{S}}^{\mathcal{S}}(pk_2)$:

$(C_0, C_1, st) \leftarrow_{\$} \mathcal{S}(1^\lambda)$
$(sk_1, pk_1) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$
$(sk_2, pk_2) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$
$G_1 \leftarrow_{\$} \mathsf{Enc}(pk_1, C_0)$
$G_2 \leftarrow_{\$} \mathsf{Enc}(pk_2, C_1)$
Construct $P_0 := P[sk_1, G_1, G_2]$
Construct $P_1 := P[sk_2, G_1, G_2]$
Return (P_0, P_1) (as two $\mathbf{NC}^1$ circuits).

Algo. $\mathcal{B}^{\mathcal{A}}(\overline{P}, st)$:

$b' \leftarrow_{\$} \mathcal{A}(\overline{P}, st)$
return b'

Why is $\overline{\mathcal{S}}$ Eq-secure?

Need to show:

$$P[sk_1, G_1, G_2] \equiv P[sk_2, G_1, G_2]$$

Recall:

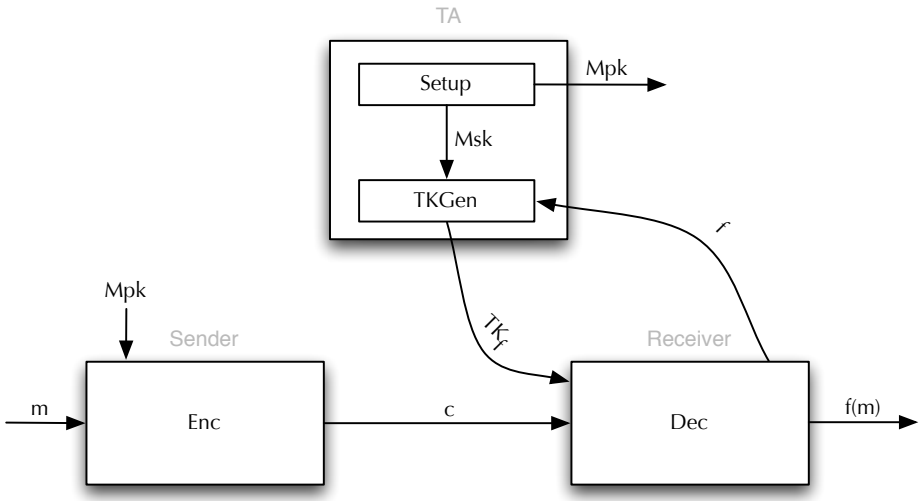
$$P[sk_1, G_1, G_2](m, E_1, E_2, \pi) := \text{if } \mathsf{V}((m, E_1, E_2), \pi) \text{ ret. } \mathsf{Dec}(sk_1, E_1)$$
$$\text{else return } \perp$$

This follows from:

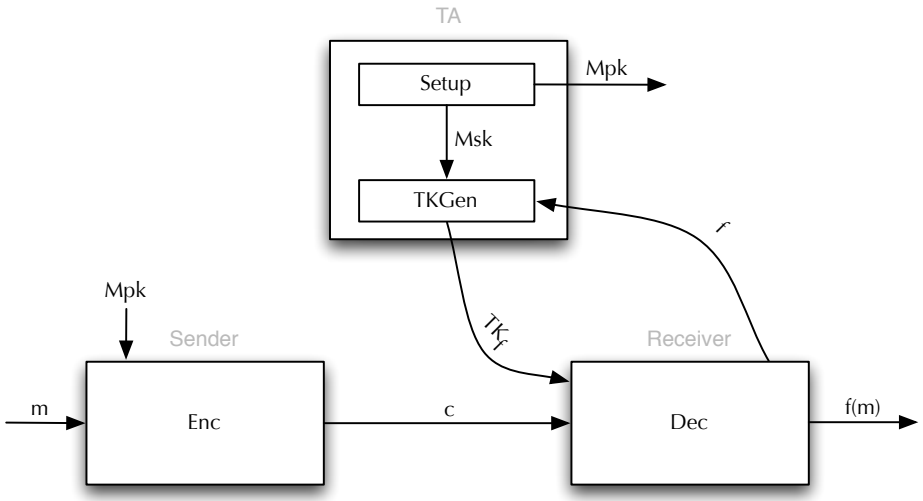
- Proof π says E_1 and E_2 are correctly evaluated.
- The proof system is **perfectly** sound.
- FHE is **perfectly** correct.
- Circuits C_0 and C_1 are functionally equivalent.

Applications of iO

Functional Encryption (FE)



Functional Encryption (FE)



$$
\begin{aligned}
(Msk, Mpk) &\leftarrow\$ \quad \mathsf{Setup}(1^\lambda) \\
TK_f &\leftarrow\$ \quad \mathsf{TKGen}(f, Msk) \\
c &\leftarrow\$ \quad \mathsf{Enc}(\mathsf{m}, Mpk) \\
f(\mathsf{m}) / \perp &= \quad \mathsf{Dec}(c, TK_f)
\end{aligned}
$$

Attribute-Based Encryption

Fine-grained access to encrypted data.

$$f[\mathrm{P}](\mathrm{A}, m) := \begin{cases} m & \text{if } \mathrm{P}(\mathrm{A}) = 1 \ ; \\ \perp & \text{otherwise.} \end{cases}$$

Attribute-Based Encryption

Fine-grained access to encrypted data.

$$f[\mathrm{P}](\mathrm{A}, m) := \begin{cases} m & \text{if } \mathrm{P}(\mathrm{A}) = 1 \ ; \\ \bot & \text{otherwise.} \end{cases}$$

For instance:

- TA checks users' age $A \in [100]$ and issues secret key sk_A
- Policy can be set to

$$\mathrm{P}(\mathrm{A}) := (A \geq 18)$$

Attribute-Based Encryption

Fine-grained access to encrypted data.

$$f[\mathrm{P}](\mathrm{A}, m) := \begin{cases} m & \text{if } \mathrm{P}(\mathrm{A}) = 1 \ ; \\ \bot & \text{otherwise.} \end{cases}$$

For instance:

- TA checks users' age $A \in [100]$ and issues secret key sk_A
- Policy can be set to

$$\mathrm{P}(\mathrm{A}) := (A \geq 18)$$

- As a result, only adults can decrypt ciphertexts.

FE Security

Intuition: Requires that:

$$(\mathsf{Enc}(m_0), TK_{f_1}, \ldots, TK_{f_n}) \overset{c}{\approx} (\mathsf{Enc}(m_1), TK_{f_1}, \ldots, TK_{f_n})$$

as long as $f_i(m_0) = f_i(m_1)$ for all i.

FE Security

Intuition: Requires that:

$$(\mathsf{Enc}(m_0), TK_{f_1}, \ldots, TK_{f_n}) \overset{c}{\approx} (\mathsf{Enc}(m_1), TK_{f_1}, \ldots, TK_{f_n})$$

as long as $f_i(m_0) = f_i(m_1)$ for all i.

$\underline{\text{IND-CPA}^{\mathcal{A}}_{\mathsf{FE}}(1^\lambda)}$:
$(m_0, m_1, st) \leftarrow\$\ \mathcal{A}_1^{\text{TOKEN}}(mpk)$
$(msk, mpk) \leftarrow\$\ \mathsf{Setup}(1^\lambda)$
$b \leftarrow\$\ \{0,1\}$
$c \leftarrow\$\ \mathsf{Enc}(mpk, m_b)$
$b' \leftarrow\$\ \mathcal{A}_2^{\text{TOKEN}}(c, st)$
return $(b = b')$

$\underline{\text{TOKEN}(f)}$:
$TK \leftarrow\$\ \mathsf{TKGen}(msk, f)$
$\mathsf{TKList} \leftarrow \mathsf{TKList} : f$
return TK

FE Security

Intuition: Requires that:

$$(\mathsf{Enc}(m_0), TK_{f_1}, \ldots, TK_{f_n}) \overset{c}{\approx} (\mathsf{Enc}(m_1), TK_{f_1}, \ldots, TK_{f_n})$$

as long as $f_i(m_0) = f_i(m_1)$ for all i.

$\underline{\text{IND-CPA}^{\mathcal{A}}_{\mathsf{FE}}(1^\lambda)}$:
$(m_0, m_1, st) \leftarrow\$\ \mathcal{A}_1^{\text{TOKEN}}(mpk)$
$(msk, mpk) \leftarrow\$\ \mathsf{Setup}(1^\lambda)$
$b \leftarrow\$\ \{0,1\}$
$c \leftarrow\$\ \mathsf{Enc}(mpk, m_b)$
$b' \leftarrow\$\ \mathcal{A}_2^{\text{TOKEN}}(c, st)$
return $(b = b')$

$\underline{\text{TOKEN}(f)}$:
$TK \leftarrow\$\ \mathsf{TKGen}(msk, f)$
$\mathsf{TKList} \leftarrow \mathsf{TKList} : f$
return TK

Legitimacy:

$$\forall f \in \mathsf{TKList} \ : \ f(m_0) = f(m_1) \ .$$

FE Security

Intuition: Requires that:

$$(\mathsf{Enc}(m_0), TK_{f_1}, \ldots, TK_{f_n}) \overset{c}{\approx} (\mathsf{Enc}(m_1), TK_{f_1}, \ldots, TK_{f_n})$$

as long as $f_i(m_0) = f_i(m_1)$ for all i.

$\underline{\text{IND-CPA}^{\mathcal{A}}_{\mathsf{FE}}(1^\lambda)}$:
$(m_0, m_1, st) \leftarrow\$\ \mathcal{A}_1^{\text{TOKEN}}(mpk)$
$(msk, mpk) \leftarrow\$\ \mathsf{Setup}(1^\lambda)$
$b \leftarrow\$\ \{0,1\}$
$c \leftarrow\$\ \mathsf{Enc}(mpk, m_b)$
$b' \leftarrow\$\ \mathcal{A}_2^{\text{TOKEN}}(c, st)$
return $(b = b')$

$\underline{\text{TOKEN}(f)}$:
$TK \leftarrow\$\ \mathsf{TKGen}(msk, f)$
$\mathsf{TKList} \leftarrow \mathsf{TKList} : f$
return TK

Legitimacy:

$$\forall f \in \mathsf{TKList} \ : \ f(m_0) = f(m_1) \ .$$

Advantage:

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathsf{FE},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\text{IND-CPA}^{\mathcal{A}}_{\mathsf{FE}}(1^\lambda)\right] - 1$$

The iO-to-FE Transform

Intuition:

Setup(1^λ):	$msk = sk$ and $mpk = pk$.
Enc(mpk, m):	Encrypt using PKEnc(pk, m).
TKGen(msk, f):	$TK_f =$ Obfuscation of $f($PKDec$(sk, \cdot))$.
Dec(TK_f, c):	Run TK_f on c.

To "switch" keys/messages in the reduction, attach proofs of consistency.

FE Construction – Details

$\mathsf{Setup}(1^\lambda)$: Set $mpk := (pk_1, pk_2, \sigma)$ and $msk = sk_1$.

FE Construction – Details

$\mathsf{Setup}(1^\lambda)$: Set $mpk := (pk_1, pk_2, \sigma)$ and $msk = sk_1$.

$\mathsf{Enc}(mpk, m)$: Output $E_1 = \mathsf{PKEnc}(pk_1, m)$ and $E_1 = \mathsf{PKEnc}(pk_1, m)$
And a **NIZK proof** π of well-formedness:

$$\exists\,(m, r_1, r_2) \;: E_1 = \mathsf{PKEnc}(pk_1, m; r_1)$$
$$\wedge\; E_2 = \mathsf{PKEnc}(pk_2, m; r_2)$$

FE Construction – Details

$\mathsf{Setup}(1^\lambda)$: Set $mpk := (pk_1, pk_2, \sigma)$ and $msk = sk_1$.

$\mathsf{Enc}(mpk, m)$: Output $E_1 = \mathsf{PKEnc}(pk_1, m)$ and $E_1 = \mathsf{PKEnc}(pk_1, m)$
And a **NIZK proof** π of well-formedness:

$$\exists\,(m, r_1, r_2) \;: E_1 = \mathsf{PKEnc}(pk_1, m; r_1)$$
$$\wedge\; E_2 = \mathsf{PKEnc}(pk_2, m; r_2)$$

$\mathsf{TKGen}(msk, f)$: Output an obfuscation of:

$$P[sk_1, f, \sigma](E_1, E_2, \pi) := \text{if } \mathsf{V}(\sigma, (E_1, E_2), \pi) \text{ ret. } f(\mathsf{PKDec}(sk_1, E_1))$$
$$\text{else return } \perp$$

FE Construction – Details

$\mathsf{Setup}(1^\lambda)$: Set $mpk := (pk_1, pk_2, \sigma)$ and $msk = sk_1$.

$\mathsf{Enc}(mpk, m)$: Output $E_1 = \mathsf{PKEnc}(pk_1, m)$ and $E_1 = \mathsf{PKEnc}(pk_1, m)$
And a **NIZK proof** π of well-formedness:

$$\exists\,(m, r_1, r_2) \;: E_1 = \mathsf{PKEnc}(pk_1, m; r_1)$$
$$\wedge\; E_2 = \mathsf{PKEnc}(pk_2, m; r_2)$$

$\mathsf{TKGen}(msk, f)$: Output an obfuscation of:

$$P[sk_1, f, \sigma](E_1, E_2, \pi) := \text{if } \mathsf{V}(\sigma, (E_1, E_2), \pi) \text{ ret. } f(\mathsf{PKDec}(sk_1, E_1))$$
$$\text{else return } \perp$$

$\mathsf{Dec}(c, TK_f)$: Run TK_f on $c = (E_1, E_2, \pi)$.

Non-Interactive Zero-Knowledge Proof

A NIZK for NP relation R is a ppt triple $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ such that:

Non-Interactive Zero-Knowledge Proof

A NIZK for NP relation R is a ppt triple $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ such that:

Perfect Completeness : For all (x, w) with $R(x, w) = 1$

$$\Pr[\sigma \leftarrow\!\!\$\ \mathsf{K}(1^\lambda); \pi \leftarrow\!\!\$\ \mathsf{P}(\sigma, x, w) : \mathsf{V}(\sigma, x, \pi) = 1] = 1$$

Non-Interactive Zero-Knowledge Proof

A NIZK for NP relation R is a ppt triple $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ such that:

Perfect Completeness : For all (x, w) with $R(x, w) = 1$

$$\Pr[\sigma \leftarrow_{\$} \mathsf{K}(1^{\lambda}); \pi \leftarrow_{\$} \mathsf{P}(\sigma, x, w) : \mathsf{V}(\sigma, x, \pi) = 1] = 1$$

Statistical Soundness : For all (even unbounded) P^*

$$\Pr[\sigma \leftarrow_{\$} \mathsf{K}(1^{\lambda}); \pi \leftarrow_{\$} \mathsf{P}^*(\sigma) : \mathsf{V}(\sigma, x, \pi) = 1 \wedge x \notin R] \in \mathrm{Negl}$$

Non-Interactive Zero-Knowledge Proof

A NIZK for NP relation R is a ppt triple $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ such that:

Perfect Completeness : For all (x, w) with $R(x, w) = 1$

$$\Pr[\sigma \leftarrow_{\$} \mathsf{K}(1^{\lambda}); \pi \leftarrow_{\$} \mathsf{P}(\sigma, x, w) : \mathsf{V}(\sigma, x, \pi) = 1] = 1$$

Statistical Soundness : For all (even unbounded) P^*

$$\Pr[\sigma \leftarrow_{\$} \mathsf{K}(1^{\lambda}); \pi \leftarrow_{\$} \mathsf{P}^*(\sigma) : \mathsf{V}(\sigma, x, \pi) = 1 \wedge x \notin R] \in \mathrm{Negl}$$

Computational ZK : There is a $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ st. for all ppt $\mathcal{A}$ and all $x \in R$

$$\Pr[\sigma \leftarrow_{\$} \mathsf{K}(1^{\lambda}); \pi \leftarrow_{\$} \mathsf{P}(\sigma, x, w) : \mathcal{A}(\sigma, x, \pi) = 1]$$
$$\overset{c}{\approx} \Pr[(\sigma, \tau) \leftarrow_{\$} \mathcal{S}_1(1^{\lambda}, x); \pi \leftarrow_{\$} \mathcal{S}_2(\sigma, x, \tau) : \mathcal{A}(\sigma, x, \pi) = 1]$$

Statistical Simulation Soundness

Statistical Simulation Soundness

Stat. Sim. Soundness : For all (even unbounded) P^* and $x \in R$

$$\Pr[(\sigma,\tau) \leftarrow\!\!\$\ \mathcal{S}_1(1^\lambda, x); \pi \leftarrow\!\!\$\ \mathcal{S}_2(\sigma, x, \tau); (x', \pi') \leftarrow\!\!\$\ \mathsf{P}^*(\sigma, x, \pi) : \\ : x \neq x' \wedge \mathsf{V}(\sigma, x', \pi') = 1 \wedge x' \notin R] \in \mathrm{Negl}$$

Proof of Security

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$
4	Sim	m_1	m_1	sk_2	sk_2	

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$
4	Sim	m_1	m_1	sk_2	sk_2	IND-CPA wrt. pk_1

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$
4	Sim	m_1	m_1	sk_2	sk_2	IND-CPA wrt. pk_1
$(5, i)$	Sim	m_1	m_1	sk_1	sk_2	

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$
4	Sim	m_1	m_1	sk_2	sk_2	IND-CPA wrt. pk_1
$(5, i)$	Sim	m_1	m_1	sk_1	sk_2	iO + SSS

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$
4	Sim	m_1	m_1	sk_2	sk_2	IND-CPA wrt. pk_1
$(5, i)$	Sim	m_1	m_1	sk_1	sk_2	iO + SSS
$(5, q)$	Sim	m_1	m_1	sk_1	sk_1	$i = q$

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$
4	Sim	m_1	m_1	sk_2	sk_2	IND-CPA wrt. pk_1
$(5, i)$	Sim	m_1	m_1	sk_1	sk_2	iO + SSS
$(5, q)$	Sim	m_1	m_1	sk_1	sk_1	$i = q$
6	Real	m_1	m_1	sk_1	sk_1	

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$
4	Sim	m_1	m_1	sk_2	sk_2	IND-CPA wrt. pk_1
$(5, i)$	Sim	m_1	m_1	sk_1	sk_2	iO + SSS
$(5, q)$	Sim	m_1	m_1	sk_1	sk_1	$i = q$
6	Real	m_1	m_1	sk_1	sk_1	Zero Knowledge

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$
4	Sim	m_1	m_1	sk_2	sk_2	IND-CPA wrt. pk_1
$(5, i)$	Sim	m_1	m_1	sk_1	sk_2	iO + SSS
$(5, q)$	Sim	m_1	m_1	sk_1	sk_1	$i = q$
6	Real	m_1	m_1	sk_1	sk_1	Zero Knowledge

Proof of Security

Game	σ, π	E_1 contains	E_2 contains	$P[sk, f_i]$ knows	$P[sk, f_{i+1}]$ knows	**Remark**
0	Real	m_0	m_0	sk_1	sk_1	IND-CPA for m_0
1	Sim	m_0	m_0	sk_1	sk_1	Zero Knowledge
2	Sim	m_0	m_1	sk_1	sk_1	IND-CPA wrt. pk_2
$(3, i)$	Sim	m_0	m_1	sk_2	sk_1	iO + SSS
$(3, q)$	Sim	m_0	m_1	sk_2	sk_2	$i = q$
4	Sim	m_1	m_1	sk_2	sk_2	IND-CPA wrt. pk_1
$(5, i)$	Sim	m_1	m_1	sk_1	sk_2	iO + SSS
$(5, q)$	Sim	m_1	m_1	sk_1	sk_1	$i = q$
6	Real	m_1	m_1	sk_1	sk_1	Zero Knowledge IND-CPA for m_1

Details: Game $(3, i) \longrightarrow$ Game $(3, i+1)$

Legitimacy requires:

$$P[sk_1, f_{i+1}, \sigma] \equiv P[sk_2, f_{i+1}, \sigma]$$

Details: Game $(3, i) \longrightarrow$ Game $(3, i+1)$

Legitimacy requires:

$$P[sk_1, f_{i+1}, \sigma] \equiv P[sk_2, f_{i+1}, \sigma]$$

Let (E_1, E_2, π) be an input:

Details: Game $(3, i) \longrightarrow$ Game $(3, i+1)$

Legitimacy requires:

$$P[sk_1, f_{i+1}, \sigma] \equiv P[sk_2, f_{i+1}, \sigma]$$

Let (E_1, E_2, π) be an input:

- Verification of π fails: Both circuits output $\perp$

Details: Game $(3, i) \longrightarrow$ Game $(3, i+1)$

Legitimacy requires:

$$P[sk_1, f_{i+1}, \sigma] \equiv P[sk_2, f_{i+1}, \sigma]$$

Let (E_1, E_2, π) be an input:

- Verification of π fails: Both circuits output $\perp$
- Verification passes:

Details: Game $(3, i) \longrightarrow$ Game $(3, i+1)$

Legitimacy requires:

$$P[sk_1, f_{i+1}, \sigma] \equiv P[sk_2, f_{i+1}, \sigma]$$

Let (E_1, E_2, π) be an input:

- Verification of π fails: Both circuits output $\perp$
- Verification passes:
  - E_1 and E_2 valid (encrypt the same message m). Both output $f_{i+1}(m)$.

Details: Game $(3, i) \longrightarrow$ Game $(3, i+1)$

Legitimacy requires:

$$P[sk_1, f_{i+1}, \sigma] \equiv P[sk_2, f_{i+1}, \sigma]$$

Let (E_1, E_2, π) be an input:

- Verification of π fails: Both circuits output $\perp$
- Verification passes:
  - E_1 and E_2 valid (encrypt the same message m). Both output $f_{i+1}(m)$.
  - E_1 and E_2 are **not** valid.

Details: Game $(3, i) \longrightarrow$ Game $(3, i+1)$

Legitimacy requires:

$$P[sk_1, f_{i+1}, \sigma] \equiv P[sk_2, f_{i+1}, \sigma]$$

Let (E_1, E_2, π) be an input:

- Verification of π fails: Both circuits output $\perp$
- Verification passes:
  - E_1 and E_2 valid (encrypt the same message m). Both output $f_{i+1}(m)$.
  - E_1 and E_2 are **not** valid.
    SSS $\implies$ E_1 and E_2 are identical to challenge ciphertexts.

Details: Game $(3, i) \longrightarrow$ Game $(3, i+1)$

Legitimacy requires:

$$P[sk_1, f_{i+1}, \sigma] \equiv P[sk_2, f_{i+1}, \sigma]$$

Let (E_1, E_2, π) be an input:

- Verification of π fails: Both circuits output $\perp$
- Verification passes:
  - E_1 and E_2 valid (encrypt the same message m). Both output $f_{i+1}(m)$.
  - E_1 and E_2 are **not** valid.
    SSS $\implies$ E_1 and E_2 are identical to challenge ciphertexts.
    By the rules of FE game $f_{i+1}(m_0) = f_{i+1}(m_1)$.

Many Applications

- Witness Encryption
- Deniable Encryption
- Constrained PRFs
- Symmetric-to-asymmetric conversions
- UCE-secure hash functions
- ...

Many Applications

- Witness Encryption
- Deniable Encryption
- Constrained PRFs
- Symmetric-to-asymmetric conversions
- UCE-secure hash functions
- ...

Thank you.