Practical Fully Secure Inner Product Functional Encryption modulo p

Guilhem Castagnos[1] Fabien Laguillaumie[2] Ida Tucker[2]

[1]Université de Bordeaux, INRIA, CNRS, IMB UMR 5251,
F-33405 Talence, France.

[2]Univ Lyon, CNRS, Université Claude Bernard Lyon 1, ENS de Lyon,
INRIA, LIP UMR 5668, F-69007, LYON Cedex 07, France.

Table of contents

Functional Encryption (FE)

Traditional Encryption: All or Nothing

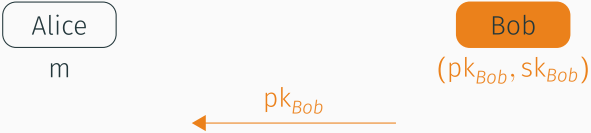
Alice

m

Bob

(pk_{Bob}, sk_{Bob})

Traditional Encryption: All or Nothing



Alice

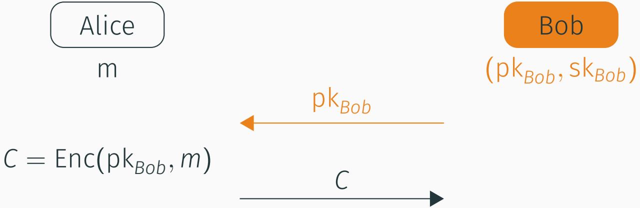
m

Bob

$(\mathrm{pk}_{Bob}, \mathrm{sk}_{Bob})$

pk_{Bob}

Traditional Encryption: All or Nothing



Alice

m

Bob

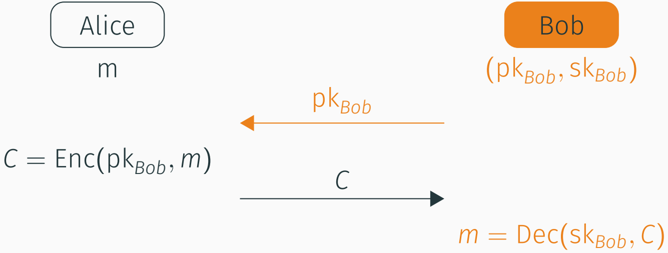
$(\mathrm{pk}_{Bob}, \mathrm{sk}_{Bob})$

pk_{Bob}

$C = \mathrm{Enc}(\mathrm{pk}_{Bob}, m)$

C

Traditional Encryption: All or Nothing



Alice

m

Bob

$(\mathrm{pk}_{Bob}, \mathrm{sk}_{Bob})$

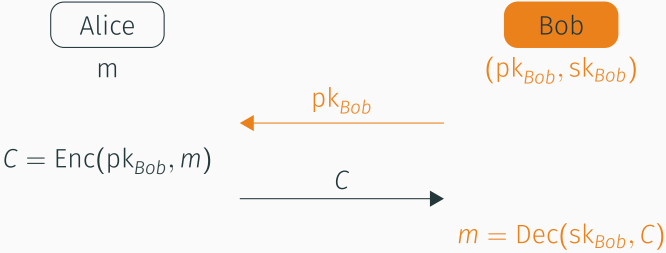
pk_{Bob}

$C = \mathrm{Enc}(\mathrm{pk}_{Bob}, m)$

C

$m = \mathrm{Dec}(\mathrm{sk}_{Bob}, C)$

Traditional Encryption: All or Nothing



Alice

m

Bob

$(\mathrm{pk}_{Bob}, \mathrm{sk}_{Bob})$

pk_{Bob}

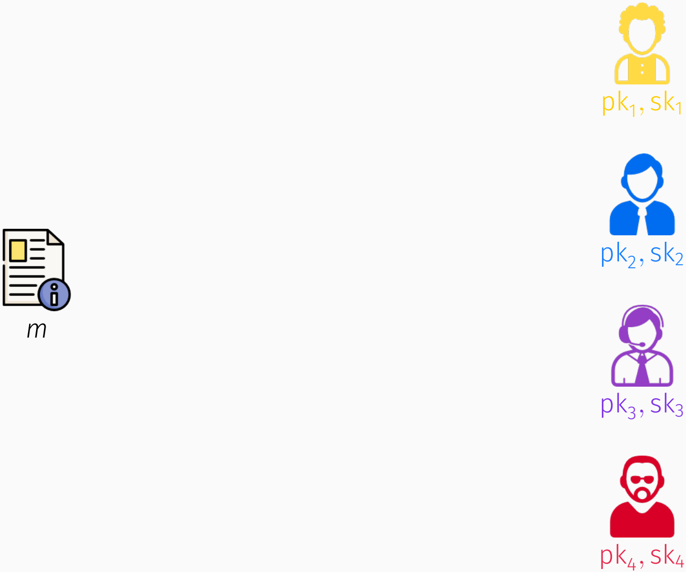
$C = \mathrm{Enc}(\mathrm{pk}_{Bob}, m)$

C

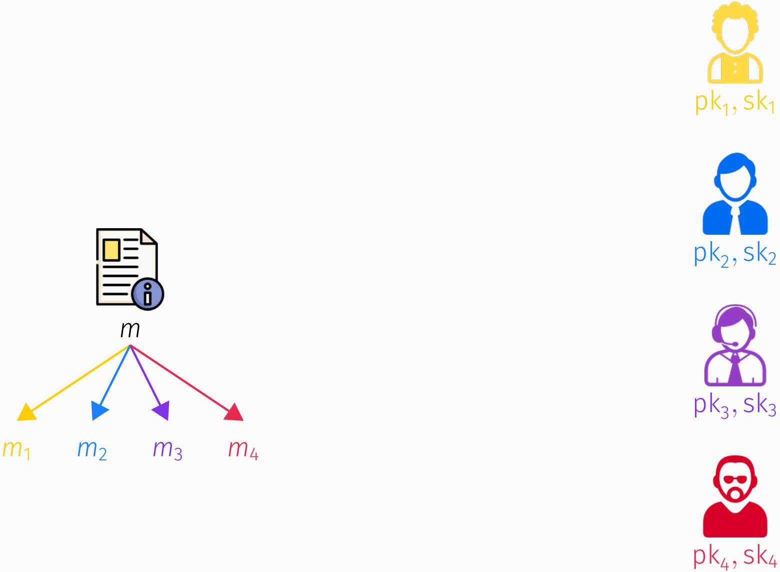
$m = \mathrm{Dec}(\mathrm{sk}_{Bob}, C)$

Bob gets **all** the information in *m*.

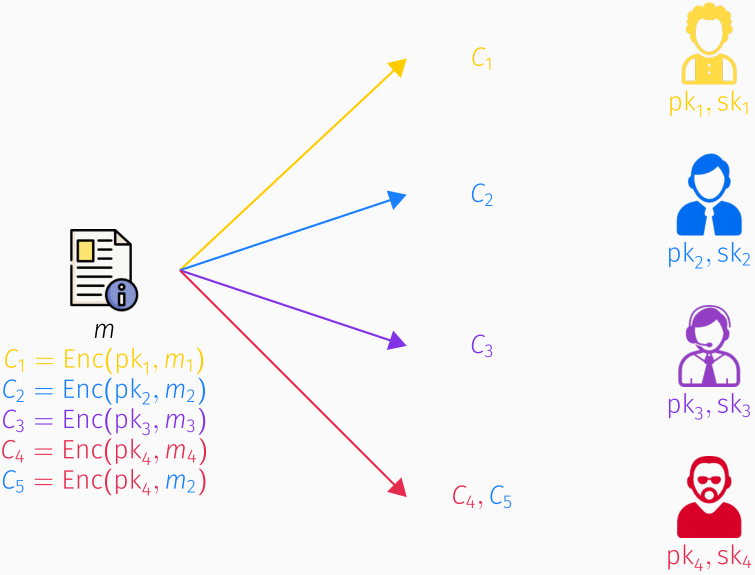
Fine Grained Access to Info with Traditional Encryption



Fine Grained Access to Info with Traditional Encryption



Fine Grained Access to Info with Traditional Encryption



m

$C_1 = \mathsf{Enc}(\mathsf{pk}_1, m_1)$
$C_2 = \mathsf{Enc}(\mathsf{pk}_2, m_2)$
$C_3 = \mathsf{Enc}(\mathsf{pk}_3, m_3)$
$C_4 = \mathsf{Enc}(\mathsf{pk}_4, m_4)$
$C_5 = \mathsf{Enc}(\mathsf{pk}_4, m_2)$

C_1

C_2

C_3

C_4, C_5

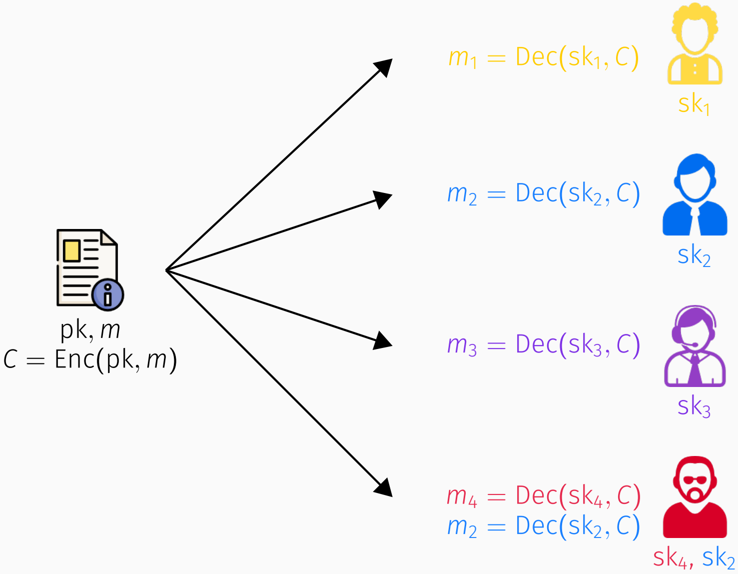
$\mathsf{pk}_1, \mathsf{sk}_1$

$\mathsf{pk}_2, \mathsf{sk}_2$

$\mathsf{pk}_3, \mathsf{sk}_3$

$\mathsf{pk}_4, \mathsf{sk}_4$

Ideal Fine Grained Access to Information



pk, m

$C = \mathsf{Enc}(\mathsf{pk}, m)$

$m_1 = \mathsf{Dec}(\mathsf{sk}_1, C)$

sk_1

$m_2 = \mathsf{Dec}(\mathsf{sk}_2, C)$

sk_2

$m_3 = \mathsf{Dec}(\mathsf{sk}_3, C)$

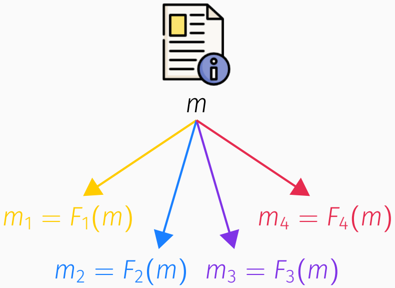
sk_3

$m_4 = \mathsf{Dec}(\mathsf{sk}_4, C)$

$m_2 = \mathsf{Dec}(\mathsf{sk}_2, C)$

sk_4, sk_2

Functional Encryption



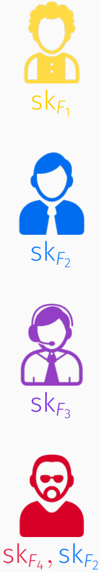
m

$m_1 = F_1(m)$

$m_2 = F_2(m)$

$m_3 = F_3(m)$

$m_4 = F_4(m)$



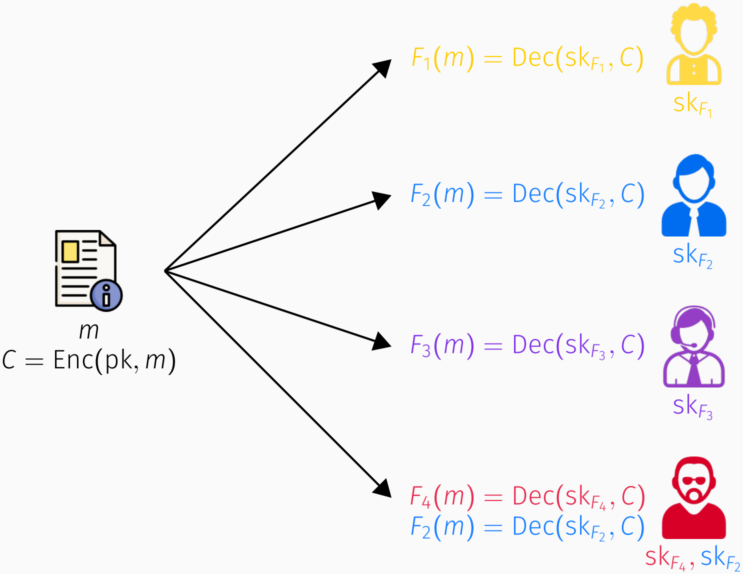
sk_{F_1}

sk_{F_2}

sk_{F_3}

$\mathsf{sk}_{F_4}, \mathsf{sk}_{F_2}$

Functional Encryption



m

$C = \mathrm{Enc}(\mathrm{pk}, m)$

$F_1(m) = \mathrm{Dec}(\mathrm{sk}_{F_1}, C)$

sk_{F_1}

$F_2(m) = \mathrm{Dec}(\mathrm{sk}_{F_2}, C)$

sk_{F_2}

$F_3(m) = \mathrm{Dec}(\mathrm{sk}_{F_3}, C)$

sk_{F_3}

$F_4(m) = \mathrm{Dec}(\mathrm{sk}_{F_4}, C)$

$F_2(m) = \mathrm{Dec}(\mathrm{sk}_{F_2}, C)$

$\mathrm{sk}_{F_4}, \mathrm{sk}_{F_2}$

Application: Spam filtering for encrypted emails

Recipient

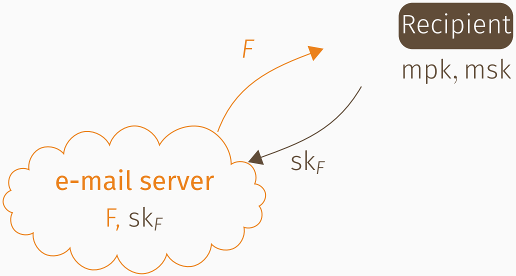
mpk, msk

e-mail server

F

$$F(m) = \begin{cases} 1 & \text{if m is spam} \\ 0 & \text{otherwise} \end{cases}$$

Application: Spam filtering for encrypted emails



Recipient

mpk, msk

F

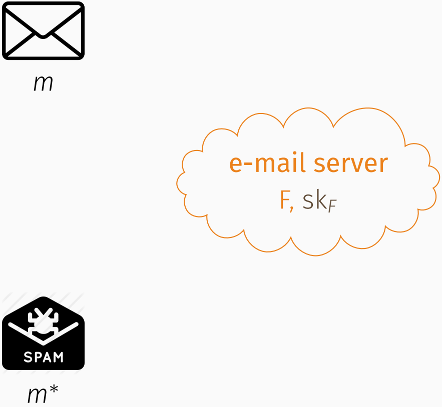
sk_F

e-mail server

F, sk_F

$$F(m) = \begin{cases} 1 & \text{if m is spam} \\ 0 & \text{otherwise} \end{cases}$$

Application: Spam filtering for encrypted emails



m

e-mail server

F, sk_F

m^*

Application: Spam filtering for encrypted emails

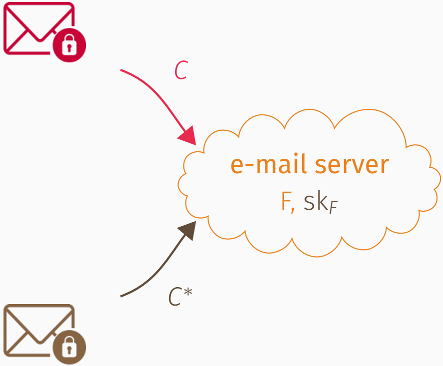
$C = \mathsf{Enc}(\mathsf{mpk}, m)$

e-mail server

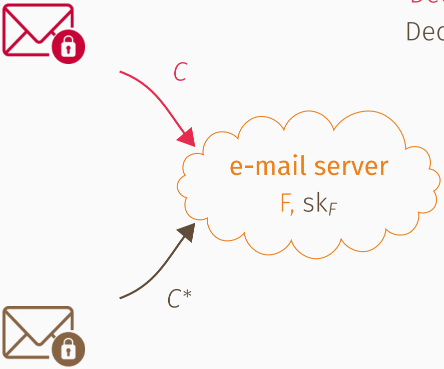
F, sk_F

$C^* = \mathsf{Enc}(\mathsf{mpk}, m^*)$

Application: Spam filtering for encrypted emails



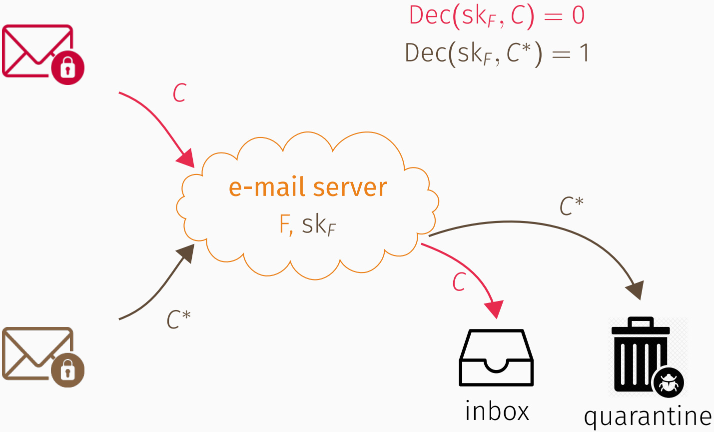
Application: Spam filtering for encrypted emails



$\mathsf{Dec}(\mathsf{sk}_F, C) = 0$

$\mathsf{Dec}(\mathsf{sk}_F, C^*) = 1$

Application: Spam filtering for encrypted emails

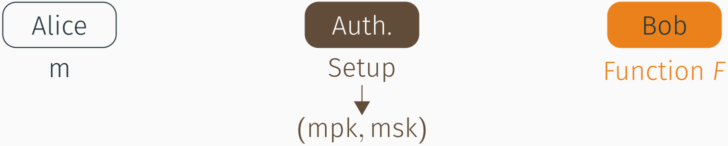


$\mathsf{Dec}(sk_F, C) = 0$

$\mathsf{Dec}(sk_F, C^*) = 1$

e-mail server learns **one bit of information**

Functional Encryption [BSW11]



Alice

m

Auth.

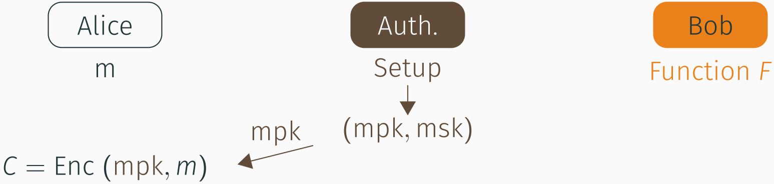
Setup

(mpk, msk)

Bob

Function *F*

Functional Encryption [BSW11]



Alice

m

Auth.

Setup

(mpk, msk)

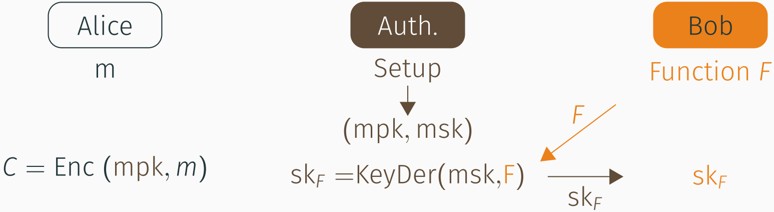
mpk

$C = \mathrm{Enc}\left(\mathrm{mpk}, m\right)$

Bob

Function F

Functional Encryption [BSW11]



Alice

m

$C = \text{Enc}\left(\text{mpk}, m\right)$

Auth.

Setup

(mpk, msk)

$\text{sk}_F = \text{KeyDer}(\text{msk}, F)$ $\xrightarrow[\text{sk}_F]{}$

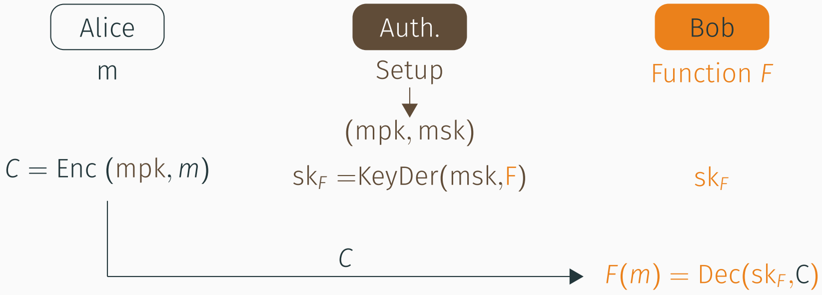
Bob

Function F

F

sk_F

Functional Encryption [BSW11]



Alice

m

Auth.

Setup

(mpk, msk)

Bob

Function F

$C = \mathrm{Enc}\,(\mathrm{mpk}, m)$

$\mathrm{sk}_F = \mathrm{KeyDer}(\mathrm{msk}, F)$

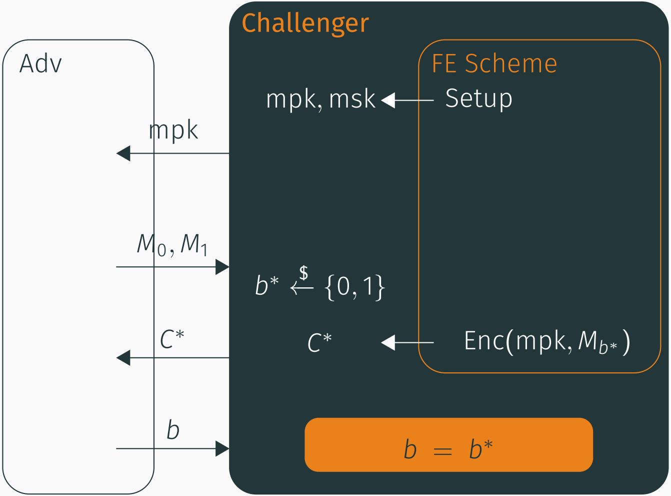
sk_F

C

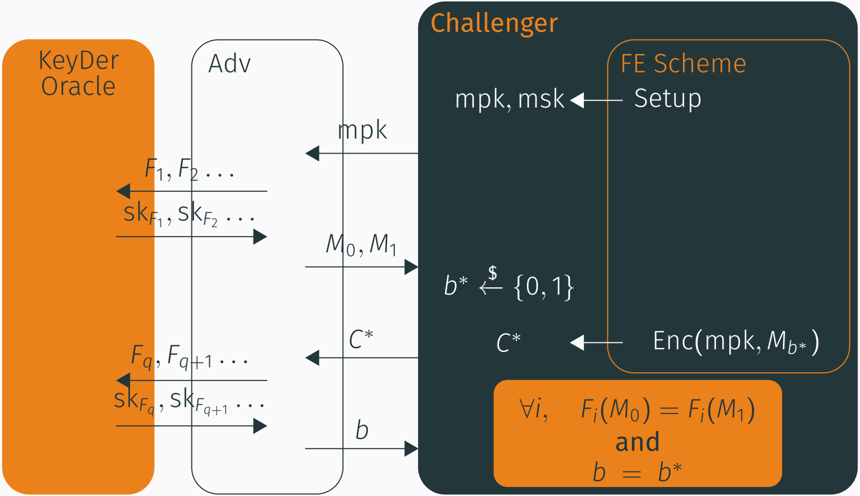
$F(m) = \mathrm{Dec}(\mathrm{sk}_F, \mathrm{C})$

Bob **only learns** $F(m)$.

FE Security – Indistinguishability



FE Security – Indistinguishability



KeyDer Oracle

Adv

Challenger

FE Scheme

mpk, msk ← Setup

mpk

$F_1, F_2 \ldots$

$\mathsf{sk}_{F_1}, \mathsf{sk}_{F_2} \ldots$

M_0, M_1

$b^* \xleftarrow{\$} \{0, 1\}$

C^*

$C^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, M_{b^*})$

$F_q, F_{q+1} \ldots$

$\mathsf{sk}_{F_q}, \mathsf{sk}_{F_{q+1}} \ldots$

b

$\forall i, \quad F_i(M_0) = F_i(M_1)$

and

$b = b^*$

Limits of General Functional Encryption

We don’t know how to build **practical** FE for general functions

Limits of General Functional Encryption

We **don't know** how to build **practical** FE for **general functions**

$\Rightarrow$ **Linear Functions:** simple with many applications

Limits of General Functional Encryption

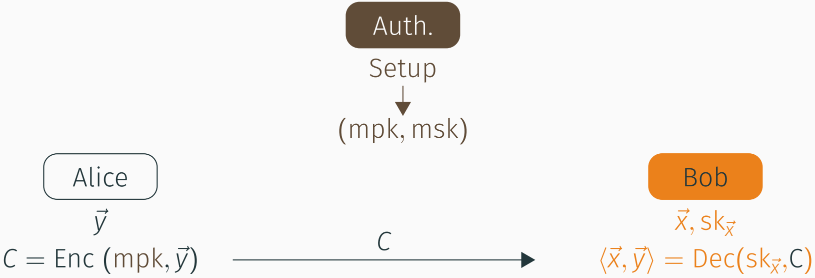
We **don't know** how to build **practical** FE for **general functions**

$\Rightarrow$ **Linear Functions: simple** with **many applications**

- Understand general FE
- Statistical analysis on encrypted data
- Evaluation of polynomials over encrypted data
- Constructing trace-and-revoke system
- etc.

The Inner Product Functionality

The inner product functionality



Auth.

Setup

$\downarrow$

$(\mathrm{mpk}, \mathrm{msk})$

Alice

$\vec{y}$

$C = \mathrm{Enc}\left(\mathrm{mpk}, \vec{y}\right)$

$\xrightarrow{C}$

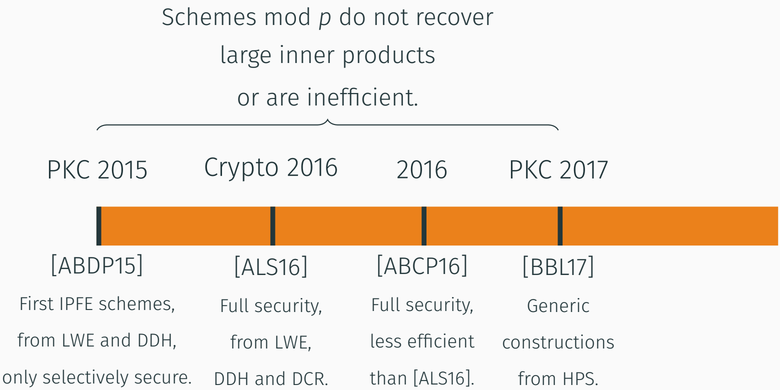
Bob

$\vec{x}, \mathrm{sk}_{\vec{x}}$

$\langle \vec{x}, \vec{y} \rangle = \mathrm{Dec}(\mathrm{sk}_{\vec{x}}, \mathrm{C})$

$$\begin{aligned} F_x : \mathcal{R}^{\ell} &\mapsto \mathcal{R} \\ y &\mapsto \langle \vec{x}, \vec{y} \rangle \end{aligned}$$

Previous work



Schemes mod p do not recover
large inner products
or are inefficient.

PKC 2015

[ABDP15]

First IPFE schemes,
from LWE and DDH,
only selectively secure.

Crypto 2016

[ALS16]

Full security,
from LWE,
DDH and DCR.

2016

[ABCP16]

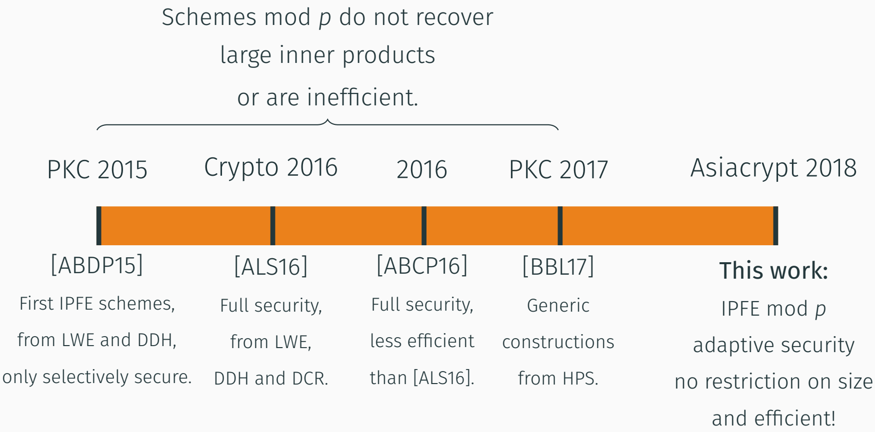
Full security,
less efficient
than [ALS16].

PKC 2017

[BBL17]

Generic
constructions
from HPS.

Previous work



Schemes mod p do not recover large inner products or are inefficient.

PKC 2015

[ABDP15]

First IPFE schemes, from LWE and DDH, only selectively secure.

Crypto 2016

[ALS16]

Full security, from LWE, DDH and DCR.

2016

[ABCP16]

Full security, less efficient than [ALS16].

PKC 2017

[BBL17]

Generic constructions from HPS.

Asiacrypt 2018

This work:

IPFE mod p adaptive security no restriction on size and efficient!

The Hard Subgroup Membership (HSM) Assumption

Framework (sketch) [CL15]

Group with an easy discrete logarithm (DL) subgroup

- $G = \langle g \rangle$ cyclic group of order $p \cdot s$ such that $\gcd(p, s) = 1$.
- p large prime
- $F = \langle f \rangle$ subgroup of G of order p.
- $G^p = \langle g_p \rangle = \{x^p, x \in G\}$ subgroup of G of order s,

$$G = F \times G^p.$$

- DL is **easy** in F (DL: given f and $h = f^x$, find $x \in \mathbf{Z}/p\mathbf{Z}$)

New Assumption

Hard Subgroup Membership problem HSM:

Hard to distinguish p-th powers in G

$$\{x \xleftarrow{\$} G\} \approx_c \{x \xleftarrow{\$} G^p\}.$$

Instantiation in class groups of an imaginary quadratic field

- $K = \mathbb{Q}(\sqrt{\Delta_K})$, $\Delta_K < 0$ and $\Delta_K \equiv 1 \mod 4$

Instantiation in class groups of an imaginary quadratic field

- $K = \mathbb{Q}(\sqrt{\Delta_K})$, $\Delta_K < 0$ and $\Delta_K \equiv 1 \mod 4$
- $\mathcal{O}_{\Delta_K}$ and $\mathcal{O}_{\Delta_p}$ s.t. $\Delta_K = -pq$, $\Delta_p = -qp^3$ with p, q primes

Instantiation in class groups of an imaginary quadratic field

- $K = \mathbb{Q}(\sqrt{\Delta_K})$, $\Delta_K < 0$ and $\Delta_K \equiv 1 \mod 4$
- $\mathcal{O}_{\Delta_K}$ and $\mathcal{O}_{\Delta_p}$ s.t. $\Delta_K = -pq$, $\Delta_p = -qp^3$ with p, q primes
- $\phi_p : C(\mathcal{O}_{\Delta_p}) \mapsto C(\mathcal{O}_{\Delta_K})$ **surjection** where $\mathrm{Ker}(\phi_\mathrm{p})$ of order p.
  - Implies $h(\mathcal{O}_{\Delta_p}) = p \times h(\mathcal{O}_{\Delta_K})$

Instantiation in class groups of an imaginary quadratic field

- $K = \mathbb{Q}(\sqrt{\Delta_K})$, $\Delta_K < 0$ and $\Delta_K \equiv 1 \mod 4$
- $\mathcal{O}_{\Delta_K}$ and $\mathcal{O}_{\Delta_p}$ s.t. $\Delta_K = -pq$, $\Delta_p = -qp^3$ with p, q primes
- $\phi_p : C(\mathcal{O}_{\Delta_p}) \mapsto C(\mathcal{O}_{\Delta_K})$ **surjection** where $\mathrm{Ker}(\phi_\mathrm{p})$ of order p.
    - Implies $h(\mathcal{O}_{\Delta_p}) = p \times h(\mathcal{O}_{\Delta_K})$
- $\mathfrak{a}$ ideal of $\mathcal{O}_\Delta$ can be written as $\mathfrak{a} = (a\mathbf{Z} + \frac{-b+\sqrt{\Delta}}{2}\mathbf{Z})$ and represented by (a, b); for $a \in \mathbf{N}, b \in \mathbf{Z}, b^2 \equiv \Delta \mod 4a$

Instantiation in class groups of an imaginary quadratic field

- $\mathfrak{t} = (p^2, p) \in \mathcal{O}_{\Delta_p}$, set $f = [\mathfrak{t}]$
  $\Rightarrow f$ generates $\mathrm{Ker}(\phi_{\mathrm{p}})$ (subgroup of order p of $C(\mathcal{O}_{\Delta_p})$), and

$$f^m = \left[p^2\mathbf{Z} + \frac{-L(m)p + \sqrt{\Delta_p}}{2}\mathbf{Z} \right]$$

$L(m)$: odd integer in $[-p, p]$ s.t. $L(m) = 1/m \mod p$

$F = <f>$ **cyclic group of order** p**, and** DL **easy**

Instantiation in class groups of an imaginary quadratic field

- $\mathfrak{t} = (p^2, p) \in \mathcal{O}_{\Delta_p}$, set $f = [\mathfrak{t}]$
  $\Rightarrow f$ generates $\mathrm{Ker}(\phi_\mathrm{p})$ (subgroup of order p of $C(\mathcal{O}_{\Delta_p})$), and

$$f^m = \left[p^2\mathbf{Z} + \frac{-L(m)p + \sqrt{\Delta_p}}{2}\mathbf{Z} \right]$$

  $L(m)$: odd integer in $[-p, p]$ s.t. $L(m) = 1/m \mod p$
  $F = < f >$ cyclic group of order p, and DL easy
- To build G^p:
  - $\hat{g} \xleftarrow{\$} C(\mathcal{O}_{\Delta_K})$ of order $s|h(\mathcal{O}_{\Delta_K})$.
  - $\gcd(p, h(\mathcal{O}_{\Delta_K})) = 1 \Rightarrow \gcd(p, s) = 1$
  - $g_p = (\phi_p^{-1}(\hat{g}))^p \in C(\mathcal{O}_{\Delta_p})$

Instantiation in class groups of an imaginary quadratic field

- $\mathfrak{t} = (p^2, p) \in \mathcal{O}_{\Delta_p}$, set $f = [\mathfrak{t}]$
  $\Rightarrow f$ generates $\mathrm{Ker}(\phi_{\mathrm{p}})$ (subgroup of order p of $C(\mathcal{O}_{\Delta_p})$), and

$$f^m = \left[p^2\mathbf{Z} + \frac{-L(m)p + \sqrt{\Delta_p}}{2}\mathbf{Z} \right]$$

  $L(m)$: odd integer in $[-p, p]$ s.t. $L(m) = 1/m \mod p$
  $F =< f >$ **cyclic group of order** p**, and** DL **easy**

- **To build** G^p**:**
  - $\hat{g} \xleftarrow{\$} C(\mathcal{O}_{\Delta_K})$ of order $s | h(\mathcal{O}_{\Delta_K})$.
  - $\mathbf{gcd}(p, h(\mathcal{O}_{\Delta_K})) = 1 \Rightarrow \mathbf{gcd}(p, s) = 1$
  - $g_p = (\phi_p^{-1}(\hat{g}))^p \in C(\mathcal{O}_{\Delta_p})$
- Set $g = g_p \cdot f$ and $G =< g >$ of order ps

Security in class groups of an imaginary quadratic field

- Security from **hardness of class number computation** and **DL problem in $C(\mathcal{O}_{\Delta_K})$**.
- Best known algos use index calculus method

  $\Rightarrow L(1/2)$ complexity
- **Shorter keys!**

	$\lambda = 112$		$\lambda = 128$	
size	this work	DCR	this work	DCR
$(p, \tilde{s})$	$(112, 684)$	$(1024, 2046)$	$(128, 924)$	$(1536, 3070)$
el^t of G	1572	4096	2084	6144
secret key	$112(\ell+1)+684$	$2048(\ell+2)$	$128(\ell+1)+924$	$3072(\ell+2)$

Sampling exponents

Problem

s **unknown**, so orders of G^p and G unknown

⇒ **Cannot sample uniformly** from G or G^p!

Sampling exponents

Problem

s **unknown**, so orders of G^p and G unknown

$\Rightarrow$ **Cannot sample uniformly** from G or G^p!

Solution

- Bound on $h(\mathcal{O}_{\Delta_K}) \Rightarrow$ upper bound $\tilde{s}$ for s
- Use $\tilde{s}$ to instantiate distributions $\mathcal{D}$ and $\mathcal{D}_p$ s.t.

$$\{g^x, x \hookleftarrow \mathcal{D}\} \approx \mathcal{U}(G),$$

$$\text{and } \{g_p^x, x \hookleftarrow \mathcal{D}_p\} \approx \mathcal{U}(G^p)$$

- **In practice:** $\mathcal{D}$ and $\mathcal{D}_p$ folded gaussian distributions with large standard deviation.

Linearly Homomorphic Public Key Encryption mod p from HSM

Homomorphic PKE scheme mod p from HSM

KeyGen Sample $t \hookleftarrow \mathcal{D}_p$ and compute $h = g_p^t$

$$\mathsf{sk} = t \text{ and } \mathsf{pk} = h$$

Homomorphic PKE scheme mod p from HSM

KeyGen Sample $t \hookleftarrow \mathcal{D}_p$ and compute $h = g_p^t$

$$\mathsf{sk} = t \text{ and } \mathsf{pk} = h$$

Enc Plaintext: $m \in \mathbf{Z}/p\mathbf{Z}$

Sample randomness $r \hookleftarrow \mathcal{D}_p$

Ciphertext:

$$(C_0, C_1) = (g_p^r, f^m \cdot h^r)$$

Homomorphic PKE scheme mod p from HSM

KeyGen Sample $t \hookleftarrow \mathcal{D}_p$ and compute $h = g_p^t$

$$\mathsf{sk} = t \text{ and } \mathsf{pk} = h$$

Enc **Plaintext:** $m \in \mathbf{Z}/p\mathbf{Z}$

Sample randomness $r \hookleftarrow \mathcal{D}_p$

Ciphertext:

$$(C_0, C_1) = (g_p^r, f^m \cdot h^r)$$

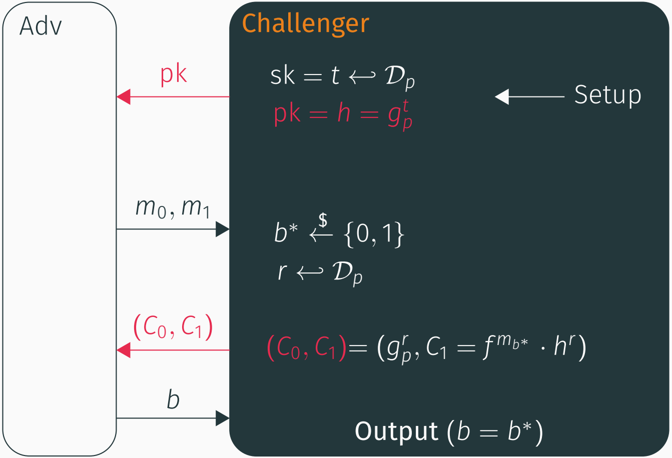
Dec **From** (C_0, C_1) **and** $\mathsf{sk} = t$ **:**

$$C_0/C_1^t \xrightarrow{\text{DL}} m \mod p$$

Security

This scheme is semantically secure under the HSM assumption.

Game 0: the original security experiment



Adv

pk

Challenger

$\mathsf{sk} = t \hookleftarrow \mathcal{D}_p$

$\mathsf{pk} = h = g_p^t$

Setup

m_0, m_1

$b^* \xleftarrow{\$} \{0,1\}$

$r \hookleftarrow \mathcal{D}_p$

(C_0, C_1)

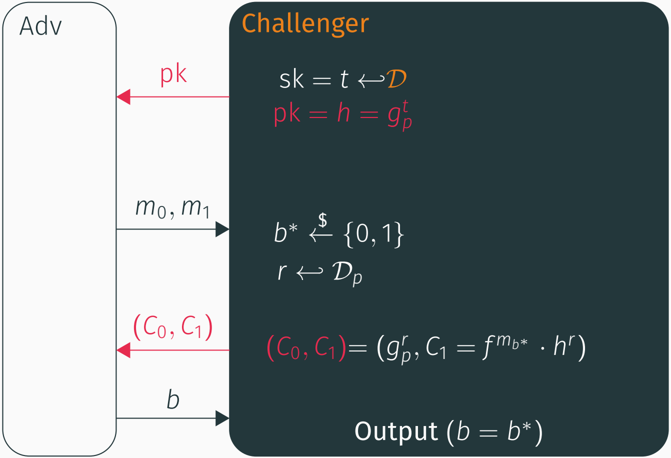
$(C_0, C_1) = (g_p^r, C_1 = f^{m_{b^*}} \cdot h^r)$

b

Output $(b = b^*)$

Game 0 is the original security experiment.

Game 1: sample t from $\mathcal{D}$



Adv

Challenger

pk

$\mathrm{sk} = t \hookleftarrow \mathcal{D}$

$\mathrm{pk} = h = g_p^t$

m_0, m_1

$b^* \overset{\$}{\leftarrow} \{0,1\}$

$r \hookleftarrow \mathcal{D}_p$

(C_0, C_1)

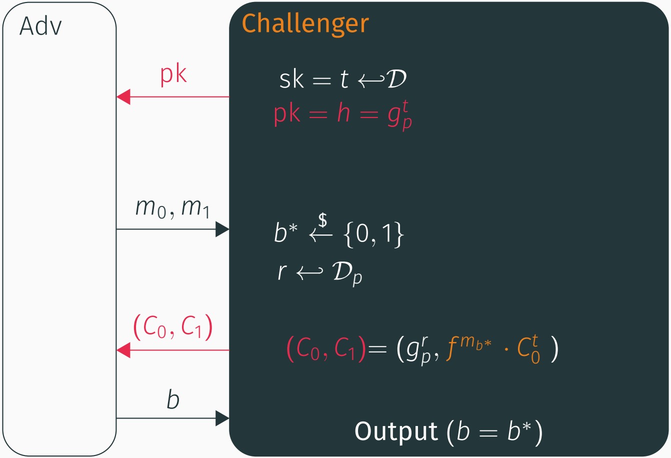
$(C_0, C_1) = (g_p^r, C_1 = f^{m_{b*}} \cdot h^r)$

b

Output $(b = b^*)$

From $\mathcal{A}$'s view, Games 0 and 1 are identical.

Game 2: use sk to compute (C_0, C_1)



Adv

Challenger

pk

$\mathsf{sk} = t \hookleftarrow \mathcal{D}$

$\mathsf{pk} = h = g_p^t$

m_0, m_1

$b^* \xleftarrow{\$} \{0, 1\}$

$r \hookleftarrow \mathcal{D}_p$

(C_0, C_1)

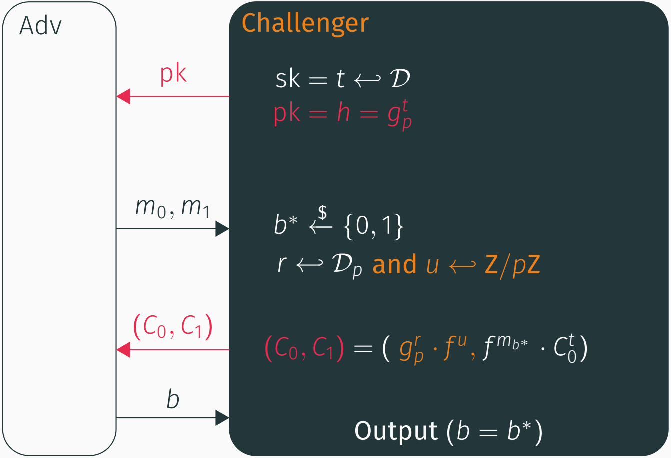
$(C_0, C_1) = (g_p^r, f^{m_{b^*}} \cdot C_0^t)$

b

Output $(b = b^*)$

From $\mathcal{A}$'s view, Games 1 and 2 are identical.

Game 3: compute $C_0 \in G \backslash G^p$



Adv

Challenger

$\mathsf{sk} = t \hookleftarrow \mathcal{D}$

$\mathsf{pk} = h = g_p^t$

pk

m_0, m_1

$b^* \xleftarrow{\$} \{0, 1\}$

$r \hookleftarrow \mathcal{D}_p$ **and** $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$

(C_0, C_1)

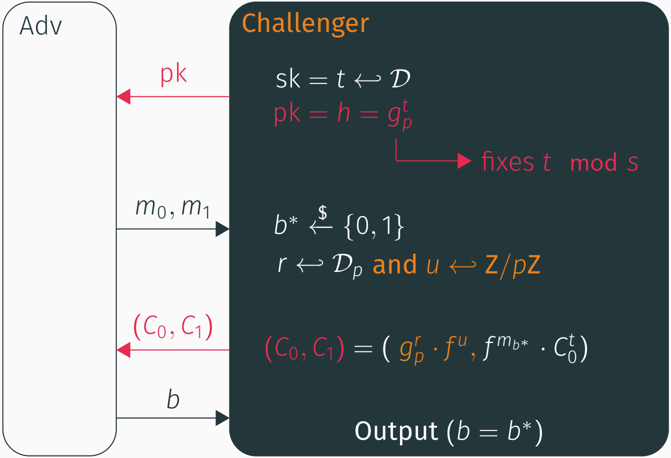
$(C_0, C_1) = (\, g_p^r \cdot f^u, f^{m_{b^*}} \cdot C_0^t)$

b

Output $(b = b^*)$

Games 2 and 3 are undistinguishable to $\mathcal{A}$
under the HSM assumption.

Game 3: compute $C_0 \in G\backslash G^p$



Adv

Challenger

pk

$\mathsf{sk} = t \hookleftarrow \mathcal{D}$

$\mathsf{pk} = h = g_p^t$

fixes $t \mod s$

m_0, m_1

$b^* \xleftarrow{\$} \{0, 1\}$

$r \hookleftarrow \mathcal{D}_p$ and $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$

(C_0, C_1)

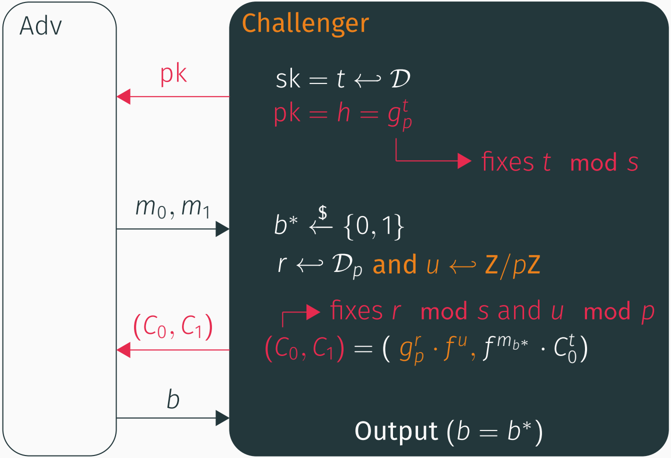
$(C_0, C_1) = (g_p^r \cdot f^u, f^{m_{b^*}} \cdot C_0^t)$

b

Output $(b = b^*)$

Games 2 and 3 are undistinguishable to $\mathcal{A}$ under the HSM assumption.

Game 3: compute $C_0 \in G \backslash G^p$



Adv

pk

m_0, m_1

(C_0, C_1)

b

Challenger

$\mathsf{sk} = t \hookleftarrow \mathcal{D}$

$\mathsf{pk} = h = g_p^t$

fixes $t \mod s$

$b^* \xleftarrow{\$} \{0, 1\}$

$r \hookleftarrow \mathcal{D}_p$ **and** $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$

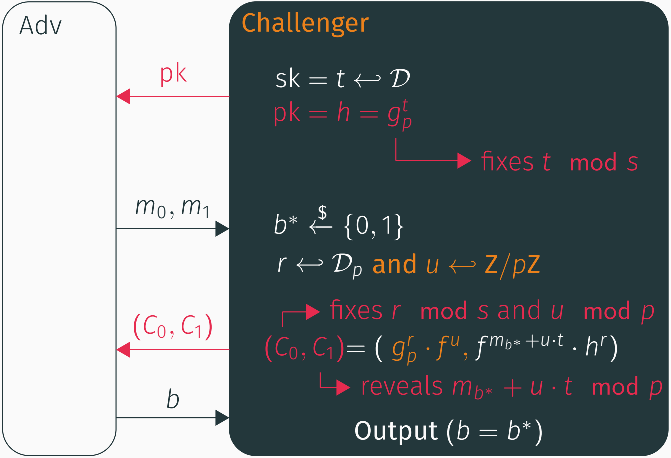
fixes $r \mod s$ and $u \mod p$

$(C_0, C_1) = (\, g_p^r \cdot f^u, f^{m_{b^*}} \cdot C_0^t)$

Output $(b = b^*)$

Games 2 and 3 are undistinguishable to $\mathcal{A}$
under the HSM assumption.

Game 3: compute $C_0 \in G \backslash G^p$



Adv

Challenger

pk

$\mathsf{sk} = t \hookleftarrow \mathcal{D}$

$\mathsf{pk} = h = g_p^t$

fixes $t \mod s$

m_0, m_1

$b^* \xleftarrow{\$} \{0, 1\}$

$r \hookleftarrow \mathcal{D}_p$ **and** $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$

fixes $r \mod s$ and $u \mod p$

(C_0, C_1)

$(C_0, C_1) = (\, g_p^r \cdot f^u, f^{m_{b^*} + u \cdot t} \cdot h^r)$

reveals $m_{b^*} + u \cdot t \mod p$

b

Output $(b = b^*)$

Games 2 and 3 are undistinguishable to $\mathcal{A}$
under the HSM assumption.

Inner Product Functional Encryption mod p from HSM

IPFE scheme mod p from HSM (simplified)

Setup Sample $\vec{t} = (t_1, \ldots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \ldots, \ell$

$\mathsf{msk} = \vec{t}$ and $\mathsf{mpk} = (h_1, \ldots, h_\ell)$

IPFE scheme mod p from HSM (simplified)

Setup Sample $\vec{t} = (t_1, \ldots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \ldots, \ell$

$\mathsf{msk} = \vec{t}$ and $\mathsf{mpk} = (h_1, \ldots, h_\ell)$

Enc Plaintext: $\vec{y} = (y_1, \ldots, y_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Sample randomness r

Ciphertext:

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r, \ldots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$$

IPFE scheme mod p from HSM (simplified)

Setup Sample $\vec{t} = (t_1, \ldots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \ldots, \ell$

$$\mathsf{msk} = \vec{t} \text{ and } \mathsf{mpk} = (h_1, \ldots, h_\ell)$$

Enc Plaintext: $\vec{y} = (y_1, \ldots, y_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Sample randomness r

Ciphertext:

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r, \ldots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$$

KeyDer Input: $\vec{x} = (x_1, \ldots, x_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Output key: $\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle$

IPFE scheme mod p from HSM (simplified)

Setup Sample $\vec{t} = (t_1, \ldots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \ldots, \ell$

$\mathsf{msk} = \vec{t}$ and $\mathsf{mpk} = (h_1, \ldots, h_\ell)$

Enc **Plaintext:** $\vec{y} = (y_1, \ldots, y_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Sample randomness r

Ciphertext:

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r, \ldots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$$

KeyDer **Input:** $\vec{x} = (x_1, \ldots, x_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Output key: $\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle$

Dec From $\vec{C}, \vec{x}$ and $\mathsf{sk}_{\vec{x}}$:

$\langle \vec{x}, \vec{y} \rangle \bmod p$

IPFE scheme mod p from HSM (simplified)

Setup Sample $\vec{t} = (t_1, \dots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \dots, \ell$

$\mathsf{msk} = \vec{t}$ and $\mathsf{mpk} = (h_1, \dots, h_\ell)$

Enc **Plaintext:** $\vec{y} = (y_1, \dots, y_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Sample randomness r

Ciphertext:

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r, \dots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$$

KeyDer **Input:** $\vec{x} = (x_1, \dots, x_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Output key: $\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle$

Dec **From** $\vec{C}, \vec{x}$ **and** $\mathsf{sk}_{\vec{x}}$ **:** $\prod_{i=1}^{\ell} C_i^{x_i}$

$\langle \vec{x}, \vec{y} \rangle \bmod p$

IPFE scheme mod p from HSM (simplified)

Setup Sample $\vec{t} = (t_1, \ldots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \ldots, \ell$

$\mathsf{msk} = \vec{t}$ and $\mathsf{mpk} = (h_1, \ldots, h_\ell)$

Enc Plaintext: $\vec{y} = (y_1, \ldots, y_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Sample randomness r

Ciphertext:

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r, \ldots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$$

KeyDer Input: $\vec{x} = (x_1, \ldots, x_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Output key: $\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle$

Dec From $\vec{C}, \vec{x}$ and $\mathsf{sk}_{\vec{x}}$: $\prod_{i=1}^{\ell} C_i^{x_i} = \prod (f^{\,y_i} \cdot h_i^r)^{x_i}$

$\langle \vec{x}, \vec{y} \rangle$ mod p

IPFE scheme mod p from HSM (simplified)

Setup Sample $\vec{t} = (t_1, \dots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \dots, \ell$

$\mathsf{msk} = \vec{t}$ and $\mathsf{mpk} = (h_1, \dots, h_\ell)$

Enc Plaintext: $\vec{y} = (y_1, \dots, y_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Sample randomness r

Ciphertext:

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r, \dots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$$

KeyDer Input: $\vec{x} = (x_1, \dots, x_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Output key: $\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle$

Dec From $\vec{C}, \vec{x}$ and $\mathsf{sk}_{\vec{x}}$: $\prod_{i=1}^{\ell} C_i^{x_i} = f^{\sum y_i x_i} \cdot g_p^{r \cdot \sum t_i x_i}$

$\langle \vec{x}, \vec{y} \rangle \bmod p$

IPFE scheme mod p from HSM (simplified)

<u>**Setup**</u> Sample $\vec{t} = (t_1, \ldots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \ldots, \ell$

$\mathsf{msk} = \vec{t}$ and $\mathsf{mpk} = (h_1, \ldots, h_\ell)$

<u>**Enc**</u> **Plaintext:** $\vec{y} = (y_1, \ldots, y_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Sample randomness r

Ciphertext:

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r, \ldots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$$

<u>**KeyDer**</u> **Input:** $\vec{x} = (x_1, \ldots, x_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Output key: $\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle$

<u>**Dec**</u> **From** $\vec{C}, \vec{x}$ **and** $\mathsf{sk}_{\vec{x}}$ **:** $\prod_{i=1}^{\ell} C_i^{x_i} = f^{\langle \vec{y}, \vec{x} \rangle} \cdot g_p^{r \cdot \langle \vec{t}, \vec{x} \rangle}$

$\langle \vec{x}, \vec{y} \rangle \bmod p$

IPFE scheme mod p from HSM (simplified)

Setup Sample $\vec{t} = (t_1, \dots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \dots, \ell$

$\mathsf{msk} = \vec{t}$ and $\mathsf{mpk} = (h_1, \dots, h_\ell)$

Enc **Plaintext:** $\vec{y} = (y_1, \dots, y_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Sample randomness r

Ciphertext:

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r, \dots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$$

KeyDer **Input:** $\vec{x} = (x_1, \dots, x_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Output key: $\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle$

Dec **From** $\vec{C}, \vec{x}$ **and** $\mathsf{sk}_{\vec{x}}$ **:** $\prod_{i=1}^{\ell} C_i^{x_i} = f^{\langle \vec{y}, \vec{x} \rangle} \cdot g_p^{r \cdot \langle \vec{t}, \vec{x} \rangle}$ **and** $C_0^{\mathsf{sk}_{\vec{x}}} = g_p^{r \cdot \langle \vec{t}, \vec{x} \rangle}$

$\langle \vec{x}, \vec{y} \rangle \bmod p$

IPFE scheme mod p from HSM (simplified)

Setup Sample $\vec{t} = (t_1, \ldots, t_\ell)$ compute $h_i = g_p^{t_i}$ for $i = 1, \ldots, \ell$

$\mathsf{msk} = \vec{t}$ and $\mathsf{mpk} = (h_1, \ldots, h_\ell)$

Enc **Plaintext:** $\vec{y} = (y_1, \ldots, y_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Sample randomness r

Ciphertext:

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r, \ldots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$$

KeyDer **Input:** $\vec{x} = (x_1, \ldots, x_\ell) \in (\mathbf{Z}/p\mathbf{Z})^\ell$

Output key: $\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle$

Dec **From** $\vec{C}, \vec{x}$ **and** $\mathsf{sk}_{\vec{x}}$ **:** $\prod_{i=1}^{\ell} C_i^{x_i} = f^{\langle \vec{y}, \vec{x} \rangle} \cdot g_p^{r \cdot \langle \vec{t}, \vec{x} \rangle}$ **and** $C_0^{\mathsf{sk}_{\vec{x}}} = g_p^{r \cdot \langle \vec{t}, \vec{x} \rangle}$

Such that:

$$\prod_{i=1}^{\ell} C_i^{x_i} / C_0^{\mathsf{sk}_{\vec{x}}} = f^{\langle \vec{x}, \vec{y} \rangle} \xrightarrow{\mathsf{DL}} \langle \vec{x}, \vec{y} \rangle \bmod p$$

Security

This scheme is secure under the HSM assumption.

Proof technique

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_{b*,1}} \cdot h_1^r, \ldots, C_\ell = f^{y_{b*,\ell}} \cdot h_\ell^r)$$

- Game 0 original security game

Proof technique

$$\vec{C} = (C_0 = g_p^r, C_1 = f^{y_{b*,1}} \cdot C_0^{t_1}, \ldots, C_\ell = f^{y_{b*,\ell}} \cdot C_0^{t_\ell})$$

- Game 0 original security game
- Game 1 use **secret key** to compute challenge ciphertext

Proof technique

$$\vec{C} = (C_0 = g_p^r f^u, C_1 = f^{y_{b*,1}} \cdot C_0^{t_1}, \ldots, C_\ell = f^{y_{b*,\ell}} \cdot C_0^{t_\ell})$$

- Game 0 original security game
- Game 1 use **secret key** to compute challenge ciphertext
- Game 2 indistinguishable from Game 1 under the HSM assumption.

Proof technique

$$\vec{C} = (C_0 = g_p^r f^u, C_1 = f^{y_{b*,1}} \cdot C_0^{t_1}, \ldots, C_\ell = f^{y_{b*,\ell}} \cdot C_0^{t_\ell})$$

- Game 0 original security game
- Game 1 use **secret key** to compute challenge ciphertext
- Game 2 indistinguishable from Game 1 under the HSM assumption.

In Game 2, from $\mathcal{A}$'s view b^* is **statistically hidden**, given

- the public key
- the challenge ciphertext
- key derivation queries

Efficiency comparison

	$\lambda = 112, \ell = 10$		$\lambda = 128, \ell = 10$	
	this work	[ALS16]	this work	[ALS16]
sk_F bitsize	1920	24592	2340	36876
Enc time	40ms	**27ms**	**78ms**	85ms
Dec time	**110ms**	301ms	**193ms**	964ms

Dependency in ℓ is linear.

Last slide!

Conclusion

- Most efficient IPFE schemes to date
- First IPFE mod a prime that recover the result whatever its size.
- Interesting framework, can be applied to other primitives.

Ongoing work

- Chosen Ciphertext Attack Secure schemes
- Threshold ECDSA using our underlying framework

Questions?

M. Abdalla, F. Bourse, A. D. Caro, and D. Pointcheval.
Better security for functional encryption for inner product evaluations.
Cryptology ePrint Archive, Report 2016/011, 2016.
`http://eprint.iacr.org/2016/011`.

M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval.
Simple functional encryption schemes for inner products.
In *PKC 2015*, *LNCS* 9020, pages 733–751. Springer, Heidelberg, March / April 2015.

S. Agrawal, B. Libert, and D. Stehlé.
Fully secure functional encryption for inner products, from standard assumptions.
In *CRYPTO 2016, Part III*, *LNCS* 9816, pages 333–362. Springer, Heidelberg, August 2016.

F. Benhamouda, F. Bourse, and H. Lipmaa.
CCA-secure inner-product functional encryption from projective hash functions.
In *PKC 2017, Part II*, *LNCS* 10175, pages 36–66. Springer, Heidelberg, March 2017.

Information $\mathcal{A}$ gets on b^* in PKE

$$m_{b^*} + u \cdot t \mod p$$

Information $\mathcal{A}$ gets on b^* in PKE

$$m_{b^*} + u \cdot t \mod p$$

Where:

(1) $u \neq 0 \mod p$ with proba $\frac{p-1}{p} \approx 1$

Information $\mathcal{A}$ gets on b^* in PKE

$$m_{b^*} + u \cdot t \mod p$$

Where:

(1) $u \neq 0 \mod p$ with proba $\frac{p-1}{p} \approx 1$

and

(2) t sampled from $\mathcal{D}$, folded gaussian, (almost) uniform mod $s \cdot p$

Information $\mathcal{A}$ gets on b^* in PKE

$$m_{b^*} + u \cdot t \mod p$$

Where:

(1) $u \neq 0 \mod p$ with proba $\frac{p-1}{p} \approx 1$

and

(2) t sampled from $\mathcal{D}$, folded gaussian, (almost) uniform mod $s \cdot p$

↓

Distribution of t (almost) uniform mod p and mod s
and $(t \mod p)$ independent of $(t \mod s)$

Information $\mathcal{A}$ gets on b^* in PKE

$$m_{b*} + u \cdot t \mod p$$

Where:

(1) $u \neq 0 \mod p$ with proba $\frac{p-1}{p} \approx 1$

and

(2) t sampled from $\mathcal{D}$, folded gaussian, (almost) uniform mod $s \cdot p$

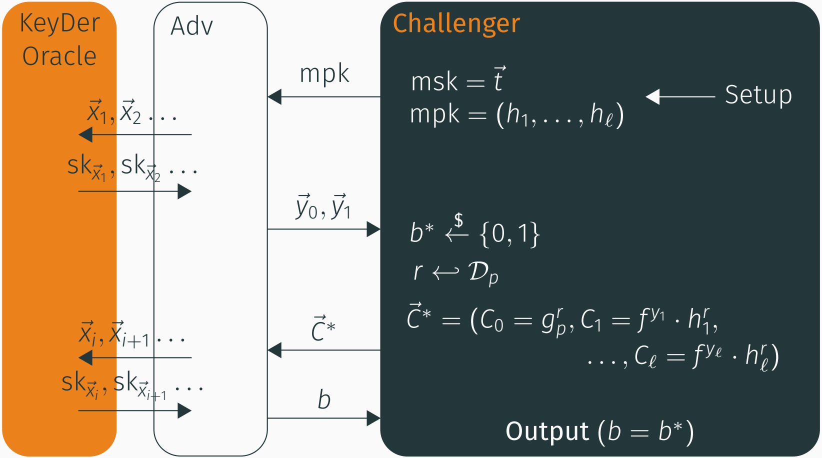
$\downarrow$

Distribution of t (almost) uniform mod p and mod s
and $(t \mod p)$ independent of $(t \mod s)$

$\downarrow$

$u \cdot t$ perfectly masks m_{b*} mod p

Game 0: the original security experiment



KeyDer Oracle

Adv

Challenger

mpk

$\mathsf{msk} = \vec{t}$

$\mathsf{mpk} = (h_1, \ldots, h_\ell)$

Setup

$\vec{x}_1, \vec{x}_2 \ldots$

$\mathsf{sk}_{\vec{x}_1}, \mathsf{sk}_{\vec{x}_2} \ldots$

$\vec{y}_0, \vec{y}_1$

$b^* \xleftarrow{\$} \{0, 1\}$

$r \hookleftarrow \mathcal{D}_p$

$\vec{C}^* = (C_0 = g_p^r, C_1 = f^{y_1} \cdot h_1^r,$

$\ldots, C_\ell = f^{y_\ell} \cdot h_\ell^r)$

$\vec{x}_i, \vec{x}_{i+1} \ldots$

$\vec{C}^*$

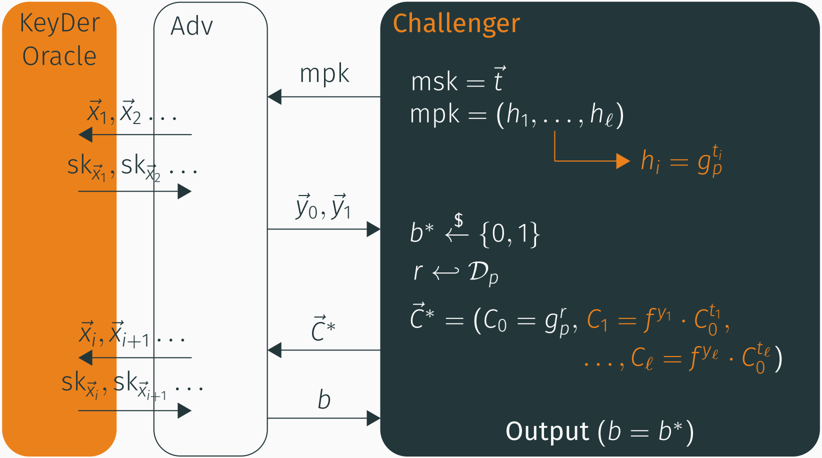
$\mathsf{sk}_{\vec{x}_i}, \mathsf{sk}_{\vec{x}_{i+1}} \ldots$

b

Output $(b = b^*)$

Game 0 is the original security experiment.

Game 1: use msk to compute $\vec{C}^*$



KeyDer Oracle

Adv

Challenger

$\vec{x}_1, \vec{x}_2 \ldots$

$\mathsf{sk}_{\vec{x}_1}, \mathsf{sk}_{\vec{x}_2} \ldots$

$\vec{x}_i, \vec{x}_{i+1} \ldots$

$\mathsf{sk}_{\vec{x}_i}, \mathsf{sk}_{\vec{x}_{i+1}} \ldots$

mpk

$\vec{y}_0, \vec{y}_1$

$\vec{C}^*$

b

$\mathsf{msk} = \vec{t}$

$\mathsf{mpk} = (h_1, \ldots, h_\ell)$

$h_i = g_p^{t_i}$

$b^* \overset{\$}{\leftarrow} \{0,1\}$

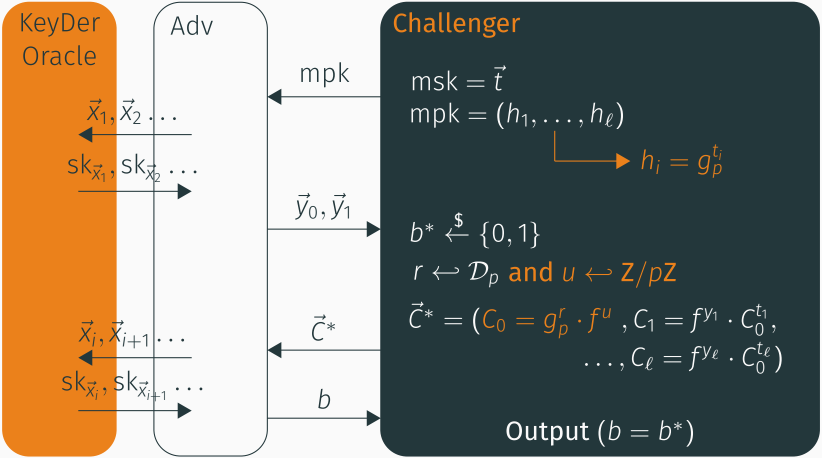
$r \hookleftarrow \mathcal{D}_p$

$$\vec{C}^* = (C_0 = g_p^r, C_1 = f^{y_1} \cdot C_0^{t_1}, \ldots, C_\ell = f^{y_\ell} \cdot C_0^{t_\ell})$$

Output $(b = b^*)$

From $\mathcal{A}$'s view, Games 0 and 1 are identical.

Game 2: compute $C_0 \in G \backslash G^p$



KeyDer Oracle

Adv

Challenger

mpk

$\mathsf{msk} = \vec{t}$

$\mathsf{mpk} = (h_1, \ldots, h_\ell)$

$h_i = g_p^{t_i}$

$\vec{x}_1, \vec{x}_2 \ldots$

$\mathsf{sk}_{\vec{x}_1}, \mathsf{sk}_{\vec{x}_2} \ldots$

$\vec{y}_0, \vec{y}_1$

$b^* \xleftarrow{\$} \{0,1\}$

$r \hookleftarrow \mathcal{D}_p$ **and** $u \hookleftarrow \mathbf{Z}/p\mathbf{Z}$

$\vec{C}^*$

$$\vec{C}^* = (C_0 = g_p^r \cdot f^u, C_1 = f^{y_1} \cdot C_0^{t_1}, \ldots, C_\ell = f^{y_\ell} \cdot C_0^{t_\ell})$$

$\vec{x}_i, \vec{x}_{i+1} \ldots$

$\mathsf{sk}_{\vec{x}_i}, \mathsf{sk}_{\vec{x}_{i+1}} \ldots$

b

Output $(b = b^*)$

Games 1 and 2 are undistinguishable to $\mathcal{A}$ under the HSM assumption.

Leaked Information in Game 2

We consider the information leaked on b^* by:

- the public key
- the challenge ciphertext
- key derivation queries

Information fixed by public key

$$\mathsf{mpk} = \{h_i = g_p^{t_i \bmod s}\}_{i \in [\ell]}$$

Information fixed by public key

$$\begin{tikzcd}
\mathsf{mpk} = \{h_i = g_p^{t_i \bmod s}\}_{i \in [\ell]} \arrow[d, "\text{Fixes}"] \\
(t_1, \ldots, t_\ell) \bmod s
\end{tikzcd}$$

Information fixed by public key

$$\mathsf{mpk} = \{h_i = g_p^{t_i \bmod s}\}_{i \in [\ell]}$$

Fixes

$\downarrow$

$(t_1, \ldots, t_\ell) \bmod s$

$(t_1, \ldots, t_\ell) \bmod p$ is still **uniformly** distributed to $\mathcal{A}$.

Information fixed by challenge ciphertext

$$\vec{C}^* = (C_0 = g_p^r \cdot f^u, \{C_i = f^{y_{b^*,i}} \cdot C_0{}^{t_i}\}_{i \in [\ell]})$$

Information fixed by challenge ciphertext

$$\begin{tikzcd}
\vec{C}^* = (C_0 = g_p^r \cdot f^u, \{C_i = f^{y_{b*,i}} \cdot {C_0}^{t_i}\}_{i \in [\ell]}) \arrow[d, "\text{Reveals}"] \\
C_i = g_p^{r \cdot t_i \mod s} \cdot f^{y_{b*,i} + u \cdot t_i \mod p}
\end{tikzcd}$$

Information fixed by challenge ciphertext

$$\vec{C}^* = (C_0 = g_p^r \cdot f^u, \{C_i = f^{y_{b*,i}} \cdot {C_0}^{t_i}\}_{i\in[\ell]})$$

Reveals

$$C_i = g_p^{r\cdot t_i \mod s} \cdot f^{y_{b*,i}+u\cdot t_i \mod p}$$

Fixes

$$\vec{y}_{b*} + u\vec{t} \bmod p$$

Information fixed by key derivation oracle

For $\vec{x}$ such that $\langle \vec{x}, \vec{y}_0 \rangle = \langle \vec{x}, \vec{y}_1 \rangle \mod p$:

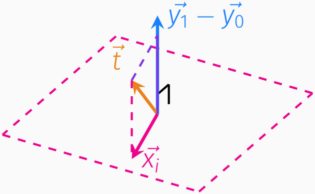
$$\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle \mod p$$

Information fixed by key derivation oracle

For $\vec{x}$ such that $\langle \vec{x}, \vec{y}_0 \rangle = \langle \vec{x}, \vec{y}_1 \rangle \mod p$:

$$\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle \mod p$$

Reveals all the information on $\vec{t}$ for directions $\perp$ to $\vec{y_0} - \vec{y_1}$.

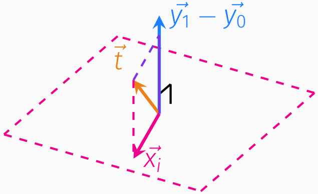


Information fixed by key derivation oracle

For $\vec{x}$ such that $\langle \vec{x}, \vec{y}_0 \rangle = \langle \vec{x}, \vec{y}_1 \rangle \mod p$:

$$\mathsf{sk}_{\vec{x}} = \langle \vec{t}, \vec{x} \rangle \mod p$$

Reveals all the information on $\vec{t}$ for directions $\perp$ to $\vec{y_0} - \vec{y_1}$.



Remaining entropy on $\vec{t}$ contained in $\langle \vec{t}, \vec{y_0} - \vec{y_1} \rangle$

$\mathcal{A}$'s success probability

From $\mathcal{A}$'s view, $\langle \vec{t}, \vec{y}_0 - \vec{y}_1 \rangle$ follows a distribution $\approx \mathcal{U}(\mathsf{Z}/p\mathsf{Z})$.

$\mathcal{A}$'s success probability

From $\mathcal{A}$'s view, $\langle \vec{t}, \vec{y}_0 - \vec{y}_1 \rangle$ follows a distribution $\approx \mathcal{U}(\mathbb{Z}/p\mathbb{Z})$.

The ciphertext reveals:

$$\vec{y}_{b*} + u\vec{t} \bmod p$$

$\mathcal{A}$'s success probability

From $\mathcal{A}$'s view, $\langle \vec{t}, \vec{y}_0 - \vec{y}_1 \rangle$ follows a distribution $\approx \mathcal{U}(\mathbb{Z}/p\mathbb{Z})$.

The ciphertext reveals:

$$\vec{y}_{b^*} + u\vec{t} \bmod p$$

The information on b^* is contained in:

$$\langle \vec{y}_{b^*}, \vec{y}_0 - \vec{y}_1 \rangle + u\langle \vec{t}, \vec{y}_0 - \vec{y}_1 \rangle \bmod p$$

$\mathcal{A}$'s success probability

From $\mathcal{A}$'s view, $\langle \vec{t}, \vec{y}_0 - \vec{y}_1 \rangle$ follows a distribution $\approx \mathcal{U}(\mathbb{Z}/p\mathbb{Z})$.

The ciphertext reveals:

$$\vec{y}_{b^*} + u\vec{t} \bmod p$$

The information on b^* is contained in:

$$\langle \vec{y}_{b^*}, \vec{y}_0 - \vec{y}_1 \rangle + u\langle \vec{t}, \vec{y}_0 - \vec{y}_1 \rangle \bmod p$$

$\mathcal{A}$ cannot guess b^* with proba $> 1/2 +$ negl