Functional Encryption

Ratna Dutta

Associate Professor

Department of Mathematics

Indian Institute of Technology, Kharagpur

ACM Summer School on Cryptology Research, ISI-Kolkata

June 4-22, 2018

Public Key Encryption (PKE)

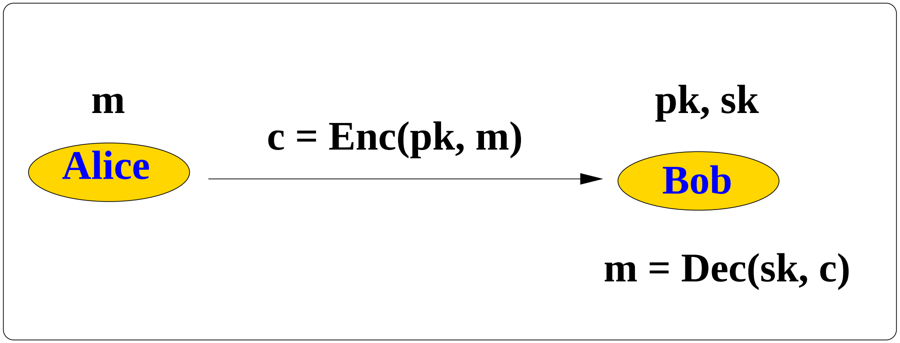
- A conventional public-key encryption scheme is comprised of two randomized algorithms Keygen, Enc and a deterministic algorithm Dec. Let $\mathcal{M}$ and $\mathcal{C}$ are the message and ciphertext space, respectively.

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(1^\rho)$$

$$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m), m \in \mathcal{M}$$

$$m \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$$

- **Correctness:** $\forall \rho, \forall m \in \mathcal{M}$
$\mathsf{Pr}[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(1^\rho) : \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m] = 1$



Drawback:

- Decryption is "all" or "nothing" affair!

Homomorphic Encryption (HE)

- Let the message space $(G, \circ)$ be a group and $\mathcal{C}$ be ciphertext space.

- A homomorphic public-key encryption scheme consists of algorithms Keygen, Enc, Dec and Eval.

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(1^\rho)$$

$$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m), m \in G$$

$$m \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$$

$$\psi \leftarrow \mathsf{Eval}(\mathsf{pk}, f, c_1, c_2, \ldots, c_t), c_i = \mathsf{Enc}(\mathsf{pk}, m_i),$$

$$m_i \in G, 1 \leq i \leq t$$

- **Correctness:** For any key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(1^\rho)$, if $f(m_1, m_2, \ldots, m_t) = m_1 \circ m_2 \circ \ldots \circ m_t$, $c_i = \mathsf{Enc}(\mathsf{pk}, m_i)$, $\forall m_i \in G$ and $\psi \leftarrow \mathsf{Eval}(\mathsf{pk}, f, c_1, c_2, \ldots, c_t)$, then $\mathsf{Pr}[\mathsf{Dec}(\mathsf{sk}, \psi) \neq f(m_1, m_2, \ldots, m_t)$ is negligible.
- i.e. $\psi = \mathsf{Enc}(\mathsf{pk}, f(m_1, m_2, \ldots, m_t))$

- If G is an additive group, then the scheme is called additively homomorphic.
- If G is a multiplicative group, then the scheme is called multiplicatively homomorphic.

- f is $+$ operator, $c_1 = \mathsf{Enc}(\mathsf{pk}, m_1)$ and $c_2 = \mathsf{Enc}(\mathsf{pk}, m_2)$ $\mathsf{Eval}(\mathsf{pk}, f, c_1, c_2) = \mathsf{Enc}(\mathsf{pk}, f(m_1, m_2)) = \mathsf{Enc}(\mathsf{pk}, m_1 + m_2)$
- Homomorphic encryption allows untrusted remote servers to perform computation on encrypted data without the data being compromised.
- This in turn facilitates outsourcing computation to untrusted servers maintained by service providers such as Dropbox, Rackspace Inc., Amazon, VMware.

Partially homomorphic cryptosystems

- Goldwasser-Micali, Bresson-Catalano-Pointcheval, Camenisch-Shoup cryptosystems are additively homomorphic
- RSA, ElGamal, Boneh-Boyen-Shacham encryption schemes are multicatively homomorphic
- Paillier cryptosystem exhibits more homomorphic properties

Fully homomorphic encryption (FHE)

- Generally speaking, FHE makes it possible to compute an encryption of $f(m)$ for some arbitrary function f, without knowing the private key.
- The result $f(m)$ of the computation remains encrypted and can only be decrypted by the party holding the private key sk.
- Delegate PROCESSING of data without giving ACCESS of it.
- Supports arbitrary computation on ciphertexts.

- FHE is first realised from lattices by Gentry in 2009
- Many improved variants have appeared in the literature following this work, all based on lattices
- **Examples:** Brakerski-Vaikuntanathan (2014), Gentry-Sahai-Waters (2013), Smart-Vercauteren (2010)

- Lattice based cryptographic constructions are potential candidates for the post-quantum era as they offer
  - apparent resistance to quantum attacks
  - security under worst-case intractability assumptions
  - efficient parallel computations
  - homomorphic computations

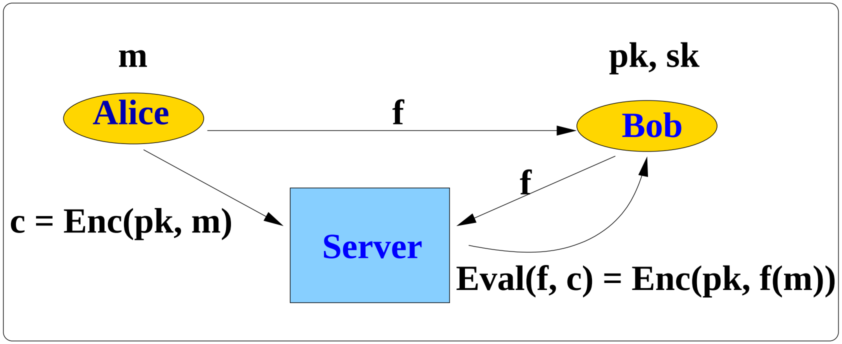
Applications of HE

- Commitment schemes
- Multiparty computation
- Election schemes
- Oblivious transfer
- Lottery protocols
- Data aggregation in wireless sensor networks

Use of Homomorphic property in Cloud computing

- Alice stores her encrypted file on Bob's Server.
- She wants to do some computation on her file.
- Alice asks Bob to perform the computation on encrypted data.
- Bob gives her an encrypted answer of her query.
- Alice uses her secret key and decrypt the answer to recover the message.

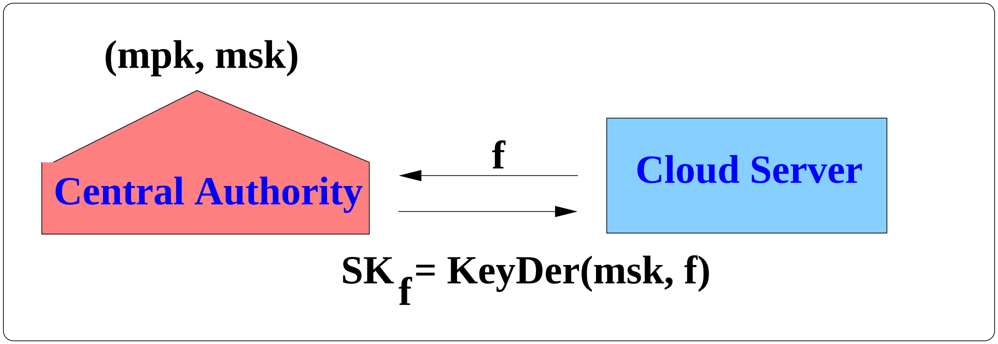
Drawback:



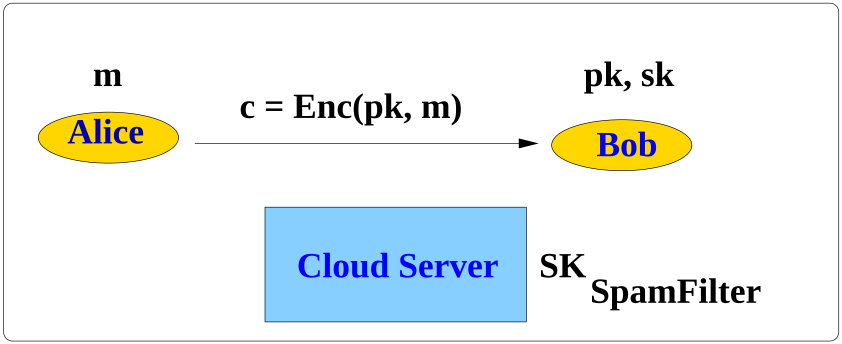
- Interaction with Bob!

Functional Encryption (FE)

(Delegates decryption capabilities)



Functional Encryption (FE)



$$\textit{if}\ \mathsf{Eval}(\mathsf{SK}_{SpamFilter}, c) = \mathsf{True}$$
$$\textit{then}\ \text{“Move to the Spam Folder”}$$

Advantages

- Decryption does not require interaction with Bob!
- Fine-Grained Access Control of Decryption Capabilities!

FE: Credit Card Transaction Alert

- Credit Card Tranaction Alert ($\mathsf{SK}_{\mathsf{Alert}}$)

$$
\begin{aligned}
&\textit{if}\, \mathsf{Eval}(\mathsf{SK}_{\mathsf{Alert}}, c) = \mathsf{True} \\
&\qquad \textit{then} \text{ “Fire an Alarm”}
\end{aligned}
$$

$\boxed{\mathsf{Alert}\text{: Transactions over Rs. 1.0 Lakhs}}$

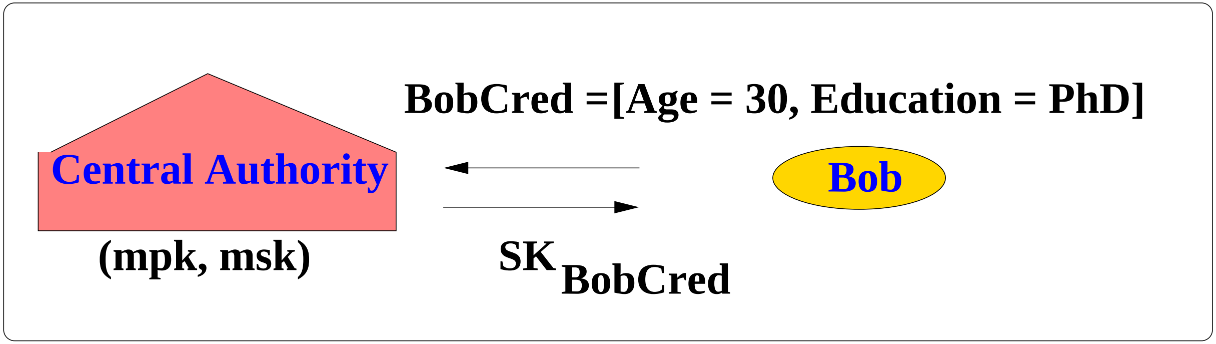
FE: Credit Card Fraud Investigation

- Credit Card Fraud Investigation ($\mathsf{SK}_{f_{\mathsf{Auditing}}}$)

$$\begin{array}{l} \textit{if}\ \mathsf{Eval}(\mathsf{SK}_{f_{\mathsf{Auditing}}}, c) = \mathsf{True} \\ \qquad \textit{then}\ \text{"Fire an Alarm"} \end{array}$$

f_{Auditing}: Transactions over Rs. 1.0 Lakhs which took place in November and originated from Kolkata.

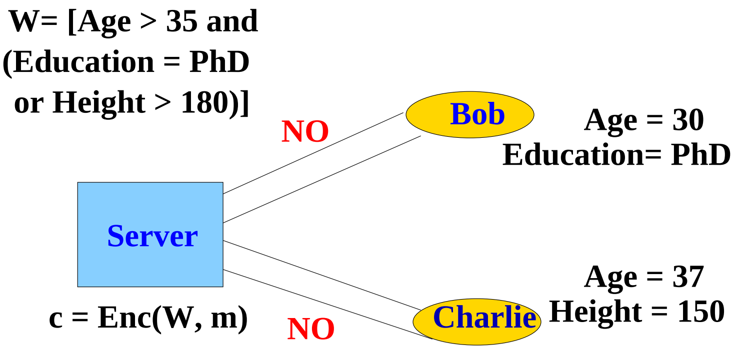
FE: Online dating



Bob has specific attributes and will receive a secret key that can only decrypt profiles for which the attributes match the dating preferences.

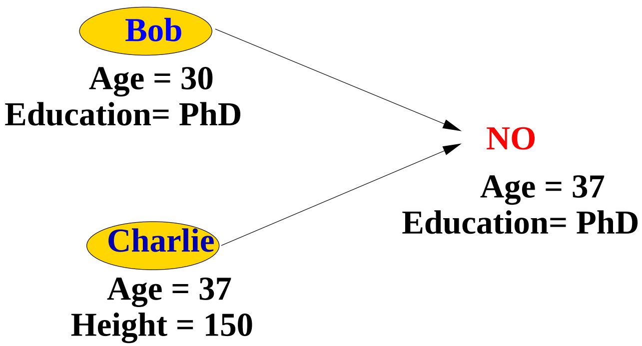
FE: Online dating

- profile m is encrypted under the dating preferences (access structure) $W = [$ Age $>$ 35 and Education $=$ PhD or Height $>$ 180)]



FE: Online dating - Collusion Resistance

- profile m is encrypted under the dating preferences (access structure) $W = [\text{ Age } > 35 \text{ and } (\text{Education } = \text{ PhD or Height } > 180)]$
- primitive should withstand collusion attack



Current Lines of Work

- Efficient functional encryption for access control
- Functional encryption for all circuits
- Efficient constructions for expressive functionalities

FE: Definition

A Functional Encryption (FE) scheme for the functionality $\mathcal{F}$ consists of the following algorithms:

$$(\mathsf{mpk}, \mathsf{msk}) \longleftarrow \mathsf{Setup}(1^\lambda, \mathcal{F})$$

$$\mathsf{SK}_f \longleftarrow \mathsf{KeyDer}(\mathsf{msk}, f)$$

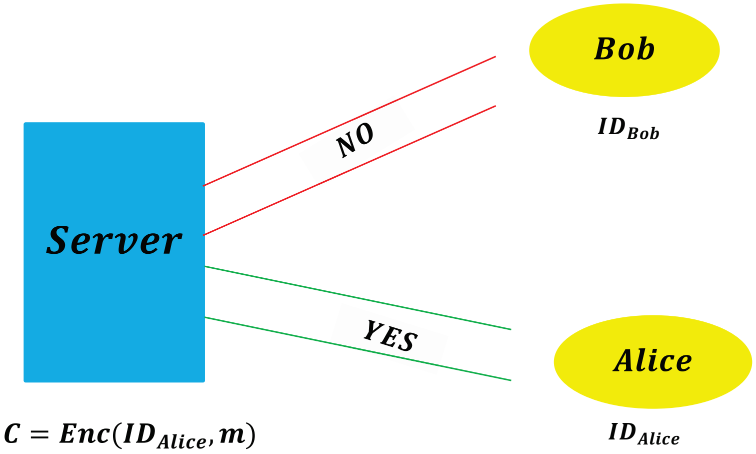
$$\mathsf{CT} \longleftarrow \mathsf{Enc}(\mathsf{mpk}, m)$$

$$f(m) \longleftarrow \mathsf{Dec}(\mathsf{SK}_f, \mathsf{CT})$$

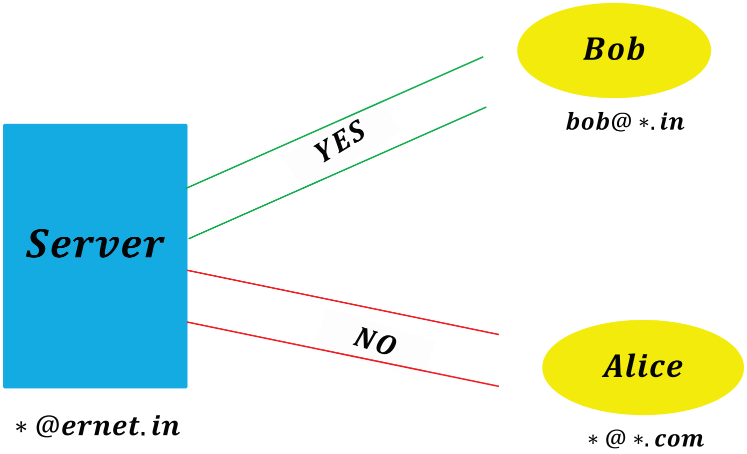
Examples of Functionalities

- (Hierarchical) Identity-Based Encryption
- Fuzzy Identity-Based Encryption
- Attribute-Based Encryption
- Predicate Encryption etc.

Identity-Based Encryption (IBE)



Generalized Hierarchical IBE (HIBE)

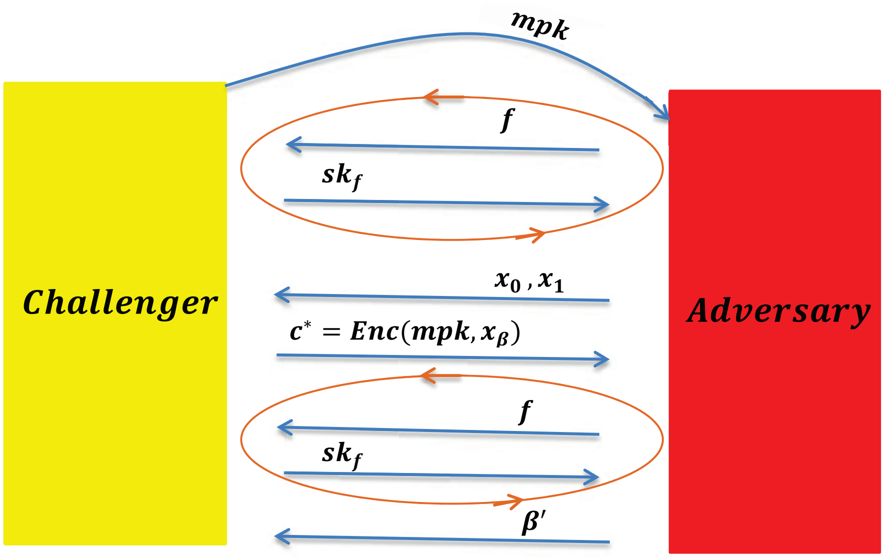


Functional Encryption: Security (Intuition)

- Secret keys $\mathsf{SK}_{f_1}, \ldots, \mathsf{SK}_{f_l}$ should only reveal $f_1(m), \ldots, f_l(m)$ when given an encryption of m.
- Indistinguishibility-based security – meaningless for certain class of functions
- Simulation-based security (strongest) – impossible to achieve for certain class of functions

Indistinguishability-Based Security

- Adversary sends two challenge ciphertexts x_0, x_1 and receives back encryption of x_β, $\beta \in \{0, 1\}$
- Adversary wins the game if its final output β' matches with β, and for all f, $f(x_0) = f(x_1)$ hold.

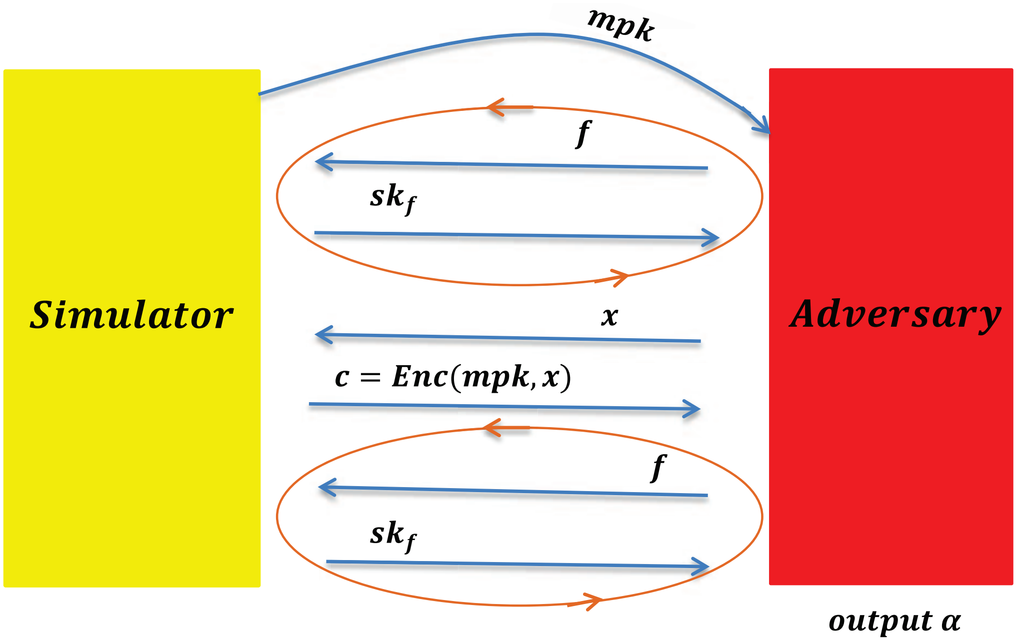


The adversary wins if $\boldsymbol{\beta = \beta'}$ ***and*** $\boldsymbol{\forall f, f(x_0) = f(x_1)}$

Simulation-Based Security

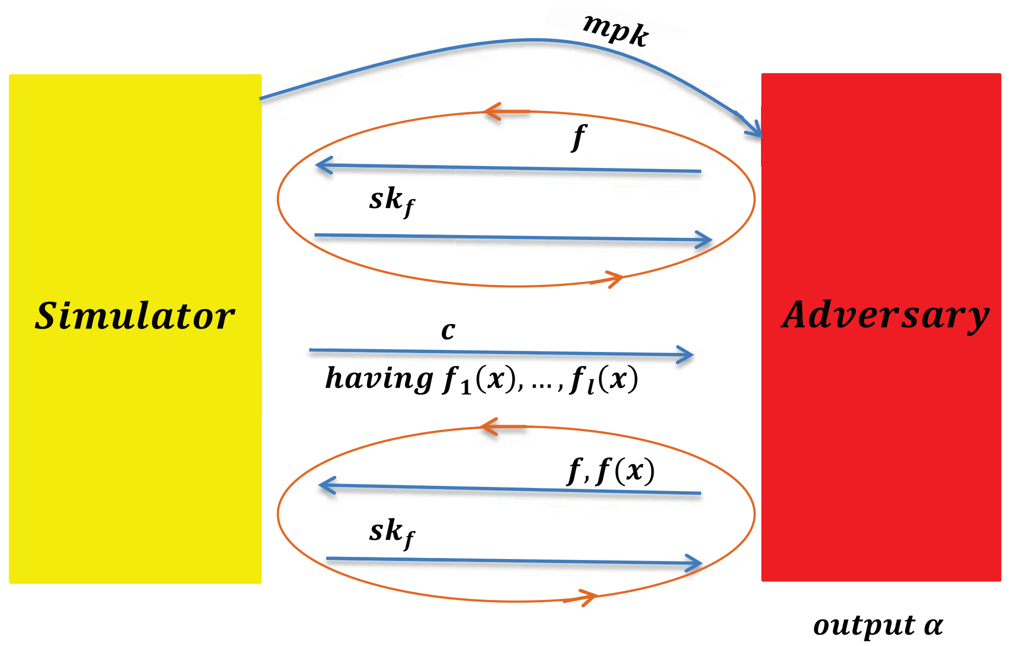
(Real world)

- uses real/ideal world paradigm
- output α of the ideal world is computationally indistinguishable from the output α in the real world


mpk
f
sk_f
Simulator
Adversary
x
$c = Enc(mpk, x)$
f
sk_f
output α

Simulation-Based Security

(Ideal world)



mpk
f
sk_f
Simulator
Adversary
c
having $f_1(x), \ldots, f_l(x)$
$f, f(x)$
sk_f
output α

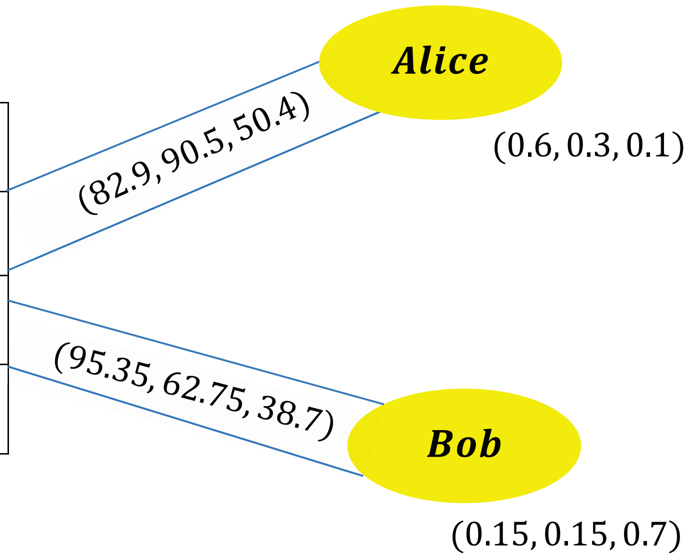
Inner-Product Functional Encryption (IP-FE)

(Course grades)

$Evaluation \; x$

	$Test$	$Quiz$	$Project$
$Oscar$	74	95	100
$Recherd$	100	85	50
$Charlie$	40	78	30

$C_x = Enc(mpk, x)$



IP-FE: Functionality

- $\mathcal{F} : Z_p^l \times Z_p^l \rightarrow Z_p$
  $(y, x) \rightarrow \langle y, x \rangle$
- Secret key for $y \in Z_p^l$: $\mathsf{SK}_y = \mathsf{KeyDer}(\mathsf{msk}, y)$
- Ciphertext for $x \in Z_p^l$: $\mathsf{CT}_x = \mathsf{Enc}(\mathsf{mpk}, x)$
- Correctness: $\mathsf{Dec}(\mathsf{SK}_y, \mathsf{CT}_x) = \langle y, x \rangle$

Functionality Properties

- Several applications
- Easy to compute – only need additions if one vector is known
- Still non-trivial – keyspace size is $\sim q^l$
- $\langle x, y \rangle$ leaks a lot of information about x
- l well chosen secret keys reveal everything

IP-FE from ElGamal

- ElGamal Encryption:

$$\mathsf{pk} = (G, g, h = g^s)$$

$$\mathsf{sk} = s$$

$$\mathsf{Enc}(\mathsf{pk}, x) = c = (c_0 = g^r, c_1 = h^r g^x)$$

$$\mathsf{Dec}(\mathsf{sk}, c) = c_1/(c_0)^{\mathsf{sk}} = h^r g^x / g^{rs} = h^r g^x / h^r = g^x$$

- Decisional Diffie-Hellman (DDH) Assumption:

$$(g, g^a, g^b, g^{ab} \sim_c (g, g^a, g^b, g^c)$$

IP-FE from ElGamal

- $\mathsf{Setup}(l)$:

$$\mathsf{mpk} = (G, g, h_1 = g^{s_1}, h_2 = g^{s_2}, \ldots, h_l = g^{s_l})$$

$$\mathsf{msk} = (s_1, s_2, \ldots, s_l)$$

- $\mathsf{Encrypt}(\mathsf{mpk}, x = (x_1, x_2, \ldots, x_l))$

$$\mathsf{CT}_x = (C_0 = g^r, C_1 = h_1^r g^{x_1}, C_2 = h_2^r g^{x_2}, \ldots, C_l = h_l^r g^{x_l})$$

- $\mathsf{KeyDer}(\mathsf{msk}, y = (y_1, y_2, \ldots, y_l))$:

$$\mathsf{SK}_y = \sum_{i \in [l]} s_i y_i$$

- $\mathsf{Decrypt}(\mathsf{SK}_y, \mathsf{CT}_x)$:

$$\prod_{i\in[l]}(C_i)^{y_i} = \prod_{i\in[l]}(h_i^r g^{x_i})^{y_i} = \prod_{i\in[l]}(g^{s_i})^{y_i r} g^{x_i y_i}$$
$$= g^{r(\sum_{i\in[l]} s_i y_i)} g^{(\sum_{i\in[l]} x_i y_i)}$$
$$= (C_0)^{\mathsf{SK}_y} g^{\langle x,y\rangle}$$

where

$$y = (y_1, y_2, \ldots, y_l), \mathsf{SK}_y = \sum_{i\in[l]} s_i y_i$$

$$\mathsf{CT}_x = (C_0 = g^r, C_1 = h_1^r g^{x_1}, C_2 = h_2^r g^{x_2}, \ldots, C_l = h_l^r g^{x_l})$$

Multi-Input Functional Encryption (MI-FE)

- Extension to multi-input functions: $f(x_1, \ldots, x_n)$
- Several encryption slots: $\mathsf{Enc}(x_1), \ldots, \mathsf{Enc}(x_n)$
- Each slot can be encrypted independently
- SK_f enables to compute $f(x_1, \ldots, x_n)$
- Several feasibility results for general circuits

(MI-FE)

- $\mathcal{F} : (Z_p^l)^n \times (Z_p^l)^n \to Z_p$
- $((\vec{y}_1, \ldots, \vec{y}_n), (\vec{x}_1, \ldots, \vec{x}_n)) \to \Sigma_{j=1}^n \langle \vec{y}_j, \vec{x}_j \rangle$
- Secret key for $\vec{y} = (\vec{y}_1, \ldots, \vec{y}_n) \in (Z_p^l)^n$:
  $\mathsf{SK}_{\vec{y}} = \mathsf{KeyDer}(\mathsf{msk}, \vec{y})$
- Ciphertext for $\vec{x} = (\vec{x}_1, \ldots, \vec{x}_n) \in (Z_p^l)^n$:
  $\mathsf{CT}_{\vec{x}} = \mathsf{Enc}(\mathsf{mpk}, \vec{x})$
- Correctness: $\mathsf{Dec}(\mathsf{SK}_{\vec{y}}, \mathsf{CT}_{\vec{x}}) = \Sigma_{j=1}^n \langle \vec{y}_j, \vec{x}_j \rangle$

Other Extensions

- Function-hiding IP-FE
- Fully secure IP-FE
- 2-input IP-FE from pairings
- Quadratic functions from pairings (in contrast to inner product which is linear function)
- n-degree functions from n-linear maps

Conclusion

- FE can be practical!
- Several extensions:
  - *Functionalities*: function hiding, multi-input, higher degrees
  - *Assumptions*: DDH, High residuosity, LWE
- Leakage should be considered more carefully

- FE has already proven useful in constructing
  - strong exponentially-efficient indistinguishibility obfuscation (SXIO)
  - randomized encoding for Turing machines
  - indistinguishibility obfuscation (IO) without multilinear maps and many more

Some open questions

- High degree polynomials from standard assumptions (e.g. LWE)
- Randomized functionalities from standard assumptions

Thank You.