Simple Functional Encryption Schemes for Inner Products

Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval

École normale supérieure, CNRS, INRIA, PSL, Paris, France

PKC 2015 — Maryland, USA
Wednesday, April 1

1. Overview of the results
   - What is Functional Encryption?
   - Inner Product functionality
   - What does simple mean? What do we achieve?
2. The Framework
   - Overview of the framework
   - Example
   - Proof of security
   - Generalization
3. Work in progress
   - What is there left to do?
   - Thank you!

Brief history

What is Functional Encryption?

Brief history

What is Functional Encryption?
Introduced by Dan Boneh, Amit Sahai and Brent Waters [BSW10]

Brief history

What is Functional Encryption?
Introduced by Dan Boneh, Amit Sahai and Brent Waters [BSW10]
Generalizes multiple concepts:

Brief history

What is Functional Encryption?
Introduced by Dan Boneh, Amit Sahai and Brent Waters [BSW10]
Generalizes multiple concepts:

- Identity-Based Encryption

Brief history

What is Functional Encryption?

Introduced by Dan Boneh, Amit Sahai and Brent Waters [BSW10]

Generalizes multiple concepts:

- Identity-Based Encryption
- Fuzzy Identity-Based Encryption

Brief history

What is Functional Encryption?

Introduced by Dan Boneh, Amit Sahai and Brent Waters [BSW10]

Generalizes multiple concepts:

- Identity-Based Encryption
- Fuzzy Identity-Based Encryption
- Attribute-Based Encryption

Brief history

What is Functional Encryption?
Introduced by Dan Boneh, Amit Sahai and Brent Waters [BSW10]
Generalizes multiple concepts:

- Identity-Based Encryption
- Fuzzy Identity-Based Encryption
- Attribute-Based Encryption
- Predicate Encryption, etc.

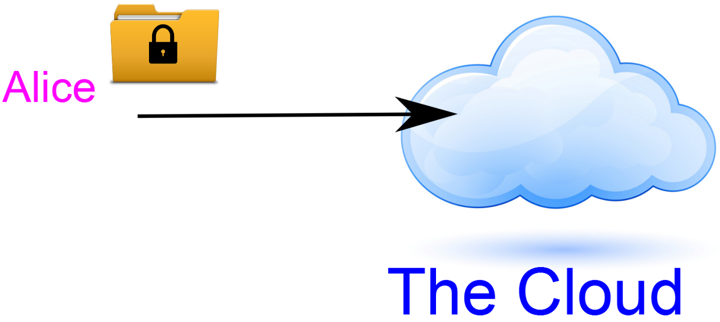
Brief history

What is Functional Encryption?
Introduced by Dan Boneh, Amit Sahai and Brent Waters [BSW10]
Generalizes multiple concepts:

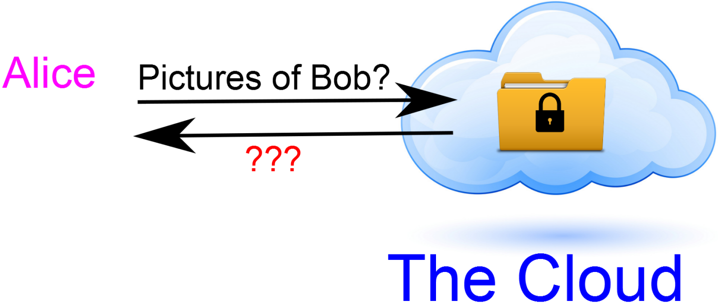
- Identity-Based Encryption
- Fuzzy Identity-Based Encryption
- Attribute-Based Encryption
- Predicate Encryption, etc.

Enables keys that give partial information.

Motivation



Motivation



Formal definition

$$\text{Functionality } \mathcal{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{M}$$
$$(k, x) \mapsto \mathcal{F}(k, x)$$

Secret key for k : $\mathbf{sk}_k \leftarrow msk$

Ciphertext for x : $\mathbf{ct}_x \leftarrow \mathbf{pk}$

Formal definition

$$\text{Functionality } \mathcal{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{M}$$
$$(k, x) \mapsto \mathcal{F}(k, x)$$

$$\text{Secret key for } k : \mathbf{sk}_k \leftarrow msk$$
$$\text{Ciphertext for } x : \mathbf{ct}_x \leftarrow \mathbf{pk}$$

Correctness

$$\text{Decrypt}(\mathbf{sk}_k, \mathbf{ct}_x) = \mathcal{F}(k, x)$$

Formal definition

$$\text{Functionality } \mathcal{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{M}$$

$$(k, x) \mapsto \mathcal{F}(k, x)$$

$$((\text{Picture,Bob}),\text{data}) \mapsto \text{Pictures of Bob}$$

$$\text{Secret key for } k : \mathbf{sk}_k \leftarrow msk$$

$$\text{Ciphertext for } x : \mathbf{ct}_x \leftarrow \mathbf{pk}$$

Correctness

$$\text{Decrypt}(\mathbf{sk}_k, \mathbf{ct}_x) = \mathcal{F}(k, x)$$

Formal definition

$$\text{Functionality } \mathcal{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{M}$$

$$(k, x) \mapsto \mathcal{F}(k, x)$$

$$((\text{Picture,Bob}),\text{data}) \mapsto \text{Pictures of Bob}$$

$$\text{Secret key for } k : \mathbf{sk}_k \leftarrow msk$$

$$\text{Ciphertext for } x : \mathbf{ct}_x \leftarrow \mathbf{pk}$$

Correctness

$$\text{Decrypt}(\mathbf{sk}_k, \mathbf{ct}_x) = \mathcal{F}(k, x)$$

Alice gets Bob's pictures in her data.

Security

Intuitively:
$\mathbf{sk}_k$ doesn't leak any more information than $\mathcal{F}(k, x)$

Even if there are collusions !
$\mathbf{sk}_k$ and $\mathbf{sk}'_k$ don't leak more information than $\mathcal{F}(k, x)$ and $\mathcal{F}(k', x)$

Security

Intuitively:

$\mathbf{sk}_k$ doesn't leak any more information than $\mathcal{F}(k, x)$

The server doesn't access Alice's private data other than needed.

Even if there are collusions !

$\mathbf{sk}_k$ and $\mathbf{sk}'_k$ don't leak more information than $\mathcal{F}(k, x)$ and $\mathcal{F}(k', x)$

Pictures of Jean and pictures of Jacques don't make pictures of Jean-Jacques.

current lines of work

- Designing efficient functional encryption for access control...

current lines of work

- Designing efficient functional encryption for access control... nothing about partial information

current lines of work

- Designing efficient functional encryption for access control... nothing about partial information
- Obtain functional encryption for all circuits...

current lines of work

- Designing efficient functional encryption for access control... nothing about partial information
- Obtain functional encryption for all circuits... construction from inefficient primitives

current lines of work

- Designing efficient functional encryption for access control... nothing about partial information
- Obtain functional encryption for all circuits... construction from inefficient primitives
- This work: figuring out what we can do with simple assumption

Inner Product functionality

$$\text{Functionality } \mathcal{F} : \mathbb{Z}_p^\ell \times \mathbb{Z}_p^\ell \to \mathbb{Z}_p$$
$$(\mathbf{y}, \mathbf{x}) \to < \mathbf{x}, \mathbf{y} >$$

Secret key for $\mathbf{y}$: $\mathbf{sk_y}$

Ciphertext for $\mathbf{x}$: $\mathbf{ct_x}$

Inner Product functionality

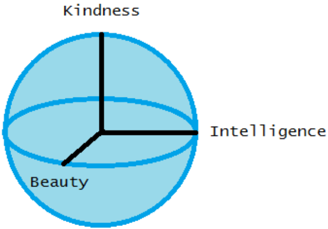
$$\text{Functionality } \mathcal{F} : \mathbb{Z}_p^\ell \times \mathbb{Z}_p^\ell \to \mathbb{Z}_p$$
$$(\mathbf{y}, \mathbf{x}) \to < \mathbf{x}, \mathbf{y} >$$

$$\text{Secret key for } \mathbf{y} : \mathbf{sk_y}$$
$$\text{Ciphertext for } \mathbf{x} : \mathbf{ct_x}$$

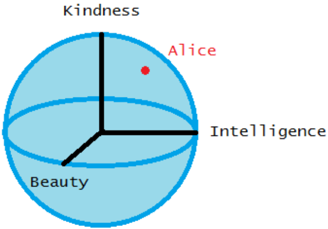
Correctness

$$\text{Decrypt}(\mathbf{y}, \mathbf{ct_x}) = < \mathbf{x}, \mathbf{y} >$$

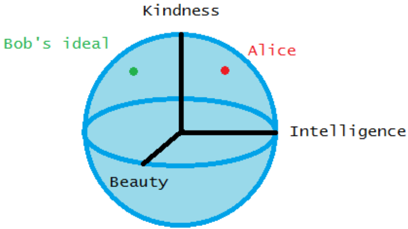
Motivation example: Online dating system



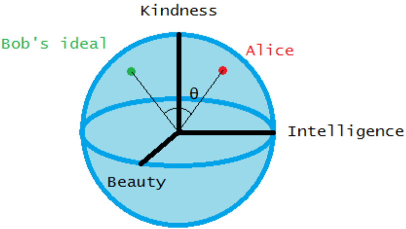
Motivation example: Online dating system



Motivation example: Online dating system



Motivation example: Online dating system



Properties

Inner product is very interesting:

- lots of applications
- easy to compute - only need additions if one vector is known
- still non-trivial: $|\mathcal{K}|$ is exponential in ℓ
- theoretically interesting problem - enables any computation in NC^0

Inherent security limitation

$< \mathbf{x}, \mathbf{y} >$ gives a lot of information about $\mathbf{x}$
ℓ well chosen secret keys reveals everything

Basic primitive: PKE with some additional structural properties

Our framework can be instantiated with different well known Public Key Encryption schemes

Basic primitive: PKE with some additional structural properties

Our framework can be instantiated with different well known Public Key Encryption schemes

Additive ElGamal, based on Decisional Diffie-Hellman (DDH) assumption

Basic primitive: PKE with some additional structural properties

Our framework can be instantiated with different well known Public Key Encryption schemes
Additive ElGamal, based on Decisional Diffie-Hellman (DDH) assumption Lattice based Public Key Encryption scheme, based on the Learning With Errors (LWE) assumption

Efficient

Ciphertext size is $\ell + 1$ elements
Key size is 1 element

Efficient

Ciphertext size is $\ell + 1$ elements

Key size is 1 element

This is really close to information theoretical optimal for correctness

Selective IND-CPA security

The resulting scheme is secure under selective chosen plaintext attacks

Security game:

- $\mathcal{A}$ submits $\mathbf{x}_0, \mathbf{x}_1$

Selective IND-CPA security

The resulting scheme is secure under selective chosen plaintext attacks

Security game:

- $\mathcal{A}$ submits $\mathbf{x}_0, \mathbf{x}_1$
- $\mathcal{A}$ receives $\mathbf{pk}, \mathbf{ct}_{\mathbf{x}_b}$

Selective IND-CPA security

The resulting scheme is secure under selective chosen plaintext attacks

Security game:

- $\mathcal{A}$ submits $\mathbf{x}_0, \mathbf{x}_1$
- $\mathcal{A}$ receives $\mathbf{pk}, \mathbf{ct}_{\mathbf{x}_b}$
- $\mathcal{A}$ sends some set of queries $\{\mathbf{y}\}$, such that $< \mathbf{x}_0, \mathbf{y} >=< \mathbf{x}_1, \mathbf{y} >$

Selective IND-CPA security

The resulting scheme is secure under selective chosen plaintext attacks

Security game:

- $\mathcal{A}$ submits $\mathbf{x}_0, \mathbf{x}_1$
- $\mathcal{A}$ receives $\mathbf{pk}, \mathbf{ct}_{\mathbf{x}_b}$
- $\mathcal{A}$ sends some set of queries $\{\mathbf{y}\}$, such that $< \mathbf{x}_0, \mathbf{y} >=< \mathbf{x}_1, \mathbf{y} >$
- $\mathcal{A}$ receives $\{\mathbf{sk}_\mathbf{y}\}$

Selective IND-CPA security

The resulting scheme is secure under selective chosen plaintext attacks

Security game:

- $\mathcal{A}$ submits $\mathbf{x}_0, \mathbf{x}_1$
- $\mathcal{A}$ receives $\mathbf{pk}, \mathbf{ct}_{\mathbf{x}_b}$
- $\mathcal{A}$ sends some set of queries $\{\mathbf{y}\}$, such that $< \mathbf{x}_0, \mathbf{y} > = < \mathbf{x}_1, \mathbf{y} >$
- $\mathcal{A}$ receives $\{\mathbf{sk}_\mathbf{y}\}$
- $\mathcal{A}$ guesses b'

How to apply our framework?

Our framework is easy to instantiate:

Pick a good Public Key Encryption scheme

requires structural properties stated later

How to apply our framework?

Our framework is easy to instantiate:

Pick a good Public Key Encryption scheme

requires structural properties stated later

Reuse Randomness to encrypt a vector

How to apply our framework?

Our framework is easy to instantiate:

Pick a good Public Key Encryption scheme
requires structural properties stated later

Reuse Randomness to encrypt a vector

Use additive homomorphism to decrypt the correct value

How to apply our framework?

Our framework is easy to instantiate:

Pick a good Public Key Encryption scheme

requires structural properties stated later

Reuse Randomness to encrypt a vector

Use additive homomorphism to decrypt the correct value

And it's done !

How to apply our framework?

Our framework is easy to instantiate:

Pick a good Public Key Encryption scheme

requires structural properties stated later

Reuse Randomness to encrypt a vector

Use additive homomorphism to decrypt the correct value

And it's done ! (and safe !)

The additively homomorphic ElGamal public key encryption scheme

$$\begin{aligned} \text{Public parameters} &: p, \mathcal{G}, g \\ \text{Secret key} &: s \\ \text{Public key} &: g^s \\ \text{Ciphertext for } m &: (g^r, g^{rs} g^m) \end{aligned}$$

The additively homomorphic ElGamal public key encryption scheme

$$\begin{aligned} \text{Public parameters} &: p, \mathcal{G}, g \\ \text{Secret key} &: s \\ \text{Public key} &: g^{s} \\ \text{Ciphertext for } m &: (g^{r}, g^{rs} g^{m}) \end{aligned}$$

Correctness

$$\frac{g^{rs} g^{m}}{(g^{r})^{s}} = g^{m}$$

Reusing randomness

$$
\begin{aligned}
\text{Public parameters} &: p, \mathcal{G}, g \\
\text{Secret key} &: s \\
\text{Public key} &: g^s \\
\text{Ciphertext for } m &: (g^r, g^{rs} g^m)
\end{aligned}
$$

Reusing randomness

$$\begin{aligned}
\text{Public parameters} &: p, \mathcal{G}, g, \ell \\
\text{Secret key} &: \vec{s} = s_1 \dots s_\ell \\
\text{Public key} &: g^{\vec{s}} = g^{s_1} \dots g^{s_\ell} \\
\text{Ciphertext for } \vec{x} &: (g^r, g^{r\vec{s}} g^{\vec{x}} = g^{rs_1} g^{x_1} \dots g^{rs_\ell} g^{x_\ell})
\end{aligned}$$

Reusing randomness

$$\begin{aligned} \text{Public parameters} &: p, \mathcal{G}, g, \ell \\ \text{Secret key} &: \vec{s} = s_1 \dots s_\ell \\ \text{Public key} &: g^{\vec{s}} = g^{s_1} \dots g^{s_\ell} \\ \text{Ciphertext for } \vec{x} &: (g^r, g^{r\vec{s}} g^{\vec{x}} = g^{rs_1} g^{x_1} \dots g^{rs_\ell} g^{x_\ell}) \end{aligned}$$

Now onto correctness...

Using homomorphism to decrypt the inner product

$$\text{Secret key}: \vec{s} = s_1 \dots s_\ell$$

$$\text{Public key}: g^{\vec{s}} = g^{s_1} \dots g^{s_\ell}$$

$$\text{Ciphertext for } \vec{x}: (g^r, g^{r\vec{s}} g^{\vec{x}} = g^{rs_1} g^{x_1} \dots g^{rs_\ell} g^{x_\ell})$$

Correctness

$$g^{rs_1} g^{x_1} g^{rs_2} g^{x_2} = g^{r(s_1+s_2)} g^{x_1+x_2}$$

Using homomorphism to decrypt the inner product

$$\text{Secret key}: \vec{s} = s_1 \dots s_\ell$$

$$\text{Public key}: g^{\vec{s}} = g^{s_1} \dots g^{s_\ell}$$

$$\text{Ciphertext for } \vec{x}: (g^r, g^{r\vec{s}} g^{\vec{x}} = g^{rs_1} g^{x_1} \dots g^{rs_\ell} g^{x_\ell})$$

Correctness

$$g^{rs_1} g^{x_1} g^{rs_2} g^{x_2} = g^{r(s_1+s_2)} g^{x_1+x_2}$$

$$\prod_i (g^{rs_i} g^{x_i})^{y_i} = (g^r)^{\sum_i y_i s_i} g^{\sum_i x_i y_i}$$

First trick

You can change easily the basis used in the whole scheme

First trick

You can change easily the basis used in the whole scheme

Given a matrix $\mathbf{P}$, a ciphertext $\mathbf{ct}_{\vec{x}}$, and the master secret key $\vec{s}$

You can generate a new ciphertext $\mathbf{ct}_{\mathbf{P}\vec{x}}$ using the homomorphism, and a new master secret key $\mathbf{P}\vec{s}$

Second trick

In the security game, there exists a basis in which the adversary cannot find the first coordinate

Second trick

In the security game, there exists a basis in which the adversary cannot find the first coordinate

Indeed, $\mathcal{A}$ can only ask secret keys for $\vec{y}$ such that $< \vec{y}, \vec{x_1} - \vec{x_0} >= 0$

So a basis having $\vec{x_1} - \vec{x_0}$ as first vector verifies this

Putting it together

Here is a simulator $\mathcal{S}$ using both tricks to solve a challenge given an adversary breaking the scheme:

Putting it together

Here is a simulator $\mathcal{S}$ using both tricks to solve a challenge given an adversary breaking the scheme:

- $\mathcal{S}$ finds a basis having $\vec{x_1} - \vec{x_0}$ as first vector

Putting it together

Here is a simulator $\mathcal{S}$ using both tricks to solve a challenge given an adversary breaking the scheme:

- $\mathcal{S}$ finds a basis having $\vec{x_1} - \vec{x_0}$ as first vector
- $\mathcal{S}$ generates $\mathbf{ct}^*$ with its input challenge in the first coordinate

Putting it together

Here is a simulator $\mathcal{S}$ using both tricks to solve a challenge given an adversary breaking the scheme:

- $\mathcal{S}$ finds a basis having $\vec{x_1} - \vec{x_0}$ as first vector
- $\mathcal{S}$ generates $\mathbf{ct}^*$ with its input challenge in the first coordinate
- $\mathcal{S}$ moves $\mathbf{ct}^*$ in the correct basis

Putting it together

Here is a simulator $\mathcal{S}$ using both tricks to solve a challenge given an adversary breaking the scheme:

- $\mathcal{S}$ finds a basis having $\vec{x_1} - \vec{x_0}$ as first vector
- $\mathcal{S}$ generates $\mathbf{ct}^*$ with its input challenge in the first coordinate
- $\mathcal{S}$ moves $\mathbf{ct}^*$ in the correct basis

$\square$

What properties do we need?

2 properties:

Randomness Reuse $g^r, g^{r\vec{s}} g^{\vec{x}}$ is safe

In this case, it is an instance of ElGamal with secret keys r and randomnesses s_i

Homomorphism of message and key

$$g^{rs_1+x_1} g^{rs_2+x_2} = g^{r(s_1+s_2)+(x_1+x_2)}$$

How to generalize?

To generalize, replace:

- $s \rightarrow sk$
- $g^s \rightarrow pk$
- $g^r \rightarrow C(r)$
- $g^{rs+x} \rightarrow Enc(pk, x; r)$

the LWE assumption

$$\begin{aligned} \text{Public parameters} &: q, n, m, \mathbf{A} \in \mathbb{Z}_q^{m\times n} \\ \text{Secret key} &: \vec{s} \in \mathbb{Z}_q^m \\ \text{Public key} &: \mathbf{A}\vec{s} + \vec{e} \in \mathbb{Z}_q^m & \vec{e} \leftarrow \chi^m \\ \text{Ciphertext for } x &: (\vec{r}\mathbf{A}, \vec{r}(\mathbf{A}\vec{s} + \vec{e}) + \lfloor \frac{q}{2} \rceil x) & \vec{r} \leftarrow \{0,1\}^{1\times m} \end{aligned}$$

the LWE assumption

$$\text{Public parameters}: \; q, n, m, \mathbf{A} \in \mathbb{Z}_q^{m\times n}$$

$$\text{Secret key}: \; \vec{s} \in \mathbb{Z}_q^{m}$$

$$\text{Public key}: \; \mathbf{A}\vec{s} + \vec{e} \in \mathbb{Z}_q^{m} \qquad \vec{e} \leftarrow \chi^m$$

$$\text{Ciphertext for } x: \; (\vec{r}\mathbf{A}, \vec{r}(\mathbf{A}\vec{s} + \vec{e}) + \lfloor \frac{q}{2} \rceil x) \qquad \vec{r} \leftarrow \{0,1\}^{1\times m}$$

Advantages

- Avoid small space restriction of additive ElGamal
- Post-quantum

the LWE assumption

$$\begin{aligned} \text{Public parameters} &: q, n, m, \mathbf{A} \in \mathbb{Z}_q^{m\times n} \\ \text{Secret key} &: \vec{s} \in \mathbb{Z}_q^m \\ \text{Public key} &: \mathbf{A}\vec{s} + \vec{e} \in \mathbb{Z}_q^m & \vec{e} \leftarrow \chi^m \\ \text{Ciphertext for } x &: (\vec{r}\mathbf{A}, \vec{r}(\mathbf{A}\vec{s} + \vec{e}) + \lfloor \frac{q}{2} \rceil x) & \vec{r} \leftarrow \{0,1\}^{1\times m} \end{aligned}$$

Advantages

- Avoid small space restriction of additive ElGamal
- Post-quantum

Inconvenients

Noisy setup - proof is more subtle

Work in progress

Work in progress

What is there left to do?

- Adaptive security
  $\mathcal{A}$ gets **pk** before choosing $\vec{x_0}$ and $\vec{x_1}$

Work in progress

Work in progress

What is there left to do?

- Adaptive security
  $\mathcal{A}$ gets **pk** before choosing $\vec{x_0}$ and $\vec{x_1}$
- Function privacy
  In private setting - $\mathcal{A}$ doesn't know what his key compute

Work in progress

Work in progress
What is there left to do?

- Adaptive security
  $\mathcal{A}$ gets **pk** before choosing $\vec{x_0}$ and $\vec{x_1}$
- Function privacy
  In private setting - $\mathcal{A}$ doesn't know what his key compute
- Find other interesting fitting PKE
  Paillier-like cryptosystem would solve the small space restrictions
- etc.

Thank you!

Thank you for your attention!