CSC 2419: Lattice-based Cryptography **Fall 2025**

Lecture 10: Functional Encryption and iO

Instructor: Akshayaram Srinivasan *Scribe: Kenny Li*

Date: 2025-11-24

10.1 Idea of Functional Encryption

What is Functional Encryption (FE)? Informally, it is a generalization of Attributed-Based Encryption (ABE) that "hide the attribute".

An Attribute-Based Encryption scheme for the class of functions is defined by a tuple of efficient algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ where KeyGen generates the master public key and the master secret key $(\mathsf{mpk}, \mathsf{msk})$, for public attribute $x \in \{0,1\}^l$ and a message m, $\mathsf{Enc}(\mathsf{mpk}, x, m)$ outputs a ciphertext ct, and for a policy $f : \{0,1\}^l \to \{0,1\}$, $\mathsf{KeyGen}(\mathsf{msk}, f)$ generates the decryption key sk_f so that $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$ output m if and only if $f(x) = 0$

The only secret in ABE is m, while in FE, given sk_f, we would like the decryption to output $f(x)$ and all information except $f(x)$ are hidden, so that the attribute x is also private given sk_f and ct.

High level idea of constructing FE under LWE: under FHEEnc, we can encrypt $f(x)$ but simply giving the secret of FHE doesn't work as a decryption because it will also reveal x. We should allow only $\mathsf{FHEEnc}(\mathsf{mpk}, f(x))$ but not any other to be decrypted. Then the decryption should be able to reveal $f(x)$ without learning other information.

The blueprint is that we use two other tools, namely garbling scheme and two-message attributed-based encryption scheme, to hide the FHEDec except for the one-time use to decrypt $f(x)$.

Remark: The FE scheme allows one to outsource an analysis f on encrypted private data x to another party, so that that party can only learn $f(x)$ without learning any other information about x.

10.2 Definition of Functional Encryption

We define Functional Encryption formally as follows:

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$ where 1^λ is security parameter
- $\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$
- $\mathsf{Keygen}(\mathsf{msk}, f) \to \mathsf{sk}_f$
- $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}) \to y$

For **correctness**, we require $y = f(x)$ with overwhelming probability

$$Pr[\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}) = f(x), (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda), \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x), \mathsf{sk}_f \leftarrow \mathsf{Keygen}(\mathsf{msk}, f)] = 1 - negl(\lambda)$$

For **security**, we want for all PPT adversary, $\exists \mathsf{Sim}$, a PPT simulator so that $\forall x, \forall f$

$$\{\mathsf{mpk}, \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x), \mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)\} \approx_C \{\mathsf{Sim}(1^\lambda, f, f(x))\}$$

Remark: we can also extend to the case where there are bounded number of functions, $\forall x, \forall f_1, ..., f_t$

$\{\mathsf{mpk}, \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x), \mathsf{sk}_{f_1} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f_1), ..., \mathsf{sk}_{f_t} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f_t)\} \approx_C \{Sim(1^\lambda, f_1, ..., f_t, f_1(x)), ..., f_t(x)\}$

However, one can show that it is impossible to achieve such a security when t is unbounded.

10.3 Garbled Circuits

To construct Functional Encryption, we are going to introduce an important tool called Garbled Circuits.

The goal of garbled circuits is to hide the input and the circuit C. Garbling scheme is a fundamental tool in cryptography, it implies a lot of things like security computation, zero knowledge proof etc...

A garbling scheme is defined as follows:

- $\mathsf{Grable}(1^\lambda, C) \to (\tilde{C}, \{label_{i,b}\}_{i\in[n],b\in\{0,1\}})$ where $\tilde{C}$ is a garbled circuit and the label can be thought of as a key associated to a single input x,that allows decryption to reveal only $C(x)$.
- $\mathsf{GbEval}(\tilde{C}, \{label_{i,b}\}_{i\in[n],b\in\{0,1\}}) \to y$

For **correctness**, we require

$$Pr[y = C(x), y \leftarrow \mathsf{GbEval}(\tilde{C}, \{label_{i,b}\}_{i\in[n],b\in\{0,1\}}), (\tilde{C}, \{label_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Grable}(1^\lambda, C)] = 1 - negl(\lambda)$$

For **security**, we want the garbling scheme reveals nothing except the size of the circuit C and the output $C(x)$, everything else should be encrypted. Namely,

$$\{(\tilde{C}, \{label_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Grable}(1^\lambda, C)\} \approx_C \{\mathsf{Sim}(1^\lambda, C(x)), 1^{|C|})\}$$

Relationship to Obfuscation: In an obfuscated program, any input can be evaluated by the program. While in the case of garbling scheme, only one input x can be evaluated using the circuit C. In this sense the garbling scheme can be thought as an one-time obfuscation program.

The fact that you are allowed an one-time use of the program, implies we can construct garbling scheme from one-way function. (In NC1, it doesn't need any assumptions)

Idea of using garbling scheme to construct FE: we can take $C = \mathsf{FHEDec}(\mathsf{sk}, \cdot)$ and we should allow only the garbling evaluates only on $\mathsf{FHEEnc}(f(x))$ but not other value. Note that given both sets of labels, we can run the garble circuit on any input, in particular, learns $\mathsf{FHEEnc}(x)$. That means we also need security on the labels in the garbling scheme. The tool we are going to achieve this is by the Two-message Attributed-Based Encryption, that is using ABE to hide the another labels.

10.4 Two-message Attributed-Based Encryption

Instead of encrypting one message in one ciphertext, we now modifies attribute-based encryption scheme to encrypt two messages m_0, m_1 in a same ciphertext such that in the decryption, m_0 is revealed if $f(x) = 0$

and m_1 is revealed if $f(x) = 1$. The security of the ABE scheme ensures the evaluator cannot decrypt any other message. In other words, $m_{f(x)}$ is revealed while $m_{1-f(x)}$ is hidden in the decryption.

For attribute $x \in \{0,1\}^n$ and a predicate $f\{0,1\}^l \to \{0,1\}$, we define Two-message Attributed-Based Encryption (ABE_2) as follows

- $\mathsf{ABE}_2\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$
- $\mathsf{ABE}_2\mathsf{Enc}(\mathsf{mpk}, x, m_0, m_1) \to \mathsf{ct}$
- $\mathsf{ABE}_2\mathsf{KeyGen}(\mathsf{msk}, f) \to \mathsf{sk}_f$
- $\mathsf{ABE}_2\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}_f) \to y$

For **correctness**, we require $y = m_{f(x)}$ with overwhelming probability

$$\begin{aligned} Pr[&\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}) = m_{f(x)}, (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{ABE}_2\mathsf{Setup}(1^\lambda), \\ &\mathsf{ct} \leftarrow \mathsf{ABE}_2\mathsf{Enc}(\mathsf{mpk}, x), \mathsf{sk}_f \leftarrow \mathsf{ABE}_2\mathsf{Keygen}(\mathsf{msk}, f)] = 1 - negl(\lambda) \end{aligned}$$

For **security**, what we want is essentially the same as the selective security of the usual ABE scheme,that is for an experiment Exp_{ABE_2} (except now encrypt (m, m_b) if $f(x) = 0$ or encrypt (m_b, m) if otherwise for randomly sampled $b \in \{0,1\}$)

$$Pr[b = b' \leftarrow Exp_{\mathsf{ABE}_2}] \leq 1/2 + negl(\lambda)$$

For more details of the selective security, refer to lecture 8 or [GKP13]

10.5 Construction of Functional Encryption

Given the garbling scheme and two-message attributed-base encryption scheme, construct functional encryption as follows: take $C = \mathsf{FHEDec}(\mathsf{sk}, \cdot)$ for garbling scheme and $\mathsf{FHEEnc}(\mathsf{pk}, x)$ as attribute for ABE_2,we encrypt the $2n$ labels of the garbling scheme by ABE_2

- $\mathsf{Setup}(1^\lambda) : (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{ABE}_2\mathsf{Setup}(1^\lambda)$
- $\mathsf{Enc}(\mathsf{mpk}, x)$:

$$\begin{aligned} Sample(\mathsf{pk}, \mathsf{sk}) &\leftarrow \mathsf{FHEKeyGen}(1^\lambda) \\ \mathsf{ct} &\leftarrow \mathsf{FHEEnc}(\mathsf{pk}, x) \\ (\tilde{C}, \{label_{i,b}\}_{i\in[n],b\in\{0,1\}}) &\leftarrow \mathsf{Grable}(1^\lambda, C) \\ \mathsf{ct}_i &\leftarrow \mathsf{ABE}_2\mathsf{Enc}(\mathsf{mpk}, \mathsf{ct}, \{label_{i,0}\}, \{label_{i,1}\}) \end{aligned}$$

- $\mathsf{KeyGen}(\mathsf{msk}, f)$:

$$\begin{aligned} f_i &\leftarrow \mathsf{FHEEval}(f, \cdot)[i] \\ \mathsf{sk}_{f_i} &\leftarrow \mathsf{ABE}_2\mathsf{KeyGen}(\mathsf{msk}, f_i) \end{aligned}$$

- $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$:

$$\begin{aligned} \{label_{i,\mathsf{FHEEnc}(f(x))[i]}\} &\leftarrow \mathsf{ABE}_2\mathsf{Dec}(\mathsf{sk}_{f_i}, ct_i) \\ f(x) &\leftarrow \mathsf{GbEval}(\tilde{C}, \{label_{i,\mathsf{FHEEnc}(f(x))[i]}\}) \end{aligned}$$

Intuitively the above scheme is secure because ABE_2 ensures the labels corresponding to $\mathsf{FHEEnc}(f(x))$ is revealed and the labels not used are hidden so that $\mathsf{FHEEnc}(x)$ is hidden hence x is hidden.

On the other hand, garbling ensure FHEDec is hidden and $f(x)$ is revealed .

Hence, with ct and sk_f, only $f(x)$ is revealed but not any other information is revealed.

proof of security: We prove by considering Hybrids.

Hybrid 0: $(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}_f)$ We cannot use security of FHE because garbling scheme is using $\mathsf{Dec}(\mathsf{sk}, \cdot)$ depending on sk not x, and we cannot use security of garbling as we are using both $\{label_{i,0}\}, \{label_{i,1}\}$

Hybrid 1:the only thing to use is ABE_2 implies $\{label_{i,\mathsf{FHEEnc}(1-f(x))[i]}\})$ is hidden, given ct and sk_f.

Hybrid 2:for $(\tilde{C}, \{label_{i,b}\}_{i\in[n],b\in\{0,1\}})$ now we can use garbling security since only one of the $\{label_{i,b}\}$ is known,and the another labels are hidden. So in this Hybrid, it is computationally indistinguishable to $\mathsf{Sim}(1^{|\mathsf{Dec}(\mathsf{sk},\cdot)|}, f(x))$

Hybrid 3: for $\mathsf{Sim}(1^{|\mathsf{Dec}(\mathsf{sk},\cdot)|}, f(x))$,just requiring $f(x)$ and generate the two message ABE and so on, security of $\mathsf{FHEEnc}(\mathsf{sk}, 0) \to \mathsf{ct}$ ensures x is hidden.

10.6 Indistinguishability Obfuscation

Functional Encryption (FE) has special place in cryptography as it is closely related to Indistinguishability Obfuscation (iO).

In last class, we saw Virtual Black-Box Obfuscation (VBB) which requires learning nothing more than treating the obfuscated program as a black-box. This is a too strong assumption in the sense that there are some program that are not able to be VBB.

Indistinguishability Obfuscation is much weaker than VBB. For $C_0 \equiv C_1$ functional equivalent such that the truth table on all inputs are the same, so we can use two different ways to implement the circuits. Then Indistinguishability Obfuscation requires just computational indistinguishable on the obfuscated program

$$iO(C_0) \approx_C iO(C_1)$$

iO implies FE for all circuits where the size of ciphertext is independent of the function, so that $|\mathsf{ct}|$ doesn't grow with f The converse is true if FE has sub-exponential security. So if we can compress the ciphertext size from $m \cdot poly(\lambda)$ to $m^{1-\epsilon} \cdot poly(\lambda)$, it will implies iO.

For x_0, x_1, $\forall i = 1, .., t$ for t possibly unbounded, it $f_i(x_0) = f_i(x_1)$ then

$$\{(\mathsf{sk}_{f_1}, ..., \mathsf{sk}_{f_t}, \mathsf{Enc}(x_0))\} \approx_C \{(\mathsf{sk}_{f_1}, ..., \mathsf{sk}_{f_t}, \mathsf{Enc}(x_1))\}$$

This is not known under the LWE assumption! This indicates the significance of iO.

Moreover, iO plus one-way functions implies most of the other cryptography schemes including non-interactive zero knowledge (NIZK), public key encryption(PKE), Hardness of computing Nash equilibrium, etc...

There are a lot of recent cryptanalysis on mutli-linear maps. A series of latest result shows that iO can be constructed from LWE, learning with parity(LPN), low complexity PRG and bilinear maps, in which LWE, LPN and loow complexity PRG are believed to be quantum resistant and bilinear maps are quantum broken. Latest result removes the assumptions of LWE and low complexity PRG replaced by sparse LPN.

iO with additive homomorphic encryption(AHE) implies FHE. Where every other candidates are lattice-based but this construction is not related to lattice.

An outstanding open problem of cryptography is to construct iO from plain LWE.

Reference [GKP13] S.Goldwasser, Y.Kalai, R.A.Popa, V.Vaikuntanathan, N.Zeldovich. Reusable Garbled Circuits and Succinct Functional Encryption.2013.