J Korea Industr Inf Syst Res Volume 18 Number 3
http://dx.doi.org/10.9723/jksiis.2013.18.3.017

Universal Composability Notion for Functional Encryption Schemes[†]

Rifki Sadikin[*], YoungHo Park[**], KilHoum Park[***], SangJae Moon[***]

Abstract We have developed an ideal functionality for security requirement of functional encryption schemes. The functionality is needed when we want to show the security of a functional encryption scheme in universal composable (UC) framework. A functionality F_{fe} was developed to represent ideal respond of a functional encryption scheme against any polynomial time active attacker. We show that UC security notion of functional encryption scheme F_{fe} is as strong as fully secure functional encryption in an indistinguishable game with chosen cipher text attack. The proof used a method that showing for any environment algorithm, it can not distinguish ideal world where the attacker play with ideal functionality F_{fe} and real world where the attacker play a fully secure functional encryption scheme.

Key Words : *universal composable framework, formal security notion, functional encryption scheme*

1. Introduction

A functional encryption scheme is a generalization of public encryption scheme that allows a fine grained relation between ciphertexts and secret keys. In a functional encryption scheme, a secret key SK_x is parametrized by a parameter x and a ciphertext CT_y is parameterized by a parameter y. Decryption for CT_y by SK_x is succeed only if a functional relation $F(x,y)$ between x and y is held. The functional relation $F(x,y)$ can realize complex relation between ciphertexts and secret keys.

Some existing encryption schemes can be expressed in functional encryption scheme. For example, ID based encryption system in [1-3] can expressed in a functional encryption scheme by setting parameter $x = ID$ and $y = ID'$ are from the same identity space and the functional relation as $F(x,y) = 1 \;\; \text{if} \;\; x = y, otherwise\ 0$. Fuzzy identity based encryption proposed in [4,5] can also be defined in a functional encryption scheme by setting x,y are sets of attribute from the same attribute space and the functional relation as $F_d(x,y) = 1 \;\; \text{if} \;\; |x \cap y| \geq d, otherwise\ 0$ where d is an integer. Moreover, any public key encryption scheme which has complex relation between ciphertexts and secret keys such as predicate encryption scheme [6,7] and attributed based encryption [8-10] can be formulated as a functional encryption scheme with a proper functional relation such as $F(\vec{x},\vec{y}) = 1 \;\; \text{if} \;\; \langle \vec{x},\vec{y} \rangle = 0, otherwise\ 0$ for inner product encryption scheme and

[†] This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(NRF-2012R1A1A4A01002603)

[*] School of Electrical Engineering and Computer Science, Kyungpook National University, First Author
[**] Department of Electronics Engineering, Kyungpook National University, Corresponding Author(parkyh@knu.ac.kr)
[***] School of Electronics Engineering, Kyungpook National University

$F(x,\Gamma_y)=1$ if $x \in \Gamma_y$, $otherwise$ 0 where Γ_y is an access structure for ciphertext policy attribute based encryption.

Many security services can be realized by a functional encryption system, For example, biometric identity based encryption which used a fuzzy identity based encryption (FIBE). Since the data acquisition in biometric prone to error, the FIBE scheme naturally compliance with biometric system [11]. Other applications of functional encryption schemes include realizing an access control mechanism without a trusted server. For example, a functional encryption scheme is used for group key distribution in a VANET [12] and for realizing access control mechanism in a cloud computer network [13].

Traditionally the security of a functional encryption scheme is proving under a indistinguishable game. There are two types of indistinguishable game: selective and full security. In selective security, the challenge ciphertext parameter is sent before the game started while there is no such restriction in full security functional encryption in indistinguishable game. However, in 2011 Boneh et al proposes much richer security framework for functional encryption schemes in simulation based framework [14].

One of simulation based security proof frameworks is universal composability (UC) model [15]. Universal compose ability framework allows one to define security requirement in an ideal functionality which contains interfaces and responses that should be realized by the real protocol. In UC, the real protocol securely realizes the functionality if any environment can't distinguish the interaction between ideal functionality and real protocol in the presence of all adversary. The first step to provide security argument in UC framework is defining an ideal functionality. Many ideal functionalities already been defined for primitive cryptographic protocol such as public encryption scheme and signature scheme [15], secure authentication [16] and ID-based encryption system [17].

In this paper, we formulated the ideal functionality for functional encryption schemes for UC framework security. Our work is extending UC notions for public key encryption scheme [15] and ID-based encryption system [17]. We generalized simulation based security framework for functional encryption scheme given in [14]. Furthermore, we also investigated the relation between UC formulation of a functional encryption and indistinguishable-game based security notions.

We organize our paper as follows: in Section 2, we recall functional encryption definition and its indistinguishable game based security definition. In Section 3, we present our proposed ideal functionality for UC-based functional encryption security framework. The relation between UC-based functional encryption and indistinguishable game based security definition is discussed in Section 4. At the end, we conclude our paper with conclusions and further studies in Section 5.

2. Functional Encryption Scheme and its Security Definitions.

In this section, we recall a functional encryption scheme definition and its security under indistinguishable games.

2.1 Functional Encryption Scheme Definition

Let us define $U=\{p_1,p_2,\ldots,p_n\}$ as a set of attributes/identities, we called U as an attribute space and p_i as an attribute and M as a message space. We denote X as a secret key space and Y as a ciphertext parameter space. X and Y might be the same as attribute space U, attributes set space U^n, or power of attributes set space 2^U. Let us we have a functional relation $F(x,y)$ that map

$x \in X$ and $y \in Y$ to a binary value $\{0,1\}$. We call the functional relation is held if $F(X, Y)$ returns 1.We define a functional encryption scheme as follows:

Definition 1. Functional Encryption Scheme. The generic functional encryption scheme Σ consists of four probabilistic polynomial time algorithms: $setup$, $extract$, $encrypt$ and $decrypt$.

- $setup(\lambda)$: The $setup$ algorithm takes λ as a security parameter and returns a master key msk and a set of public parameters pk. The set of public parameters pk should be made public and the master key msk is kept secret (known only by setup party).
- $extract(pk, msk, x)$: The $extract$ algorithm takes a set of public parameters pk, the master key msk and a secret key parameter x and returns a secret key sk_x that parameterized by x.
- $encrypt(pk, y, m)$: The $encrypt$ algorithm takes a set of public parameters pk, a ciphertext parameter y and a message $m \epsilon M$. It returns a ciphertext ct_Y that parameterized by y.
- $decrypt(pk, sk_x, ct_y)$: the $decrypt$ algorithm takes a set of public parameters pk, a secret key sk_X parameterized by X and a ciphertext ct_Y parameterized by Y. The decrypt algorithm returns a message $m \epsilon M$ or a random $\perp$.

A functional encryption scheme $\Sigma = (setup, extract, encrypt, decrypt)$ over a set of attributes $U = \{p_1, p_2, \ldots, p_n\}$, a plaintext space M, a secret key parameter space X, a ciphertext parameter space Y and a functional relation $F(x, y)$ should satisfy the correctness requirement:

- $\forall m \in M, \forall x \in X, \forall y \in Y$
- $(pk, msk) \leftarrow setup(\lambda)$
- $SK_x \leftarrow extract(pk, msk, x)$
- $CT_y \leftarrow encrypt(pk, y, m)$
- If $F(x, y) = 1$ then $m \leftarrow decrypt(pk, SK_x, CT_y)$

2.2 Security of Fuzzy Identity based Encryption

Security notion for a functional encryption scheme $\Sigma = (setup, extract, encrypt, decrypt)$ is formulated in term of an indistinguishable game with presence of an adaptive chosen plaintext attacker A. We denote this game as $FE-IND-CPA2$. The indistinguishable game consists the following phases:

- **Setup.** The challenger C runs the $setup(\lambda)$ algorithm and gives a set of public parameters pk to the attacker A.
- **Phase 1.** The attacker A sends secret key queries for secret key parameters $x_1, \ldots, x_q$. The challenger C responds by sending $SK_{x_i} \leftarrow extract(pk, msk, x_i)$ for each secret key query to the attacker A.
- **Challenge.** The attacker A choose two messages with equal length from message space $m_0, m_1 \in M$ and a target ciphertext parameter y^* with restriction for all queried secret key parameters $x_1, \ldots, x_q$ none of them satisfy $F(x_i, y^*) = 1$. The challenger C flips the random coin $b \leftarrow \{0,1\}$, and send $CT_{y^*} \leftarrow extract(pk, m_b, y^*)$ to the attacker A.
- **Phase 2.** Phase 1 repeated, with restriction and none of the quired secret key parameters $x_{q+1}, \ldots, x_{2q}$ satisfy $F(x_i, y^*) = 1$.
- **Guess.** At the end, the attacker A outputs a guess b' for b

The advantage of the attacker A in this game is defined as

$$Adv_{\Sigma, A}^{FE-IND-CPA2} = \left| \Pr[b' - b] - \frac{1}{2} \right| \qquad (1)$$

If we allow $2q$ decryption queries in **Phase 1** and **Phase 2** such that the attacker A sends queried arbitrary ciphertext CT_i, and the challenger

C responds with $m'_i \leftarrow decrypt(pk, SK_{x_i}, CT_i)$ then we achieve indistinguishable game under adaptive chosen-ciphertext attacker which we denote by $FE-IND-CCA2$. Definition for selective functional encryption security model is similar with fully functional encryption with exception the target ciphertext parameter $y*$ is given ahead before the game is started.

Definition 2. Fully (Selective) Secure Functional Encryption. A functional encryption scheme $\Sigma = (setup, extract, encrypt, decrypt)$ is called fully (selective) secure functional encryption if for all PPT chosen plaintext (ciphertext) attacker A has at most a negligible advantage in $FE-IND-CPA(CCA)2$ game.

3. Universally Composable Functional Encryption

3.1 Universal Composable Security Framework

Universal composable (UC) framework is a simulation-based formal framework for showing the security of protocol. UC framework follows ideal-real world separation. Security of protocol is defined by comparing the output of ideal and real given to environment. If the distribution ensembles of output of ideal and real given to environment are indistinguishable then the protocol run in real world emulates the ideal functionality run in ideal world [15].

Real world simulation consists of several PPT machines: an environment Z, l parties $P_1, \ldots, P_l$ that runs a protocol π, and an attacker A. The execution in the real world is as follows: the initial machine is the environment Z, Z invokes the adversary A with security parameter λ and an input k. The adversary A can perform deliver messages to any party and corrupt a party action. Then, Z activated party $P_1, \ldots, P_l$ to run the protocol π. Let us denote $REAL_{\pi, A, Z}(\lambda, k)$ as ensembles of outputs in real world execution.

In ideal world, the simulation includes a special party called ideal functionality. Ideal world consists of several PPT machines: an environment Z, a special machine F that runs the ideal functionality, an ideal adversary S that simulates all possible real life attacker and l dummy parties $\widehat{P}_1, \ldots, \widehat{P}_l$ that are doing nothing but forwarding messages they received. Let us denote $IDEAL_{F, A, S}(\lambda, k)$ as ensembles of outputs in real world execution.

Definition 3 UC securely realize, [15]. Let us define F as an ideal functionality and π is a multi-party real protocol. We say that π UC securely realizes F if for any PPT adversary A there exist an ideal simulator S such that for any environment Z:

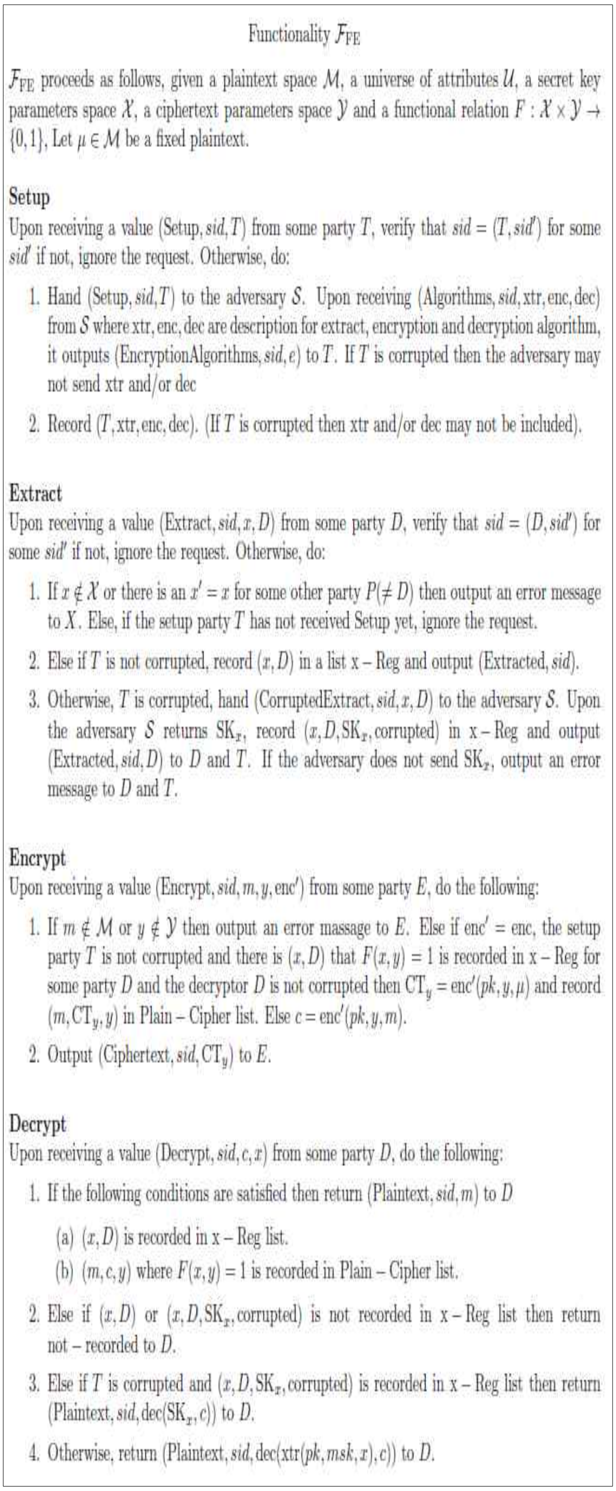
$$|REAL_{\pi,A,Z}(\lambda,k) - IDEAL_{F,S,Z}(\lambda,k)| < v(\lambda) \qquad (2)$$

where $v(\lambda)$ is a negligible function.

3.2 Functional Encryption Ideal Functionality

We construct an ideal functional encryption schemes functionality F_{FE}. The functionality F_{FE} ,presented in Figure 1, is inspired by ideal functionality for public key and identity based encryption schemes [15,17]. We modified ideal functionality for public key and identity based encryption in such it can accommodate functional relation that is used in a functional encryption scheme.

Some points about the functionality F_{FE}: If no party is corrupted in setup, extract, encrypt and decrypt then F_{FE} will provide information theoretically secure encryption (no relation between ciphertexts and plaintexts). F_{FE} can be realized by protocols that only have local communication. The functionality allows parties to extract many secret key parameter $x_1, \ldots, x_q$ as long as x_i has not been extracted by other party. In encryption request by a party with a secret key parameter x_i, the functionality requires the secret key SK_{x_i} already been extracted.



Functionality $\mathcal{F}_{\mathrm{FE}}$

$\mathcal{F}_{\mathrm{FE}}$ proceeds as follows, given a plaintext space $\mathcal{M}$, a universe of attributes $\mathcal{U}$, a secret key parameters space $\mathcal{X}$, a ciphertext parameters space $\mathcal{Y}$ and a functional relation $F : \mathcal{X} \times \mathcal{Y} \rightarrow \{0,1\}$, Let $\mu \in \mathcal{M}$ be a fixed plaintext.

Setup

Upon receiving a value (Setup, sid, T) from some party T, verify that $sid = (T, sid')$ for some sid' if not, ignore the request. Otherwise, do:

1. Hand (Setup, sid, T) to the adversary $\mathcal{S}$. Upon receiving $(\mathrm{Algorithms}, sid, \mathrm{xtr}, \mathrm{enc}, \mathrm{dec})$ from $\mathcal{S}$ where xtr, enc, dec are description for extract, encryption and decryption algorithm, it outputs $(\mathrm{EncryptionAlgorithms}, sid, e)$ to T. If T is corrupted then the adversary may not send xtr and/or dec
2. Record $(T, \mathrm{xtr}, \mathrm{enc}, \mathrm{dec})$. (If T is corrupted then xtr and/or dec may not be included).

Extract

Upon receiving a value $(\mathrm{Extract}, sid, x, D)$ from some party D, verify that $sid = (D, sid')$ for some sid' if not, ignore the request. Otherwise, do:

1. If $x \notin \mathcal{X}$ or there is an $x' = x$ for some other party $P(\neq D)$ then output an error message to X. Else, if the setup party T has not received Setup yet, ignore the request.
2. Else if T is not corrupted, record (x, D) in a list $\mathrm{x-Reg}$ and output $(\mathrm{Extracted}, sid)$.
3. Otherwise, T is corrupted, hand $(\mathrm{CorruptedExtract}, sid, x, D)$ to the adversary $\mathcal{S}$. Upon the adversary $\mathcal{S}$ returns SK_x, record $(x, D, \mathrm{SK}_x, \mathrm{corrupted})$ in $\mathrm{x-Reg}$ and output $(\mathrm{Extracted}, sid, D)$ to D and T. If the adversary does not send SK_x, output an error message to D and T.

Encrypt

Upon receiving a value $(\mathrm{Encrypt}, sid, m, y, \mathrm{enc}')$ from some party E, do the following:

1. If $m \notin \mathcal{M}$ or $y \notin \mathcal{Y}$ then output an error massage to E. Else if $\mathrm{enc}' = \mathrm{enc}$, the setup party T is not corrupted and there is (x, D) that $F(x, y) = 1$ is recorded in $\mathrm{x-Reg}$ for some party D and the decryptor D is not corrupted then $\mathrm{CT}_y = \mathrm{enc}'(pk, y, \mu)$ and record (m, CT_y, y) in $\mathrm{Plain-Cipher}$ list. Else $c = \mathrm{enc}'(pk, y, m)$.
2. Output $(\mathrm{Ciphertext}, sid, \mathrm{CT}_y)$ to E.

Decrypt

Upon receiving a value $(\mathrm{Decrypt}, sid, c, x)$ from some party D, do the following:

1. If the following conditions are satisfied then return $(\mathrm{Plaintext}, sid, m)$ to D
   (a) (x, D) is recorded in $\mathrm{x-Reg}$ list.
   (b) (m, c, y) where $F(x, y) = 1$ is recorded in $\mathrm{Plain-Cipher}$ list.
2. Else if (x, D) or $(x, D, \mathrm{SK}_x, \mathrm{corrupted})$ is not recorded in $\mathrm{x-Reg}$ list then return $\mathrm{not-recorded}$ to D.
3. Else if T is corrupted and $(x, D, \mathrm{SK}_x, \mathrm{corrupted})$ is recorded in $\mathrm{x-Reg}$ list then return $(\mathrm{Plaintext}, sid, \mathrm{dec}(\mathrm{SK}_x, c))$ to D.
4. Otherwise, return $(\mathrm{Plaintext}, sid, \mathrm{dec}(\mathrm{xtr}(pk, msk, x), c))$ to D.

Figure 1. Functionality F_{FE}.

4. Relation between and UC-secure and Fully Secure Functional Encryption

Given a functional encryption scheme $\Sigma = (setup, extract, encrypt, decrypt)$, we can transform protocol Σ to a protocol π_Σ such π_Σ that has the same interface as F_{FE}. By this transformation, we can investigate the relation between protocol π_Σ to ideal functionality F_{FE}. The transformation proceeds as follows:

- When a party T who is running protocol π_Σ receives $(Setup, sid, T)$, it runs $(pk, msk) \leftarrow setup(\lambda)$ from Σ and binds extraction algorithm $xtr = extract(pk, mk, .)$, encryption algorithm $enc = encrypt(pk, ., .)$ and decryption algorithm $dec = encrypt(pk, ., .)$. T outputs $(EncryptionAlgorithm, sid, enc)$ and keeps extraction algorithm xtr and decryption dec for itself.
- When a party D who is running protocol π_Σ receives an input $(Extract, sid, x, D)$. If $x \not\in X$, there is a party P has extracted x or the setup party T has not run $(Setup, sid, T)$, D outputs error message. Otherwise, T runs $SK_x \leftarrow extract(pk, mk, x)$ and sends the private key SK_x securely to D. When D receives SK_x, it outputs $(Extracted, sid, D)$ otherwise outputs error message.
- When a party E running protocol π_Σ receives an input $(Encrypt, sid, m, y, enc)$. If $m \not\in M$, or $y \not\in Y$, it outputs an error message. Otherwise, E runs $c \leftarrow encrypt(pk, y, m)$ and outputs $(Ciphertext, sid, c)$.
- When a party D who is running protocol π_Σ. It receives an input $(Decrypt, sid, c, x)$, If D does not have private key SK_x, where $F(x, y) = 1$ then it outputs $not-recorded$, otherwise D executes $m \leftarrow decrypt(pk, SK_x, c)$ and outputs $(Plaintext, sid, m)$.

Theorem 1. Let $\Sigma=(setup,extract,encrypt,decrypt)$ is a functional encryption scheme, π_{Σ} securely realizes the functionality F_{FE} in a presence of non-adaptive adversary if and only if Σ is a fully secure functional encryption with adaptive chosen ciphertext attacker.

Proof. Firstly, we give a proof that if π_{Σ} UC-realizes F_{FE} then Σ is fully secure functional encryption with adaptive chosen ciphertext. In other word, if we have an attacker G that wins in $FE-IND-CCA2$ game with advantage $Adv_{\Sigma,G}^{FE-IND-CPA2} > v(\lambda)$ where $v(\lambda)$ is a negligible function then we can construct an environment Z that can that can distinguish whether it interacts with ideal world or real world more than negligible probability $v(\lambda)$ by using G. Z executes as follows:

(1) Z activates setup party T with input $(Setup, sid, T)$. When receiving $(EncryptionAlgorithm, sid, enc)$ with algorithm enc includes public parameter pk, Z forwards enc to G.

(2) When G sends an extraction query for a secret key parameter x_i to protocol Σ, environment Z sends $(Extract, sid, x_i, D)$ to a decryption party G. In this step, Z will receive secret key SK_{x_i}. Z can not tell the difference whether it interacts with ideal world or real world since they are returning the same value SK_{x_i}. At the end, Z passes SK_{x_i} to G.

(3) When G sends a decryption query for a pair of secret key parameter dan a arbitrary ciphertext (x_i, CT_i) to protocol Σ, the environment Z sends $(Decrypt, sid, CT_i, x_i)$ to a decryption party D. If the secret key SK_{x_i} has not been extracted Z select other uncorrupted decryptor party $\widetilde{D}$, and inputs $(Extract, sid, x_i, D)$. After receiving $(Extracted, sid, D)$, the environment Z send $(Decrypt, sid, CT_i, x_i)$ to decryption party $\widetilde{D}$. When Z receives $(Plaintext, sid, m_i)$, the environment Z sends m_i to G.

(4) When G challenges the protocol Σ by selecting a target ciphertext parameter y^* and 2 equal length messages m_0 and m_1. The environment Z flips a random bit $b \leftarrow \{0,1\}$ and sends $(Encrypt, sid, m_b, y^*, enc)$ to the encryption party E. When Z receives $(Ciphertext, sid, CT_{y^*})$, it sends $(Ciphertext, sid, CT_{y^*})$ to G.

(5) Repeat 2,3 with restriction none of the queried secret key parameter x_i satisfy functional relation $F(x_i, y^*) = 1$ and none of ciphertext queried CT_i satisfy $CT_i = CT_{y^*}$.

(6) When G outputs a guess b' for b, the environment $b' = b$ outputs to guess which the world it interacted.

Now we analyzing the advantage of the environment Z in distinguishing real world (π_{Σ}, A) or ideal world (F_{FE}, S). When Z interacts with the real world (π_{Σ}, A), the challenge ciphertext is $CT_{y^*} \leftarrow encrypt(pk, m_b, y^*)$. Therefore, the advantage of the environment Z is the same as the advantage of G: $Adv_G^{FE-IND-CPA2} = \left|\Pr[b=b'] - \frac{1}{2}\right| > v(\lambda)$.

While, when Z interacts with ideal world (F_{FE}, S). The functionality F_{FE} returns $CT_{y^*} \leftarrow encrypt(pk, \mu, y^*)$ where $\mu \neq m_b$ then the advantage of Z is $\left|\Pr[b=b'] - \frac{1}{2}\right| = 0$. Therefore, we can conclude the advantage of Z in distinguishing real world and ideal world is

$$|REAL_{\pi,A,Z}(\lambda,k) - IDEAL_{F,S,Z}(\lambda,k)| > v(\lambda) \qquad (3)$$

Hence the environment Z can distinguish whether it interacts with real world (π_Σ, A) or ideal world (F_{FE}, S).

Secondly, we prove the statement "if Σ is fully secure functional encryption in $FE-IND-CCA2$ game then π_Σ UC-realizes F_{FE}". This statement can be proved by constructing an adversary G that has advantage $Adv_G^{FE-IND-CPA2} > v(\lambda)$ in fully functional encryption security game $FE-IND-CCA2$ by using an environment Z which can distinguish whether it interacts with a real world (π_Σ, A) or an ideal world (F_{FE}, S) where the attacker A is a non-adaptive attacker. In other words, Z has property: $|REAL_{\pi,A,Z}(\lambda,k) - IDEAL_{F,S,Z}(\lambda,k)| > v(\lambda)$.

Since the attacker A in real world (π_Σ, A) is a non-adaptive adversary setting, the adversary A cannot corrupt any party while the execution in process. The environment Z can instruct the adversary to corrupt a setup party T or a decryption party D or encryption party E only the beginning of execution. We can argue when T, D or E was corrupted non-adaptively by A the environment Z cannot distinguish the two worlds. When no party was corrupted then we can build the adversary G by using a challenger C running the protocol Σ as follows:

(1) When Z sends $(Setup, sid, T)$ to a setup party T, G sends $Setup$ query to the challenger C after receiving a set of public parameters pk from the challenger. G builds up the encryption algorithm $enc = encrypt(pk,.,.)$ and G sets the output of T as $(EncryptionAlgorithm, sid, enc)$.

(2) When Z inputs $(Extract, sid, x_i, D)$ for some party T. G continues the simulation by sending x_i to the challenger C. The challenger C responds by executing $SK_x \leftarrow extract(pk, mk, x)$. At the end, G lets D outputs $(Extracted, sid, D)$.

(3) When Z inputs $(Decrypt, sid, CT_i, x_i)$ for some party D. G responds by sending (x_i, CT_i) to the challenger C. The challenger C responds by executing $m \leftarrow decrypt(pk, SK_{x_i}, CT_i)$. Then, G lets D outputs $(Plaintext, sid, m_i)$.

(4) When Z inputs $(Encrypt, sid, m_i, y_i, enc)$ for the first $h-1$ times for an encryption party$E$. G runs $CT_{y_i} \leftarrow encrypt(pk, m_i, y_i)$ and lets E output $(Ciphertext, sid, CT_{y_i})$.

(5) At the h-th time Z inputs $(Encrypt, sid, m_h, y_h, enc)$ to a party E. G sends $m_0 = m_h, m_1 = \mu$ and $y^* = y_h$ to challenger C. The challenger flips a random coin b and returns $CT_{y^*} \leftarrow encrypt(pk, y^*, m_b)$ to G. G lets E outputs $(Ciphertext, sid, CT_{y^*})$.

(6) When Z inputs $(Encrypt, sid, m_i, y_i, enc)$ for $h+1$-th to l-th for an encryption party E. G runs $CT_{y_i} \leftarrow encrypt(pk, m_i, \mu)$ and lets E output $(Ciphertext, sid, CT_{y_i})$.

(7) When Z halts, it outputs $b' \leftarrow \{0,1\}$, G outputs whatever Z outputs as its guess.

The advantage of G can be computed by analyzing the environment Z. When Z inputs $(Setup, sid, T)$, $(Extract, sid, x_i, D)$ and $(Decrypt, sid, CT_i, x_i)$ the environment Z cannot distinguish between the real world (π_Σ, A) or the ideal world (F_{FE}, S) since both return the same output. However, when Z inputs $(Encrypt, sid, m_i, y_i, enc)$ messages l times we have the following analysis:

Let us define H_i is the probability environment Z outputs 1 when it interacts with ideal world (F_{FE}, S) with exception the first i encryption queries used real plaintexts. Therefore,

$H_0 = IDEAL_{F_{FE},S,Z}$ and $H_l = REAL_{\pi_\Sigma,A,Z}$. Notice that in h-th time of encryption query we have two cases when $b=0$, G used a real plaintext m_h then $H_h = \Pr[G \leftarrow 1 | CT_{y^*} \leftarrow encrypt(pk, y^*, m_h)]$ and otherwise G used μ then we $H_{h-1} = \Pr[G \leftarrow 1 | CT_{y^*} \leftarrow encrypt(pk, y^*, \mu)]$.

Now, we can confer the relation between the advantage of the environment Z to distinguish the real world and ideal world and the advantage of the adversary G in $FE-IND-CCA2$ game as follows:

Notice that we have,

$$\sum_{i=1}^{l} |H_{i-1} - H_i| \geq \left| \sum_{i=1}^{l} H_{i-1} - H_i \right| \\ = |H_0 - H_1| = |IDEAL_{F_{FE},S,Z} - REAL_{\pi_\Sigma,A,Z}| > v(\lambda) \tag{4}$$

Therefore, there exists $h \in \{1,\ldots,l\}$ such that $|H_{h-1} - H_h| > v(\lambda)/l$. Since $H_{h-1} = \Pr[G \leftarrow 1 | CT_{y^*} \leftarrow encrypt(pk, y^*, \mu)]$ and $H_{h-1} = \Pr[G \leftarrow 1 | CT_{y^*} \leftarrow encrypt(pk, y^*, \mu)]$ than we can rewrite $|H_{h-1} - H_h| > v(\lambda)/l$ as

$$|\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]| > v(\lambda)/l \tag{5}$$

We can change the equation (5) to became

$$\left| \Pr[b' = b] - \frac{1}{2} \right| = Adv_{\Sigma,G}^{FE-\ IND-CCA2} > v(\lambda)/l \tag{6}$$

Since $v(\lambda)$ is a negligible function than the simulator G has advantage in $FE-IND-CCA2$ more than negligible function $v(\lambda)$. This completes our proof of Theorem 1.

5. Conclusions

Universal composable notation for functional encryption schemes was proposed. A functionality that capture ideal requirement for for a functional encryption scheme is presented in a functionality which we denote F_{FE}. We prove that a functional encryption scheme that UC realizes F_{FE} is as secure as fully secure functional encryption scheme with an adaptive chosen ciphertext attacker. This show that our proposed universal composable functionality capture the strongest security notion of functional encryption schemes. Despite this, our works can be extended to capture attribute hiding property of a functional encryption scheme.

References

[1] Shamir A., "Identity-based cryptosystems and signature schemes", *Proceedings of CRYPTO 84 on Advances in cryptology*, pp.47-53, Springer Verlag, 1985.

[2] Boneh D. and Franklin M., "Identity-Based Encryption from the Weil Pairing", *SIAM Journal of Computing*, vol 32, no 3, pp. 586-615, 2003.

[3] Gentry C., "Practical identity-based encryption without random oracle", *Proceedings of the 24th annual international conference on The Theory and Applications of Cryptographic Techniques*, pp. 445-464, St. Petersburg, Russia, 2006.

[4] Sahai A. and Waters B., "Fuzzy identity-based encryption". *EUROCRYPT, Lecture Notes in Computer Science*, Vol. 3949, pp. 457 - 473, Springer, 2005.

[5] Baek J., Susilo W. and Zhou J., "New constructions of fuzzy identity based encryption". *In Proceedings of the 2nd ACM symposium on information, computer, and communications security*, pp. 369-370, 2007.

[6] Katz J., Sahai A. and Waters B., "Predicate encryption supporting disjunctions, polynomial equations, and inner products", *Proceedings of the theory and applications of cryptographic*

techniques 27th annual international conference on Advances in cryptology, Istanbul, Turkey, pp. 146-162, 2008.

[7] Okamoto, T. and Takahima, K., "Adaptively-Hiding (Hierarchical) Inner Product Encryption", *Advances in Cryptology EUROCRYPT 2012-LNCS,* vol. 7273, 2012.

[8] Bethencourt, J., Sahai, A., Waters, B., "Ciphertext-Policy Attribute-Based Encryption," *IEEE Symposium on Security and Privacy,* pp. 321-334, 2007.

[9] Goyal V., Pandey O., and Sahai A., "Attribute-based encryption for fine-grained access control of encrypted data", *Proceedings of the 13th ACM conference on Computer and communications security,* pp. 89-98, 2006.

[10] Lewko, A., Okamoto, T., Sahai, A., Takashima, K., and Waters, B., "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption", *Advances in Cryptology - EUROCRYPT 2010 LNCS,* vol. 6110, pp. 62-91, 2010.

[11] Li F., Khan M. K., "A biometric identity-based signcryption scheme", *Future Generation Computer Systems*, vol. 28, no. 1, pp. 306-310, 2012.

[12] Dijiang H., and Mayank V., ASPE: attribute-based secure policy enforcement in vehicular ad hoc networks, *Ad Hoc Networks*, vol. 7, no. 8, pp. 1526-1535, 2009.

[13] Guojun W., Qin L., Jie W., and Minyi G, Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers, *Computers & Security,* vol. 30, no. 5, pp. 320-331, 2011.

[14] Boneh D., Sahai A., and Waters B., "Functional encryption: definition and challenges", Theory of Cryptography Lecture Notes in Computer Science, vol. 6597, pp. 253-273, 2011.

[15] Canetti, R., "Universally composable security: a new paradigm for cryptographic protocols," *Foundations of Computer Science, Proceedings. 42nd IEEE Symposium on* , pp. 136-145, 2001.

[16] Canetti, Ran and Krawczyk, Hugo, "Universally Composable Notions of Key Exchange and Secure Channels", *Advances in Cryptology – EUROCRYPT 2002 LNCS,* vol. 2332, pp. 337-351, 2002.

[17] Nishimaki R., Manabe Y., and Okamoto T., "Universally composable identity-based encryption", *IEICE Trans. Fundamental Communication Computer Science,* vol. E91-A, no. 1, pp. 262-271, 2008.

Rifki Sadikin received his B.S. in electrical engineering from Gadjah Mada University, Yogyakarta, Indonesia in 1999, M.S. degree in computer science from Indonesia University, Jakarta, Indonesia in 2004. From 2009 until now, he is a PhD student in School of Electrical Engineering and Computer Science at Kyungpook National University, Korea. His research interests include information security, computer network, and distributed system.

YoungHo Park received his BS, MS, and Ph. D degrees in electronic engineering from Kyungpook National University, Daegu, Korea in 1989, 1991, and 1995, respectively. He is currently a professor in the Department of Electronics Engineering at Kyungpook National University. In 1996-2008, he was a professor in the School of Electronics and Electrical Engineering at Sangju National University, Korea. In 2003-2004, he was a visiting scholar in the School of Electrical Engineering and Computer Science at Oregon State University, USA. His research interests include computer networks and information security.

Kil-Houm Park received his B.S. degree from Kyungpook National University, Daegu, Korea, in 1982 and his M.S. and Ph.D. degrees in electronic engineering from KAIST, Daejeon, Korea, in 1984 and 1990. He has been a Professor with the School of Electronics Engineering, Kyungpook National University, since 1984. His current research interests include medical image processing, fingerprint recognition, computer vision, and information security.

SangJae Moon received the B.E. (1972) and M.E. (1974) degrees in electronic engineering from Seoul National University, Korea, and the PhD (1984) degree in communication engineering from the Department of Electrical Engineering of the University of California, Los Angeles. He is currently a professor in the School of Electronics Engineering, Kyungpook National University, Korea. He was president of the Korea Institute of Information Security and Cryptology from February 2001 to January 2002. His research interests currently are in the areas of cryptography, network security, and security applications.

논문접수일 : 2013년 05월 09일
1차수정완료일 : 2013년 06월 07일
2차수정완료일 : 2013년 06월 19일
게재확정일 : 2013년 06월 21일