Generic Constructions for Chosen-Ciphertext Secure Attribute Based Encryption

Shota Yamada[1], Nuttapong Attrapadung[2], Goichiro Hanaoka[2], and Noboru Kunihiro[1]

[1] The University of Tokyo
{yamada@it.,kunihiro@}k.u-tokyo.ac.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST)
{n.attrapadung,hanaoka-goichiro}@aist.go.jp

Abstract. In this paper we propose generic conversions for transforming a chosen-plaintext (CPA) secure attribute-based encryption (ABE) to a chosen-ciphertext (CCA) secure ABE. The only known generic conversion, to the best of our knowledge, was presented by Goyal et al. in ACM-CCS 2006, which itself subsumes the well-known IBE-to-PKE conversion by Canetti, Halevi, and Katz proposed in Eurocrypt 2004. The method by Goyal et al. has some restrictions that it assumes the delegatability of the original ABE and can deal only with the key-policy type of ABE with large attribute universe. In contrast, our methodology is applicable also to those ABE schemes without known delegatability. Furthermore, it works for both key-policy or ciphertext-policy flavors of ABE and can deal with both small and large universe scheme. More precisely, our method assumes only either delegatability or a newly introduced property called verifiability of ABE. We then exhaustively check the verifiability of existing ABE schemes and found that most of them satisfy such a property, hence CCA-secure versions of these schemes can be obtained automatically.

1 Introduction

Background. Attribute-based encryption (ABE) is a generalized cryptographic primitive from normal public key encryption (PKE) that provides an access control mechanism over encrypted data using access policies and ascribed attributes among private keys and ciphertexts. ABE was introduced first by Sahai and Waters [30]. In an ABE system, a user in the system possesses a key associated with an access policy, stating what kind of ciphertext that she can decrypt. On the other hand, a ciphertext is associated with a set of attributes. The decryption can then be done if the policy associated to the key is satisfied by the attribute set associated to the ciphertext. This setting of ABE is called key-policy ABE (KP-ABE) since a key is associated with a policy. Its dual notion in which the role of policy and attribute set is swapped is called ciphertext-policy ABE (CP-ABE). In this setting, a policy will be associated to a ciphertext while an attribute set will be associated to a key.

D. Catalano et al. (Eds.): PKC 2011, LNCS 6571, pp. 71–89, 2011.

Most of the proposed ABE schemes [21,5,29] in the literature focused on the aspect of extending the expressiveness of policies so as to achieve fine-grained access control (see below). Some other schemes focused on extending to the multi-authority setting [13,14,3,25], while the most recent achievements in this research area were the schemes that attain adaptive security [26,28].

In this paper we focus on another important issue namely the aspect of attaining security against chosen-ciphertext attack (CCA) for ABE in the standard model. The first CCA-secure KP-ABE has already appeared in the seminal paper for the first expressive KP-ABE scheme by Goyal et al. [21]. Their CCA-secure scheme extends the methodology of converting any identity-based encryption (IBE) scheme to CCA-secure PKE scheme Canetti, Halevi, and Katz [12]. Such a technique relies on delegatability, which is the property that allows using a key of one policy $\mathbb{A}$ to construct a key for another policy $\mathbb{A}'$ that is more restricted than $\mathbb{A}$. Their construction is generic: it is a conversion that transforms any CPA-secure KP-ABE with delegation to a CCA-secure one. For the case of CCA-secure CP-ABE, Bethencourt, Sahai, and Waters [5] mentioned that a similar methodology can be used but they omitted to describe the details. We note that [5] also uses another method for their CCA-secure CP-ABE namely the Fujisaki-Okamoto conversion [18] but this can be proven only in the random oracle model. Some specific CCA-secure construction for CP-ABE with only AND-gates was proposed in [15].

To the best of our knowledge, the only generic and standard-model construction for CCA-secure ABE available in the literature is the aforementioned one by Goyal et al. [21]. The scheme works for the key-policy flavor and can deal only with the scheme for a large attribute universe. It works roughly as follows. To encrypt to an attribute set S, Bob first generates a signing and verification key pair (sk, vk) of a one-time signature scheme. Bob then constructs a ciphertext for $S \cup \{vk\}$ of the original scheme, where vk is treated as a dummy attribute, and signs this with sk. Upon receiving, Alice checks the validity of signature and then delegates her key from policy $\mathbb{A}$ to policy $\mathbb{A}$ AND vk. Alice then uses the latter key to decrypt the ciphertext.

Our Contributions. We propose eight generic conversions that transform CPA-secure ABE to CCA-secure ABE. The eights conversions comprise all the combinations by three categorization: (1) whether we consider CP-ABE or KP-ABE, (2) the original scheme deals with a small or large universe of attributes, and (3) the conversion uses which methodology out of two that we propose. One methodology is based on delegatability of ABE, while the other is based on a new property called verifiability of ABE. The former methodology is a reminiscent of the method by Goyal et al. [21] as described above. On the other hand, the latter can be considered as its "dual". Roughly speaking, while the delegatability-based method utilizes the AND functionality of ABE, the new verifiability-based method uses the OR functionality.

Before moving further, we point out an apparent strength of our thesis: one methodology has an advantage over the other in the sense that it requires only either one of the two additional properties. The new verifiability-based one does not require delegatability of ABE. It thus applies to those ABE without known delegation, such as the linear secret sharing based KP-ABE of Goyal et al. [21] and non-monotonic KP-ABE of Ostrovsky et al. [29] for instances.

Another advantage is that our methodology is generic: it converts the underlying CPA-secure ABE in the black-box manner. Readers who are familiar with ABE may argue that CCA-secure version of any ABE can be rather easy to construct since, to the best of our knowledge, all the available schemes so far were based on bilinear pairing and with this tool there are some well-known, but specific, techniques such as [12,9] (in the context of IBE) to attain CCA security. However, using such specific techniques requires researchers to construct and prove the security individually each time a new ABE is proposed, which is not quite convenient. Besides, we believe that some new ABE which is not based on pairing will be proposed in the future.

Our Approach. The new verifiability-based method works roughly as follows. We briefly describe here for the case of KP-ABE (with a large universe). The scheme is indeed similar to the aforementioned method, where Bob encrypts to $S \cup \{vk\}$, while Alice holds a key for a policy $\mathbb{A}$, with the only difference in decryption algorithms. Instead of delegating the key, Alice will use the verifiability to check a kind of well-formed-ness of ciphertext before decrypting. Such a verifiability allows to check whether a ciphertext will decrypt to the same result when using either a key for policy $\mathbb{A}$ or a key for the singleton policy $\{\{vk\}\}$. The latter key will be used only in the proof. The ability to use either key to decrypt can be considered intuitively as an (implicit) or functionality. For the case of CP-ABE, the utilization of or will become more clear: there, we explicitly use a policy of the form $\mathbb{A}$ or vk.

The use of or and some form of verifiability to attain CCA security can be traced back to the classic Naor-Yung two-key paradigm [27] in the context of CCA-secure PKE. However, their scheme poses a strong requirement: the existence of non-interactive zero knowledge proofs, and thus, enhanced trapdoor permutations. In contrast, our newly defined verifiability for ABE is indeed quite a weak requirement. Regarding this, we show the gap between our verifiability and the commonly defined public verifiability. Furthermore, for pairing-based schemes, the verifiability comes for almost free in many schemes.

Finally, we note that the described methods assume that the original ABE can deal with large universe (super-polynomial size). This is since we have to treat vk as dummy attributes. In our methodology, we further propose how to deal with small universe schemes by introducing a twist similar to the well-known technique by Dwork, Dolev, and Naor [17] (in the context of PKE).

Related Works on ABE. ABE was first introduced by Sahai and Waters [30] in the context of a generalization of IBE called Fuzzy IBE, which is an ABE that allows only single threshold access structures. The first KP-ABE scheme that allows any monotone access structures was proposed by Goyal et al. [21]. The first such CP-ABE scheme which allows the same expressiveness was proposed by Bethencourt, Sahai, and Waters [5], albeit the security of their scheme was only proved in the generic bilinear group model. Ostrovsky, Sahai, and

Waters [29] then subsequently extended both schemes to handle also any non-monotone structures. Towards constructing a CP-ABE in the standard model, Cheung and Newport [15] proposed a CP-ABE scheme that allows only AND gate, while Goyal et al. [20] proposed a CP-ABE scheme which allows only a-priori bounded size of gates (bounded CP-ABE). Waters [31] then proposed the first fully expressive CP-ABE in the standard model. Herranz et al. [23] proposed the first constant-size ciphertext scheme for CP-ABE allowing threshold gates. Recently, Attrapadung and Libert [1] proposed the first fully expressive KP-ABE with constant-size ciphertexts. All of these works were limited to deal with selective adversaries [11,6] until only two recent works by Lewko et al. [26] and Takashima and Okamoto [28], where they obtained adaptively secure ABE schemes. The aforementioned ABE schemes deal only with single authority, which is the setting that we focus here as well. Some extensions to multi-authority schemes were considered in [13,14,3,25]. It is also worth mentioning that dual-policy ABE, which is a combination of the mentioned two flavors of ABE, was proposed in [2].

Organization of the Paper. In the rest of this paper, we first give syntax and security notion of FE in Sec. 2, give the definition of verifiability and delegatability of FE in Sec. 3, show our general construction in Sec. 4, prove the security of our constructions for the case of CP-ABE in Sec. 5, 6, show instantiations of our generic construction in Sec. 7, show that our definition of verifiability is strictly weaker notion than usual public verifiability in Sec. 8.

2 Definitions

We capture notions of CP-ABE and KP-ABE by providing a unified definition and security notion for functional encryption[1] here and then instantiating to both primitives in the next subsection.

2.1 Syntax and Security Definition for Functional Encryption

Syntax. Let $R : \Sigma_k \times \Sigma_e \rightarrow \{0,1\}$ be a boolean function where Σ_k and Σ_e denote "key attribute" and "ciphertext attribute" spaces. A functional encryption (FE) scheme for R consists of the following algorithms: **Setup**, **KeyGen**, **Encrypt**, **Decrypt**.

Setup$(\lambda, \mathrm{des}) \rightarrow (PK, MSK)$**:** The setup algorithm takes as input a security parameter λ and a scheme description des and outputs a public key PK and a master secret key MSK.

KeyGen$(MSK, PK, X) \rightarrow SK_X$**:** The key generation algorithm takes in the master secret key MSK, the public key PK, and a key attribute $X \in \Sigma_k$. It outputs a private key SK_X.

[1] Our definition of FE is not the fully generalized one, as recently defined in [10]. It can be considered as the class of predicate encryption with public index in [10].

Encrypt$(PK, M, Y) \rightarrow CT$**:** The encryption algorithm takes as input a public key PK, the message M, and a ciphertext attribute $Y \in \Sigma_e$. It will output a ciphertext CT. We assume that Y is implicitly included in CT.

Decrypt$(PK, CT, SK_X) \rightarrow M$ or $\perp$**:** The decryption algorithm takes in the public parameters PK, a ciphertext CT, and a private key SK_X. It outputs the message M or $\perp$ which represents that the ciphertext is not in a valid form.

We require the standard correctness of decryption: that is, for all λ, all (PK, MSK) output by **Setup**(λ, des), all $X \in \Sigma_k$, all SK_X output by **KeyGen**(MSK, PK, X), and $Y \in \Sigma_e$,

- If $R(X, Y) = 1$, then **Decrypt**$(PK,$ **Encrypt**$(PK, M, Y)), SK_X) = M$.
- If $R(X, Y) = 0$, then **Decrypt**$(PK,$ **Encrypt**$(PK, M, Y)), SK_X) = \perp$.

Security Notion. We now give the definition of indistinguishability under chosen ciphertext attack (CCA-security) for FE scheme Π. This is described by a game between a challenger and attacker $\mathcal{A}$. The game proceeds as follows:

Setup. The challenger runs the setup algorithm and gives PK to $\mathcal{A}$.

Phase1. $\mathcal{A}$ may adaptively make queries of the following types:

– ***Key-extraction query.*** $\mathcal{A}$ submits X to the challenger. If the challenger already extracted a private key SK_X for X, then returns it. Otherwise the challenger runs $SK_X \leftarrow$ **KeyGen**(MSK, PK, X) and returns it.

– ***Decryption query.*** $\mathcal{A}$ submits (CT, X) to the challenger and ask for the decryption result of ciphertext CT under private key for X. If the challenger has not previously extracted a private key SK_X for X, then extract it by $SK_X \leftarrow$ **KeyGen**(MSK, PK, X). Then, the challenger returns the output of **Decrypt**(PK, CT, SK_X) to $\mathcal{A}$.

Challenge. $\mathcal{A}$ declares two equal length messages M_0 and M_1 and target ciphertext attribute $Y^* \in \Sigma_e$. Y^* cannot satisfy $R(X, Y^*) = 1$ for any attribute sets X such that $\mathcal{A}$ already queried private key for X. Then the challenger flips a random coin $\beta \in \{0, 1\}$, runs **Encrypt**$(PK, M_\beta, Y^*) \rightarrow CT^*$ and gives challenge ciphertext CT^* to $\mathcal{A}$.

Phase2. $\mathcal{A}$ may adaptively make queries as the same as in **Phase1** with following added restriction. $\mathcal{A}$ cannot query to extract a private key SK_X for X such that $R(X, Y^*) = 1$. $\mathcal{A}$ cannot submit (CT, X) such that $R(X, Y^*) = 1$ and $CT = CT^*$.

Guess. $\mathcal{A}$ outputs a guess β' for β.

We say that $\mathcal{A}$ succeeds if $\beta' = \beta$ and denote the probability of this event by $\Pr^{FE}_{\mathcal{A},\Pi}$. The advantage of an attacker $\mathcal{A}$ is defined as $Adv^{FE}_{\mathcal{A},\Pi} = \Pr^{FE}_{\mathcal{A},\Pi} - \frac{1}{2}$.

Definition 1. *We say that an FE scheme Π is $(\tau, \epsilon, q_D, q_E)$* CCA-secure *if for all τ-time algorithms $\mathcal{A}$ who make a total of q_D decryption queries and a total of q_E key-extraction queries, we have that $Adv^{FE}_{\mathcal{A},\Pi} < \epsilon$. We say that an FE scheme Π is* CCA-secure *if for all polynomial τ, q_D, q_E and for all nonnegligible ϵ, Π is $(\tau, \epsilon, q_D, q_E)$ CCA-secure.*

Definition 2. *We say that an FE scheme Π is (τ, ϵ, q_E)* CPA-secure *if Π is $(\tau, \epsilon, 0, q_E)$* CCA-secure*. We say that an FE scheme Π is CPA-secure if for all polynomial τ, q_E, for all nonnegligible ϵ, Π is (τ, ϵ, q_E) CPA-secure.*

We say that the FE scheme is selectively CCA/CPA-secure if we add an Initial stage **Init** before the setup where the adversary submits the ciphertext attribute $Y^* \in \Sigma_e$.

2.2 Definitions for Attribute-Based Encryption

Definition 3 (Access Structures). *Consider a set of parties $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$. A collection $\mathbb{A} \subseteq 2^{\mathcal{P}}$ is said monotone if for all B, C we have that if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (resp., monotonic access structure) is a collection (resp., monotone collection) $\mathbb{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.*

Definition 4 (KP-ABE). *Let U be an attribute space. A key-policy attribute-based encryption (KP-ABE) for a collection $\mathcal{A}$ of access structures over U is a functional encryption for $R^{\mathsf{KP}} : \mathcal{A} \times 2^U \rightarrow \{0, 1\}$ defined by $R^{\mathsf{KP}}(\mathbb{A}, \omega) \mapsto 1$ iff $\omega \in \mathbb{A}$. Furthermore, the description* des *consists of the attribute universe U, $\Sigma_k = \mathcal{A}$, and $\Sigma_e = 2^U$.*

Definition 5 (CP-ABE). *A ciphertext-policy attribute-based encryption (CP-ABE) is the dual variant of KP-ABE. More precisely, if we let U be the attribute space, a CP-ABE for a collection $\mathcal{A}$ of access structures over U is a functional encryption for $R^{\mathsf{CP}} : 2^U \times \mathcal{A} \rightarrow \{0, 1\}$ defined by $R^{\mathsf{CP}}(\omega, \mathbb{A}) \mapsto 1$ iff $\omega \in \mathbb{A}$. Furthermore, the description* des *consists of the attribute universe U, $\Sigma_k = 2^U$, and $\Sigma_e = \mathcal{A}$.*

Some Terminologies. We define some terminologies and properties related to access structures here. Any monotonic (resp., non-monotonic) access structure $\mathbb{A}$ can be represented by a corresponding boolean formula (resp., with negation), which we denote by $\psi(\mathbb{A})$, over variables in U. This is naturally defined in the sense that $S \in \mathbb{A}$ holds iff the evaluation of $\psi(\mathbb{A})$ with the assignment that sets all variables in S to 1 and other variables outside S to 0 yields the value 1.

Consider the case where $\mathbb{A}$ is a monotonic access structure over U. If we denote a minimal representation of $\mathbb{A}$ by $\mathsf{min}(\mathbb{A}) = \{S \in \mathbb{A} \mid \text{there exists no } B \in \mathbb{A} \text{ such that } B \subset S\}$. Then, it is straightforward to see that $\psi(\mathbb{A}) = \vee_{S' \in \mathsf{min}(\mathbb{A})} (\wedge_{P \in S'} P)$.

Next we consider the case where $\mathbb{A}$ that is a non-monotonic access structure over U. We proceed similarly to Ostrovsky et al. [29]. For each $P \in U$ we define another primed attribute P'. Let $\bar{U} = \{P' \mid P \in U\}$. As in [29], we define a monotonic access structure $\tilde{\mathbb{A}}$ over $U \cup \bar{U}$ in such a way that $S \in \mathbb{A} \Leftrightarrow S \cup \{P' \in \bar{U} \mid P \in U \setminus S\} \in \tilde{\mathbb{A}}$. Then, it is not hard to see that $\psi(\mathbb{A})$ can be written as $\psi(\tilde{\mathbb{A}})$ with each primed variable P' being replaced by the negation of P.

For simplicity, we will use the access structure $\mathbb{A}$ and its corresponding boolean formula $\psi(\mathbb{A})$ interchangeably when specifying a policy.

3 Two Properties: Verifiability and Delegatability

In our constructions, we need FE (CP/KP-ABE) to have either verifiability or delegatability. In this section we define both properties. While the former is a new one defined in this paper, the latter one was already defined in [21,5,7] for the KP-ABE, CP-ABE and general FE cases respectively. We note that the notion of delegation for FE subsumes that of hierarchical IBE [22,6]. We also note that similar notion to the verifiability "committing" is defined in the IBE setting in [19].

Verifiability. Intuitively, we say that an FE scheme has verifiability if it is possible to verify whether a ciphertext will be recovered into the same decryption result under two different decryption keys with two specific attributes.

Definition 6. *An FE scheme* $\Pi = (\mathbf{Setup}, \mathbf{KeyGen}, \mathbf{Encrypt}, \mathbf{Decrypt})$*, is said to have verifiability if there also exists a polynomial time algorithm* **Verify** *that takes as inputs* PK, CT, X, X' *and outputs* 0 *or* 1 *or* $\perp$ *according to the following properties. Let* Y *be obtained from parsing* CT*. We require that first if* $R(X, Y) = 0$ *or* $R(X', Y) = 0$*, then* **Verify** *outputs* $\perp$*.*

Second if $R(X, Y) = R(X', Y) = 1$ *then if we let* $\mathbf{Setup}(\lambda, des) \rightarrow (PK, MSK)$ *then we have the following.*

(Soundness). *For all* CT *in the ciphertext space (which might be invalid),*

$$Pr\left[\begin{array}{r} \mathbf{Decrypt}(PK, CT, SK_X) = \\ \mathbf{Decrypt}(PK, CT, SK_{X'}) \end{array} \middle| \begin{array}{l} \mathbf{Verify}(PK, CT, X, X') = 1, \\ SK_X \leftarrow \mathbf{KeyGen}(PK, MSK, X), \\ SK_{X'} \leftarrow \mathbf{KeyGen}(PK, MSK, X') \end{array}\right] = 1.$$

(Completeness). *For all* M *in the message space,*

$$Pr\left[\mathbf{Verify}(PK, CT, X, X') = 1 \mid CT \leftarrow \mathbf{Encrypt}(PK, M, Y)\right] = 1.$$

Note that our definition of verifiability is weaker notion than usual public verifiability. Namely, in our definition, validity of the ciphertext is not needed to be checked. See Sec. 8 for the gap between the (standard) public verifiability and our verifiability. Moreover, for our conversion, the above definition of verifiability is sufficient but not necessary. See Appendix B for a weaker (but complicated) variant of our verifiability.

Delegatability. Intuitively, delegatability is the capability to use a key for some key attribute X to derive another key for key attribute X' which is possible if X' is inferior than X when considering a well-defined partial order relation $\succeq$ over the key attribute domain Σ_k. More precisely, one can derive $SK_{X'}$ from SK_X if $X \succeq X'$. In the both of CP-ABE and KP-ABE cases, we define the partial order relation as $X \succeq X'$ iff $X \supseteq X'$. The formal definition is as follows.

Definition 7. *An FE scheme* $\Pi = (\mathbf{Setup}, \mathbf{KeyGen}, \mathbf{Encrypt}, \mathbf{Decrypt})$*, is said to have delegatability if there also exists a polynomial time algorithm* **Delegate** *such that the output of* $\mathbf{Delegate}(PK, SK_X, X, X')$ *and of* $\mathbf{KeyGen}(MSK, PK, X')$ *have the same probability distribution for all* $X, X' \in \Sigma_k$ *such that* $X' \preceq X$ *and for all* SK_X *output by* $\mathbf{KeyGen}(MSK, PK, X)$*.*

Indeed, this definition is stronger than needed to apply our conversion for the KP-ABE case. In such a case, it suffices to require only that the output of $\mathbf{Delegate}(PK, SK_{\mathbb{A}}, \psi(\mathbb{A}), \psi(\mathbb{A}) \wedge P)$ and of $\mathbf{KeyGen}(MSK, PK, \psi(\mathbb{A}) \wedge P)$ have the same probability distribution for all access structure $\mathbb{A}$, an attribute P, and all $SK_{\psi(\mathbb{A})}$ output by $\mathbf{KeyGen}(MSK, PK, \psi(\mathbb{A}))$.

4 General Constructions of CCA-Secure ABE

In this section, we show eight conversions to convert a CPA-secure FE scheme Π to a CCA-secure FE scheme Π'. The eight conversions consist of the combinations of whether the original FE scheme Π is CP-ABE or KP-ABE, Π has verifiability or delegatability, and Π deals with a small or large universe. To describe all conversions in a concise way, we write them all in one construction template below. Each conversion then differs in only the definitions of specific variables in Π' namely X' for key attribute, Y' for ciphertext attribute, W for dummy attribute universe, and a procedure called **Subroutine** used in decryption algorithms. We define W below, while the rest are given in Table 1.

Attribute Universes. ABE can be categorized by the size of the attribute universe that such a scheme can deal with: whether it is of polynomial or super-polynomial size. These are called small and large universe scheme respectively. In our conversions, the converted scheme Π' will be able to deal with the same type as that of its original scheme Π. Suppose that we construct a scheme Π' to work with a universe U, we will utilize a set W of *dummy attributes*, which is disjointed from U. The original scheme Π is then required to deal with universe $U \cup W$. A set of dummy attributes will then be associated to a verification key vk of a one-time signature scheme used in the conversion (see Appendix C.2). We assume that for all vk, $vk \in \{0,1\}^{\ell}$. The set W is defined as follows.

- If Π is a small universe scheme, we set $W = \{P_{1,0}, P_{1,1}, P_{2,0}, P_{2,1}, \ldots, P_{\ell,0}, P_{\ell,1}\}$. We set a dummy attribute set $S_{vk} \subset W$ by setting $S_{vk} = \{P_{1,vk_1}, P_{2,vk_2}, \ldots, P_{\ell,vk_\ell}\}$, where we denote by vk_j the j-th bit of vk.
- If Π is a large universe scheme, we set $W = \{0,1\}^{\ell}$. We set a dummy attribute set $S_{vk} \subset W$ by simply letting $S_{vk} = \{vk\}$.

Construction Template. Given a CPA-secure FE scheme $\Pi = (\mathbf{Setup}, \mathbf{KeyGen}, \mathbf{Encrypt}, \mathbf{Decrypt})$ with verifiability *or* delegatability, we construct another FE scheme $\Pi' = (\mathbf{Setup}', \mathbf{KeyGen}', \mathbf{Encrypt}', \mathbf{Decrypt}')$ which is CCA-secure as follows. Let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a one-time signature scheme.

Setup$'(\lambda, U)$**.** It outputs $\mathbf{Setup}(\lambda, U \cup W) \to (PK, MSK)$.

KeyGen$'(MSK, PK, X)$**.** It outputs $\mathbf{KeyGen}(MSK, PK, X') \to SK_{X'}$. Hence $SK'_X = SK_{X'}$.

Encrypt$'(PK, M, Y)$**.** It first creates a one-time signature key pair by running $\mathcal{G}(\lambda) \to (vk, sk)$. It then runs $\mathbf{Encrypt}(PK, M, Y') \to CT$ and $\mathcal{S}(sk, CT) \to \sigma$. It outputs $CT' = (vk, CT, \sigma)$.

Table 1. How to setup X', Y' and **Subroutine** in each case

Conversion CP-ABE1	Conversion KP-ABE1
CPA CP-ABE w/ verifiability ⇒ CCA CP-ABE	CPA KP-ABE w/ verifiability ⇒ CCA KP-ABE
Attribute set $X' = X$ **Policy** $Y' = Y \vee (\wedge_{P \in S_{vk}} P)$ **Subroutine** If **Verify**$(PK, CT, X, S_{vk}) = 0$ or $\perp$ Return $\perp$. Else Return **Decrypt**$(PK, CT, SK_{X'})$.	**Policy** $X' = X$ **Attribute set** $Y' = Y \cup S_{vk}$ **Subroutine** If **Verify**$(PK, CT, X, \wedge_{P \in S_{vk}} P) = 0$ or $\perp$ Return $\perp$. Else Return **Decrypt**$(PK, CT, SK_{X'})$.
Conversion CP-ABE2	Conversion KP-ABE2
CPA CP-ABE w/ delegation ⇒ CCA CP-ABE	CPA KP-ABE w/ delegation ⇒ CCA KP-ABE
Attribute set $X' = X \cup W$ **Policy** $Y' = Y \wedge (\wedge_{P \in S_{vk}} P)$ **Subroutine** Run **Delegate**$(PK, SK'_X, X \cup W, X \cup S_{vk})$ $\rightarrow SK_{X \cup S_{vk}}$. Return **Decrypt**$(PK, CT, SK_{X \cup S_{vk}})$.	**Policy** $X' = X$ **Attribute set** $Y' = Y \cup S_{vk}$ **Subroutine** Run **Delegate**$(PK, SK'_X, X, X \wedge (\wedge_{P \in S_{vk}} P))$ $\rightarrow SK_{X \wedge (\wedge_{P \in S_{vk}} P)}$. Return **Decrypt**$(PK, CT, SK_{X \wedge (\wedge_{P \in S_{vk}} P)})$.

Decrypt$'(PK, CT', SK'_X)$**.** It parses the ciphertext CT' as (vk, CT, σ). If $\mathcal{V}(vk, CT, \sigma) = 0$, then it outputs $\perp$. Otherwise, it runs a subroutine **Subroutine** and outputs its returned value.

Theorem 1. *Let Π be $(\tau, \epsilon_{ABE}, q)$ CPA-secure CP/KP-ABE scheme with verifiability/delegatability, and Σ be a (τ, ϵ_{OTS}) secure one-time signature scheme, then Π' constructed as above is $(\tau - o(\tau), \epsilon_{ABE} + \epsilon_{OTS}, q_D, q_E)$ CCA-secure CP/KP-ABE scheme where $q \geq q_D + q_E$.*

The theorem can be proven from Lemma 1, 2.

Correctness. We prove the correctness of all the conversions as follows.

- In the case of CP-ABE1 and CP-ABE2, assume that the attribute set X satisfies the policy Y (that is $R^{\mathsf{CP}}(X, Y) = 1$). In CP-ABE1, **Verify** outputs 1 since S_{vk} trivially satisfies $\wedge_{P \in S_{vk}} P$ therefore both X and S_{vk} satisfy $Y \vee (\wedge_{P \in S_{vk}} P)$. The correctness then follows from that of the original ABE. In CP-ABE2, since $X \cup W \supseteq X \cup S_{vk}$, **Delegate** outputs secret key for $X \cup S_{vk}$ correctly and it can be easily seen that $X \cup S_{vk}$ satisfies $Y \wedge (\wedge_{P \in S_{vk}} P)$. The correctness follows similarly.
- In the case of KP-ABE1 and KP-ABE2, assume that the attribute set Y satisfies the policy X (that is $R^{\mathsf{KP}}(X, Y) = 1$). In KP-ABE1, **Verify** outputs 1

since $\wedge_{P \in S_{vk}} P$ is trivially satisfied by S_{vk} therefore both X and $\wedge_{P \in S_{vk}} P$ is satisfied by $Y \cup S_{vk}$. The correctness then follows from that of the original ABE. In KP-ABE2, since $X \succeq X \wedge (\wedge_{P \in S_{vk}} P)$, **Delegate** outputs secret key for $X \wedge (\wedge_{P \in S_{vk}} P)$ correctly and it can be easily seen that $X \wedge (\wedge_{P \in S_{vk}} P)$ is satisfied by $Y \cup S_{vk}$. The correctness follows similarly.

Remark. We propose another two variants which are conversions for CP-ABE and KP-ABE based on verifiability in Appendix A. We also note that the conversion KP-ABE2 for the large universe case is exactly the one in [21]. We include it here to cover the big picture of the whole framework.

Efficiency Consideration. We first consider the expansion of attribute sets. This only occurs in CP-ABE2, where we define a key for set $X' = X \cup W$. A problem may occur for the large universe case, since W is of super-polynomial size the key size may also expand enormously depending on the underlying ABE. If such a problem occurs, we use W as defined in the small universe case.

Next we consider the expansion of policies. In all of available constructions of ABE in the literature, an access structure is represented by either of two methods namely an access tree ([21,5]) or a linear-secret sharing scheme (LSSS) matrix ([21,29,31,26,28]). The efficiency, in particular, key sizes and ciphertext sizes, of these respective ABE schemes tend to depend on the size of access trees or LSSS matrices used in such schemes. (See the definition of LSSS in Appendix C.1). Our conversions particularly use policies of the form $\psi(\mathbb{A}) \vee (\wedge_{P \in S_{vk}} P)$ and $\psi(\mathbb{A}) \wedge (\wedge_{P \in S_{vk}} P)$. Therefore, we have to check whether $\psi(\mathbb{A})$ when augmented to each of both forms still can be represented efficiently or not. To this end, the efficiency is guaranteed from the following two observations.

Proposition 1. *Let access structures $\mathbb{A}$ and $\mathbb{B}$ be expressed by access trees using the method in [21] with h_a, h_b nodes and ℓ_a, ℓ_b leaves respectively. Then access structure corresponding to $\psi(\mathbb{A}) \wedge \psi(\mathbb{B})$ and $\psi(\mathbb{A}) \vee \psi(\mathbb{B})$ can be expressed by access trees both with $h_a + h_b + 1$ nodes and $\ell_a + \ell_b$ leaves.*

Proposition 2. *Let access structures $\mathbb{A}$ and $\mathbb{B}$ be expressed by an $\ell_a \times m_a$ and an $\ell_b \times m_b$ LSSS matrix by using the LSSS of [4] respectively. Then the LSSS matrix corresponding to $\psi(\mathbb{A}) \wedge \psi(\mathbb{B})$ and $\psi(\mathbb{A}) \vee \psi(\mathbb{B})$ can be expressed by an $(\ell_a + \ell_b) \times (m_a + m_b)$ and an $(\ell_a + \ell_b - 1) \times (m_a + m_b)$ LSSS matrix respectively.*

To conclude, since $|S_{vk}| = \ell = poly(\lambda)$ in the large-universe construction and $|S_{vk}| = 1$ in the small-universe construction, our conversions can be efficiently implementable.

Selective security. We remark that our conversion can be also applied to selectively (CPA-)secure ABE schemes, and in such cases, resulting CCA-secure schemes are only selectively (CCA-)secure as well.

5 Security of Our Constructions from Verifiability

Security of our constructions from verifiability, i.e. CP-ABE1 and KP-ABE1 is addressed as follows:

Lemma 1. *Let Π be $(\tau, \epsilon_{ABE}, q)$ CPA-secure CP/KP-ABE scheme with verifiability and Σ be a (τ, ϵ_{OTS}) secure one-time signature scheme, then Π' constructed as in Sec. 4 (CP/KP-ABE1) is $(\tau - o(\tau), \epsilon_{ABE} + \epsilon_{OTS}, q_D, q_E)$ CCA-secure CP/KP-ABE scheme where $q \geq q_D + q_E$.*

In the rest of this section, we prove Lemma 1 for the case of CP-ABE. The lemma can also be proven similar way in the case of KP-ABE.

Proof of Lemma 1 for the case of CP-ABE. Assume we are given an adversary $\mathcal{A}$ which breaks CCA-security of the scheme Π' (CP-ABE1) with running time τ, advantage ϵ, q extraction queries, and, q_D decryption queries. We use $\mathcal{A}$ to construct another adversary $\mathcal{B}$ which breaks CPA-security of the scheme Π. Define adversary $\mathcal{B}$ as follows:

Setup. The challenger runs $\mathbf{Setup}(\lambda, U \cup W) \rightarrow (PK, MSK)$. Then $\mathcal{B}$ is given PK and gives it to $\mathcal{A}$. $\mathcal{B}$ also runs $\mathcal{G}(\lambda) \rightarrow (vk^*, sk^*)$.

Phase1. $\mathcal{A}$ may adaptively make queries of the following types:

– ***Key-extraction query.*** When $\mathcal{A}$ submits S, then $\mathcal{B}$ submits same S to challenger. $\mathcal{B}$ is given private key SK_S for S and gives it to $\mathcal{A}$.

– ***Decryption query.*** When $\mathcal{A}$ submits (CT', S) such that $CT' = (vk, CT, \sigma)$, $\mathcal{B}$ respond to $\mathcal{A}$ as follows. First, $\mathcal{B}$ checks whether $\mathcal{V}(vk, CT, \sigma) = 1$ holds. If it does not hold, then $\mathcal{B}$ returns $\perp$. If it holds and $vk^* = vk$, then $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ checks whether $\mathbf{Verify}(PK, CT, S_{vk}, S) = 1$. If it does not hold, then $\mathcal{B}$ returns $\perp$. Otherwise $\mathcal{B}$ submits S_{vk} to the challenger and is given $SK_{S_{vk}}$. Then $\mathcal{B}$ returns output of $\mathbf{Decrypt}(PK, CT, SK_{S_{vk}})$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ declares two equal length messages M_0 and M_1 and an access structure $\mathbb{A}^*$. Then $\mathcal{B}$ declares the same messages M_0, M_1 and $\mathbb{A}^{*\prime}$ for the challenger, where $\mathbb{A}^{*\prime}$ is an access structure such that $\psi(\mathbb{A}^{*\prime}) = \psi(\mathbb{A}^*) \vee (\wedge_{P \in S_{vk^*}} P)$. The challenger flips a random coin $\beta \in \{0, 1\}$, runs $\mathbf{Encrypt}(PK, M_\beta, \psi(\mathbb{A}^{*\prime})) \rightarrow CT^*$ and gives CT^* to $\mathcal{B}$. Then $\mathcal{B}$ runs $\mathcal{S}(sk^*, CT^*) \rightarrow \sigma^*$, and gives $CT^{*\prime} = (vk^*, CT^*, \sigma^*)$ to $\mathcal{A}$ as challenge ciphertext.

Phase2. $\mathcal{B}$ responds to $\mathcal{A}$'s query as the same as in **Phase1**.

Guess. Finally, $\mathcal{A}$ outputs a guess β' for β. Then $\mathcal{B}$ outputs β' as its guess.

Let **Win** denote the event that $\mathcal{A}$ correctly guess β, **Abort** denote the event that $\mathcal{B}$ aborts. If **Abort** does not occur, from the verifiability of the scheme, $\mathcal{B}$'s simulation is perfect. So, $\mathcal{B}$'s advantage for guessing β is estimated as $Pr[\mathcal{B}$ correctly guesses $\beta] - \frac{1}{2} = Pr[\mathbf{Win}|\overline{\mathbf{Abort}}]Pr[\overline{\mathbf{Abort}}] - \frac{1}{2} \geq Pr[\mathbf{Win}] - Pr[\mathbf{Abort}] - \frac{1}{2} \geq \epsilon - Pr[\mathbf{Abort}]$. Since $Pr[\mathbf{Abort}] \leq \epsilon_{OTS}$ holds due to unforgeability of the one-time-signature, the proof is completed. □

6 Security of Our Construction from Delegatability

Security of our constructions from delegatability, i.e. CP-ABE2 and KP-ABE2 is addressed as follows. In this section, we prove Lemma 2 for the case of CP-ABE. The lemma can also be proven by similar way in the case of KP-ABE.

Lemma 2. *Let Π be a $(\tau, \epsilon_{ABE}, q)$ CPA-secure CP/KP-ABE scheme with delegatability and Σ be (τ, ϵ_{OTS}) secure one-time signature, then Π' constructed as in section4 (CP/KP-ABE2) is $(\tau - o(\tau), \epsilon_{ABE} + \epsilon_{OTS}, q_D, q_E)$ CCA-secure CP/KP-ABE scheme where $q \geq q_D + q_E$.*

Proof of Lemma 2 for the case of CP-ABE. Assume we are given an adversary $\mathcal{A}$ which breaks CCA-security of the scheme Π' (CP-ABE2) with running time τ, advantage ϵ, q_E key-extraction queries, and, q_D decryption queries. We use $\mathcal{A}$ to construct another adversary $\mathcal{B}$ which breaks CPA-security of the scheme Π. Define adversary $\mathcal{B}$ as follows:

Setup. The challenger runs $\mathbf{Setup}(\lambda, U \cup W) \rightarrow (PK, MSK)$. Then $\mathcal{B}$ is given PK and gives it to $\mathcal{A}$. $\mathcal{B}$ also runs $\mathcal{G}(\lambda) \rightarrow (vk^*, sk^*)$.

Phase1. $\mathcal{A}$ may adaptively make queries of the following types:

– ***Key-extraction query.*** When $\mathcal{A}$ submits S, then $\mathcal{B}$ submits $S \cup W$ to the challenger. $\mathcal{B}$ is given private key $SK_{S \cup W}$ for $S \cup W$ and gives it to $\mathcal{A}$.

– ***Decryption query.*** When $\mathcal{A}$ submits (CT', S) such that $CT' = (vk, CT, \sigma)$, $\mathcal{B}$ respond to $\mathcal{A}$ as follows. First, $\mathcal{B}$ checks whether $\mathcal{V}(vk, CT, \sigma) = 1$ holds. If it does not hold, then $\mathcal{B}$ returns $\perp$. If it holds and $vk^* = vk$, then $\mathcal{B}$ aborts. Otherwise $\mathcal{B}$ submits $S \cup S_{vk}$ to the challenger and is given $SK_{S \cup S_{vk}}$. Then $\mathcal{B}$ rerandomize it by $SK_{S \cup S_{vk}} \leftarrow \mathbf{Delegate}(PK, SK_{S \cup S_{vk}}, S \cup S_{vk}, S \cup S_{vk})$ and returns output of $\mathbf{Decrypt}(PK, CT, SK_{S \cup S_{vk}})$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ declares two equal length messages M_0, M_1 and $\mathbb{A}^*$. Then $\mathcal{B}$ declares the same messages M_0, M_1, and $\mathbb{A}^{*\prime}$ for the challenger, where $\mathbb{A}^{*\prime}$ is an access structure such that $\psi(\mathbb{A}^{*\prime}) = \psi(\mathbb{A}^*) \wedge (\wedge_{P \in S_{vk^*}} P)$. The challenger flips a random coin $\beta \in \{0, 1\}$, runs $\mathbf{Encrypt}(PK, M_\beta, \psi(\mathbb{A}^{*\prime})) \rightarrow CT^*$ and gives CT^* to $\mathcal{B}$. Then $\mathcal{B}$ runs $\mathcal{S}(sk^*, CT^*) \rightarrow \sigma^*$ and gives $CT^{*\prime} = (vk^*, CT^*, \sigma^*)$ to $\mathcal{A}$ as challenge ciphertext.

Phase2. $\mathcal{B}$ responds to $\mathcal{A}$'s query as the same as in **Phase1**.

Guess. Finally, $\mathcal{A}$ outputs a guess β' for β. Then $\mathcal{B}$ outputs β' as its guess.

Similar analysis to the previous section shows that $Pr[\mathcal{B} \text{ correctly guess } \beta] - \frac{1}{2} \geq \epsilon - \epsilon_{OTS}$. □

7 Applications to Existing Schemes

7.1 The Case of ABE by Lewko et al.

In this section, we show some applications of our conversions to the recent CPA-secure CP-ABE by Lewko et al. [26] to achieve CCA-secure schemes. We observe first that neither delegation was presented nor verifiability is available in their ABE. However, we show here that only a slight modification will allow both properties. For self-containment, we briefly describe their scheme here.

Description for CP-ABE of [26]. The scheme works in a bilinear group of composite order $N = p_1p_2p_3$. Denote $\mathbb{G}_{p_j}$ the subgroup of order p_j of $\mathbb{G}$. The master key is $MSK = (\alpha \in \mathbb{Z}_N, X_3 \in \mathbb{G}_{p_3})$, while the public key is of the form

$PK = N, g, g^a, e(g,g)^\alpha, \{T_i = g^{s_i}\}_{i \in U}$ where $g \in \mathbb{G}_{p_1}, a, s_i \in \mathbb{Z}_N$. Note that the scheme works with a small universe U. A secret key for set $S \subset U$ is of the form $SK_S = (S,\ K = g^\alpha g^{at} R_0,\ L = g^t R'_0,\ \{K_i = T_i^t R_i\}_{i \in S})$ for random $R_0, R'_0, R_i \in \mathbb{G}_{p_3}, t \in \mathbb{Z}_N$. Denote $B(\mathbb{A}) = \cup_{S \in \mathbb{A}} S$. A ciphertext for policy $\mathbb{A}$ is of the form $CT = (C = Me(g,g)^{s\alpha},\ C' = g^s,\ \{C_x = g^{aA_x \cdot v} T_{\rho(x)}^{-r_x}, D_x = g^{r_x}\}_{x \in B(\mathbb{A})})$ for some random $s, r_x \in \mathbb{Z}_N$ and where $A_x \cdot v$ is the random share for x of the secret s in the LSSS scheme representing the policy $\mathbb{A}$. Decryption can be done if $S \in \mathbb{A}$ by recovering $e(C', K)/\prod_{\rho(x) \in S}(e(C_x, L)e(D_x, K_{\rho(x)}))^{\omega_x} = e(g,g)^{\alpha s}$ where $\{\omega_x\}$ is the reconstruction coefficient of the LSSS scheme.

Slight Modification. The above scheme seems not to have neither delegatability nor verifiability. This is mainly due to the fact that one cannot check whether a ciphertext consists of only elements in $\mathbb{G}_{p_1 p_2}$ or not. Thus, for achieving delegatability and verifiability, we modify the above ABE scheme by simply including also the generator $X_3 \in \mathbb{G}_{p_3}$ in PK. We argue that this modified scheme is still CPA-secure. This can be easily seen since all the three underlying hard problems in [26] that the scheme is based on contains a generator of $\mathbb{G}_{p_3}$ as an input. In the following, we show verifiability and delegatability of the resulting scheme.

Delegatability. We define **Delegate** of the modified scheme as follows.
Delegate$(PK, SK_S, S'(\subseteq S))$ It chooses random $u \in \mathbb{Z}_N$ and random elements $R_0, R'_0, R_i \in \mathbb{G}_{p_3}$. It computes the key for S' as $SK_{S'} = (S',\ K' = Kg^{au}R_0,$ $L' = Lg^u R'_0,\ \{K'_i = K_i T_i^u R_i\}_{i \in S'})$.
It is straightforward to see that the output of **Delegate**$(PK, SK_S, S'(\subseteq S))$ and that of **KeyGen**(MSK, PK, S') have the same probability distribution.

Verifiability. We define **Verify** of the modified scheme as follows.
Verify(PK, CT, S, S') It parses $PK = (N, g, g^a, e(g,g)^\alpha, \{T_i\}_{i \in \mathcal{U}}, X_3)$ and $CT = (\mathbb{A}, C, C', \{C_x, D_x\}_{x \in B(\mathbb{A})})$ then outputs V as

$$V = \begin{cases} \perp \text{ if } S \notin \mathbb{A} \text{ or } S' \notin \mathbb{A}. \\ 1 \text{ if } \prod_{\rho(x) \in S} (e(C_x, g)e(D_x, T_{\rho(x)}))^{w_{x,S}} \\ \qquad = \prod_{\rho(x) \in S'} (e(C_x, g)e(D_x, T_{\rho(x)}))^{w_{x,S'}} = e(g^a, C'), \qquad (1) \\ \quad \text{and } \ e(C', X_3) = 1,\ e(C_x, X_3) = e(D_x, X_3) = 1 \text{ for all } x \in B(\mathbb{A}). \quad (2) \\ 0 \text{ otherwise.} \end{cases}$$

Here $\omega_{x,S}$ and $\omega_{x,S'}$ are reconstruction coefficients in the LSSS. Hence we have $\sum_{\rho(x) \in S} \omega_{x,S} A_x = \sum_{\rho(x) \in S'} \omega_{x,S'} A_x = (1, 0, \ldots, 0)$. We now prove that **Verify** algorithm defined as above satisfies soundness and completeness properties.

Proving Soundness. Consider $S, S' \in \mathbb{A}$. Assume that **Verify**$(PK, CT, S, S') =$ 1 and that $SK_S, SK_{S'}$ are correctly generated. We will prove that **Decrypt** (PK, CT, SK_S) = **Decrypt**$(PK, CT, SK_{S'})$ holds. To this end, we parse

$SK = (S, K, L, \{K_i\}_{i \in S})$, and see that **Decrypt**(PK, CT, SK_S) outputs the following.

$$\begin{aligned}
&\Big(C \cdot \prod\nolimits_{\rho(x)\in S}(e(C_x, L)e(D_x, K_{\rho_x}))^{\omega_x}\Big)/e(C', K) && \text{by def}\\
&= \frac{\Big(C \cdot \prod_{\rho(x)\in S}(e(C_x, g^t \cdot R'_0)e(D_x, T^t_{\rho(x)}R_{\rho(x)}))^{\omega_x}\Big)}{e(C', g^\alpha \cdot g^{at} \cdot R_0)} && \text{by def of } SK_S\\
&= \Big(C \cdot \prod\nolimits_{\rho(x)\in S}(e(C_x, g)e(D_x, T_{\rho(x)}))^{t\omega_x}\Big)/e(C', g^\alpha \cdot g^{at}) && \text{by (2)}\\
&= C \cdot e(g, C')^{at}/(e(C', g)^\alpha e(g, C')^{at}) = C/e(g, C')^\alpha. && \text{by (1)}
\end{aligned}$$

Now since S is arbitrary, the same result holds for S', which concludes the proof.

Proving Completeness. Assume that a ciphertext CT is correctly generated. We will prove that **Verify**$(PK, CT, S, S') = 1$. A correctly generated ciphertext is the form of $CT = (C = Me(g, g)^{s\alpha},\ C' = g^s,\ \{C_x = g^{aA_x \cdot v}T^{-r_x}_{\rho(x)}, D_x = g^{r_x}\}_{x \in B(\mathbb{A})})$. Since all elements are in $\mathbb{G}_{p_1p_2}$, (2) holds. Equation (1) also holds by straightforward calculation.

Resulting CCA-secure Schemes. We now compare the two CCA-secure CP-ABE constructions converted from the above (slightly modified) CP-ABE of [26] by using the CP-ABE1 (required verifiability) and CP-ABE2 (required delegatability). As for the public key length, ciphertext length, and encryption cost, it seems that former is as efficient as latter. (Ciphertext length and encryption cost depend on the underlying LSSS matrix.) Secret key length of former is shorter than that of latter. As for the decryption cost, latter is more efficient than former since the **Verify** algorithm contains many pairing computation as opposed to the **Delegate** algorithm.

Table 2. ABE with delegatability or verifiability. In the table, "Deleg." and "Verif." denote delegatability and verifiability respectively.

Schemes		KP/CP	Universe	Deleg.	Verif.	Security	Assumption
Goyal et al.	[21, Sect.4]	KP	small	U	✓	selective	DBDH
Goyal et al.	[21, Sect.5]	KP	large	✓	✓	selective	DBDH
Goyal et al.	[21, Sect.A]	KP	small	U	✓	selective	DBDH
Ostrovsky et al.	[29, Sect.3]	KP	large	U	✓	selective	DBDH
Bethencourt et al.	[5]	CP	large	✓	✓	selective	Generic group
Goyal et al.	[20]	CP	small	U	✓	selective	DBDH
Waters	[31, Sect.3]	CP	small	✓	✓	selective	DPBDHE
Lewko et al.	[24, Sect.6]	KP	large	U	✓	selective	q-MEBDH
Attrapadung et al.	[1]	KP	large	U	✓	selective	DBDHE
Lewko et al.	[26, Sect.2]	CP	small	U	U	full	3 assumptions
Section 7.1 (modified from [26])		CP	small	✓	✓	full	3 assumptions
Lewko et al.	[26, Sect.A]	KP	small	U	U	full	3 assumptions
Okamoto et al.	[28]	KP	large	U	U	full	DLIN
Slightly modified	[28]	KP	large	✓	✓	full	DLIN

We remark that KP-ABE scheme in [26] also could be modified to have verifiability by similar technique to the case of CP-ABE.

7.2 Summary for Applications to Existing Schemes

In Table 2, we give an overview of existing ABE schemes and their properties, and from this table, one can see that many of these schemes satisfy verifiability and/or delegatability. We remark that similarly to [26], Okamoto and Takashima's scheme [28] can be also modified to have both delegatability and verifiability. See the full version of our paper for details. In the table, ✓ denotes there is verify or delegate algorithm that satisfies our definition, "U" denotes there is unknown such algorithm.

8 Remark on Verifiability

Our definition of verifiability is considered weaker than that of the standard public verifiability where roughly speaking, we say that an encryption scheme satisfies *public verifiability* if any third party (who does not have any secret) can always verify whether a given ciphertext is one of possible outputs of the encryption algorithm or not. To see this, we show an FE scheme which has verifiability, but does not have public verifiability. We construct such FE scheme $\Pi' = (\textbf{Setup}', \textbf{KeyGen}', \textbf{Encrypt}', \textbf{Decrypt}', \textbf{Verify}')$ from FE scheme $\Pi = (\textbf{Setup}, \textbf{KeyGen}, \textbf{Encrypt}, \textbf{Decrypt}, \textbf{Verify})$ with verifiability, an one-way function $f : \{0,1\}^n \rightarrow \{0,1\}^{n'}$, and a hardcore function $h : \{0,1\}^n \rightarrow \{0,1\}$ for f. Here, n and n' are polynomials of λ. $\textbf{Setup}'$ and $\textbf{KeyGen}'$ are the same as $\textbf{Setup}$ and $\textbf{KeyGen}$ respectively. $\textbf{Encrypt}'$ is slightly different from $\textbf{Encrypt}$. $\textbf{Encrypt}'(PK, M, Y)$ first runs $\textbf{Encrypt}(PK, M, Y) \rightarrow CT$ and picks a random $x \leftarrow \{0,1\}^n$ independently. Then it compute $(f(x), h(x)) \in \{0,1\}^{n'+1}$ and returns final ciphertext $CT' = (CT, f(x), h(x))$. $\textbf{Decrypt}'(PK, CT', SK_X)$ first parses CT' as (CT, y, b) and returns $\textbf{Decrypt}(PK, CT, SK_X)$ where $y \in \{0,1\}^{n'}$, $b \in \{0,1\}$. $\textbf{Verify}'(PK, CT', X, X')$ first parses CT' as (CT, y, b) as the same as above, then returns $\textbf{Verify}(PK, CT, X, X')$.

It is clear that Π' has verifiability since **Verify** algorithm defined as above works correctly. However, Π' does not have public verifiability since for verifying validity of a ciphertext in the sense of public verifiability, one has to correctly guess the hardcore bit $h(x)$ from only $f(x)$.

References

1. Attrapadung, N., Libert, B.: Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In: Catalano, D., et al. (eds.) PKC 2011. LNCS, vol. 6571. Springer, Heidelberg (2011)
2. Attrapadung, N., Imai, H.: Dual-Policy Attribute Based Encryption. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 168–185. Springer, Heidelberg (2009)

3. Attrapadung, N., Imai, H.: Conjunctive Broadcast and Attribute-Based Encryption. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 248–265. Springer, Heidelberg (2009)
4. Benaloh, J.C., Leichter, J.: Generalized Secret Sharing and Monotone Functions. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 27–35. Springer, Heidelberg (1990)
5. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy (S&P), pp. 321–334 (2007)
6. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
7. Boneh, D., Hamburg, M.: Generalized Identity Based and Broadcast Encryption Schemes. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)
8. Boneh, D., Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
9. Boyen, X., Mei, Q., Waters, B.: Direct Chosen Ciphertext Security from Identity-Based Techniques. In: ACM CCS 2005, pp. 320–329 (2005)
10. Boneh, D., Sahai, A., Waters, B.: Functional Encryption: Definitions and Challenges. In: TCC 2011. LNCS, vol. 6597. Springer, Heidelberg (to appear)
11. Canetti, R., Halevi, S., Katz, J.: A Forward-Secure Public-Key Encryption Scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 254–271. Springer, Heidelberg (2003)
12. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
13. Chase, M.: Multi-authority Attribute Based Encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)
14. Chase, M., Chow, S.: Improving privacy and security in multi-authority attribute-based encryption. In: ACM CCS 2009, pp. 121–130 (2009)
15. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: ACM CCS 2007, pp. 456–465 (2007)
16. Damgård, I., Thorbek, R.: Linear integer secret sharing and distributed exponentiation. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 75–90. Springer, Heidelberg (2006)
17. Dolev, D., Dwork, C., Naor, M.: Non-Malleable Cryptography (Extended Abstract). In: STOC 1991, pp. 542–552 (1991)
18. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
19. Green, M., Hohenberger, S.: Blind Identity-Based Encryption and Simulatable Oblivious Transfer. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 265–282. Springer, Heidelberg (2007)
20. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded Ciphertext Policy Attribute Based Encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
21. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM CCS 2006, pp. 89–98 (2006)

22. Gentry, C., Silverberg, A.: Hierarchical ID-Based Cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
23. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant Size Ciphertexts in Threshold Attribute-Based Encryption. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 19–34. Springer, Heidelberg (2010)
24. Lewko, A., Sahai, A., Waters, B.: Revocation Systems with Very Small Private Keys. In: IEEE Symposium on Security and Privacy (S&P), pp. 273–285 (2010)
25. Lewko, A., Waters, B.: Decentralizing Attribute-Based Encryption. Cryptology ePrint Archive: Report 2010/351 (2010)
26. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
27. Naor, M., Yung, M.: Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In: STOC 1990, pp. 427–437 (1990)
28. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
29. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM CCS 2007, pp. 195–203 (2007)
30. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.). EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
31. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In: Catalano, D., et al. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heibelberg (2011)

A Variants of CP-ABE2 and KP-ABE2

In Table 3, we show CP-ABE3 and KP-ABE3 which are variants of CP-ABE2 and KP-ABE2, respectively. These schemes are very similar, differences are that CP-ABE3 and KP-ABE3 are constructed from verifiability whereas CP-ABE2 and KP-ABE2 are constructed from delegatability.

Table 3. How to setup X', Y' and **Subroutine** in CP/KP-ABE3

Conversion CP-ABE3	Conversion KP-ABE3
CPA CP-ABE w/ verifiability ⇒ CCA CP-ABE	CPA KP-ABE w/ verifiability ⇒ CCA KP-ABE
Attribute set $X' = X \cup W$ **Policy** $Y' = Y \wedge (\wedge_{P \in S_{vk}} P)$	**Policy** $X' = X$ **Attribute set** $Y' = Y \cup S_{vk}$
Subroutine If **Verify**$(PK, CT, X \cup W, X \cup S_{vk}) = 0$ or $\perp$ Return $\perp$. Else Return **Decrypt**$(PK, CT, SK_{X'})$.	**Subroutine** If **Verify**$(PK, CT, X, X \wedge_{P \in S_{vk}} P) = 0$ or $\perp$ Return $\perp$. Else Return **Decrypt**$(PK, CT, SK_{X'})$.

B Weaker Verifiability

Our conversion still works even if the underlying FE scheme does not satisfy our verifiability but a weaker notion than it. This weaker variant of our verifiability is defined as follows.

Definition 8. *A FE scheme Π is said to have weaker verifiability if there exists a polynomial time algorithm* **Verify** *that takes as We require that if $R(X, Y) = 0$ or $R(X', Y) = 0$, then* **Verify** *outputs $\perp$. Here, Y is obtained from parsing CT.*

1. *If SK'_X is output of $\mathbf{KeyGen}(PK, MSK, X)$, then $Pr[\mathbf{Decrypt}$ $(PK, CT, SK_X) = \mathbf{Decrypt}(PK, CT, SK_{X'})|$*
   $SK_X \leftarrow \mathbf{KeyGen}(PK, MSK, X),\ SK_{X'}\mathbf{KeyGen}(PK, MSK, X'),$
   $\mathbf{Verify}(PK, CT, X, X', SK'_X) = 1] = 1$ always holds.
2. *If $R(X, Y) = R(X', Y) = 1$, then it holds that for all correctly generated PK and for all CT (which might be invalid),*
   $Pr[\mathbf{Verify}(PK, CT, X, X', SK_X) = \mathbf{Verify}(PK, CT, X', X, SK_{X'})|$
   $SK_X \leftarrow \mathbf{KeyGen}(PK, MSK, X),\ SK_{X'} \leftarrow \mathbf{KeyGen}(PK, MSK, X')] = 1.$
3. *If SK_X is output of $\mathbf{KeyGen}(PK, MSK, X)$ and $R(X, Y) = R(X', Y) = 1$, then It holds that $Pr[\mathbf{Verify}(PK, CT, X, X', SK_X) = 1|$*
   $SK_X \leftarrow \mathbf{KeyGen}(PK, MSK, X),\ CT \leftarrow \mathbf{Encrypt}(PK, M, Y)] = 1$

C Some Omitted Definitions and Descriptions

C.1 Linear Secret Sharing Schemes

Definition 9 (Linear Secret Sharing Scheme). *Let $\mathcal{P}$ be a set of parties. Let M be a $\ell \times k$ matrix. Let $\pi : \{1, \ldots, \ell\} \rightarrow \mathcal{P}$ be a function that maps a row to a party for labeling. A secret sharing scheme Π for access structure $\mathbb{A}$ over a set of parties $\mathcal{P}$ is a linear secret-sharing scheme (LSSS) in $\mathbb{Z}_p$ and is represented by (M, π) if it consists of two efficient algorithms:*

Share$_{(M,\pi)}$: *The algorithm takes as input $s \in \mathbb{Z}_p$ which is to be shared. It chooses $a_2, \ldots, a_k \in \mathbb{Z}_p$ and let $\boldsymbol{a} = (s, a_2, \ldots, a_k)^\top$. It outputs $M \cdot \boldsymbol{a}$ as the vector of ℓ shares. The share $\lambda_i := \langle \boldsymbol{M_i}, \boldsymbol{a} \rangle$ belongs to party $\pi(i)$, where $\boldsymbol{M_i}^\top$ denotes the i^{th} row of M.*

Recon$_{(M,\pi)}$: *The algorithm takes as input an access set $S \in \mathbb{A}$. Let $I = \{i | \pi(i) \in S\}$. It outputs a set of constants $\{(i, \mu_i)\}_{i \in I}$ which has a linear reconstruction property: $\sum_{i \in I} \mu_i \cdot \lambda_i = s$.*

C.2 One-Time-Signature

A one-time-signature scheme consists of the following three algorithms, $\mathcal{G}$, $\mathcal{S}$, and $\mathcal{V}$. The key generation algorithm $\mathcal{G}(\lambda)$ takes as input the security parameter λ, and outputs a verification key vk and a signing key sk. The sign algorithm

$\mathcal{S}(sk, m)$ takes as input sk and a message m, and outputs a signature σ. The verify algorithm $\mathcal{V}(vk, m, \sigma)$ takes as input vk, m, and σ, and outputs a bit $b \in \{0, 1\}$. We require that for all honestly generated sk, all m in the message space, and all σ, output by $\mathcal{S}(sk, m)$, we have $\mathcal{V}(vk, m, \sigma) = 1$. Next, we define strong unforgery of a (one-time) signature scheme Σ against chosen message attacks. Security is defined using the following game between an attacker $\mathcal{A}$ and a challenger. Both the challenger and attacker are given λ as input. First, the challenger runs $\mathcal{G}(\lambda)$ to obtain vk and sk. It gives $\mathcal{A}$ vk. Next, $\mathcal{A}$ may issue at most one signing query m^*. The challenger responds with $\sigma^* = \mathcal{S}(sk, m^*)$. Finally, $\mathcal{A}$ outputs (m, σ). We say that $\mathcal{A}$ succeeds to forge if $\mathcal{A}$ outputs (m, σ) such that $(m, \sigma) \neq (m^*, \sigma^*)$ and $\mathcal{V}(vk, m, \sigma) = 1$, and denote the probability of this event by $Adv^{OTS}_{\mathcal{A},\Sigma}$.

Definition 10. *We say that a one-time-signature scheme Σ is (τ, ϵ)*-secure *if for all τ-time algorithms $\mathcal{A}$ we have that $Adv^{OTS}_{\mathcal{A},\Sigma} < \epsilon$.*