Short CCA-Secure Ciphertext-Policy Attribute-Based Encryption

Hiroaki Anada
Department of Information Security
University of Nagasaki
Nagasaki, Japan
Email: anada@sun.ac.jp

Seiko Arita
Graduate School of Information Security
Institute of Information Security
Yokohama, Japan
Email: arita@iisec.ac.jp

***Abstract*—We propose a technique of individually modifying an attribute-based encryption scheme (ABE) secure against chosen-plaintext attacks (CPA) into an ABE scheme secure against chosen-ciphertext attacks (CCA) in the standard model. We demonstrate the technique in the case of the Waters ciphertext-policy ABE (CP-ABE). Our technique is helpful when a Diffie-Hellman tuple to be verified is in the terminal group of a bilinear map. We utilize the Twin Diffie-Hellman Trapdoor Test of Cash, Kiltz and Shoup, and it results in expansion of secret key length and decryption cost of computation by a factor of four, whereas public key length, ciphertext length and encryption cost of computation remain almost the same. In the case that the size of attribute sets are small, those lengths and costs are smaller than those of the CP-ABE obtained via the generic transformation of Yamada et al. in PKC 2011.**

I. Introduction

Attribute-based encryption (ABE) was first proposed by Sahai and Waters [10] to realize fine-grained access control by encryption, where attributes mean authorized credentials. In ciphertext-policy ABE (CP-ABE) introduced by the subsequent work of Goyal, Pandey, Sahai and Waters [7], ciphertexts are associated with access policies over attributes, while secret keys are associated with sets of attributes. A secret key works to decrypt a ciphertext if and only if the associated set of attributes satisfies the associated access policy. Since the proposal, it has been studied to attain certain properties such as indistinguishability against chosen-plaintext attacks (IND-CPA) in the standard model [11] and adaptive security against adversary's choice of a target access structure [9].

In this paper, we work through a problem of constructing a shorter ABE scheme that attains indistinguishability against chosen-ciphertext attacks (IND-CCA) in the standard model. Let us recall the case of identity-based encryption (IBE). The CHK transformation [4] is a generic tool for obtaining IND-CCA secure IBE scheme. It transforms any hierarchical IBE (HIBE) scheme that is selective-ID IND-CPA secure into an IBE scheme that is adaptive-ID IND-CCA secure. In contrast, direct chosen-ciphertext security technique for IBE of Boyen, Mei and Waters [3] is an individual technique for obtaining an IND-CCA secure IBE scheme. It converts a HIBE scheme that is adaptive-ID IND-CPA secure into an IBE scheme that is adaptive-ID IND-CCA secure. Though the technique needs to treat each scheme individually, the obtained scheme attains better performance than that obtained by the generic tool (the CHK transformation). Let us transfer into the case of ABE. The transformation of Yamada et al. [12] is a generic tool for obtaining IND-CCA secure ABE scheme. It transforms any ABE scheme that is IND-CPA secure into an ABE scheme that is IND-CCA secure. Notice here that developing direct chosen-ciphertext security technique for ABE (in the standard model) is a missing piece. One of the reason seems that there is an obstacle that a Diffie Hellman tuple to be verified is in the terminal group of a bilinear map. In that situation, the bilinear map looks of no use.

A. Our Contribution

A first contribution is that we fill in the missing piece of direct chosen-ciphertext security for ABE. We develop a technique and apply it to the Waters CP-ABE scheme [11] to obtain IND-CCA security. A second technical contribution is as follows. To overcome the above obstacle, we employ and apply the Twin Diffie-Hellman Trapdoor Test of Cash, Kiltz and Shoup [5]. In addition to that, we also utilize the algebraic trick of Boneh and Boyen [2] and Kiltz [8] to reply for adversary's decryption query. In total, we develop the technique to realize direct chosen-ciphertext security.

B. Related Works

Waters [11] pointed out that IND-CCA security would be attained by the CHK transformation. Gorantla, Boyd and Nieto [6] constructed a IND-CCA secure CP-ABKEM in the random oracle model. Yamada et al. [12] proposed a generic transformation of a IND-CPA secure ABE scheme into a IND-CCA secure ABE scheme. Their transformation is considered to be an ABE-version and versatile. Especially, it can be applied to non-pairing-based scheme.

The Waters CP-ABE [11] can be captured as a CP-ABKEM: the blinding factor can be considered as a random one-time key. In addition, the CP-ABKEM is IND-CPA secure because the Waters CP-ABE is proved to be IND-CPA secure. For theoretical simplicity, we will provide *a scheme of KEM first, and then an encryption scheme*.

C. Efficiency Comparison

We compare efficiency of our `CP-ABKEM` to the original Waters $\texttt{CP-ABKEM}_{\text{cpa}}$. We also compare efficiency of the CP-

978-1-5090-6517-2/17/$31.00 ©2017 IEEE

TABLE I
EFFICIENCY COMPARISON OF IND-SEL-CCA SECURE CP-ABKEMS OBTAINED FROM THE WATERS $\texttt{CP-ABKEM}_{\text{CPA}}$.

Scheme	L(PK)	L(SK_S)	L(CT)	C(Encap)	C(Decap)
Generic transform of Yamada et al. [12]	$+2\lambda^2(\mathbb{G})$	$+2\lambda^2(\mathbb{G})$	$+3\lambda^2$(bit)	$+2\lambda^2$exp.($\mathbb{G}$)	$+2\lambda^2$pair.(e)
Our individual modification (CP-ABKEM)	$+3(\mathbb{G}_T)$	$\times 4$	$+2(\mathbb{G}_T)$	$+4$exp.($\mathbb{G}_T$)	$\times 4$

1) L(data) denotes length of data, C(algorithm) denotes computational amount of algorithm.
2) $+$ and $\times$ mean increment and multiplier to the length or computational amount of the Waters $\texttt{CP-ABKEM}_{\text{cpa}}$.
3) ($\mathbb{G}$), ($\mathbb{G}_T$) and (bit) mean elements in $\mathbb{G}$, elements in $\mathbb{G}_T$ and bits, respectively.
4) exp.($\mathbb{G}$) and pair.(e) mean a computational amount of one exponentiation in $\mathbb{G}$ and one pairing computation by the map e, respectively.

ABKEM obtained by the generic transformation of Yamada et al. [12]. Here the generic transformation [12] is considered in the setting of small attribute universe [7], delegation case and the Lamport one-time signature case. Table I shows these comparison. Our individual technique results in expansion of secret key length and decryption cost of computation by a factor of four, while public key length, ciphertext length and encryption cost of computation are almost the same as those of the Waters. In the case that the size of attribute sets are up to the square of the security parameter λ, lengths and costs of our CP-ABKEM are smaller than those of the CP-ABE obtained via the generic transformation of Yamada et al. [12].

II. PRELIMINARIES

The security parameter is denoted λ. A prime of bit length λ is denoted p. A cyclic group of order p is denoted $\mathbb{G}$.

A. Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and e be a bilinear map, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The bilinear map e has the following properties: 1. Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$. 2. Non-degeneracy: $e(g, g) \neq \text{id}_{\mathbb{G}_T}$ (: the identity element of the group $\mathbb{G}_T$). Parameters of a bilinear map are generated by a probabilistic polynomial time (PPT) algorithm Grp on input λ: $(p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \texttt{Grp}(\lambda)$. Hereafter we assume that the group operation in $\mathbb{G}$ and $\mathbb{G}_T$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ are computable in PT in λ.

B. Access Structure

Let $\mathcal{U} = \{1, \ldots, u\}$ be a set of attributes. An *access structure* is defined as a collection $\mathbb{A}$ of non-empty subsets of $\mathcal{U}$; that is, $\mathbb{A} \subset 2^{\mathcal{U}} \backslash \{\phi\}$. An access structure $\mathbb{A}$ is called *monotone* if for any $B \in \mathbb{A}$ and $B \subset C$, $C \in \mathbb{A}$ holds. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets. We will consider in this paper *only monotone access structures*.

C. CP-ABKEM

A ciphertext-policy attribute-based key encapsulation mechanism (CP-ABKEM) consists of four PPT algorithms (Setup, Encap, Keygen, Decap).

1) $\mathbf{Setup}(\lambda, \mathcal{U})$*:* A setup algorithm Setup takes as input the security parameter λ and the attribute universe $\mathcal{U} = \{1, \ldots, u\}$. In this paper, we assume the *small universe case* [7]. It returns a public key PK and a master secret key MSK.

2) $\mathbf{Encap}(PK, \mathbb{A})$*:* An encapsulation algorithm Encap takes as input the public key PK and an access structure $\mathbb{A}$. It returns a random string κ and its encapsulation ψ.

3) $\mathbf{KeyGen}(PK, MSK, S)$*:* A key generation algorithm KeyGen takes as input the public key PK, the master secret key MSK and an attribute set S. It returns a secret key SK_S corresponding to S.

4) $\mathbf{Decap}(PK, SK_S, \psi)$*:* A decapsulation algorithm Decap takes as input the public key PK, an encapsulation (we also call it a ciphertext according to context) ψ and a secret key SK_S. It first checks whether $S \in \mathbb{A}$, where S and $\mathbb{A}$ are contained in SK_S and ψ, respectively. If the check result is FALSE, it puts $\hat{\kappa} = \perp$. It returns a decapsulation result $\hat{\kappa}$.

5) Chosen-Ciphertext Attack on CP-ABKEM [6]: A *selective game on a target access structure* (IND-sel-CCA game) is described as the following experiment of an adversary $\mathcal{A}$.

$\mathbf{Experiment}^{\text{ind-sel-cca}}_{\mathcal{A},\texttt{CP-ABKEM}}(\lambda, \mathcal{U})$
$\mathbb{A}^* \leftarrow \mathcal{A}(\lambda, \mathcal{U}), (\text{PK}, \text{MSK}) \leftarrow \mathbf{Setup}(\lambda, \mathcal{U})$
$\epsilon \leftarrow \mathcal{A}^{\mathbf{KeyGen}(\text{PK},\text{MSK},\cdot),\mathbf{Decap}(\text{PK},\text{SK}_\cdot,\cdot)}(\text{PK})$
$(\kappa^*, \psi^*) \leftarrow \mathbf{Encap}(\text{PK}, \mathbb{A}^*), \kappa \leftarrow \mathit{KeySp}(\lambda), b \leftarrow \{0, 1\}$
If $b = 1$ then $\tilde{\kappa} = \kappa^*$ else $\tilde{\kappa} = \kappa$
$b' \leftarrow \mathcal{A}^{\mathbf{KeyGen}(\text{PK},\text{MSK},\cdot),\mathbf{Decap}(\text{PK},\text{SK}_\cdot,\cdot)}(\tilde{\kappa}, \psi^*)$
If $b' = b$ then return WIN else return LOSE.

Two kinds of queries are issued by $\mathcal{A}$. One is key-extraction queries. Indicating S_i, $\mathcal{A}$ queries its key-extraction oracle $\text{KeyGen}(\text{PK}, \text{MSK}, \cdot)$ for SK_{S_i}. (We do not require S_{i_1} and S_{i_2} to be distinct). Another is decapsulation queries. Indicating a pair (S_j, ψ_j), $\mathcal{A}$ queries its decapsulation oracle $\text{Decap}(\text{PK}, \text{SK}_\cdot, \cdot)$ for the decapsulation $\hat{\kappa}_j$. An access structure $\mathbb{A}_j$, which is used to generate an encapsulation ψ_j, is implicitly included in ψ_j. In the case that $S \notin \mathbb{A}$, $\hat{\kappa}_j = \perp$ is replied to $\mathcal{A}$. Both kinds of queries are at most q_{k} and q_{d} times in total, respectively, which are polynomial in λ. The access structure $\mathbb{A}^*$ declared by $\mathcal{A}$ is called a *target access structure*. Two restrictions are imposed on $\mathcal{A}$ concerning $\mathbb{A}^*$. In key-extraction queries, each attribute set S_i must satisfy $S_i \notin \mathbb{A}^*$. In decapsulation queries, each pair (S_j, ψ_j) must satisfy $S_j \notin \mathbb{A}^*$ in the phase before the declaration of $\mathbb{A}^*$ and each pair (S_j, ψ_j) must satisfy $S_j \notin \mathbb{A}^* \vee \psi_j \neq \psi^*$ in the phase after the declaration of $\mathbb{A}^*$. The *advantage* of the adversary $\mathcal{A}$ over CP-ABKEM in the IND-CCA game is defined as: $\mathbf{Adv}^{\text{ind-sel-cca}}_{\mathcal{A},\texttt{CP-ABKEM}}(\lambda, \mathcal{U}) = \Pr[\mathbf{Experiment}^{\text{ind-sel-cca}}_{\mathcal{A},\texttt{CP-ABKEM}}(\lambda, \mathcal{U}) \text{ returns WIN}]$. CP-ABKEM

is called *secure against chosen-ciphertext attacks* if, for any PPT adversary $\mathcal{A}$ and for any attribute universe $\mathcal{U}$, $\mathbf{Adv}^{\text{ind-sel-cca}}_{\mathcal{A},\texttt{CP-ABKEM}}(\lambda,\mathcal{U})$ is negligible in λ. In the indistinguishability game against *chosen-plaintext attack* (IND-CPA game), the adversary $\mathcal{A}$ issues no decapsulation query (that is, $q_{\rm d}=0$). The *advantage* $\mathbf{Adv}^{\text{game}}_{\mathcal{A},\text{scheme}}(\lambda,\mathcal{U})$ of the adversary $\mathcal{A}$ over a scheme in a game is defined in the same way as above.

6) Ciphertext-Policy Attribute-Based Encryption Scheme: In the case of a ciphertext-policy attribute-based encryption scheme (CP-ABE), Encap$(\text{PK},\mathbb{A})$ and Decap$(\text{PK},\text{SK}_S,\psi)$ are replaced by PPT algorithms Encrypt$(\text{PK},\mathbb{A},m)$ and Decrypt$(\text{PK},\text{SK}_S,\text{CT})$, respectively, where m and CT mean a message and a ciphertext, respectively. The IND-CCA game for CP-ABE is defined in the same way as for CP-ABKEM above, except the following difference. In Challenge phase, the adversary $\mathcal{A}$ submits two equal length messages (plaintexts) m_0 and m_1. Then the challenger flips a coin $b\in\{0,1\}$ and gives an encryption result CT of m_b to $\mathcal{A}$. In Guess phase, the adversary $\mathcal{A}$ returns $b'\in\{0,1\}$. If $b'=b$, then $\mathcal{A}$ wins in the IND-CCA game. Otherwise, $\mathcal{A}$ loses.

D. The Twin Diffie-Hellman Technique

A 6-tuple $(g,X_1,X_2,Y,Z_1,Z_2)\in\mathbb{G}^6$ is called a *twin Diffie-Hellman tuple* if the tuple is written as $(g,g^{x_1},g^{x_2},g^y,g^{x_1y},g^{x_2y})$ for some elements x_1,x_2,y in $\mathbb{Z}_p$. In other words, a 6-tuple (g,X_1,X_2,Y,Z_1,Z_2) is a twin Diffie-Hellman tuple (twin DH tuple, for short) if $Y=g^y$ and $Z_1=X_1^y$ and $Z_2=X_2^y$. The following lemma of Cash, Kiltz and Shoup will be used in the security proof to decide whether a tuple is a twin DH tuple or not.

Lemma 1 (Cash, Kiltz and Shoup [5] "Trapdoor Test"): Let X_1,r,s be mutually independent random variables, where X_1 takes values in $\mathbb{G}$, and each of r,s is uniformly distributed over $\mathbb{Z}_p$. Define the random variable $X_2=X_1^{-r}g^s$. Suppose that $\hat{Y},\hat{Z}_1,\hat{Z}_2$ are random variables taking values in $\mathbb{G}$, each of which is defined independently of r. Then the probability that the truth value of $\hat{Z}_1{}^r\hat{Z}_2=\hat{Y}^s$ does not agree with the truth value of $(g,X_1,X_2,\hat{Y},\hat{Z}_1,\hat{Z}_2)$ being a twin DH tuple is at most $1/p$. Moreover, if $(g,X_1,X_2,\hat{Y},\hat{Z}_1,\hat{Z}_2)$ is a twin DH tuple, then $\hat{Z}_1{}^r\hat{Z}_2=\hat{Y}^s$ certainly holds.

Note that Lemma 1 is a statistical property. To be precise, we consider the following experiment of an algorithm *Cheat* with *unbounded computational power*, where *Cheat*, given a triple (g,X_1,X_2), tries to generate a 6-tuple $(g,X_1,X_2,\hat{Y},\hat{Z}_1,\hat{Z}_2)$ that passes the "Trapdoor Test" though it is *not* a twin DH tuple.

$\mathbf{Experiment}^{\text{twinDH-test}}_{\mathit{Cheat},\mathbb{G}}(\lambda)$

$(g,X_1)\leftarrow\mathbb{G}^2,(r,s)\leftarrow\mathbb{Z}_p^2,X_2=X_1^{-r}g^s$

$\mathbb{G}^3\ni(\hat{Y},\hat{Z}_1,\hat{Z}_2)\leftarrow \mathit{Cheat}(g,X_1,X_2)$

If $\hat{Z}_1{}^r\hat{Z}_2=\hat{Y}^s\wedge(g,X_1,X_2,\hat{Y},\hat{Z}_1,\hat{Z}_2)$ is NOT a twin DH, then return WIN else return LOSE

Let us define the advantage of *Cheat* over $\mathbb{G}$ as follows.

$$\mathbf{Adv}^{\text{twinDH-test}}_{\mathit{Cheat},\mathbb{G}}(\lambda)=\Pr[\mathbf{Experiment}^{\text{twinDH-test}}_{\mathit{Cheat},\mathbb{G}}(\lambda)\text{ returns WIN}].$$

Now we are ready to complement Lemma 1.

Lemma 2 (a Complement to [5] "Trapdoor Test"): For any algorithm *Cheat* with unbounded computational power, $\mathbf{Adv}^{\text{twinDH-test}}_{\mathit{Cheat},\mathbb{G}}(\lambda)$ is at most $1/p$.

A proof of Lemma 2 is given in the full version paper [1].

III. Securing the Waters CP-ABKEM against Chosen-Ciphertext Attacks

In this section, we describe our direct chosen-ciphertext security technique by applying it to the Waters CP-ABE [11].

1) Overview of Our Technique: The Waters CP-ABE is proved to be secure in the IND-sel-CPA game in [11]. We convert it into a scheme that is secure in the IND-sel-CCA game by employing the Twin Diffie-Hellman technique of Cash, Kiltz and Shoup [5] and the algebraic trick of Boneh and Boyen [2] and Kiltz [8]. In our encryption, a ciphertext become to contain additional two elements (d_1,d_2), which function in decryption as a "check sum" to verify that a tuple is certainly a twin DH tuple. In our security proof, the Twin Diffie-Hellman Trapdoor Test does the function instead. It is noteworthy that we can not use the bilinear map instead because the tuple to be verified is in the terminal group. In addition, the algebraic trick enables to answer for adversary's decryption queries.

2) KEM and Encryption Scheme: The Waters CP-ABE can be captured as a CP-ABKEM: the blinding factor of the form $e(g,g)^{\alpha s}$ in the Waters CP-ABE can be considered as a random one-time key. So we call it the Waters CP-ABKEM hereafter and denote it as $\texttt{CP-ABKEM}_{\text{cpa}}$. Likewise, we distinguish parameters and algorithms of $\texttt{CP-ABKEM}_{\text{cpa}}$ by the index $_{\text{cpa}}$. For theoretical simplicity, we first develop a KEM CP-ABKEM.

A. Our Construction

Our CP-ABKEM consists of the following four PPT algorithms (Setup, Encap, KeyGen, Decap). Roughly speaking, the Waters original scheme $\texttt{CP-ABKEM}_{\text{cpa}}$ (the first scheme in [11]) corresponds to the case $k=1$ below excluding the "check sum" (d_1,d_2).

1) $\mathbf{Setup}(\lambda,\mathcal{U})$*:* Setup takes as input the security parameter λ and the attribute universe $\mathcal{U}=\{1,\ldots,u\}$. It runs $\texttt{Grp}(\lambda)$ to get $(p,\mathbb{G},\mathbb{G}_T,g,e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order p, $e:\mathbb{G}\to\mathbb{G}_T$ is a bilinear map and g is a generator of $\mathbb{G}$. These become public parameters. Then Setup takes u random group elements $h_1,\ldots,h_u\in\mathbb{G}$ that are associated with the u attributes. In addition, it chooses random exponents $\alpha_k\in\mathbb{Z}_p,k=1,\ldots,4,a\in\mathbb{Z}_p$ and a hash key $\eta\in\mathit{HKey}(\lambda)$. The public key is published as $\text{PK}=(g,g^a,h_1,\ldots,h_u,e(g,g)^{\alpha_1},\ldots,e(g,g)^{\alpha_4},\eta)$. The authority sets $\text{MSK}=(g^{\alpha_1},\ldots,g^{\alpha_4})$ as the master secret key.

2) $\mathbf{Encap}(\mathit{PK},\mathbb{A})$*:* The encapsulation algorithm Encap takes as input the public key PK and an LSSS access structure $\mathbb{A}=(M,\rho)$, where M is an $l\times n$ matrix and ρ is the function which maps each row i of M to an attribute in $\mathcal{U}=\{1,\ldots,u\}$. Encap first chooses a random value $s\in\mathbb{Z}_p$ that is the encryption exponent s and random values $y_2,\ldots,y_n\in\mathbb{Z}_p$. Then Encap forms a vector $\vec{v}=(s,y_2,\ldots,y_n)$. For $i=1$

to l, it calculates $\lambda_i = \vec{v} \cdot M_i$, where M_i denotes the i-th row vector of M. In addition, Encap chooses random values $r_1, \ldots, r_l \in \mathbb{Z}_p$. Then, a pair of a random one-time key and its encapsulation (κ, ψ) is computed as follows.

Put $C' = g^s$; For $i = 1$ to $l : C_i = g^{a\lambda_i} h_{\rho(i)}^{-r_i}, D_i = g^{r_i}$;
$\psi_{\text{cpa}} = (\mathbb{A}, C', ((C_i, D_i); i = 1, \ldots, l)), \tau \leftarrow H_\eta(\psi_{\text{cpa}})$;
For $k = 1$ to $4 : \kappa_k = e(g,g)^{\alpha_k s}; d_1 = \kappa_1^\tau \kappa_3, d_2 = \kappa_2^\tau \kappa_4$;
$(\kappa, \psi) = (\kappa_1, (\psi_{\text{cpa}}, d_1, d_2))$.

3) $\mathbf{KeyGen}(MSK, PK, S)$*:* The key generation algorithm KeyGen takes as input the master secret key MSK, the public key PK and a set S of attributes. KeyGen first chooses a random $t_k \in \mathbb{Z}_p, k = 1, \ldots, 4$. It creates the secret key SK_S as follows.

$1 \leqslant k \leqslant 4 : K_k = g^{\alpha_k} g^{at_k}, L_k = g^{t_k}, [x \in S : K_{k,x} = h_x^{t_k}]$;
$\text{SK}_S = ((K_k, L_k, (K_{k,x}; x \in S)); k = 1, \ldots, 4)$.

4) $\mathbf{Decap}(PK, \psi, SK_S)$*:* The decapsulation algorithm Decap takes as input the public key PK, an encapsulation ψ for the access structure $\mathbb{A} = (M, \rho)$ and a private key SK_S for an attribute set S. It first checks whether $S \in \mathbb{A}$. If the result is FALSE, put $\hat{\kappa} = \perp$. else, let $I_S = \rho^{-1}(S) \subset \{1, \ldots, l\}$ and let $\{\omega_i \in \mathbb{Z}_p; i \in I_S\}$ be a set of linear reconstruction constants. Then, the decapsulation $\hat{\kappa}$ is computed as follows.

Parse ψ into $(\psi_{\text{cpa}} = (\mathbb{A}, C', ((C_i, D_i); i = 1, \ldots, l)), d_1, d_2)$;
$\tau \leftarrow H_\eta(\psi_{\text{cpa}})$;
$1 \leqslant k \leqslant 4 : \hat{\kappa_k} = e(C', K_k) / \prod_{i \in I_S} (e(C_i, L_k) e(D_i, K_{k,\rho(i)}))^{\omega_i}$
$= e(g,g)^{\alpha_k s}$,
If $\hat{\kappa_1}^\tau \hat{\kappa_3} \neq d_1 \vee \hat{\kappa_2}^\tau \hat{\kappa_4} \neq d_2$, then put $\hat{\kappa} = \perp$, else put $\hat{\kappa} = \hat{\kappa_1}$.

B. Security and its Proof

Theorem 1: If the Waters $\texttt{CP-ABKEM}_{\text{cpa}}$ [11] is selectively secure against chosen-plaintext attacks and an employed hash function family *Hfam* has target collision resistance, then our CP-ABKEM is selectively secure against chosen-ciphertext attacks. More precisely, for any given PPT adversary $\mathcal{A}$ that attacks CP-ABKEM in the IND-sel-CCA game where decapsulation queries are at most q_{d} times, and for any attribute universe $\mathcal{U}$, there exist a PPT adversary $\mathcal{B}$ that attacks $\texttt{CP-ABKEM}_{\text{cpa}}$ in the IND-sel-CPA game and a PPT target collision finder $\mathcal{CF}$ on *Hfam* that satisfy the following tight reduction.

$$\mathbf{Adv}^{\text{ind-sel-cca}}_{\mathcal{A},\texttt{CP-ABKEM}}(\lambda, \mathcal{U}) \leqslant \mathbf{Adv}^{\text{ind-sel-cpa}}_{\mathcal{B},\texttt{CP-ABKEM}_{\text{cpa}}}(\lambda, \mathcal{U}) + \mathbf{Adv}^{\text{tcr}}_{\mathcal{CF},Hfam}(\lambda) + \frac{q_{\text{d}}}{p}.$$

The definition of the target collision resistance game and the advantage of $\mathcal{CF}$ are given in [1].

Proof. Given any adversary $\mathcal{A}$ that attacks our scheme CP-ABKEM in the IND-sel-CCA game, we construct an adversary $\mathcal{B}$ that attacks the Waters scheme $\texttt{CP-ABKEM}_{\text{cpa}}$ in the IND-sel-CPA game as follows.

1) Commit a Target Access Structure: $\mathcal{B}$ is given $(\lambda, \mathcal{U})$ as inputs, where λ is the security parameter and $\mathcal{U} = \{1, \ldots, u\}$ is the attribute universe. $\mathcal{B}$ invokes $\mathcal{A}$ on $(\lambda, \mathcal{U})$ and gets a target access structure $\mathbb{A}^* = (M^*, \rho^*)$ from $\mathcal{A}$, where M^* is of size $l^* \times n^*$. $\mathcal{B}$ uses $\mathbb{A}^*$ as the target access structure of itself and outputs $\mathbb{A}^*$.

2) Set up: In return to outputting $\mathbb{A}^*$, $\mathcal{B}$ receives the public key PK_{cpa} for $\texttt{CP-ABKEM}_{\text{cpa}}$, which consists of the following components.

$$\text{PK}_{\text{cpa}} = (g, g^a, h_1, \ldots, h_u, e(g,g)^\alpha).$$

To set up a public key PK for CP-ABKEM, $\mathcal{B}$ *herein* needs a challenge instance: $\mathcal{B}$ queries its challenger and gets a challenge instance $(\tilde{\kappa}, \psi^*_{\text{cpa}})$. It consists of the following components.

$\tilde{\kappa} = e(g,g)^{\alpha s^*}$ OR a random one-time key $\kappa \in \mathit{KeySp}(\lambda)$,
$\psi^*_{\text{cpa}} = (\mathbb{A}^*, C'^* = g^{s^*}, ((C_i^*, D_i^*); i = 1, \ldots, l^*))$.

Then $\mathcal{B}$ makes the rest of parameters of PK as follows.

Pick up $\eta \leftarrow \mathit{HKey}(\lambda)$, take $\tau^* \leftarrow H_\eta(\psi^*_{\text{cpa}})$;
Put $e(g,g)^{\alpha_1} = e(g,g)^\alpha$;
Pick up $\gamma_1, \gamma_2 \leftarrow \mathbb{Z}_p$, put $e(g,g)^{\alpha_2} = e(g,g)^{\gamma_2} / e(g,g)^{\alpha_1 \gamma_1}$;
Pick up $\mu_1, \mu_2 \leftarrow \mathbb{Z}_p$, put $e(g,g)^{\alpha_3} = e(g,g)^{\mu_1} / e(g,g)^{\alpha_1 \tau^*}$,
$e(g,g)^{\alpha_4} = e(g,g)^{\mu_2} / e(g,g)^{\alpha_2 \tau^*}$.

Note that we have set implicitly the following relations:

$$\alpha_2 = \gamma_2 - \alpha_1\gamma_1, \quad \alpha_3 = \mu_1 - \alpha_1\tau^*,$$
$$\alpha_4 = \mu_2 - \alpha_2\tau^* = \mu_2 - (\gamma_2 - \alpha_1\gamma_1)\tau^*. \tag{1}$$

A public key PK for CP-ABKEM become:

$$\text{PK} = (\text{PK}_{\text{cpa}}, e(g,g)^{\alpha_2}, e(g,g)^{\alpha_3}, e(g,g)^{\alpha_4}, \eta).$$

Then $\mathcal{B}$ inputs PK into $\mathcal{A}$. Note that PK determines the corresponding MSK uniquely.

3) Phase 1: $\mathcal{B}$ answers for two types of $\mathcal{A}$'s queries as follows.

(1) Key-Extraction Queries. In the case that $\mathcal{A}$ issues a key-extraction query for an attribute set $S \subset \mathcal{U}$, $\mathcal{B}$ has to simulate $\mathcal{A}$'s challenger. To do so, $\mathcal{B}$ issues key-extraction queries to $\mathcal{B}$'s challenger for S *repeatedly up to four times*. As replies, $\mathcal{B}$ gets four secret keys of the Waters $\texttt{CP-ABKEM}_{\text{cpa}}$ for a single attribute set S:

$$\text{SK}_{\text{cpa},S,k} = (K_{\text{cpa},k}, L_{\text{cpa},k}, (K_{\text{cpa},k,x}; x \in \mathcal{S})), k = 1, \ldots, 4.$$

We remark that, according to the randomness in the key-generation algorithm of the Waters $\texttt{CP-ABKEM}_{\text{cpa}}$, all four secret keys $\text{SK}_{\text{cpa},S,1}, \ldots, \text{SK}_{\text{cpa},S,4}$ are random and mutually independent. To reply a secret key SK_S of our CP-ABKEM to $\mathcal{A}$, $\mathcal{B}$ converts the four secret keys as follows.

Put $K_1 = K_{\text{cpa},1}$, $L_1 = L_{\text{cpa},1}$, $K_{1,x} = K_{\text{cpa},1,x}, x \in S$;
Put $K_2 = g^{\gamma_2} K_{\text{cpa},2}^{-\gamma_1}$, $L_2 = L_{\text{cpa},2}^{-\gamma_1}$, $K_{2,x} = K_{\text{cpa},2,x}^{-\gamma_1}$, $x \in S$;
Put $K_3 = g^{\mu_1} K_{\text{cpa},3}^{-\tau^*}$, $L_3 = L_{\text{cpa},3}^{-\tau^*}$, $K_{3,x} = K_{\text{cpa},3,x}^{\tau^*}$, $x \in S$;
Put $K_4 = g^{\mu_2 - \gamma_2\tau^*} K_{\text{cpa},4}^{\gamma_1\tau^*}, L_4 = L_{\text{cpa},4}^{\gamma_1\tau^*}, K_{4,x} = K_{\text{cpa},4,x}^{\gamma_1\tau^*}, x \in S$.

Then $\mathcal{B}$ replies $\mathrm{SK}_S = ((K_k, L_k, (K_{k,x}; x \in S)); k = 1, \ldots, 4)$ to $\mathcal{A}$.

(2) Decapsulation Queries. In the case that $\mathcal{A}$ issues a decapsulation query for (S, ψ), where $S \subset \mathcal{U}$ is an attribute set and $\psi = (\psi_{\mathrm{cpa}}, d_1, d_2)$ is an encapsulation concerning $\mathbb{A}$, $\mathcal{B}$ has to simulate $\mathcal{A}$'s challenger. To do so, $\mathcal{B}$ computes the decapsulation result $\hat{\kappa}$ as follows.

If $S \notin \mathbb{A}$ then put $\hat{\kappa} = \perp$,
else Take $\tau \leftarrow H_\eta(\psi_{\mathrm{cpa}})$;
Put $\hat{Y} = e(C', g)^{\tau - \tau^*}$,
$\hat{Z}_1 = d_1 / e(C', g)^{\mu_1}, \hat{Z}_2 = d_2 / e(C', g)^{\mu_2}$;
If $\hat{Z}_1{}^{\gamma_1} \hat{Z}_2 \neq \hat{Y}^{\gamma_2}$ (: call this check TwinDH-Test)
then put $\hat{\kappa} = \hat{\kappa_1} = \perp$
else If $\tau = \tau^*$ then abort (: call this case Abt)
else $\hat{\kappa} = \hat{\kappa_1} = \hat{Z_1}^{1/(\tau - \tau^*)}$.

Then $\mathcal{B}$ replies $\hat{\kappa}$ to $\mathcal{A}$.

4) Challenge: In the case that $\mathcal{A}$ queries its challenger for a challenge instance, $\mathcal{B}$ makes a challenge instance as follows.

Put $d_1^* = e(C'^*, g)^{\mu_1}, d_2^* = e(C'^*, g)^{\mu_2}$;
Put $\psi^* = (\psi_{\mathrm{cpa}}^*, d_1^*, d_2^*)$.

Then $\mathcal{B}$ feeds $(\tilde{\kappa}, \psi^*)$ to $\mathcal{A}$ as a challenge instance.

5) Phase 2: The same as in Phase 1.

6) Guess: In the case that $\mathcal{A}$ returns $\mathcal{A}$'s guess $\tilde{b}$, $\mathcal{B}$ returns $\tilde{b}$ itself as $\mathcal{B}$'s guess.

In the above construction of $\mathcal{B}$, $\mathcal{B}$ can perfectly simulate the real view of $\mathcal{A}$ until the case Abt happens, except for a negligible case, and hence the algorithm $\mathcal{A}$ works as designed. To see the perfect simulation with a negligible exceptional case, we are enough to prove the following seven claims.

Claim 1: The reply $\mathrm{SK}_S = ((K_k, L_k, (K_{k,x}; x \in S)); k = 1, 2, 3, 4)$ for a key-extraction query is a perfect simulation.

Proof. We must consider the implicit relations (1). For the index 2, we have implicitly set the randomness $t_2 = t_{\mathrm{cpa},2}(-\gamma_1)$ and we get: $K_2 = g^{\gamma_2} K_{\mathrm{cpa},2}^{-\gamma_1} = g^{\gamma_2} (g^{\alpha_1} g^{a t_{\mathrm{cpa},2}})^{-\gamma_1} = g^{\alpha_2} g^{a t_2}, L_2 = L_{\mathrm{cpa},2}^{-\gamma_1} = (g^{t_{\mathrm{cpa},2}})^{-\gamma_1} = g^{t_2}, K_{2,x} = K_{\mathrm{cpa},2,x}^{-\gamma_1} = (h_x^{t_{\mathrm{cpa},2}})^{-\gamma_1} = h_x^{t_2}, x \in S$. For the index 3 and 4, see [1]. □

Claim 2: $(e(g,g), e(g,g)^{\alpha_1}, e(g,g)^{\alpha_2}, \hat{Y}, \hat{Z}_1, \hat{Z}_2)$ is a twin Diffie-Hellman tuple if and only if $(e(g,g), e(g,g)^{\alpha_1 \tau}\ e(g,g)^{\alpha_3},\ e(g,g)^{\alpha_2 \tau}\ e(g,g)^{\alpha_4}, e(C', g), d_1, d_2)$ is a twin Diffie-Hellman tuple.

Proof. This claim can be proved by a short calculation. See [1]. □

Claim 3: If $(e(g,g), e(g,g)^{\alpha_1}, e(g,g)^{\alpha_2}, \hat{Y}, \hat{Z}_1, \hat{Z}_2)$ is a twin Diffie-Hellman tuple, then $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$ certainly passes the TwinDH-Test: $\hat{Z}_1{}^{\gamma_1} \hat{Z}_2 = \hat{Y}^{\gamma_2}$.

Proof. This claim is a direct consequence of Lemma 1. □

Claim 4: Consider the following event OVL_i:

In the i-th TwinDH-Test, the following condition holds:

$$\begin{cases} \hat{Z}_1{}^{\gamma_1} \hat{Z}_2 = \hat{Y}^{\gamma_2} \text{ holds and} \\ (e(g,g), e(g,g)^{\alpha_1}, e(g,g)^{\alpha_2}, \hat{Y}, \hat{Z}_1, \hat{Z}_2) \text{ is NOT a twin DH.} \end{cases}$$

Then, for at most q_{d} times decapsulation queries of $\mathcal{A}$, the probability that at least one OVL_i occurs is negligible in λ. More precisely, the following inequality holds:

$$\Pr[\vee_{i=1}^{q_{\mathrm{d}}} \mathrm{OVL}_i] \leqslant q_{\mathrm{d}} / p. \tag{2}$$

Proof. To apply Lemma 2, we construct an algorithm $\mathit{Cheat}_{\lambda,\mathcal{U}}$ with unbounded computational power, which takes as input $(e(g,g), e(g,g)^{\alpha_1}, e(g,g)^{\alpha_2})$ and returns $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$ employing $\mathcal{A}$ as a subroutine. Fig. 1 shows the construction.

Given $(e(g,g), e(g,g)^{\alpha_1}, e(g,g)^{\alpha_2})$ as input :
Set up
Initialize the inner state and put TABLE $= \phi$;
Get a target access structure $\mathbb{A}^* \leftarrow \mathcal{A}(\lambda, \mathcal{U})$;
Compute the base $g \in \mathbb{G}$ from $(e(g,g), e)$;
Pick up $a \in \mathbb{Z}_p$ and $h_1, \ldots, h_u \in \mathbb{G}$;
Put $\mathrm{PK}_{\mathrm{cpa}} = (g, g^a, h_1, \ldots, h_u, e(g,g)^{\alpha_1})$;
Get $(\kappa^*, \psi_{\mathrm{cpa}}^*) \leftarrow \mathbf{Encap}_{\mathrm{cpa}}(\mathrm{PK}_{\mathrm{cpa}}, \mathbb{A}^*)$;
Pick up $\eta \leftarrow \mathit{HKey}(\lambda)$ and compute $\tau^* \leftarrow H_\eta(\psi_{\mathrm{cpa}}^*)$;
Compute discrete logarithms $\alpha_1, \alpha_2 \in \mathbb{Z}_p$ of $e(g,g)^{\alpha_1}, e(g,g)^{\alpha_2}$ to the base $e(g,g)$;
Pick up $\mu_1, \mu_2 \leftarrow \mathbb{Z}_p$
and put $\alpha_3 = \mu_1 - \alpha_1 \tau^*, \alpha_4 = \mu_2 - \alpha_2 \tau^*$;
Put $\mathrm{PK} = (\mathrm{PK}_{\mathrm{cpa}}, e(g,g)^{\alpha_2}, e(g,g)^{\alpha_3}, e(g,g)^{\alpha_4}, \eta)$,
$\mathrm{MSK} = (g^{\alpha_1}, g^{\alpha_2}, g^{\alpha_3}, g^{\alpha_4})$; Give PK to $\mathcal{A}$;
Phase 1
In the case $\mathcal{A}$ makes a key-extraction query for $S \subset \mathcal{U}$;
Reply SK_S to $\mathcal{A}$ in the same way as **KeyGen** using MSK;
In the case $\mathcal{A}$ makes a decapsulation query for
$(\mathbb{A}, \psi = (\psi_{\mathrm{cpa}}, d_1, d_2), S)$;
Reply $\hat{\kappa}$ to $\mathcal{A}$ in the same way as **Decap** does using MSK;
Compute $\hat{Y} = e(C', g)^{\tau - \tau^*}$,
$\hat{Z}_1 = d_1 / e(C', g)^{\mu_1}, \hat{Z}_2 = d_2 / e(C', g)^{\mu_2}$;
Update TABLE $=$ TABLE $\cup (\hat{Y}, \hat{Z}_1, \hat{Z}_2)$;
Challenge
In the case $\mathcal{A}$ makes a challenge instance query;
Put $d_1^* = e(C'^*, g)^{\mu_1}, d_2^* = e(C'^*, g)^{\mu_2}, \psi^* = (\psi_{\mathrm{cpa}}^*, d_1^*, d_2^*)$;
Pick up $\kappa \leftarrow \mathit{KeySp}(\lambda), b \leftarrow \{0, 1\}$;
If $b = 1$ then put $\tilde{\kappa} = \kappa^*$ else put $\tilde{\kappa} = \kappa$;
Reply $(\tilde{\kappa}, \psi^*)$ to $\mathcal{A}$;
Phase 2
The same as in Phase 1;
Return
In the case $\mathcal{A}$ returns its guess b^*;
Choose one triple $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$ from TABLE at random;
Return $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$.

Fig. 1. An algorithm $\mathit{Cheat}_{\lambda,\mathcal{U}}$ with unbounded computational power.

First, note that the view of $\mathcal{A}$ in $\mathit{Cheat}_{\lambda,\mathcal{U}}$ is the same as the real view of $\mathcal{A}$ and hence the algorithm $\mathcal{A}$ works as designed.

Second, note that the return $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$ of $\mathit{Cheat}_{\lambda,\mathcal{U}}$ is randomized in TABLE. Hence:

$$\sum_{i=1}^{q_{\mathrm{d}}} \frac{1}{q_{\mathrm{d}}} \Pr[\mathrm{OVL}_i] = \frac{1}{q_{\mathrm{d}}} \sum_{i=1}^{q_{\mathrm{d}}} \Pr[\mathrm{OVL}_i] = \mathbf{Adv}^{\text{twinDH-test}}_{\mathit{Cheat}_{\lambda,\mathcal{U}},\mathbb{G}}(\lambda). \tag{3}$$

Third, applying Lemma 2 to $\mathit{Cheat}_{\lambda,\mathcal{U}}$, we get:

$$\mathbf{Adv}^{\text{twinDH-test}}_{\mathit{Cheat}_{\lambda,\mathcal{U}},\mathbb{G}}(\lambda) \leqslant 1/p. \tag{4}$$

Combining (3) and (4), we have:

$$\Pr[\vee_{i=1}^{q_\mathrm{d}} \mathrm{OVL}_i] \leqslant \sum_{i=1}^{q_\mathrm{d}} \Pr[\mathrm{OVL}_i] \leqslant q_\mathrm{d}\mathbf{Adv}^{\text{twinDH-test}}_{\mathit{Cheat}_{\lambda,\mathcal{U}},\mathbb{G}}(\lambda) \leqslant \frac{q_\mathrm{d}}{p}.\square$$

Claim 5: The probability that OVL_i never occurs in TWINDH-TEST for each i and ABT occurs is negligible in λ. More precisely, the following inequality holds:

$$\Pr[\left(\wedge_{i=1}^{q_\mathrm{d}} \neg \mathrm{OVL}_i\right) \wedge \mathrm{ABT}] \leqslant \mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{CF},\mathit{Hfam}}(\lambda). \tag{5}$$

Proof. This claim is proved by constructing a collision finder $\mathcal{CF}$ on *Hfam*. See [1]. □

Claim 6: The reply $\hat{\kappa}$ to $\mathcal{A}$ as an answer for a decapsulation query is correct.

Claim 7: The challenge instance $\psi^* = (\psi^*_{\mathrm{cpa}}, d_1^*, d_2^*)$ is correctly distributed.

Proof. These two claims are proved by a direct calculation. See [1]. □

Now we are ready to evaluate the advantage of $\mathcal{B}$ in the IND-sel-CPA game. That $\mathcal{A}$ wins in the IND-sel-CCA game means that $(\tilde{\kappa}, \psi^* = (\psi^*_{\mathrm{cpa}}, d_1^*, d_2^*))$ is correctly guessed. This is equivalent to that $(\tilde{\kappa}, \psi^*_{\mathrm{cpa}})$ is correctly guessed because ψ^*_{cpa} determines the consistent blinding factor $\kappa^* = e(g,g)^{\alpha s^*}$ uniquely. This means that $\mathcal{B}$ wins in the IND-sel-CPA game.

Therefore, the probability that $\mathcal{B}$ wins is equal to the probability that $\mathcal{A}$ wins, OVL_i never holds in TWINDH-TEST for each i and ABT never occurs. So we have:

$$\begin{aligned}
&\Pr[\mathcal{B} \text{ wins}] = \Pr[(\mathcal{A} \text{ wins}) \wedge \left(\wedge_{i=1}^{q_\mathrm{d}} \neg \mathrm{OVL}_i\right) \wedge (\neg \mathrm{ABT})] \\
=&\Pr[\mathcal{A} \text{ wins}] - \Pr[(\mathcal{A} \text{ wins}) \wedge \neg\Big(\Big(\wedge_{i=1}^{q_\mathrm{d}} \neg \mathrm{OVL}_i\Big) \wedge (\neg \mathrm{ABT})\Big)] \\
\geqslant&\Pr[\mathcal{A} \text{ wins}] - \Pr[\neg\Big(\Big(\wedge_{i=1}^{q_\mathrm{d}} \neg \mathrm{OVL}_i\Big) \wedge (\neg \mathrm{ABT})\Big)] \\
=&\Pr[\mathcal{A} \text{ wins}] - (\Pr[\vee_{i=1}^{q_\mathrm{d}} \mathrm{OVL}_i] + \Pr[\Big(\wedge_{i=1}^{q_\mathrm{d}} \neg \mathrm{OVL}_i\Big) \wedge \mathrm{ABT}]).
\end{aligned}$$

Substituting (2), (5) and advantages into the above, we have:

$$\begin{aligned}
\mathbf{Adv}^{\text{ind-sel-cpa}}_{\mathcal{B},\texttt{CP-ABKEM}_{\mathrm{cpa}}}(\lambda,\mathcal{U}) \geqslant\ & \mathbf{Adv}^{\text{ind-sel-cca}}_{\mathcal{A},\texttt{CP-ABKEM}}(\lambda,\mathcal{U}) \\
& - \frac{q_\mathrm{d}}{p} - \mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{CF},\mathit{Hfam}}(\lambda).
\end{aligned}$$

This is what we should prove in Theorem 1. □

C. Encryption Version

It is straightforward to construct our encryption scheme CP-ABE from CP-ABKEM. The IND-sel-CCA security of CP-ABE is proved based on IND-sel-CPA security of the Waters *KEM* $\texttt{CP-ABKEM}_{\mathrm{cpa}}$. See [1].

IV. CONCLUSIONS

We developed a technique of the so-called direct chosen-ciphertext security *for ABE* in the standard model in the case of the Waters scheme ($\texttt{CP-ABKEM}_{\mathrm{cpa}}$, $\texttt{CP-ABE}_{\mathrm{cpa}}$). We utilized the Twin Diffie-Hellman Trapdoor Test of Cash, Kiltz and Shoup and the algebraic trick of Boneh and Boyen [2] and Kiltz [8]. Our technique is helpful when a Diffie-Hellman tuple to be verified is in the terminal group of a bilinear map.

ACKNOWLEDGMENT

This work is partially supported by kakenhi Grant-in-Aid for Scientific Research (C) JP15K00029 from Japan Society for the Promotion of Science.

REFERENCES

[1] H. Anada and S. Arita, "Short cca-secure ciphertext-policy attribute-based encryption," *the full version, to appear in IACR Cryptology ePrint Archive*, 2017.

[2] D. Boneh and X. Boyen, "Efficient selective-id secure identity-based encryption without random oracles," in *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, 2004, pp. 223–238. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24676-3_14

[3] X. Boyen, Q. Mei, and B. Waters, "Direct chosen ciphertext security from identity-based techniques," in *Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005, Alexandria, VA, USA, November 7-11, 2005*, 2005, pp. 320–329. [Online]. Available: http://doi.acm.org/10.1145/1102120.1102162

[4] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, 2004, pp. 207–222. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24676-3_13

[5] D. Cash, E. Kiltz, and V. Shoup, "The twin diffie-hellman problem and applications," in *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, 2008, pp. 127–145. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-78967-3_8

[6] M. C. Gorantla, C. Boyd, and J. M. G. Nieto, "Attribute-based authenticated key exchange," in *Information Security and Privacy - 15th Australasian Conference, ACISP 2010, Sydney, Australia, July 5-7, 2010. Proceedings*, 2010, pp. 300–317. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-14081-5_19

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, 2006, pp. 89–98. [Online]. Available: http://doi.acm.org/10.1145/1180405.1180418

[8] E. Kiltz, "Chosen-ciphertext security from tag-based encryption," in *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, 2006, pp. 581–600. [Online]. Available: http://dx.doi.org/10.1007/11681878_30

[9] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, 2010, pp. 62–91. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13190-5_4

[10] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, 2005, pp. 457–473. [Online]. Available: http://dx.doi.org/10.1007/11426639_27

[11] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, 2011, pp. 53–70. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19379-8_4

[12] S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro, "Generic constructions for chosen-ciphertext secure attribute based encryption," in *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, 2011, pp. 71–89. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19379-8_5