1-2020

Key regeneration-free ciphertext-policy attribute-based encryption and its application

Hui CUI

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Baodong QIN

Jian WENG

Citation

CUI, Hui; DENG, Robert H.; QIN, Baodong; and WENG, Jian. Key regeneration-free ciphertext-policy attribute-based encryption and its application. (2020). *Information Sciences*. 517, 217-229.
Available at: https://ink.library.smu.edu.sg/sis_research/5061

Key regeneration-free ciphertext-policy attribute-based encryption and its application

Hui Cui[a,e,*], Robert H. Deng[b], Baodong Qin[c], Jian Weng[d]

[a] *Discipline of Information Technology, Mathematics and Statistics, Murdoch University, Perth, Australia*
[b] *School of Information Systems, Singapore Management University, Singapore*
[c] *National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, China*
[d] *Department of Computer Science, Jinan University, Guangzhou, China*
[e] *Data61, CSIRO, Melbourne, Australia*

ARTICLE INFO

Keywords:
ABE
Key regeneration-free
Key randomization
Generic transformation

ABSTRACT

Attribute-based encryption (ABE) provides a promising solution for enabling scalable access control over encrypted data stored in the untrusted servers (e.g., cloud) due to its ability to perform data encryption and decryption defined over descriptive attributes. In order to bind different components which correspond to different attributes in a user's attribute-based decryption key together, key randomization technique has been applied in most existing ABE schemes. This randomization method, however, also empowers a user the capability of regenerating a newly randomized decryption key over a subset of the attributes associated with the original decryption key. Because key randomization breaks the linkage between this newly generated key and the original key, a malicious user could leak the new decryption key to others without taking any responsibility for the key abuse. To solve this problem, we think of key regeneration-free ABE to disallow a user from randomizing his/her decryption key in any manner, i.e., a user can only delegate his/her decryption key in exactly the same form without any modification so that any abused or pirated key can be traced back to its original owner. Motivated by strongly unforgeable signature, we first define a security notion called strong key unforgeability, and show that ABE schemes equipped with the strong key unforgeability are immune to key regeneration. We then provide a generic transformation to convert ciphertext-policy ABE (CP-ABE) schemes of certain type to key regeneration-free CP-ABE schemes, and show how the transformation works by presenting two concrete constructions.

1. Introduction

Cloud computing allows data owners with limited resources to outsource data storage and processing to the cloud where massive storage and computational capacities are available. Since cloud service providers are usually commercial enterprises which are in different trust domains from data owners, data security and privacy in the cloud has received growing attention in both the academic and the public. A promising solution for data protection in the untrusted cloud server is using encryption, i.e., data owners encrypt the data and upload the ciphertexts to the cloud.

[*] Corresponding author.
E-mail address: hui.cui@murdoch.edu.au (H. Cui).

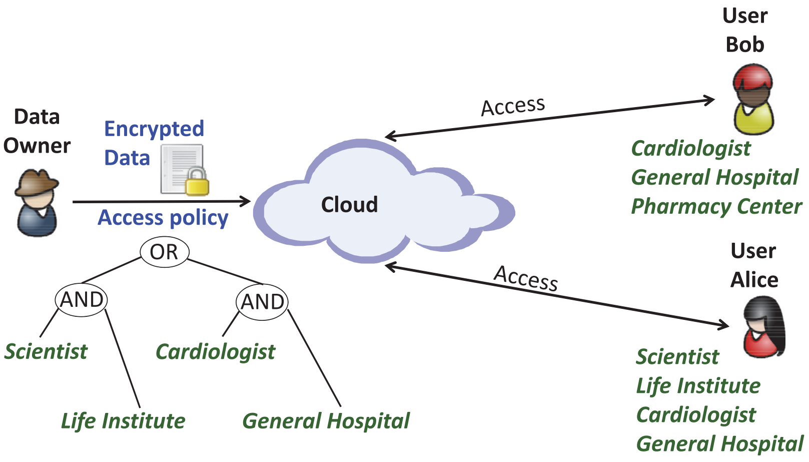

Fig. 1. A system architecture for accessing EMRs using CP-ABE.

In a traditional public-key encryption scheme, encrypted data is targeted for decryption by a single user, so they lack the expressiveness needed for the advanced data sharing. Attribute-based encryption (ABE) [1] is a one-to-many public-key encryption technique with the ability to perform data encryption and decryption defined over some descriptive attributes. It has two complementary forms [2]: Ciphertext-policy ABE (CP-ABE) where attributes (or credentials) are associated with user's decryption keys and access policies (or access structures) over these attributes are attached to ciphertexts, and key-policy ABE (KP-ABE) where attributes are used to annotate ciphertexts and access policies over these attributes are ascribed to users' decryption keys. In both CP-ABE and KP-ABE, decryption is possible if and only if the attributes satisfy the corresponding access structure. While it is possible to transform one type into the other, CP-ABE appears more aligned with the setting of access control to the encrypted data in the cloud where data owners directly specify access policies under which the encrypted data can be decrypted. In reality, users' access privileges are often granted based on their functional roles in an organization, where a role can be specified by a set of attributes. Concerning this fact, CP-ABE enables access control over the encrypted data with respect to the functional roles, rather than the usual concept of individuals inherent in the normal public-key encryption. An illustrative system architecture for accessing the encrypted electronic medical records (EMRs) using CP-ABE is shown in Fig. 1, in which a data owner encrypts an EMR using the encryption algorithm of a CP-ABE scheme, and then uploads the ciphertext together with the access policy to the cloud. The access policy requires that only "Scientist" at "Life Institute" or "Cardiologist" at "General Hospital" can decrypt the ciphertext and get access to the EMR. During the rest of the paper, we will focus on CP-ABE systems unless otherwise stated explicitly.

In a CP-ABE scheme, the decryption keys are defined over a set of attributes shared by multiple users to implement efficient one-to-many encryption, i.e., a data owner can decide who are privileged recipients by encrypting a piece of data with an access policy over attributes rather than exactly specifying each individual receiver. The key challenge in constructing an ABE scheme is to make the decryption keys collusion resistant such that if multiple users collude with each other, they are not able to decrypt a ciphertext if none of them could decrypt this ciphertext using their own decryption keys. In prior ABE schemes, this problem has been addressed by the key randomization technique which binds various components (each corresponding to a different attribute) of a user's decryption key using the randomness. Key randomization, however, also equips a user with the capability of key regeneration such that a user possessing a decryption key over an attribute set $\mathbf{A}$ is able to create a new decryption key for an attribute set $\mathbf{A}' \subseteq \mathbf{A}$. Users with ill intentions might be reluctant to share their original decryption keys with others, but they would be fearless to hand out piratical decryption keys different from their original keys, since re-randomization during key regeneration breaks the linkage between the new and the original decryption keys. Take the system in Fig. 1 as an instance, in which a user Alice has a decryption key corresponding to her attribute set $\mathbf{A} = \{$General Hospital, Cardiologist, Life Institute, Scientist$\}$. Using the key randomization, Alice could regenerate a new decryption key for the attribute subset $\mathbf{A}' = \{$Life Institute, Scientist$\}$ and sell this key to an insurance company for financial benefits without being traced. To overcome this inherent drawback of the key regeneration in ABE schemes, security notions such as traceability [3,4] and accountability [5,6] have been introduced, where a user's identification information is bound to the attribute-based decryption key such that the user's identity will be exposed from an input decryption key. This approach efficiently deters malicious users from sharing their decryption privileges, because they will confront the risk of being caught when committing adversarial key delegations. However, to the best of our knowledge, there are two limitations in prior constructions (e.g., [3–9]). First, they are built on the basis of some specific schemes and second, most of them conquer the issue of key abuse or regeneration in ABE at the price of the user privacy, since a user's identity or private information will be revealed in order to make a user accountable. In other words, on behalf of the individual privacy, the existing approach of using the identifier de facto deters users from regenerating their decryption keys. With these two limitations in mind, we ask the question: is it possible to provide an efficient and generic methodology to construct

ABE schemes in which key regeneration is computationally infeasible? In this paper, we give an affirmative answer to this question.

1.1. Our contributions

The security requirement of indistinguishability in CP-ABE implies that an adversary, who is given a series of decryption keys associated with a range of attribute sets of its choice, is not able to produce a new decryption key for an attribute set matching some access structure which cannot be satisfied by those previously queried sets of attributes. We refer to this property of CP-ABE as *key unforgeability*. For any standard CP-ABE scheme that is indistinguishably secure, it is easy to conclude that this scheme is endowed with the property of key unforgeability. However, such a scheme enables a user possessing a decryption key over an attribute set $\mathbf{A}$ to regenerate a newly randomized decryption key over a subset of attributes $\mathbf{A}$, which could be abused by users to gain financial profits from selling their keys.

To solve the problem, in this paper, we come up with *key regeneration-free CP-ABE*. Different from CP-ABE, key regeneration-free CP-ABE disallows a user to randomize or regenerate his/her decryption key in any manner. In key regeneration-free CP-ABE, key delegation is immutable and all-or-nothing. That is, a user can only delegate the decryption key in exactly the original form without any modification. From a technical point of view, we regard the decryption key generation algorithm run by an attribute authority (AA) as a signature scheme without verification, and view a decryption key as a signature signed by the AA over a set of attributes. Thus, the key regeneration-freeness in CP-ABE, which we call *strong key unforgeability*, resembles the notion of strong unforgeability for a signature scheme. Based on this observation and motivated by the technique of constructing strongly unforgeable signature schemes[1] in [11], we detail how to build efficient key regeneration-free CP-ABE schemes. Specifically, our contributions in this paper are threefold. First, we explore and define security properties that a key regeneration-free CP-ABE scheme must satisfy. Then, we introduce a generic construction to convert standard CP-ABE schemes into key regeneration-free CP-ABE schemes. Finally, we apply this generic approach to construct two concrete key regeneration-free CP-ABE schemes supporting expressive access structures. We also show that our construction is very efficient and can be applied to support CP-ABE with white-box traceability [3] without incurring much additional computational costs.

1.2. Related work

Attribute-based encryption. Sahai and Waters [1] first introduced the notion of attribute-based encryption (ABE), and then Goyal et al. formulated KP-ABE and CP-ABE as two complimentary forms of ABE [2]. The first KP-ABE construction given in [2] realized the monotonic access structures for key policies, and the first KP-ABE system that supports the expression of non-monotone formulas was presented in [12] to enable more viable access policies. Bethencourt, Sahai and Waters [13] proposed the first CP-ABE construction secure under the generic group model. Cheung and Newport [14] presented a CP-ABE scheme that is proved to be secure under the standard model, but it only supports the AND access structures. A CP-ABE system under more advanced access structures was proposed by Goyal et al. [15] based on the number theoretic assumption. The problem of key regeneration in ABE is also known as key cloning [7], malicious key delegation [3,9] or key abuse [5]. To deal with this problem, various ABE schemes such as ABE with key cloning protection [7], accountable ABE [5,6], traceable ABE [3,4,8] and key regeneration free ABE [9] have been put forward with concrete constructions. Specifically, the KP-ABE scheme in [4] achieves black-box traceability, the CP-ABE schemes in [5–7,9] only support ciphertext-policies using AND gate, the CP-ABE in [3] supports any monotone access structure but it is secure over a composite-order group, and the CP-ABE scheme in [8] is over the prime-order group but is built on the basis of [16] rather than a general construction. Notice that the CP-ABE scheme given in [9] also solved the key abuse problem by disabling the key regeneration capability, but the proposed scheme in [9] is a concrete construction and only supports access policies expressed in AND gates, while the construction given in the paper is a generic one.

Strong unforgeability. Strong unforgeability [17] is a useful notion for building chosen-ciphertext secure encryption systems [18] as well as group signatures [19]. In the random oracle model, schemes based on the full domain hash [20–22] and several other methods [20,23,24] are strongly unforgeable. Also, there are several constructions [25–28] which are strongly unforgeable without random oracles but depend on relatively strong assumptions. Tree-based signatures [29–34] can be proven strongly unforgeable without random oracles and based on standard assumptions, but they are less efficient than the strongly unforgeable scheme in [11], which is a generic transformation without random oracles based on the standard computational Diffie–Hellman problem in bilinear groups.

1.3. Organization

The remainder of this paper is organized as follows. In Section 2, we briefly review some of the notions and definitions to be used in the paper. In Section 3, after presenting the system framework, we define the security model of key

[1] It may be suggested to append a strongly unforgeable signature on the attribute-based decryption key to obtain key regeneration-free ABE, but signature schemes in the standard model still have their price [10] in the performance.

regeneration-free CP-ABE in terms of ciphertext indistinguishability and strong key unforgeability. In Section 4, we present a generic approach to transform a standard CP-ABE scheme to a key regeneration-free CP-ABE scheme and formally prove its security. We give two instantiations of key regeneration-free CP-ABE in Section 5, and compare them with other existing constructions in Section 6. We conclude the paper in Section 7.

2. Preliminaries

In this section, we review some basic cryptographic notions and definitions that are to be used in this paper.

2.1. Bilinear pairings and complexity assumptions

Let G be a group of a prime order p with a generator g. We define $\hat{e} : G \times G \rightarrow G_1$ to be a bilinear map if it has the following properties [35]:

1. Bilinear: for all $g \in G$, and $a, b \in Z_p$, we have $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$.
2. Non-degenerate: $\hat{e}(g, g) \neq 1$.

We say that G is a bilinear group if the group operation in G is efficiently computable and there exists a group G_1 and an efficiently computable bilinear map $\hat{e} : G \times G \rightarrow G_1$ as above.

Computational DL. The computational discrete log (DL) problem is that for any probabilistic polynomial-time (PPT) algorithm, it is difficult to compute a given (g, g^a), where $g \in G$, $a \in Z_p$ are chosen independently and uniformly at random.

Decisional parallel BDHE. The decisional q-parallel bilinear Diffie–Hellman exponent (BDHE) problem is that for any PPT algorithm, given $\overrightarrow{y} =$

$$\begin{aligned} & g, g^s, g^a, \ldots, g^{a^q}, g^{a^{q+2}}, \ldots, g^{a^{2q}}, \\ & \forall\ 1 \leq j \leq q \quad g^{s \cdot b_j}, g^{a/b_j}, \ldots, g^{a_q/b_j}, g^{a^{q+2}/b_j}, \ldots, g^{a^{2q}/b_j}, \\ & \forall\ 1 \leq j, k \leq q, k \neq j \quad g^{a \cdot s \cdot b_k/b_j}, \ldots, g^{a^q \cdot s \cdot b_k/b_j}, \end{aligned}$$

it is difficult to distinguish $(\overrightarrow{y}, \hat{e}(g, g)^{a^{q+1}s})$ from $(\overrightarrow{y}, Z)$, where $g \in G$, $Z \in G_1$, a, s, $b_1, \ldots, b_q \in Z_p$ are chosen independently and uniformly at random.

There is also a variant of the decisional q-parallel BDHE assumption, which is called the $(q-1)$ assumption. The $(q-1)$ problem is that for any PPT algorithm, given $\overrightarrow{y} =$

$$\begin{aligned} & g, g^s, \\ & g^{a^i}, g^{b_j}, g^{s \cdot b_j}, g^{a^i b_j}, g^{a^i/b_j^2} \quad \forall\ (i, j) \in [q, q], \\ & g^{a^i/b_j} \quad \forall\ (i, j) \in [2q, q] \text{ with } i \neq q+1, \\ & g^{a^i b_j/b_{j'}^2} \quad \forall\ (i, j, j') \in [2q, q, q] \text{ with } j \neq j', \\ & g^{sa^i b_j/b_{j'}}, g^{sa^i b_j/b_{j'}^2} \quad \forall\ (i, j, j') \in [q, q, q] \text{ with } j \neq j', \end{aligned}$$

it is difficult to distinguish $(\overrightarrow{y}, \hat{e}(g, g)^{a^{q+1}s})$ from $(\overrightarrow{y}, Z)$, where $g \in G$, $Z \in G_1$, a, s, $b_1, \ldots, b_q \in Z_p$ are chosen independently and uniformly at random.

2.2. Collision-resistant hash functions

Let $\{H_k\}$ be a family of keyed hash functions H_k: $\{0, 1\}^* \rightarrow \{0, 1\}^n$ indexed by $k \in \mathcal{K}$ (where $\mathcal{K}$ is the key space) that can map data of arbitrary length to data of a fixed length n. A family of collision resistant hash functions is a set of hash functions with the following properties [36]:

1. There is a PPT algorithm, which on input a security parameter, uniformly and randomly selects a member of the family with the given value attached.
2. All functions in the family are computable in polynomial time.
3. The problem of finding $x \neq y$ such that $H_k(x) = H_k(y)$ for a randomly chosen H_k in the family is computationally impossible to solve.

2.3. Access structures and linear secret sharing

We review the notions of access structures and linear secret sharing schemes in [37,38] as follows.

Definition 1 (Access structure). Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, \ldots, P_n\}}$ is monotone if $\forall B$, C: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \subseteq \mathbb{A}$. An access structure (respectively, monotone access structure) $\mathbb{A}$ is a collection (respectively, monotone collection) of non-empty subsets of $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, \ldots, P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

Definition 2 (Linear secret sharing schemes). Let P be a set of parties. Let $\mathbb{M}$ be a matrix of size $l \times n$. Let ρ: $\{1, \ldots, l\} \rightarrow P$ be a function that maps a row to a party for labeling. A secret sharing scheme Π over a set of parties P is a linear secret-sharing scheme (LSSS) over Z_p if

1. The shares for each party form a vector over Z_p.
2. There exists a matrix $\mathbb{M}$ which has l rows and n columns called the share-generating matrix for Π. For $x = 1, \ldots, l$, the x-th row of the matrix $\mathbb{M}$ is labeled by a party $\rho(x)$, where ρ: $\{1, \ldots, l\} \rightarrow P$ is a function that maps a row to a party for labeling. Considering that the column vector $v = (\mu, r_2, \ldots, r_n)$, where $\mu \in Z_p$ is the secret to be shared and $r_2, \ldots, r_n \in Z_p$ are randomly chosen, then $\mathbb{M}v$ is the vector of l shares of the secret μ according to Π. The share $(\mathbb{M}v)_x$ belongs to party $\rho(x)$.

It has been noted in [37] that every LSSS also enjoys the linear reconstruction property. Suppose that Π is an LSSS for an access structure $\mathbb{A}$. Let $\mathbf{A}$ be an authorized set, and define $I \subseteq \{1, \ldots, l\}$ as $I = \{i | \rho(i) \in \mathbf{A}\}$. Then the vector $(1, 0, \ldots, 0)$ is in the span of rows of matrix $\mathbb{M}$ indexed by I, and there exist constants $\{w_i \in Z_p\}_{i \in I}$ such that, for any valid shares $\{v_i\}$ of a secret μ according to Π, we have $\Sigma_{i \in I} w_i v_i = \mu$. These constants $\{w_i\}$ can be found in the polynomial time with respect to the size of the share-generating matrix $\mathbb{M}$ [39].

Boolean formulas [37]. Access policies can also be described in terms of monotonic boolean formulas. LSSS access structures are more general, and can be derived from representations as boolean formulas. There are standard techniques to convert any monotonic boolean formula into a corresponding LSSS matrix. A boolean formula can be represented as an access tree, where the interior nodes are AND and OR gates, and the leaf nodes correspond to attributes. The number of rows in the corresponding LSSS matrix is the same as the number of leaf nodes in the access tree.

3. Framework and security model

In this section, we first present a framework for key regeneration-free CP-ABE and then formally define its security.

3.1. Framework

At a high level, a key regeneration-free CP-ABE scheme works the same as a standard CP-ABE scheme, and consists of a setup algorithm Setup, a decryption key generation algorithm KeyGen, an encryption algorithm Encrypt and a decryption algorithm Decrypt.

- Setup$(1^\lambda) \rightarrow (pars, msk)$. Taking a security parameter λ as the input, this setup algorithm outputs a public parameter *pars* and a master private key *msk* for the system.
- KeyGen$(pars, msk, \mathbf{A}) \rightarrow sk_{\mathbf{A}}$. Taking the public parameter *pars*, the master private key *msk* and an attribute set $\mathbf{A}$ as the input, this decryption key generation algorithm generates the attribute-based decryption key $sk_{\mathbf{A}}$.
- Encrypt$(pars, M, \mathbb{A}) \rightarrow C$. Taking the public parameter *pars*, a message M and an access structure $\mathbb{A}$ over the universe of attributes as the input, this encryption algorithm outputs a ciphertext C.
- Decrypt$(pars, C, \mathbf{A}, sk_{\mathbf{A}}) \rightarrow M/\perp$. Taking the public parameter *pars*, a ciphertext C with an access structure $\mathbb{A}$ and a decryption key $sk_{\mathbf{A}}$ corresponding to an attribute set $\mathbf{A}$ as the input, this decryption algorithm outputs either the message M when the attribute set $\mathbf{A}$ satisfies the access structure $\mathbb{A}$ of the ciphertext C, or a symbol $\perp$ indicating the failure of the decryption.

We require that a key regeneration-free CP-ABE scheme is correct, meaning that the decryption algorithm correctly decrypts a ciphertext having an access structure $\mathbb{A}$ with a decryption key over $\mathbf{A}$, only if $\mathbf{A}$ is an authorized set of $\mathbb{A}$. Formally, for all messages M, and all attribute sets $\mathbf{A}$ and access structures $\mathbb{A}$ with authorized $\mathbf{A}$ satisfying $\mathbb{A}$, if $(pars, msk) \leftarrow$ Setup(1^λ), $sk_{\mathbf{A}} \leftarrow$ KeyGen$(pars, msk, \mathbf{A})$, $C \leftarrow$ Encrypt$(pars, M, \mathbb{A})$, then Decrypt$(pars, C, \mathbf{A}, sk_{\mathbf{A}}) = M$.

Notice that with respect to a concrete construction, the last input of the encryption algorithm Encrypt will be set to be $(\mathbb{M}, \rho)$ rather than $\mathbb{A}$.

3.2. Security definitions

Below we present the security definition for a key regeneration-free CP-ABE scheme in terms of two security games: the traditional security game for ciphertext indistinguishability and an additional security game for strong key unforgeability. The ciphertext indistinguishability requires the adversary to be unable to distinguish a ciphertext intended for one message from a ciphertext intended for another message, while the strong key unforgeability prevents the adversary from producing a new decryption key for any previously queried set of attributes.

The ciphertext indistinguishability is described by a security game between a challenger algorithm $\mathcal{C}$ and an adversary algorithm $\mathcal{A}$ under chosen-plaintext attacks, where algorithm $\mathcal{A}$ will be challenged on an encryption to an access structure $\mathbb{A}^*$, and can issue decryption key queries on any set of attributes $\mathbf{A}$ such that $\mathbf{A}$ does not satisfy $\mathbb{A}^*$.

- Setup. Algorithm $\mathcal{C}$ runs the Setup algorithm, and gives the public parameter *pars* to algorithm $\mathcal{A}$.
- Phase 1. Algorithm $\mathcal{A}$ issues decryption key queries for the sets of attributes $\mathbf{A}_1, \ldots, \mathbf{A}_{q_1}$. For each attribute set $\mathbf{A}_i$, algorithm $\mathcal{C}$ runs the KeyGen algorithm, and sends $sk_{\mathbf{A}_i}$ to algorithm $\mathcal{A}$.

- Challenge. Algorithm $\mathcal{A}$ sends an access structure $\mathbb{A}^*$ and two messages M_0, M_1 of equal length to algorithm $\mathcal{C}$, with the restriction that none of the sets $\mathbf{A}_1, \ldots, \mathbf{A}_{q_1}$ from Phase 1 satisfy the access structure $\mathbb{A}^*$. Algorithm $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$, and encrypts M_b under $\mathbb{A}^*$. The challenge ciphertext C^* is given to algorithm $\mathcal{A}$.
- Phase 2. Algorithm $\mathcal{A}$ continues issuing decryption key queries as in Phase 1 with the restriction that none of the sets of attributes $\mathbf{A}_{q_1+1}, \ldots, \mathbf{A}_q$ satisfy the access structure $\mathbb{A}^*$ in the challenge phase.
- Guess. Algorithm $\mathcal{A}$ outputs a guess b' of b.

A key regeneration-free CP-ABE scheme is ciphertext indistinguishable if all PPT adversaries have at most a negligible advantage in the security parameter λ in the above game, where the advantage of algorithm $\mathcal{A}$ is defined as $\Pr[b' = b] - 1/2$.

This model can be easily extended to deal with chosen-ciphertext attacks by allowing decryption queries in Phase 1 and Phase 2. Also, a key regeneration-free CP-ABE system is said to be selectively indistinguishable if an Initialization phase is added before the Setup phase where algorithm $\mathcal{A}$ commits a challenge access structure $\mathbb{A}^*$ which it aims to attack.

The ciphertext indistinguishability defined above implies that an adversary, who is given decryption keys corresponding to a series of attribute sets $\{\mathbf{A}_1, \ldots, \mathbf{A}_q\}$ of its choice, is not able to produce a decryption key for a new set of attributes $\mathbf{A}'$ satisfying an access structure which is not satisfied by any one of the previous keys. We call this property as key unforgeability.

Next we define strong key unforgeability, which is described as a game between a challenger algorithm $\mathcal{C}$ and an adversary algorithm $\mathcal{A}$ under chosen attribute set attacks, where algorithm $\mathcal{A}$ is allowed to issue decryption key queries on any set of attributes $\mathbf{A}$, and is challenged to output a new decryption key for a set of attributes $\mathbf{A}'$ which is a subset of $\mathbf{A}$. Suppose that algorithm $\mathcal{A}$ obtains a pair ($\mathbf{A}$, $sk_{\mathbf{A}}$) along with other attribute set and decryption key pairs of its choice. Algorithm $\mathcal{A}$ wins the game if it is able to create a new decryption key for a subset of attributes $\mathbf{A}$ such that $sk'_{\mathbf{A}} \not\subseteq sk_{\mathbf{A}}$. In other words, assuming that algorithm $\mathcal{A}$ obtains a pair ($\mathbf{A}$, $sk_{\mathbf{A}}$) along with other attribute set and decryption key pairs of its choice, then the system is strongly unforgeable if the adversary cannot create a new decryption key $sk'_{\mathbf{A}}$ for a subset of $\mathbf{A}$.

- Setup. Algorithm $\mathcal{C}$ runs the Setup algorithm, and gives the public parameter *pars* to algorithm $\mathcal{A}$.
- Key queries. Algorithm $\mathcal{A}$ issues decryption key queries for sets of attributes $\mathbf{A}_1, \ldots, \mathbf{A}_q$. For each attribute set $\mathbf{A}_i$, algorithm $\mathcal{C}$ runs the KeyGen algorithm, and sends $sk_{\mathbf{A}_i}$ to algorithm $\mathcal{A}$. These queries could be asked adaptively so that each query on $\mathbf{A}_i$ may depend on the responds to $\mathbf{A}_1, \ldots, \mathbf{A}_{i-1}$.
- Output. Algorithm $\mathcal{A}$ outputs a pair ($\mathbf{A}$, $sk_{\mathbf{A}}$). Algorithm $\mathcal{A}$ wins if $sk_{\mathbf{A}}$ is a valid decryption key over $\mathbf{A}$ according to the decryption algorithm, i.e., Decrypt(*pars*, C, $\mathbf{A}$, $sk_{\mathbf{A}}$) $\rightarrow$ M for C being any encryption of M with $\mathbf{A}$ satisfying the access structure attached to C, and ($\mathbf{A}$, $sk_{\mathbf{A}}$) is not among the pairs ($\mathbf{A}_i$, $sk_{\mathbf{A}_i}$) generated during the query phase or $\mathbf{A}$ is a subset of $\mathbf{A}_i$ for $i \in [1, q]$.

We define the advantage of algorithm $\mathcal{A}$ in attacking the strong key unforgeability of a key regeneration-free CP-ABE scheme as the probability that algorithm $\mathcal{A}$ wins the above game, taken over the random bits of algorithm $\mathcal{C}$ and algorithm $\mathcal{A}$. A key regeneration-free CP-ABE scheme is strongly key unforgeable under adaptive chosen attribute set attacks if all PPT adversaries have at most a negligible advantage in the security parameter λ in the above game.

4. From CP-ABE to key regeneration-free CP-ABE

In this section, we present a general transformation that converts a standard CP-ABE scheme into a key regeneration-free CP-ABE scheme.

4.1. Generic construction

In a standard CP-ABE scheme, a user's decryption key $sk_{\mathbf{A}}$ over an attribute set $\mathbf{A}$ consists of multiple components (each corresponding to an attribute in $\mathbf{A}$). In order to prevent multiple users from combining their key components to form a decryption key beyond their respective decryption capabilities, the key randomization is employed to securely bind the components of a user's decryption key together; however, given a key $sk_{\mathbf{A}}$, the user can regenerate a new key $sk'_{\mathbf{A}}$ over an attribute set $\mathbf{A}'$ which is a subset of $\mathbf{A}$ by re-randomizing $sk_{\mathbf{A}}$. As alluded to in Section 1, since the new key $sk'_{\mathbf{A}}$ and the original key $sk_{\mathbf{A}}$ are statistically uncorrelated, the user could misuse the new key, e.g., sell it in a black market without being held accountable.

In key regeneration-free CP-ABE, we aim to avoid the above key regeneration problem by averting the gadget of key randomization in the key generation algorithm of CP-ABE. Our solution is to twist the key generation algorithm such that the decryption key is extracted over the attribute set, as well as the chosen randomness. Thus, when performing re-randomization, the user has to generate the decryption key over an attribute set which has an element relating to the new randomness (this new element does not belong to the original attribute set). According to the security of the original key generation algorithm, such a kind of key re-randomization will fail.

Let R be a random space and $\mathcal{L}$ be the label associated with an attribute set $\mathbf{A}$. In order to convert a standard CP-ABE scheme into a key regeneration-free CP-ABE, we require the key generation algorithm and the decryption algorithm in the standard CP-ABE scheme to have the following properties.

- Property 1. The key generation algorithm can be partitioned into two algorithms F_1 and F_2 so that a decryption key over a set of attributes $\mathbf{A}$ and a label $\mathcal{L}$ under a master private key msk is computed as follows.
  1. Randomly select $r \leftarrow R$.
  2. Set $sk_1 \leftarrow F_1(\mathbf{A}, r, msk \circ \mathcal{L})$ and $sk_2 \leftarrow F_2(r, msk)$.
  3. Output the decryption key $sk_{\mathbf{A} \circ \mathcal{L}} \leftarrow (sk_1, sk_2)$.
- Property 2. Given $\mathbf{A}$, $\mathcal{L}$ and sk_2, there is at most one sk_1 such that (sk_1, sk_2) is a valid decryption key corresponding to $\mathbf{A}$ and the public parameter $pars$.
- Property 3. The decryption algorithm can successfully decrypt a ciphertext under a decryption key $sk_{\mathbf{A} \circ \mathcal{L}}$ as long as the set of attributes $\mathbf{A}$ of the decryption key $sk_{\mathbf{A} \circ \mathcal{L}}$ satisfies the access structure of the ciphertext.[2]

Notice that sk_2 does not depend on the set of attributes $\mathbf{A}$ and the label $\mathcal{L}$. Moreover, given $\mathbf{A}$, $\mathcal{L}$ and sk_2, the decryption key is fully determined.

Now we are ready to describe the transformation from a standard CP-ABE scheme into a key regeneration-free CP-ABE scheme in terms of ABE systems under the prime-order groups, since there exist techniques [41] to convert pairing based schemes from composite-order groups to prime-order groups. Let G be a group of a prime order p. Let $H = \{H_k\}$ be a family of collision-resistant hash functions where H_k: $\{0, 1\}^* \rightarrow \{0, 1\}^n$ indexed by $k \in \mathcal{K}$ with $\mathcal{K}$ being the key space. Assume that $p \geq 2^n$ so that hash outputs can be viewed as elements of Z_p, and each element of Z_p has a unique encoding. Denote $x||y$ as the concatenation of two strings x and y. For brevity, we use $\mathbf{A}$ to denote the concatenation of all elements in $\mathbf{A}$.

Let $\mathcal{ABE}$ = (Setup, KeyGen, Encrypt, Decrypt) be a CP-ABE scheme over a prime-order group, where KeyGen is partitioned using functions F_1 and F_2. We build a new CP-ABE scheme $\mathcal{ABE}_{\text{new}} = (\text{Setup}_{\text{new}}, \text{KeyGen}_{\text{new}}, \text{Encrypt}_{\text{new}}, \text{Decrypt}_{\text{new}})$ via the following transformation. Notice that this transformation is simple and efficient. In particular, a user's decryption key in the new scheme has the same size as the key in the original scheme plus a short string.

- $\text{Setup}_{\text{new}}(1^\lambda) \rightarrow (par', msk')$. Randomly choose generators $g, h \in G$, a hash key $k \in \mathcal{K}$. Run Setup to obtain the public parameter $pars$ and the master private key msk. Output the public parameter $pars' = (pars, g, h, k)$, and the master private key $msk' = msk$ for the new scheme.
- $\text{KeyGen}_{\text{new}}(pars', msk', \mathbf{A}) \rightarrow sk_{\mathbf{A}}$. A user's decryption key over a set of attributes $\mathbf{A} = \{a_1, \ldots, a_l\}$ is generated as follows.
  1. Randomly choose $r \in R$, and set $sk_2 \leftarrow F_2(r, msk)$.
  2. Compute $t \leftarrow H_k(\mathbf{A}||sk_2)$ and view t as an element of Z_p.
  3. Randomly choose $s \in Z_p$, and compute $\mathcal{L} \leftarrow g^t h^s$.
  4. Compute $sk_1 \leftarrow F_1(\mathbf{A}, r, msk \circ \mathcal{L})$, and output the decryption key $sk_{\mathbf{A}} \leftarrow (sk_1, sk_2, s)$, where "$\circ$" is an operation depending on the specific key generation algorithm in the original CP-ABE scheme.
- $\text{Encrypt}_{\text{new}}(pars', M, \mathbb{A}) \rightarrow C$. Run Encrypt to produce a ciphertext C.
- $\text{Decrypt}_{\text{new}}(pars', C, \mathbf{A}, sk_{\mathbf{A}}) \rightarrow M/\perp$. Compute $\mathcal{L}$ from $sk_{\mathbf{A}}$, and run Decrypt with the additional $\mathcal{L}$ to output a message M or a failure symbol $\perp$.

To conclude, our approach to achieve strong key unforgeability in CP-ABE is as follows. Firstly, an intermediate value $\mathcal{L}$ is derived in Step 3, and then the key component on the attribute set $\mathbf{A}$ is extracted by the underlying key generation algorithm in Step 4. Since $\mathcal{L}$ is derived from the attribute set $\mathbf{A}$ and sk_2 with sk_2 being derived from the randomness r, the decryption key is essentially generated over the attribute set $\mathbf{A}$ and the chosen randomness r. Thus, the user cannot directly "re-randomize" the decryption key. This may suggest that the resulting system is strongly key unforgeable. However, creating this circularity [11] makes the key extraction on $\mathcal{L}$ and $\mathbf{A}$ dependent on the randomness r, breaking the proof of security for the underlying key generation algorithm of ABE. To overcome this challenge, an additional hashing step, Step 3, is introduced where a chameleon hash [42] is used to hash again. The extra randomness s of the chameleon hash breaks the circularity in the proof of security [11], and makes it possible to prove the security based on the key unforgeability of the underlying algorithm in ABE. It is worthy noting that the randomness of the chameleon hash is indispensable to respond the key generation queries from a Type 3 adversary in the proof of the security, which is presented in the section below.

4.2. Proof of strong key unforgeability

Theorem 1. *The $\mathcal{ABE}_{\text{new}}$ constructed above is a secure key regeneration-free CP-ABE scheme in the sense that it is strongly key unforgeable assuming that the underlying ABE scheme $\mathcal{ABE}$ is key unforgeable, the discrete log assumption holds in G, and H_k is collision resistant.*

Proof. Suppose that algorithm $\mathcal{A}$ is an adversary that is given the public parameter $pars' = (pars, g, h, k)$ and breaks the strong key unforgeability of $\mathcal{ABE}_{\text{new}}$.

Algorithm $\mathcal{A}$ issues decryption key queries on attribute sets $\mathbf{A}_1, \ldots, \mathbf{A}_q$ and is given the corresponding keys $sk_{\mathbf{A}_i} = (sk_{i,1}, sk_{i,2}, s_i)$ for $i = 1, \ldots, q$. Let $t_i = H_k(\mathbf{A}_i||sk_{i,2})$, $\mathcal{L}_i = g^{t_i} h^{s_i}$ for $i = 1, \ldots, q$. Let $(\mathbf{A}', sk_{\mathbf{A}'} = (sk'_1, sk'_2, s')$ be the forged key produced by algorithm $\mathcal{A}$, with $t' = H_k(\mathbf{A}'||sk'_2)$ and $\mathcal{L}' = g^{t'} h^{s'}$. We consider the following three types of key forgeries.

[2] Please refer to the Instantiations for more details about this property. In [40], the similar property is applied the design of the scheme.

- Type 1. A key forgery where $\mathcal{L}' = \mathcal{L}_i$ and $t' = t_i$ for some $i \in \{1, \ldots, q\}$.
- Type 2. A key forgery where $\mathcal{L}' = \mathcal{L}_i$ and $t' \neq t_i$ for some $i \in \{1, \ldots, q\}$.
- Type 3. Any other key forgery $\mathcal{L}' \neq \mathcal{L}_i$ for all $i \in \{1, \ldots, q\}$.

If algorithm $\mathcal{A}$ is successful, it must output a key forgery of Type 1, Type 2, or Type 3. We show that a Type 1 forgery can be used to break the collision-resistance of H_k, a Type 2 forgery can be used to solve the discrete log problem in G, and a Type 3 forgery can be used to break the key unforgeability of the underlying CP-ABE scheme $\mathcal{ABE}$. Assume that a simulator algorithm can flip a coin at the beginning of the simulation to guess which type of forgery algorithm $\mathcal{A}$ will yield and set up the simulation environment appropriately.

- Type 1 adversary. Suppose that algorithm $\mathcal{A}$ is an adversary of Type 1 that breaks the strong key unforgeability of $\mathcal{ABE}_{\text{new}}$. We construct an algorithm $\mathcal{B}$ that breaks the collision-resistance of H_k. Algorithm $\mathcal{B}$ is given a random key $k' \in \mathcal{K}$, and its goal is to output a pair of values (x_1, x_2) such that $x_1 \neq x_2$ and $H_{k'}(x_1) = H_{k'}(x_2)$. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ as follows.
  - Setup. Algorithm $\mathcal{B}$ sets $k \leftarrow k'$, and generates the public parameter and the master private key according to the Setup$_{\text{new}}$ algorithm. Algorithm $\mathcal{B}$ gives algorithm $\mathcal{A}$ the public parameter $pars' = (pars, g, h, k)$ and keeps the master private key msk'.
  - Key generation queries. Algorithm $\mathcal{A}$ issues up to q key generation queries. Algorithm $\mathcal{B}$ responds to a query on a set of attributes $\mathbf{A}_i$ by running the KeyGen$_{\text{new}}$ algorithm and returning the corresponding attribute-based decryption key $sk_{\mathbf{A}_i}$ to algorithm $\mathcal{A}$.
  - Output. Algorithm $\mathcal{A}$ outputs a key forgery $(\mathbf{A}', sk_{\mathbf{A}'}) = (sk'_1, sk'_2, s')$ such that $(\mathbf{A}', sk_{\mathbf{A}'}) \notin \{(\mathbf{A}_1, sk_{\mathbf{A}_1}), \ldots, (\mathbf{A}_q, sk_{\mathbf{A}_q})\}$ and $\mathbf{A}'$ is a subset of any $\mathbf{A}_i \in \{\mathbf{A}_1, \ldots, \mathbf{A}_q\}$, $\mathcal{L}' = \mathcal{L}_i$ and $t' = t_i$ for some $i \in \{1, \ldots, q\}$. Specifically, $t' = t_i$ means that $H_k(\mathbf{A}'||sk'_2) = H_k(\mathbf{A}_i||sk_{i,2})$, $\mathcal{L}' = \mathcal{L}_i$ means that $g^{t'}h^{s'} = g^{t_i}h^{s_i}$. Then algorithm $\mathcal{B}$ outputs the pair $(\mathbf{A}'||sk'_2, \mathbf{A}_i||sk_{i,2})$ as a collision on H_k.

  It is easy to see that when algorithm $\mathcal{A}$ produces a key forgery of Type 1, algorithm $\mathcal{B}$ succeeds in finding a collision in H_k. Since $H_k(\mathbf{A}'||sk'_2) = H_k(\mathbf{A}_i||sk_{i,2})$, it remains to show that $\mathbf{A}'||sk'_2 \neq \mathbf{A}_i||sk_{i,2}$.

  Consider a contradiction that $\mathbf{A}' = \mathbf{A}_i$ and $sk'_2 = sk_{i,2}$. Since $t' = t_i$ and $\mathcal{L}' = \mathcal{L}_i$, we have $s' = s_i$ (recall that any exponent $s \in Z_p$ has a unique encoding). Since $sk'_2 = sk_{i,2}$ and $\mathcal{L}' = \mathcal{L}_i$, the second property of partitioned key generation implies that $sk'_1 = sk_{i,1}$. Hence, we have $\mathbf{A}' = \mathbf{A}_i$ and $sk_{\mathbf{A}'} = sk_{\mathbf{A}_i}$, which contradicts the fact that $(\mathbf{A}', sk_{\mathbf{A}'})$ is a strong key forgery. Therefore, $\mathbf{A}'||sk'_2 \neq \mathbf{A}_i||sk_{i,2}$, implying that whenever algorithm $\mathcal{A}$ produces a key forgery of Type 1, algorithm $\mathcal{B}$ finds a collision in H_k.

- Type 2 adversary. Suppose that algorithm $\mathcal{A}$ is an adversary of Type 2 that breaks the strong key unforgeability of $\mathcal{ABE}_{\text{new}}$. We construct an algorithm $\mathcal{B}$ that solves the discrete log problem in G. Algorithm $\mathcal{B}$ is given a pair (g', h'), and its goal is to output a such that $h' = (g')^a$. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ as follows.
  - Setup. Algorithm $\mathcal{B}$ sets $g \leftarrow g'$, $h \leftarrow h'$, and generates the public parameter and the master private key according to the Setup$_{\text{new}}$ algorithm. Algorithm $\mathcal{B}$ gives algorithm $\mathcal{A}$ the public parameter $pars' = (pars, g, h, k)$ and keeps the master private key msk'.
  - Key generation queries. Algorithm $\mathcal{A}$ issues up to q key generation queries. Algorithm $\mathcal{B}$ responds to a query on a set of attributes $\mathbf{A}_i$ by running KeyGen$_{\text{new}}$ algorithm and returning the corresponding decryption key $sk_{\mathbf{A}_i}$ to algorithm $\mathcal{A}$.
  - Output. Algorithm $\mathcal{A}$ outputs a key forgery $(\mathbf{A}', sk_{\mathbf{A}'}) = (sk'_1, sk'_2, s')$ such that $(\mathbf{A}', sk_{\mathbf{A}'}) \notin \{(\mathbf{A}_1, sk_{\mathbf{A}_1}), \ldots, (\mathbf{A}_q, sk_{\mathbf{A}_q})\}$ and $\mathbf{A}'$ is a subset of any $\mathbf{A}_i \in \{\mathbf{A}_1, \ldots, \mathbf{A}_q\}$, $\mathcal{L}' = \mathcal{L}_i$ and $t' \neq t_i$ for some $i \in \{1, \ldots, q\}$. Since $\mathcal{L}' = \mathcal{L}_i$, we have

  $$g^{t'}h^{s'} = g^{t_i}h^{s_i} \Rightarrow g^{t'}(g^a)^{s'} = g^{t_i}(g^a)^{s_i} \Rightarrow a = \frac{t' - t_i}{s_i - s'}.$$

  Since $s' = s_i$ and $g^{t'}h^{s'} = g^{t_i}h^{s_i}$, implying $t' = t_i$, we have $s' - s_i \neq 0$. Therefore, algorithm $\mathcal{B}$ outputs a, solving the computational DL problem.

- Type 3 adversary. Suppose that algorithm $\mathcal{A}$ is an adversary of Type 3 that breaks the strong key unforgeability of $\mathcal{ABE}_{\text{new}}$. We construct an algorithm $\mathcal{B}$ that breaks the key unforgeability of $\mathcal{ABE}$. Algorithm $\mathcal{B}$ is given the public parameter $pars$, and its goal is to output a pair $(\mathbf{A}, sk_{\mathbf{A}})$ such that $sk_{\mathbf{A}}$ is a valid key for $\mathbf{A}$, and $\mathbf{A}$ is not among $\mathbf{A}_i$ for $i \in [1, q]$ which are attribute sets queried by algorithm $\mathcal{A}$ before and $\mathbf{A}$ is not a subset of $\mathbf{A}_i$ for $i \in [1, q]$. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ as follows.
  - Setup. Algorithm $\mathcal{B}$ generates the public parameter $pars'$.
    1. Chooses a random generator $g \in G$.
    2. Chooses a random exponent $a \in Z_p$, and sets $h \leftarrow g^a$.
    3. Chooses a random hash key $k \in \mathcal{K}$.
    4. Sends the public $pars' \leftarrow (pars, g, h, k)$ to algorithm $\mathcal{A}$.
  - Key generation queries. Algorithm $\mathcal{A}$ issues up to q key generation queries. Algorithm $\mathcal{B}$ responds to a query on a set of attributes $\mathbf{A}$ as follows.

1. Chooses a random exponent $w \in Z_p$, and sets $\mathcal{L} \leftarrow g^w$.
2. Sends a key generation query on $\mathbf{A}$ and $\mathcal{L}$ to its own challenger algorithm, and gets a decryption key (sk_1, sk_2) on $\mathbf{A}$ and $\mathcal{L}$.
3. Computes $t \leftarrow H_k(\mathbf{A}||sk_2)$, and sets $s \leftarrow (w-t)/a$.
4. Sends $sk_{\mathbf{A}} \leftarrow (sk_1, sk_2, s)$ to algorithm $\mathcal{A}$.

Observe that $\mathcal{L} = g^w = g^{as+t} = g^t h^s$, and s is uniform in Z_p, as required. As a result, $sk_{\mathbf{A}}$ is a valid decryption key on $\mathbf{A}$.

- Output. Finally, algorithm $\mathcal{A}$ outputs ($\mathbf{A}'$, (sk_1', sk_2', s')) as a key forgery. Algorithm $\mathcal{B}$ produces a key forgery on the underlying scheme $\mathcal{ABE}$ as follows.
  1. Computes $t' \leftarrow H_k(\mathbf{A}'||sk_2')$, and $\mathcal{L}' \leftarrow g^{t'}h^{s'}$.
  2. Outputs (($\mathbf{A}'$, $\mathcal{L}'$), (sk_1', sk_2')).

  It is not difficult to see that $\mathcal{L}' \notin \{\mathcal{L}_1, \ldots, \mathcal{L}_q\}$, because if $\mathcal{L}' = \mathcal{L}_i$ for some $i \in \{1, \ldots, q\}$, then either $t' = t_i$ (a key forgery of Type 1) or $t' \neq t_i$ (a key forgery of Type 2). Therefore, when algorithm $\mathcal{A}$ produces a key forgery of Type 3, algorithm $\mathcal{B}$ outputs a key forgery on a new attribute set $\mathbf{A}'$ (which is not a subset of any queried attribute set) for the underlying scheme $\mathcal{ABE}$.

In summary, we show that three types of key forgeries can be used to break collision-resistance of H_k, discrete log assumption in G and key unforgeability of the underlying ABE scheme $\mathcal{ABE}$, respectively. This completes the proof of Theorem 1. □

5. Concrete constructions

In this section, we present two concrete key regeneration-free CP-ABE schemes based on the generic transformation introduced in the previous section.

5.1. Instantiation 1

Below we apply our generic transformation technique to the most efficient CP-ABE system in [38] to obtain a key regeneration-free CP-ABE scheme. It is easy to see that the key generation algorithm in [38] can be partitioned into functions F_1 and F_2 such as

$$F_1(\mathbf{A}, r, msk \circ \mathcal{L}) = \left\{msk^{\mathcal{L}} \cdot (g^a)^r, \quad \forall i \in \mathbf{A} \ \ h_i^r\right\}, \qquad F_2(r, msk) = g^r,$$

where $g^a, h_1, \ldots, h_{|\mathbf{A}|}$ are elements of the public parameter. Given $\mathbf{A}$, $\mathcal{L}$ and $sk_2 = F_2(r, msk)$, there exists only one $sk_1 = F_1(\mathbf{A}, r, msk \circ \mathcal{L})$ which leads the success in the decryption, reflecting the second property of the partitioned key generation. In addition, the decryption algorithm can decrypt the ciphetext using the given decryption key (sk_1, sk_2) with a label $\mathcal{L}$ by sending $\mathcal{L}$ to the exponent (which will be shown in the concrete scheme). Notice that every element in G is assumed to have a unique encoding; otherwise an adversary can invalidate property 2 by changing the encoding of a group element [11].

Let G be a group of a prime order p with a generator g, $\hat{e}: G \times G \to G_1$ denote the bilinear map, and $\{H_k\}$ be a family of collision-resistant hash functions H_k: $\{0, 1\}^* \to \{0, 1\}^n$ indexed by $k \in \mathcal{K}$ with the hash outputs being viewed as elements of Z_p. We transform Waters CP-ABE scheme [38] as follows, in which some algorithms are the same as the original ones.

- Setup. This algorithm takes the security parameter λ as the input. It firstly chooses U random group elements $h_1, \ldots, h_U \in G$ that are associated with the U attributes in the system. Then, it chooses random exponents $\alpha, a \in Z_p$. The public parameter is $pars = \{g, \hat{e}(g, g)^\alpha, g^a, h_1, \ldots, h_U\}$, and the master private key is $msk = g^\alpha$. Finally, it randomly chooses a generator $h \in G$, a hash key $k \in \mathcal{K}$, and outputs the system parameter $pars' = (pars, h, k)$.
- KeyGen. This algorithm takes the system parameter $pars'$, the master private key msk and an attribute set $\mathbf{A}$ as the input. It randomly chooses $r, s \in Z_p$, and computes

  $$sk_2 = g^r, \quad t = H_k(\mathbf{A}||sk_2), \quad \mathcal{L} = H_k\left(g^t h^s\right),$$
  $$sk_1 = \left(sk_1' = (g^\alpha)^{\mathcal{L}}(g^a)^r, \ sk_1^{(i)} = h_i^r \ \ \forall i \in \mathbf{A}\right).$$

  It outputs the attribute-based decryption key $sk_{\mathbf{A}} = (sk_1, sk_2, s)$.
- Encrypt. The encryption algorithm takes the system parameter $pars'$, a message M and an LSSS access structure ($\mathbb{M}$, ρ) where $\mathbb{M}$ is an $l \times n$ matrix and the function ρ maps the rows of $\mathbb{M}$ to attributes as the input. It randomly chooses a vector $\overrightarrow{v} = (\mu, y_2, \ldots, y_n) \in Z_p^n$. These values will be used to share the encryption exponent μ. For $i = 1$ to l, it calculates $v_i = \overrightarrow{v} \cdot \mathbb{M}_i$, where $\mathbb{M}_i$ is the vector corresponding to the i-th row of $\mathbb{M}$. In addition, the algorithm chooses random $z_1, \ldots, z_l \in Z_p$, and computes

  $$C = M \cdot \hat{e}(g, g)^{\alpha\mu}, \quad C' = g^\mu, \quad \forall i \in \mathbf{A} \ \ C_i = g^{av_i} h_{\rho(i)}^{-z_i}, \ \ D_i = g^{z_i}.$$

  It outputs the ciphertext CT = (($\mathbb{M}$, ρ), C, C', $\{C_i, D_i\}_{i \in [1,l]}$).

- Decrypt. This algorithm takes the system parameter $pars'$, a ciphertext CT with an access structure $(\mathbb{M}, \rho)$ and a decryption key $sk_{\mathbf{A}}$ corresponding to an attribute set $\mathbf{A}$ as the input. Suppose that the attribute set $\mathbf{A}$ satisfies the access structure $(\mathbb{M}, \rho)$. Let I be defined as $I = \{i\colon \rho(i) \in \mathbf{A}\}$. Denote by $\{w_i \in Z_p\}_{i \in I}$ a set of constants such that if $\{v_i\}$ are valid shares of any secret μ according to $\mathbb{M}$, then $\Sigma_{i \in I} w_i v_i = \mu$. The algorithm computes

$$\frac{\hat{e}\left(C', sk'_1\right)}{\prod_{i\in I}\left(\hat{e}(C_i, sk_2)\hat{e}\left(D_i, sk_1^{\rho(i)}\right)\right)^{w_i}} = \frac{\hat{e}(g,g)^{\alpha\mu\mathcal{L}}\hat{e}(g,g)^{a\mu r}}{\prod_{i\in I}\hat{e}(g,g)^{rav_iw_i}} = \hat{e}(g,g)^{\alpha\mu\mathcal{L}},$$

and $\mathcal{L} = H_k(g^t h^s),\ t = H_k(\mathbf{A}||sk_2)$. Finally, it cancels out the value $(\hat{e}(g,g)^{\alpha\mu\mathcal{L}})^{1/\mathcal{L}}$ from C and obtains the message M.

Lemma 1. *The CP-ABE system above is selectively indistinguishable and strongly key unforgeable assuming that the decisional q-parallel BDHE assumption holds in G and that H_k is collision resistant.*

Proof. The Waters scheme [38] is known to be selectively indistinguishable assuming that the decisional q-parallel BDHE assumption holds in G. The proof for the selective indistinguishability is the same as that in [38] and is omitted here. When the q-parallel BDHE assumption holds in G, the discrete log assumption holds in G. Besides, the key generation algorithm and the decryption algorithm in [38] satisfy the required properties. Hence, all requirements of Theorem 1 are met which implies that the new CP-ABE system above is strongly key unforgeable. □

5.2. Instantiation 2

Below we apply our generic transformation to convert the first large universe CP-ABE scheme in [16] to a large universe CP-ABE scheme with strong key unforgeability. It is straightforward that the key generation algorithm in this construction is partitioned, and the functions F_1 and F_2 are

$$F_1\left(\mathbf{A}, \{r, r_1, \ldots r_{|\mathbf{A}|}\}, msk \circ \mathcal{L}\right) = \left\{msk^{\mathcal{L}} \cdot w^r,\ \forall i \in \mathbf{A}\ (u^{A_i}h)^{r_i}v^{-r}\right\},$$
$$F_2\left(\{r, r_1, \ldots r_{|\mathbf{A}|}\}, msk\right) = \{g^r,\ \forall i \in \mathbf{A}\ g^{r_i}\},$$

where w, h, u, v are elements belonging to the public parameter. Given $\mathbf{A}$, $\mathcal{L}$ and $sk_2 = F_2(r, msk)$, there exists only one sk_1 making the decryption successful, reflecting the second property of ABE scheme. Also, the third property is achieved (and to be shown in the concrete scheme) in that the decryption algorithm is able to recover the plaintext if the attributes $\mathbf{A}$ satisfy the access structure of the ciphertext by assigning $\mathcal{L}$ to the exponent.

Let G be a bilinear group of a prime order p with a generator g, and $\hat{e} : G \times G \to G_1$ denote the bilinear map. Let $\{H_k\}$ be a family of collision-resistant hash functions H_k: $\{0, 1\}^* \to \{0, 1\}^n$ indexed by $k \in \mathcal{K}$ with the hash outputs being viewed as elements of Z_p. We transform the Rouselakis–Waters CP-ABE scheme as follows, in which some algorithms are exactly the same as the original ones.

- Setup. This algorithm takes the security parameter λ as the input. It firstly chooses random group elements $u, h, v, w \in G$. Then, it chooses a random exponent $\alpha \in Z_p$. The public parameter is $pars = \{g, u, h, w, v, \hat{e}(g,g)^{\alpha}\}$, and the master private key is $msk = \alpha$. Finally, it randomly chooses a generator $h \in G$, a hash key $k \in \mathcal{K}$, and outputs the system parameter $pars' = (pars, k)$.
- KeyGen. This algorithm takes the system parameter $pars'$, the master private key msk and an attribute set $\mathbf{A} = \{A_1, \ldots, A_k\}$ as the input. It randomly chooses $r, r_1, \ldots, r_k, s \in Z_p$, and computes

$$sk'_2 = g^r,\quad t = H_k\left(\mathbf{A}||sk'_2||\{sk_2^{(i)}\}\right),\quad \mathcal{L} = H_k(g^t h^s),\quad sk'_1 = (g^{\alpha})^{\mathcal{L}}w^r,$$
$$\forall i \in [1,k]\ \ sk_2^{(i)} = g^{r_i},\quad sk_1^{(i)} = (u^{A_i}h)^{r_i}v^{-r}.$$

It outputs the attribute-based decryption key $sk_{\mathbf{A}} = (sk_1, sk_2, s)$ for $sk_1 = (sk'_1, \{sk_1^{(i)}\}_{i\in[1,k]})$, $sk_2 = (sk'_2, \{sk_2^{(i)}\}_{i\in[1,k]})$.
- Encrypt. This algorithm takes the system parameter $pars'$, a message M and an LSSS access structure $(\mathbb{M}, \rho)$ where the function ρ associates the rows of $\mathbb{M}$ to attributes as the input. Let $\mathbb{M}$ be an $l \times n$ matrix. It randomly chooses a vector $\vec{v} = (\mu, y_2, \ldots, y_n) \in Z_p^n$. These values will be used to share the encryption exponent μ. For $i = 1$ to l, it calculates $v_i = \vec{v} \cdot \mathbb{M}_i$, where $\mathbb{M}_i$ is the vector corresponding to the i-th row of $\mathbb{M}$. In addition, it randomly chooses $z_1, \ldots, z_l \in Z_p$, and computes

$$C = M \cdot \hat{e}(g,g)^{\alpha\mu},\quad C' = g^{\mu},$$
$$\forall i \in [1,l]\ \ C_i = w^{v_i}v^{z_i},\ D_i = g^{z_i},\ E_i = (u^{\rho(i)}h)^{-z_i}.$$

It outputs the ciphertext $\mathrm{CT} = ((\mathbb{M}, \rho), C, C', \{(C_i, D_i, E_i)\}_{i \in [1,l]})$.
- Decrypt. This algorithm takes the system parameter $pars'$, a ciphertext CT for an access structure $(\mathbb{M}, \rho)$ and a decryption key key $sk_{\mathbf{A}}$ for an attribute set $\mathbf{A}$ as the input. Suppose that the attribute set $\mathbf{A}$ satisfies the access structure $(\mathbb{M}, \rho)$. Let I be defined as $I = \{i : \rho(i) \in \mathbf{A}\}$. Denote by $\{w_i \in Z_p\}_{i \in I}$ a set of constants such that if $\{v_i\}$ are valid shares

Table 1
Comparison of efficiency among several ABE systems. Note that to make the comparison clearer, the scheme in [3] is transformed from composite-order groups to prime-order groups.

	[3]	Modified [3]	Construction 1	[8]	Construction 2
$\lvert par' \rvert$	$m+5$	$m+5$	$m+5$	7	6
$\lvert sk \rvert$	$k+4$	$k+4$	$k+3$	$2k+4$	$2k+3$
$\lvert \mathrm{CT} \rvert$	$2l+3$	$2l+3$	$2l+2$	$3l+3$	$3l+2$
Decrypt	$(2l+1)P$ $+(l+2)E$	$(2l+1)P$ $+(l+2)E$	$(2l+1)P$ $+(l+3)E$	$(3l+1)P$ $+(l+2)E$	$(3l+1)P$ $+(l+3)E$
Access Structure	AND, OR	AND, OR	AND, OR	AND, OR	AND, OR
Group order	composite	prime	prime	prime	prime
Security	full	selective	selective	selective	selective

of any secret μ according to $\mathbb{M}$, then $\Sigma_{i\,\in\,I} w_i v_i = \mu$. The algorithm computes

$$\frac{\hat{e}\big(C', sk'_1\big)}{\prod_{i\in I}\left(\hat{e}\big(C_i, sk'_2\big)\hat{e}\Big(D_i, sk_1^{\rho(i)}\Big)\hat{e}\Big(E_i, sk_2^{\rho(i)}\Big)\right)^{w_i}} = \frac{\hat{e}(g,g)^{\alpha\mu\mathcal{L}}\hat{e}(g,w)^{\mu r}}{\prod_{i\in I}\hat{e}(g,w)^{r v_i w_i}} = \hat{e}(g,g)^{\alpha\mu\mathcal{L}},$$

and $\mathcal{L} = H_k(g^t h^s)$, $t = H_k(\mathbf{A}||sk'_2||\{sk_2^{(i)}\})$. It then cancels out the value $(\hat{e}(g,g)^{\alpha\mu\mathcal{L}})^{1/\mathcal{L}}$ from C and obtains the message M.

Lemma 2. *The key regeneration-free CP-ABE scheme above is selectively indistinguishable and strongly key unforgeable assuming that the* $(q-1)$ *assumption holds in* G *and that* H_k *is collision resistant.*

Proof. The Rouselakis–Waters scheme [16] is known to be selectively indistinguishable assuming that the decisional q-parallel BDHE assumption holds in G. The proof for the selective indistinguishability of the new scheme is the same as that in [16] and hence is omitted. When the $(q-1)$ assumption holds in G, the discrete log assumption holds in G. Besides, the key generation algorithm in the Rouselakis–Waters scheme is partitioned, and the decryption algorithm is able to successfully decrypt a ciphertext with an additional label $\mathcal{L}$. Thus, all requirements of Theorem 1 are satisfied. Therefore, the new CP-ABE scheme above is strongly key unforgeable. □

6. Discussion

In this section, we first compare our key regeneration-free CP-ABE with several previous CP-ABE schemes aiming to solve the key abuse problem. We then discuss how to extend our scheme to traceable CP-ABE.

6.1. Comparison

To the best of our knowledge, most of the existing techniques [3–8] addressing the key misuse problem in ABE are proposed on the basis of some concrete constructions such that users will be meticulous to share decryption privileges since users' identities can be traced from the shared decryption keys. Though the scheme given in [9] solves the key regeneration issue by making the CP-ABE scheme key regeneration free, it does not support expressive access structures (expressed in AND and OR gates). Our technique presented in this paper is a generic mechanism, which can transform any CP-ABE system with partitioned key generation into key regeneration-free CP-ABE which prevents any modification to the original decryption key.

In Table 1, we compare the two instantiations derived from our generic construction (applying the underlying CP-ABE schemes used in [3] and [8], respectively, to the generic framework) with two practical CP-ABE schemes presented in [3] and [8], respectively. The scheme in [3] is the first traceable CP-ABE scheme supporting policies in the monotone access structures and is fully secure over a composite order-group, while the scheme in [8] is a practical large universe CP-ABE system supporting white-box traceability in which the number of the attributes allowed in the system is unbounded. Let "prime" and "composite" denote prime-order group and composite-order group, respectively, "full" and "selective" denote fully secure and selectively secure, respectively, "P" and "E" denote the paring calculation and exponentiation calculation, respectively. Let $|sk|$, $|\mathrm{CT}|$ and $|par'|$ denote the sizes of user's decryption key, ciphertext overhead and system parameter, respectively. Let l be the number of attributes in an access structure, k be the size of an attribute set, and m be the maximum size allowed for k. From Table 1, we can see that our instantiations have more or less the same efficiency as the two schemes in [3] and [8], i.e., the proposed approach of making CP-ABE schemes key regeneration free to overcome the key regeneration abuse challenge is almost as efficient as those applying tracing mechanisms.

6.2. Application to White–Box traceable attribute-based encryption

In a white-box traceable CP-ABE scheme (e.g, [3]), a user's identity is embedded in the decryption key in such a way that the user cannot modify the decryption key to another one embedded with a different identity. In the case that the user maliciously shares the decryption privilege with others, the user can be traced back based on the identity in a pirated decryption key. Our key regeneration-free CP-ABE schemes can be easily applied to achieve white-box traceability by embedding a user's identity in his/her decryption key. A straightforward method is to set the component s in the decryption key as a value related to a user identity (similar to the role of c in the decryption key defined in [3]), and thus for any decryption key with an element s, the user's identity can be traced by searching among all the users' identities to see which one's identifying value matches the given s (we assume that there exists a table storing all users' identities and their corresponding identifying values (id, s)). We omit the details here, please refer to [3] for the full description about the white-box tracing function.

7. Conclusions

The key randomization is an important technique in the standard attribute-based encryption (ABE) schemes to prevent collusion attacks among multiple users such that malicious users are not able to combine their decryption keys to decrypt ciphertexts that none of them could decrypt on their own. This randomization method, however, also endows a user with the capability of regenerating a newly randomized decryption key over a subset of the attributes associated with the original decryption key. Because key randomization breaks the linkage between the newly generated key and the original key, an adversarial user could leak the new decryption key to others without being held responsible for any abuse of the leaked key. This paper focused on removing this key regeneration property from ABE while preserving all advantages of ABE, so-called *Key Regeneration-Free ABE*. Motivated by the observation that generating a decryption key in ABE can be regarded as issuing a "signature" by the attribute authority (AA), we introduced a security notion called strong key unforgeability, and showed that the key regeneration property inherent in the standard ABE schemes can be avoided by equipping an ABE scheme with the strong key unforgeability. In a key regeneration-free ABE scheme, a user possessing a decryption key is disallowed to randomize the original key in any manner, i.e., a user can only delegate the decryption key in exactly the same form without any modification so that an abused or pirated key can be traced back to its original owner. We gave a generic transformation for converting a standard CP-ABE scheme to a key regeneration-free CP-ABE scheme. We also presented two concrete constructions of key regeneration-free CP-ABE schemes by applying the transformation to the standard CP-ABE schemes in [3] and [8], respectively.

Acknowledgments

This research effort is supported by the Singapore National Research Foundation under the NCR Award number NRF2014NCR-NCR001-012. This work is also supported by the National Natural Science Foundation of China under the grant number 61872292.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ins.2019.12.025.

References

[1] A. Sahai, 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, in: Lecture Notes in Computer Science, 3494, Springer, 2005, pp. 457–473.

[2] V. Goyal, O. Pandey, A. Sahai, 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, in: Lecture Notes in Computer Science, 5126, Springer, 2006, pp. 89–98.

[3] Z. Liu, Z. Cao, D.S. Wong, White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures, IEEE Trans. Inf. ForensicsSecur. 8 (1) (2013) 76–88.

[4] Z. Liu, Z. Cao, D.S. Wong, Fully collusion-resistant traceable key-policy attribute-based encryption with sub-linear size ciphertexts, in: Information Security and Cryptology - 10th International Conference, Inscrypt 2014, Beijing, China, December 13–15, 2014, Revised Selected Papers, in: Lecture Notes in Computer Science, 8957, Springer, 2014, pp. 403–423.

[5] J. Li, K. Ren, B. Zhu, Z. Wan, Privacy-aware attribute-based encryption with user accountability, in: Information Security, 12th International Conference, ISC 2009, Pisa, Italy, September 7–9, 2009. Proceedings, 2009, pp. 347–362.

[6] J. Li, Q. Huang, X. Chen, S.S.M. Chow, D.S. Wong, D. Xie, Multi-authority ciphertext-policy attribute-based encryption with accountability, in: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2011, Hong Kong, China, March 22–24, 2011, ACM, 2011, pp. 386–390.

[7] M.J. Hinek, S. Jiang, R. Safavi-Naini, S.F. Shahandashti, Attribute-based encryption with key cloning protection, IACR Cryptol. ePrint Archive 2008 (2008) 478.

[8] J. Ning, Z. Cao, X. Dong, L. Wei, X. Lin, Large universe ciphertext-policy attribute-based encryption with white-box traceability, in: Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7–11, 2014. Proceedings, Part II, in: Lecture Notes in Computer Science, 8713, Springer, 2014, pp. 55–72.

[9] Y. Jiang, W. Susilo, Y. Mu, F. Guo, Ciphertext-policy attribute-based encryption with key-delegation abuse resistance, in: Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4–6, 2016, Proceedings, Part I, in: Lecture Notes in Computer Science, 9722, Springer, 2016, pp. 477–494.

[10] D. Boneh, R. Canetti, S. Halevi, J. Katz, Chosen-ciphertext security from identity-based encryption, SIAM J. Comput. 36 (5) (2007) 1301–1328.

[11] D. Boneh, E. Shen, B. Waters, Strongly unforgeable signatures based on computational Diffie–Hellman, in: Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography, New York, NY, USA, April 24–26, 2006, Proceedings, in: Lecture Notes in Computer Science, 3958, Springer, 2006, pp. 229–240.

[12] R. Ostrovsky, A. Sahai, B. Waters, Attribute-based encryption with non-monotonic access structures, in: Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28–31, 2007, ACM, 2007, pp. 195–203.

[13] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: 2007 IEEE Symposium on Security and Privacy (S&P 2007), 20–23 May 2007, Oakland, California, USA, IEEE Computer Society, 2007, pp. 321–334.

[14] L. Cheung, C.C. Newport, Provably secure ciphertext policy ABE, in: Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28–31, 2007, ACM, 2007, pp. 456–465.

[15] V. Goyal, A. Jain, O. Pandey, A. Sahai, Bounded ciphertext policy attribute based encryption, in: Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7–11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations, in: Lecture Notes in Computer Science, 5126, Springer, 2008, pp. 579–591.

[16] Y. Rouselakis, B. Waters, Practical constructions and new proof methods for large universe attribute-based encryption, in: 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4–8, 2013, ACM, 2013, pp. 463–474.

[17] J.H. An, Y. Dodis, T. Rabin, On the security of joint signature and encryption, in: Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28, - May 2, 2002, Proceedings, in: Lecture Notes in Computer Science, 2332, Springer, 2002, pp. 83–107.

[18] D. Dolev, C. Dwork, M. Naor, Non-malleable cryptography (extended abstract), in: Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5–8, 1991, New Orleans, Louisiana, USA, ACM, 1991, pp. 542–552.

[19] G. Ateniese, J. Camenisch, M. Joye, G. Tsudik, A practical and provably secure coalition-resistant group signature scheme, in: Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20–24, 2000, Proceedings, in: Lecture Notes in Computer Science, 1880, Springer, 2000, pp. 255–270.

[20] M. Bellare, P. Rogaway, The exact security of digital signatures - how to sign with RSA and rabin, in: Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12–16, 1996, Proceeding, in: Lecture Notes in Computer Science, 1070, Springer, 1996, pp. 399–416.

[21] J. Coron, On the exact security of full domain hash, in: Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20–24, 2000, Proceedings, in: Lecture Notes in Computer Science, 1880, Springer, 2000, pp. 229–235.

[22] D. Boneh, B. Lynn, H. Shacham, Short signatures from the weil pairing, J. Cryptol. 17 (4) (2004) 297–319.

[23] E. Goh, S. Jarecki, A signature scheme as secure as the Diffie–Hellman problem, in: Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003, Proceedings, in: Lecture Notes in Computer Science, 2656, Springer, 2003, pp. 401–415.

[24] S. Micali, L. Reyzin, Improving the exact security of digital signature schemes, J. Cryptol. 15 (1) (2002) 1–18.

[25] R. Gennaro, S. Halevi, T. Rabin, Secure hash-and-sign signatures without the random oracle, in: Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2–6, 1999, Proceeding, in: Lecture Notes in Computer Science, 1592, Springer, 1999, pp. 123–139.

[26] S. Micali, M.O. Rabin, S.P. Vadhan, Verifiable random functions, in: 40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17–18 October, 1999, New York, NY, USA, IEEE Computer Society, 1999, pp. 120–130.

[27] R. Cramer, V. Shoup, Signature schemes based on the strong RSA assumption, ACM Trans. Inf. Syst. Secur. 3 (3) (2000) 161–185.

[28] D. Boneh, X. Boyen, Short signatures 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, in: Lecture Notes in Computer Science, 3027, Springer, 2004, pp. 56–73.

[29] O. Goldreich, Two remarks concerning the Goldwasser-Micali-Rivest signature scheme, in: Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings, in: Lecture Notes in Computer Science, 263, Springer, 1986, pp. 104–110.

[30] S. Goldwasser, S. Micali, R.L. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, SIAM J. Comput. 17 (2) (1988) 281–308.

[31] M. Naor, M. Yung, Universal one-way hash functions and their cryptographic applications, in: Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14–17, 1989, Seattle, Washigton, USA, ACM, 1989, pp. 33–43.

[32] R. Cramer, I. Damgård, New generation of secure and practical rsa-based signatures, in: Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18–22, 1996, Proceedings, in: Lecture Notes in Computer Science, 1109, Springer, 1996, pp. 173–185.

[33] C. Dwork, M. Naor, An efficient existentially unforgeable signature scheme and its applications, J. Cryptol. 11 (3) (1998) 187–208.

[34] D. Boneh, I. Mironov, V. Shoup, A secure signature scheme from bilinear maps, in: Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13–17, 2003, Proceedings, in: Lecture Notes in Computer Science, 2612, Springer, 2003, pp. 98–110.

[35] D. Boneh, M. Franklin, Identity-based encryption from the weil pairing, in: CRYPTO, in: Lecture Notes in Computer Science, 2139, Springer-Verlag, 2001, pp. 213–219.

[36] I. Damgård, Collision free hash functions and public key signature schemes, in: Advances in Cryptology - EUROCRYPT '87, Workshop on the Theory and Application of Cryptographic Techniques, Amsterdam, The Netherlands, April 13–15, 1987, Proceedings, in: Lecture Notes in Computer Science, 304, Springer, 1987, pp. 203–216.

[37] A.B. Lewko, B. Waters, Decentralizing attribute-based encryption, in: Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15–19, 2011. Proceedings, in: Lecture Notes in Computer Science, 6632, Springer, 2011, pp. 568–588.

[38] 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, in: Lecture Notes in Computer Science, 6571, Springer, 2011, pp. 53–70.

[39] A. Beimel, Secure Schemes for Secret Sharing and Key Distribution, Israel Institute of Technology, Israel Institute of Technology, 1996 Ph.D. thesis.

[40] H. Cui, R.H. Deng, Y. Li, B. Qin, Server-aided revocable attribute-based encryption, in: Computer Security - ESORICS 2016 - 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26–30, 2016, Proceedings, Part II, in: Lecture Notes in Computer Science, 9879, Springer, 2016, pp. 570–587.

[41] D.M. Freeman, Converting pairing-based cryptosystems from composite-order groups to prime-order groups, 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, in: Lecture Notes in Computer Science, 6110, Springer, 2010, pp. 44–61.

[42] H. Krawczyk, T. Rabin, Chameleon signatures, in: Proceedings of the Network and Distributed System Security Symposium, NDSS 2000, San Diego, California, USA, The Internet Society, 2000.