Using Threshold Attribute-Based Encryption for Practical Biometric-Based Access Control

Deholo Nali, Carlisle Adams, Ali Miri
(Corresponding author: Deholo Nali)

School of Information Technology and Engineering, University of Ottawa
800 King Edward Avenue, P.O. Box 450, Station A, Ottawa, Ontario, Canada K1N 6N5
(Email: {deholo, cadams, samiri}@site.uottawa.ca)

(Received June 30, 2005; revised and accepted Aug. 2, 2005)

Abstract

Threshold attribute-based encryption (thABE) is a variant of identity-based encryption which views identities as sets of descriptive attributes. If a thABE ciphertext c is computed for a set ω of attributes, then, to decrypt c, a user must have keys associated with a sufficiently large subset of ω. One application of thABE is biometric-based access control (BBAC). Practical BBAC applications impose the following constraints on the design of thABE schemes: first, a suitable thABE scheme must have an efficient decryption procedure; second, the proposed scheme must prevent colluding users from being able to decrypt ciphertexts which none of them could decrypt; third, the designed scheme must provide a mechanism whereby encryptors can, *at encryption time*, specify multiples sets of attributes with their corresponding threshold values. To the best of our knowledge, no scheme is known that simultaneously satisfies the aforementioned requirements. This paper describes an efficient and collusion-resistant thABE scheme featuring dynamically-specifiable threshold values. The proposed scheme is proven secure in the random oracle model, and its efficiency and flexibility are compared with Sahai and Waters'thABE scheme.

Keywords: Access control, biometric-based encryption, fuzzy identity-based encryption

1 Introduction

Cryptography has long been suggested as a method to support access control. The idea is to encrypt confidential documents according to sets of privileges, and to give the corresponding privilege decryption keys to users who have these privileges. Thus, it is expected that only authorized users can cryptographically access (decrypt) protected documents. In the following section, we present issues which must be addressed when cryptography is used to support biometric-based access control (BBAC).

1.1 Practical BBAC Constraints

In BBAC, the biometric identity of a person is often composed of multiple sets of attributes. This is true for one's voice [18], iris [8], and fingerprint [13]. Moreover, the number of attributes used to represent biometric identities is often large (15 in the case of Monrose et al.'s key-stroke-based algorithm [17], 46 in the case of Monrose et al.'s voice-based method [18], 249 in the case of Daugman's iris-based algorithm [8], and 648 in the case of Jain et al.'s fingerprint-based method [13]). Furthermore, the number of attributes used to identify users evolves with both time (i.e. research discoveries) and the accuracy of specific input devices [21]. (This accuracy can only be determined at encryption time.) Consequently, practical cryptographic BBAC applications call for the use of mechanisms which enable users to encrypt confidential documents with respect to multiple and varying sets of attributes.

1.2 Threshold Attribute-Based Encryption

Recently introduced by Sahai and Waters [21], threshold attribute-based encryption (thABE) has been suggested as a cryptographic primitive suitable for BBAC. More precisely, let d be a threshold value, and c be a ciphertext intended for an identity ω (where ω is a set of descriptive attributes). Let also ω' be the identity of a user $\mathcal{U}$. Then, $\mathcal{U}$ must have decryption keys associated with a d-element subset of ω in order decrypt c. (In other words, $|\omega \cap \omega'| \geq d$ must be satisfied in order for $\mathcal{U}$ to decrypt c.) Thus, if attributes represent features extracted from biometric readings, and if users are given decryption keys corresponding to their biometric features, then thABE can be used to support BBAC.

1.3 Concrete Example of thABE Application to BBAC

In BBAC, each user is uniquely identified by sets of attributes representing features of her biometric identity (such as her iris or her thumb). Since biometric measurements are noisy, the use of thABE significantly improves the possibility of using biometrics to authenticate human users, and thereby control access to information addressed to these users. For instance, a user $\mathcal{A}$ could go to a Driver Licensing Agency $\mathcal{D}$, and identify herself via an iris scan, *under the ongoing surveillance of a trained agent*. $\mathcal{D}$ could then use this scan to encrypt $\mathcal{A}$'s information (e.g. an annual driver's license), when this information needs to be securely sent to $\mathcal{A}$ (e.g. via the Web, or via storage of ciphertexts on a smart card presented by (a representative of) $\mathcal{A}$). In order to obtain her biometric private keys, $\mathcal{A}$ would have to go in person to a trusted third party (e.g. a state agency) which would deliver keys via the same authenticating procedure as that used by $\mathcal{D}$. $\mathcal{A}$ could then decrypt ciphertexts addressed to her, using a thABE scheme.

It is important to note that physical biometric attributes do not have to be remembered (like passwords) or stored in a transportable tamper-resistant device. Moreover, not that, when biometrics are read under the surveillance of trained agents, the odds of user impersonation in the process of obtaining biometric-based decryption keys is significantly reduced. Moreover, the aforementioned identification method is simple[1], does not require prior relationship with the authenticating party, and does not require corroborating credentials (which may have been illegally forged). Furthermore, each user decides to whom her biometric identity is revealed, and the use of biometrics to generate cryptographic keys provides an efficient method to assign unique identifiers to human users. Hence, the aforementioned biometric-based access control process offers many security benefits.

It should be noted, however, that biometric-based decryption keys are not as easy to change as other classes of decryption keys. Indeed, humans have a limited number of physical attributes (e.g. eyes and fingers). Note, nevertheless, that advances in biometric-based authentication regularly discover new biometric features which can be used to better authenticate users.

1.4 Certificate-Based vs. Identity-Based Encryption

thABE is a variant of IBE [6, 22]. While certificate-based encryption (CBE) forces encryptors to obtain the public-key certificates of their recipients, IBE requires that encryptors obtain only one set of system-wide certified public parameters. Since digital certificates are known to be difficult to manage and distribute in large user-communities, it follows that, in some situations, IBE is more suitable to encryption-based access control than CBE. In particular, IBE is preferable to CBE when known techniques to revoke identity-based (ID-based) decryption keys are acceptable. One such technique consists of associating attribute identifiers with time periods [6]. In such a case, users are given time-period-bound decryption keys, at the beginning of each period, upon satisfaction of a number of criteria. Another ID-based key revocation method consists of using online cryptographic agents which must be involved in each decryption process [4, 5, 15]. These agents (which can be associated with many users) partially process each user's decryption request, if and only if the user has all required security privileges. Thus, thABE can be used to support BBAC and RBAC, without requiring encryptors to obtain public-key certificates.

[1] It consists of a mere biometric reading

1.5 Difficulty of Designing Efficient and Flexible thABE Schemes

Designing efficient and flexible thABE schemes is not trivial. Indeed, if New-thABE is such a scheme, then the following challenges must be simultaneously addressed.

C1 New-thABE should be resistent to collusion attacks whereby users having various attribute keys attempt to pool their keys in order to decrypt ciphertexts which none of them could decrypt on her own. This means that private attribute keys should be "personalized" while public attribute keys should not be tied with users, so that New-thABE be scalable to large user populations.

C2 Encryption for an identity ω should target each of the possible sufficiently large subsets of ω. If ω has ℓ elements and d is a given threshold value, then there are $\frac{\ell!}{d!(\ell-d)!}$ possible combinations. However, the cost of encryption should remain low (i.e. ideally constant, but at most linear with respect to d).

C3 If d is a threshold value associated with a ciphertext c, then a user with identity ω should be given private keys in such a way that at least d of them must be jointly used in order to decrypt c. This is challenging to achieve because private keys are given before encryption time. Hence, the use of a mere secret sharing scheme at decryption-key granting time is not adequate.

C4 Encryptors should be able to select attributes from multiple sets whose threshold values are different.

With respect to C3, remark that the decryption procedure of Sahai and Waters's second scheme [21] requires $2d$ pairing (i.e. expensive) operations, where d is a system-wide threshold parameters. (Note that Sahai and Waters presented two schemes in [21]. However, only the second scheme is scalable when the number of attributes grows. Moreover, only the second scheme features a small set of certified public parameters. Consequently, the phrase

Sahai and Waters'scheme will henceforth refer to Sahai and Waters'*second* scheme.) One is interested in a scheme whose decryption algorithm involves a small constant number of pairing computations. Furthermore, remark that Sahai and Waters suggested two methods to address C4. First, it was suggested to use different thABE systems with different threshold values. Second, it was proposed to use a maximal threshold value for all ciphertexts, and to use dummy attributes (whose secret keys are given by default to all decryptors) in order to decrease the effective threshold value associated with a given ciphertext. The first proposed option is not necessarily convenient, but most importantly, it increases the length of ciphertexts associated with multiple sets of attributes (by a factor equal to the number of sets). The second proposed option is not necessarily adequate either, as illustrated by the following example. Suppose that $d_{max} = 4$, and that one wants to encrypt a message m for an identity $\omega = \{\alpha_1, \alpha_2\}$ with threshold value $d = 2$. Let δ_1 and δ_2 be two dummy attributes. If m is encrypted for ω with δ_1 and δ_2, then the recipient can decrypt the ciphertext without having the secret keys associated with α_1 and α_2. Moreover, if m is encrypted for with ω with δ_1, then the recipient only needs to have the secret key associated with α_1 or α_2.

Thus, while Sahai and Waters addressed C1 and C2 simultaneously, designing a scheme which efficiently addresses C1 through C4 remains an open question. The goal of this paper is to address each of the aforementioned challenges. Our contributions are presented in Section 1.7. Section 1.6 reviews past research related to thABE.

1.6 Related Work

thABE schemes form a subclass of fuzzy ID-based encryption (FIBE) schemes [21]. As thABE schemes, FIBE schemes view identities as sets of descriptive attributes, and use attributes to encrypt messages. More precisely, a user $\mathcal{U}$ with identity ω' is able to decrypt a ciphertext encrypted with identity ω if and only if ω and ω' are within a certain distance from each other, as stipulated by a specifiable metric. One such metric is the *set-overlap* distance, which forces each decryptor of any given ciphertext c to have keys corresponding to a sufficiently large subset of the identity used to obtain c. When a FIBE scheme uses the *set-overlap* distance, then this scheme is called a *threshold attributed-based encryption* (thABE) scheme. IBE was introduced by Shamir [22], in 1984, but the first efficient and provably secure IBE scheme was described by Boneh and Franklin [6], in 2001. thABE schemes distinguish themselves from known *threshold IBE* (thIBE) schemes [2, 4, 15], by the following two features: first, thIBE schemes view each identity as one string (instead of a set of attribute strings); second, thIBE schemes require one decryptor to interact with a threshold number of servers in order to complete decryption procedures. With regards to ID-based access control, Nali et al. [19] presented a thIBE scheme involving encryption with multiple attributes, but their scheme is not collusion-resistant. Yao et al. [23] also proposed a collusion-resistant ID-based encryption scheme which encrypts to multiple hierarchical identities, but this scheme is computationally expensive (the number of pairings in the decryption procedure grows linearly with the number of hierarchical identities used to encrypt a message.)

Much research dealing with the use of biometrics to derive cryptographic secret keys has also been conducted [7, 9, 14, 16, 17, 18]. In these systems, as pointed out by Sahai and Waters [21], the capture of one's biometric reading enables full impersonation of the corresponding person. On the contrary, FIBE allow biometric measurements to be public. In fact, identities used to encrypt messages are also made public.

1.7 Contributions

The main contribution of this paper is to present a flexible and collusion-resistant thABE scheme featuring significantly lower computational requirements than Sahai and Waters'scheme (henceforth referred to as SW-thABE). More precisely, let d be the predefined threshold value associated with SW-thABE. (Take, for instance, $d = 249$ in the case of Daugman's iris recognition algorithm [8]). Then the proposed scheme's decryption procedure requires only 2 pairing computations, compared with $2d$ pairings in the case of SW-thABE's decryption algorithm. Moreover, the proposed thABE scheme (henceforth referred to as New-thABE) is flexible because it allows to specify, at encryption time, various sets of attributes with their associated threshold values. Furthermore, if t denotes the number of attributes used to encrypt a message, an optimization of New-thABE is proposed which drastically reduces its ciphertext length, from $O(t)$ to $O(1)$, when decryptors must have *all* the attributes of a given set. Moreover, we note that, when a ciphertext's threshold value is maximal, New-thABE provides the first efficient collusion-resistant IBE scheme.

We examine the security of New-thABE in the random oracle model [3], and show that this scheme can be extended (via Fujisaki-Okamoto padding [10]) into a scheme that provides semantic security against adaptive chosen ciphertext attacks, in the random oracle model, if the bilinear Diffie-Hellman problem is intractable. For comparison, note that SW-thABE achieves a weaker security result (i.e. semantic security against selective-ID attacks[2]), in the standard model.

1.8 Outline

The sequel is organized as follows: Section 2 reviews standard terminology concerning thABE, its security, and related number theoretic assumptions. Section 3 describes our proposed thABE scheme, and Section 4 discusses

[2]Selective-ID attacks model attacker which choose, in advance, the target identity they wish to attack.

its computational requirements, in comparison with SW-thABE. Section 5 summarizes the security guarantees of our scheme, and Section 6 concludes the paper.

2 Preliminaries

In this section, we present fundamental definitions related to thABE schemes, their security, bilinear pairing and standard number theoretic assumptions. Readers familiar with [21] may go to Section 3.

2.1 Identities

For the description of thABE schemes, each *identity* is viewed as a set ω of *attributes*. Each user with identity ω is given a set of private *attribute keys* each of which corresponds to an element of ω. These attributes keys are secretly granted by an entity known as the private key generator (PKG), upon careful inspection of users'identities (e.g. via surveillance of on-site biometric measurement).

2.2 Threshold Attribute-Based Encryption Scheme

Each thABE scheme is composed of four algorithms whose functions are described below:

1) `Setup`: Given a security parameter k, this algorithm is used by the PKG to return a tuple *params* of system parameters. These parameters include a description of the message space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$, along with a secret piece of data s_0 called the *master secret key*. Other parameters are allowed, such as *attribute* parameters. Some parameters may be public (including those describing $\mathcal{M}$ and $\mathcal{C}$), while others remain secret (including the master secret key).

2) `Key Generation`: Given the scheme's public and private parameters, and an arbitrary identity ω, this algorithm returns a set D_ω of private keys corresponding to ω.

3) `Encrypt`: Given a threshold parameter d, a message $m \in \mathcal{M}$, the identity ω of an intended recipient, and the scheme's public parameters, this algorithm returns a ciphertext $c \in \mathcal{C}$ corresponding to m, ω and d.

4) `Decrypt`: Given a ciphertext $c \in \mathcal{C}$ (including a threshold parameter d and a target identity ω), the private key set $D_{\omega'}$ of a recipient, and the scheme's public parameters, this algorithm returns the message $m \in \mathcal{M}$ associated with c if $|\omega \cap \omega'| \geq d$.

Encryption and decryption must satisfy the following consistency constraint:

$$\forall\, m \in \mathcal{M}\ \text{Decrypt}(d, c, pubParams, D_{\omega'}) = m,$$
$$\text{if } |\omega \cap \omega'| \geq d \text{ and } c = \text{Encrypt}(d, m, \omega, pubParams).$$

2.3 Threat Model

To examine the security of thABE schemes, the related notions of adaptive chosen ciphertext security and chosen plaintext security are first defined in Section 2.3.1 and Section 2.3.2. These notions are used to define the security of thABE schemes.

2.3.1 Chosen Ciphertext Security

Let Ψ be a thABE scheme. Note that Ψ's ability to encrypt for multiple sets of attributes is not intrinsically related to Ψ's security. Hence, we shall assume (for simplicity) that Ψ encrypts for only one set of attributes. Thus, the following game, initiated by a challenger $\mathcal{C}h$ against an attacker $\mathcal{A}$, may be considered:

Setup: From a security parameter k, $\mathcal{C}h$ uses Ψ's *Setup* algorithm to generate the cryptosystem's public and private parameters - keeping the private parameters secret while giving the public ones to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ issues to $\mathcal{C}h$ a polynomially bounded number of queries of the following types:

- *Key Extraction query*: Given an identity ω_i, $\mathcal{C}h$ must return the private key set D_{ω_i} associated with ω_i.
- *Decryption query*: Given an identity ω_i', and a ciphertext c_i encrypted for an identity ω_i such that $|\omega_i \cap \omega_i'| \geq d_i$ (where d_i is the threshold parameter associated with c_i), $\mathcal{C}h$ must return a message m_i associated with c_i.

The above queries may be issued adaptively: each request may depend on its predecessors.

Challenge: Once *Phase 1* is over, $\mathcal{A}$ issues a threshold parameter d^*, an identity ω^* of its choice, and a pair (m_0, m_1) of equal-length plaintexts, such that, in *Phase 1*, no *Key Extraction* queries were issued on an identity ω such that $|\omega \cap \omega^*| \geq d^*$. $\mathcal{C}h$ then picks a random bit $\beta \in \{0, 1\}$, computes the encryption c^* of m_β for ω^*, and sends c^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ issues a polynomially bounded number of *Phase 1* types of queries, under the following restrictions:

- No *Key Extraction* queries are issued on an identity ω_i such that $|\omega_i \cap \omega^*| \geq d^*$.
- No *Decryption* query is issued with c^* as an argument.

The queries may be issued adaptively: each request may depend on its predecessors.

Guess: Once *Phase 2* is over, $\mathcal{A}$ submits a guess bit β' and wins the game if $\beta' = \beta$.

Definition 1. *The above game is called the* IND-thABE-CCA *attack game. The quantity* $|Pr[\beta' = \beta] - \frac{1}{2}|$ *– representing the advantage of* $\mathcal{A}$ *over any challenger* $\mathcal{C}h$ *in the game – is denoted by* $Adv^{thABE-CCA}_{\mathcal{A},\mathcal{C}h}$. *A thABE scheme is said to be secure against adaptive chosen ciphertext attacks (*IND-thABE-CCA *secure, in short) if no polynomially bounded attacker can be found that has non-negligible advantage in the above* IND-thABE-CCA *game.*

2.3.2 Chosen Plaintext Security

The notion of chosen plaintext security is similar to (and weaker than) the notion of *chosen ciphertext security.* To define semantic security for thABE schemes, one may consider a game (called the *IND-thABE-CPA* game), which is identical to the *IND-thABE-CCA* game, except that the attacker $\mathcal{A}$ is unable to issue *Decryption* queries.

Definition 2. *The value* $|Pr[\beta' = \beta] - \frac{1}{2}|$ *– representing the advantage of* $\mathcal{A}$ *over any challenger* $\mathcal{C}h$ *in the* IND-thABE-CPA *game – is denoted by* $Adv^{thABE-CPA}_{\mathcal{A},\mathcal{C}h}$. *A thABE scheme is said to be secure against adaptive chosen plaintext attacks (*IND-thABE-CPA *secure, in short) if no polynomially bounded attacker can be found that has non-negligible advantage in a* IND-thABE-CPA *game.*

2.4 Bilinear Pairing and Diffie-Hellman Problems

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two Abelian groups of prime order q, where $\mathbb{G}_1$ is additive and $\mathbb{G}_2$ is multiplicative. Let $P_0^{(1)} \in \mathbb{G}_1^*$ be a generator of $\mathbb{G}_1$. A *Bilinear pairing* $\hat{e}$ is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ such that $\hat{e}(aP_0^{(1)}, bP_0^{(1)}) = \hat{e}(P_0^{(1)}, P_0^{(1)})^{ab}$ for all $a, b \in \mathbb{Z}_q^*$. (Bilinear pairings can be constructed using *Weil pairings* (cf. section 5 of [6]) and – more efficiently – using *Tate pairings* on elliptic curves.) The map $\hat{e}$ is said to be an *admissible pairing* if it is a *non-degenerate* (i.e. $\hat{e}$ does not send all pairs of points in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$), *computable* (i.e. $\hat{e}$ efficiently computes the image of any pair of points in $\mathbb{G}_1 \times \mathbb{G}_1$) *Bilinear pairing.* Let $\mathcal{A}$ be an attacker modelled as a probabilistic Turing machine.

The *computational Diffie-Hellman* (*CDH*) problem [6] is that in which $\mathcal{A}$ is to compute $abP_0^{(1)}$, given $(\mathbb{G}_1, q, P_0^{(1)}, aP_0^{(1)}, bP_0^{(1)})$ and a security parameter k, where $a, b \in \mathbb{Z}_q^*$ are unknown. The success (or *advantage*) of $\mathcal{A}$ is then defined as $Pr[\mathcal{A}$ computes $abP_0^{(1)}]$. The *decisional Diffie-Hellman* (*DDH*) problem [6] is that in which $\mathcal{A}$ is to guess whether $cP_0^{(1)} = abP_0^{(1)}$, given $(\mathbb{G}_1, q, P_0^{(1)}, aP_0^{(1)}, bP_0^{(1)}, cP_0^{(1)})$ and a security parameter k, where $a, b, c \in \mathbb{Z}_q^*$ are unknown. The success (or *advantage*) of $\mathcal{A}$ is then defined as $Pr[\mathcal{A}$ makes a right guess that $cP_0^{(1)} = abP_0^{(1)}]$. $\mathbb{G}_1$ is called a *Gap-Diffie-Hellman group* if the CDH is intractable in $\mathbb{G}_1$, but the the DDH can be solved in polynomial time in $\mathbb{G}_1$. The *Bilinear Diffie-Hellman* (*BDH*) problem [6] is that in which $\mathcal{A}$ is to compute $\hat{e}(P_0^{(1)}, P_0^{(1)})^{abc}$ given a security parameter k, the tuple $(\mathbb{G}_1, q, P_0^{(1)}, aP_0^{(1)}, bP_0^{(1)}, cP_0^{(1)})$ where $a, b, c \in \mathbb{Z}_q^*$ are unknown, and given the fact that the *CDH* problem cannot be solved in polynomial time with non-negligible advantage in both $\mathbb{G}_1$ and $\mathbb{G}_2$. The success (or *advantage*) of $\mathcal{A}$ is then defined as $Pr[\mathcal{A}$ outputs $\hat{e}(P_0^{(1)}, P_0^{(1)})^{abc}]$.

3 Proposed Threshold Attribute-Based Encryption Scheme

The thABE scheme presented below (i.e. New-thABE) shows how to encrypt for two attribute sets whose threshold values can be dynamically specified by the encryptor. Encryption to polynomially many attribute sets could also be handled, using a similar methodology. However, for simplicity, we only describe the case in which there are two attribute sets.

1) `Instance Generator` (k). This procedure, denoted by IG, is a randomized algorithm which takes a security parameter $k > 0$, runs in $O(k)$, and outputs $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are two Abelian groups of prime order $q \geq 2^k$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is an admissible pairing with respect to which $\mathbb{G}_1$ and $\mathbb{G}_2$ are Gap-Diffie-Hellman groups.

2) `Setup` (k): Given a security parameter $k > 0$, the *PKG*:

   a. runs IG with input k and obtains $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$.

   b. sets $\delta = 2$, and computes both $n = poly_1(k)$ and $\ell = poly_2(k)$, where $poly_1$ and $poly_2$ are polynomials over the positive integers; (δ is the number of attribute sets used for encryption, n is the message length, and ℓ is the total number of attributes;)

   c. picks, randomly and uniformly[3], $P_0^{(1)}, P_0^{(2)} \in \mathbb{G}_1$ and $s_0 \in \mathbb{Z}_q^*$;

   d. computes $P_{pub} = s_0 P_0^{(1)}$, $g = \hat{e}(P_0^{(2)}, P_{pub})$;

   e. chooses two cryptographic hash functions $\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $\mathcal{H}_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$.

   Let N_ℓ denote the set $\{1, \cdots, \ell\}$. The message space is $\mathcal{M} = \{0,1\}^n$ and the ciphertext space is $\mathcal{C} = N_\ell^\delta \times \{0,1\}^{\delta \cdot \ell} \times \mathbb{G}_1^{\ell+1} \times \{0,1\}^n$. The system's public parameters (which must be certified) are $pubParams = (\delta, q, n, \hat{e}, P_0^{(1)}, g, P_{pub}, \mathcal{H}_1, \mathcal{H}_2)$, and the *PKG* keeps s_0 and $P_0^{(2)}$ secret, while $params = (pubParams, s_0, P_0^{(2)})$.

3) `Key Generation` $(\omega, params)$: Let $\omega = \{i_j\}_{j=1}^t$ be the identity of a user $\mathcal{U}$, where each i_j is the index of an attribute. Let also ID_{i_j} be the string identifier of the i_j^{th} attribute, and $Q_{i_j} = \mathcal{H}_1(ID_{i_j})$ for $1 \leq j \leq t$.

[3] In the sequel, we shall use the notation $x \in_R X$ to indicate that the element x is chosen uniformly at random from the set X.

Then, the PKG picks $\mu \in_R \mathbb{Z}_q^*$, and secretly gives $\mathcal{U}$ her private key $D_\omega = (S_\omega, T_\mu, U_\mu)$, where $U_\mu = \mu P_0^{(1)}$, $T_\mu = s_0 P_0^{(2)} - \mu s_0 P_0^{(1)}$, and $S_\omega = \{D_{i_j}\}_{j=1}^{t}$ with $D_{i_j} = \mu \cdot Q_{i_j}$ for $1 \leq j \leq t$.

4) `Encrypt` $(d_1, d_2, m, \omega_1, \omega_2, pubParams)$: Given two threshold parameters d_1 and d_2, a message $m \in \mathcal{M}$, $pubParams$, and the identities $\omega_1 = \{i_j\}_{j=1}^{t_1}$ and $\omega_2 = \{u_v\}_{v=1}^{t_2}$ of an intended recipient, the encryptor:

   a. selects δ random polynomials F_1 and F_2, where $F_p(x) = P_{pub} + (\sum_{z=1}^{d_p-1} r_{(p,z)} x^z) P_0^{(1)}$ (with $p = 1, 2$ and $r_{(p,z)} \in_R \mathbb{Z}_q^*$ for $1 \leq z \leq d_p$);

   b. picks $r \in_R \mathbb{Z}_q^*$;

   c. computes X, $(Y_{i_j}^{(1)})_{j=1}^{t_1}$, and $(Y_{u_v}^{(2)})_{v=1}^{t_2}$ as follows: $X = rP_0^{(1)}$, $Q_j = \mathcal{H}_1(ID_j)$, and $Y_j^{(p)} = r(F_p(j) + Q_j)\ \forall\ j \in \omega_p$ for $p = 1, 2$;

   d. computes $\sigma_1, \sigma_2 \in \{0,1\}^\ell$ such that the e^{th} bit of σ_p is 1 if and only if $e \in \omega_p$ (with $p = 1, 2$);

   e. computes $Z = m \oplus \mathcal{H}_2(g^r)$ and outputs $c = (d_1, d_2, \sigma_1, \sigma_2, X, (Y_{i_j}^{(1)})_{j=1}^{t_p}, (Y_{u_v}^{(2)})_{v=1}^{t_2}, Z)$.

5) `Decrypt` $(c, D_{\omega'}, pubParams)$: Let $c = (d_1, d_2, \sigma_1, \sigma_2, X, (Y_{i_j}^{(1)})_{j=1}^{t_1}, (Y_{u_v}^{(2)})_{v=1}^{t_2}, Z)$ be a given ciphertext computed for the identities ω_1 and ω_2. Let also $D_{\omega'} = (S_{\omega'}, T_\mu, U_\mu)$ be a recipient's private key, where $S_{\omega'} = \{D_j\}_{j \in \omega'}$. For $p = 1, 2$, let Φ_p be a d_p-element subset of $\omega_p \cap \omega'$. Then, the recipient outputs $m = Z \oplus \mathcal{H}_2[\hat{e}(\frac{1}{\delta}(\sum_{j\in\Phi_1} \phi_j^{\omega_1} D_j + \sum_{j\in\Phi_2} \phi_j^{\omega_2} D_j) - T_\mu, X)^{-1} \cdot \hat{e}(U_\mu, \frac{1}{\delta}(\sum_{j\in\Phi_1} \phi_j^{\omega_1} Y_j + \sum_{j\in\Phi_2} \phi_j^{\omega_2} Y_j))]$, where each $\phi_j^{\omega_p} = \prod_{z \in \omega_p \setminus \{j\}} \frac{-z}{j-z}$ is the zero evaluation of the Lagrange coefficient of j with respect to ω_p.

For correctness, assume that $\delta = 1$ (since the case $\delta = 2$ follows similarly), and remark that:

$$\begin{aligned} & \hat{e}(\sum_{j\in\Phi_1} \phi_j^{\omega_1} D_j - T_\mu, X)^{-1} \\ = \ & \hat{e}(\mu \sum_{j\in\Phi_1} \phi_j^{\omega_1} Q_j, rP_0^{(1)})^{-1} \cdot \hat{e}(s_0 P_0^{(2)}, rP_0^{(1)}) \\ & \qquad \cdot \hat{e}(\mu s_0 P_0^{(1)}, rP_0^{(1)})^{-1}; \end{aligned}$$

$$\begin{aligned} & \hat{e}(U_{\omega'}, \sum_{j\in\Phi_1} \phi_j^{\omega_1} Y_j) \\ = \ & \hat{e}(\mu P_0^{(1)}, r \sum_{j\in\Phi_1} \phi_j^{\omega_1} F(j)) \cdot \hat{e}(\mu P_0^{(1)}, r \sum_{j\in\Phi_1} \phi_j^{\omega_1} Q_j) \\ = \ & \hat{e}(\mu P_0^{(1)}, rs_0 P_0^{(1)}) \cdot \hat{e}(\mu \sum_{j\in\Phi_1} \phi_j^{\omega_1} Q_j, rP_0^{(1)}); \end{aligned}$$

$$\begin{aligned} & \hat{e}(\sum_{j\in\Phi_1} \phi_j^{\omega_1} D_j - T_\mu, X)^{-1} \cdot \hat{e}(U_{\omega'}, \sum_{j\in\Phi_1} \phi_j^{\omega_1} Y_j) \\ = \ & \hat{e}(s_0 P_0^{(2)}, rP_0^{(1)}) \\ = \ & g^r. \end{aligned}$$

Optimization when Threshold Value is Maximal

In some access control scenarios, users are required to have *all* the attributes of a given attribute set in order to access a given resource. For instance, some attributes of a person may appear in all her biometric readings. These attributes may therefore be grouped in a special set, and used when encrypting documents for this person. When thABE is used to enforce BBAC, the aforementioned scenario can be modelled as follows: let ω be one set of t attributes; then decryptors must have keys associated with the t elements of ω. The scheme presented below shows how to obtain a thABE scheme with short and constant-size ciphertexts, when a user must have all the attributes of a given set in order to decrypt a particular ciphertext.

1) `Instance Generator` (k). As in New-thABE.

2) `Setup` (k). As in New-thABE with $\delta = 1$ and $\mathcal{C} = \{0,1\}^\ell \times \mathbb{G}_1^2 \times \{0,1\}^n$.

3) `Encrypt` $(d, m, \omega, pubParams)$: Given a message $m \in \mathcal{M}$, $pubParams$, and the identity $\omega = \{i_j\}_{j=1}^{t}$ of an intended recipient, the encryptor:

   a. picks $r \in_R \mathbb{Z}_q^*$;

   b. computes $X = rP_0^{(1)}$ and $Y = r(P_{pub} + \sum_{j=1}^{t} Q_{i_j})$ (where $Q_{i_j} = \mathcal{H}_1(ID_{i_j})\ \forall j = 1, \cdots, t$);

   c. computes $\sigma \in \{0,1\}^\ell$ such that the e^{th} bit of σ is 1 if and only if $e \in \omega$;

   d. computes $Z = m \oplus \mathcal{H}_2(g^r)$ and outputs $c = (\sigma, X, Y, Z)$.

4) `Decrypt` $(c, D_{\omega'}, pubParams)$: Let $c = (\sigma, X, Y, Z)$ be a given ciphertext computed for the identity ω (where $|\omega| = t$). Let also $D_{\omega'} = (S_{\omega'}, T_\mu, U_\mu)$ be a recipient's private key, where $S_{\omega'} = \{D_j\}_{j\in\omega'}$. Let $\Phi = \omega' \cap \omega$. Then, the recipient outputs

$$m = Z \oplus \mathcal{H}_2\left(\hat{e}(\sum_{j\in\Phi} D_j - T_{\omega'}, X)^{-1} \cdot \hat{e}(U_{\omega'}, Y)\right).$$

4 Efficiency

Table 1 compares the computational requirements of New-thABE with those of SW-thABE. d denotes the (sum of all) threshold value(s), while t denotes both number of attributes used for encryption and the number of attributes which identify an arbitrary user. M_X and A_X respectively denote computational costs of scalar multiplication and addition in the Abelian group X. R_X denotes the computational cost of uniformly selecting a random element in the set X. The computational cost of exponentiation in the group X is denoted by Ex_X, and $\mathbf{P}$ denotes the computational cost of a bilinear pairing operation. Note that pairings computations are, by far,

Table 1: Efficiency Comparison of New-thABE with SW-thABE.

Schemes		**Computational Requirements**	**Features**
SW-thABE	Public Parameters	$M_{\mathbb{G}_1} + 2R_{\mathbb{G}_1} + R_{\mathbb{Z}_q^*}$	1 Attribute Set 1 Threshold Value
	Key Generation	$(t^2+t)A_{\mathbb{G}_1} + t(d-1)Ex_{\mathbb{Z}_q^*} + (4t+t^2)M_{\mathbb{G}_1} + t(d-1)M_{\mathbb{Z}_q^*} + (d+t-1)R_{Zqs}$	
	Encryption	$tA_{\mathbb{G}_1} + Ex_{\mathbb{G}_2} + (2t+2)M_{\mathbb{G}_1} + \mathbf{P} + R_{\mathbb{Z}_q^*}$	
	Decryption	$d \cdot Ex_{\mathbb{G}_2} + d \cdot Inv_{\mathbb{G}_2} + (1+d)M_{\mathbb{G}_2} + 2d \cdot \mathbf{P}$	
New-thABE	Public Parameters	$M_{\mathbb{G}_1} + \mathbf{P} + 2R_{\mathbb{G}_1} + R_{\mathbb{Z}_q^*}$	Many Attribute Sets Flexible Threshold Values
	Key Generation	$A_{\mathbb{Z}_q^*} + (3+t) \cdot M_{\mathbb{G}_1}$	
	Encryption	$(2t)A_{\mathbb{G}_1} + Ex_{\mathbb{G}_2} + t(d-1)E_{\mathbb{Z}_q^*} + (1+2t) \cdot M_{\mathbb{G}_1} + t(d-1)M_{\mathbb{Z}_q^*} + d \cdot R_{\mathbb{Z}_q^*}$	
	Decryption	$(2d-1)A_{\mathbb{G}_1} + Inv_{\mathbb{G}_2} + (2d+2)M_{\mathbb{G}_1} + M_{\mathbb{G}_2} + 2\mathbf{P}$	

the most expensive of the operations considered above. For instance, [12] indicates that, for parameters providing practical security assurance (i.e. equivalent to 1024-bit RSA [20]), $\mathbf{P} \approx 4M_{\mathbb{G}_1}$, $M_{\mathbb{G}_1} \approx 100A_{\mathbb{G}_1}$, $A_{\mathbb{G}_1} \approx 14M_{\mathbb{Z}_q^*}$, $Inv_{\mathbb{Z}_q^*} \approx 16M_{\mathbb{Z}_q^*}$, and $M_{\mathbb{Z}_q^*}$ takes about 15 micro seconds in software. Moreover, the cost of an operation in $\mathbb{G}_2$ can be assumed to be comparable to the cost of the same operation in $\mathbb{Z}_q^*$. Furthermore, note that neither the computational cost of hash functions, nor that of additions in $\mathbb{Z}_q^*$ and exclusive OR operations are taken into account in Table 1.

Table 1 shows that the *Setup* procedure of SW-thABE and New-thABE have similar computational requirements. In particular, both feature a short constant set of public and private parameters. However, New-thABE's *Key Generation* algorithm is significantly less computationally expensive than SW-thABE's *Key Generation* procedure. This computational difference is partially shifted to the cost requirement of New-thABE's encryption algorithm. Indeed, SW-thABE's *Encryption* procedure requires much more computation than New-thABE's *Encryption* algorithm. Note however that the cost of New-thABE's *Encryption* algorithm is lower than SW-thABE's *Key Generation* algorithm, especially when the number of attributes used for encryption is high. The main advantage of New-thABE comes from its *Decryption* procedure which involves only 2 pairings, compared with $2d$ pairings in the case of SW-thABE's decryption algorithm. This gain is achieved by replacing pairing computations with scalar multiplications in $\mathbb{G}_1$. (The latter class of operations are significantly less expensive than the former one.) Furthermore, Table 1 emphasizes that New-thABE handles multiple attribute sets whose threshold values can be specified *at encryption time*.

5 Security

This section presents the security guarantees of our proposed thABE scheme.

Theorem 1. *Let k be a security parameter. Assume that the hash functions of New-thABE are random oracles. Suppose also that there exists an attacker $\mathcal{A}$ which has non-negligible advantage $\varepsilon(k)$, in time τ, against any challenger $\mathcal{C}h$, in the* IND-thABE-CPA *game. Then, there exists an algorithm $\mathcal{B}$ which solves the BDH problem, in time $O(\tau)$, with non-negligible advantage at least $\frac{\varepsilon(k)}{q_{\mathcal{H}_2}}$, where $q_{\mathcal{H}_2}$ is the number of $\mathcal{H}_2$ queries issued by $\mathcal{A}$ in the attack game.*

Proof: In this proof, we show how to construct $\mathcal{B}$ using $\mathcal{A}$. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two Abelian groups of prime order q, where $\mathbb{G}_1$ is additive and $\mathbb{G}_2$ is multiplicative. Let $P_0^{(1)} \in \mathbb{G}_1^*$ be a generator of $\mathbb{G}_1$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a bilinear pairing on $\mathbb{G}_1$, such that the BDH is assumed to be hard with respect to $\hat{e}$, and let $(\mathbb{G}_1, q, P_0^{(1)}, aP_0^{(1)}, bP_0^{(1)}, cP_0^{(1)})$ be a tuple for which a, b, c are unknown to $\mathcal{B}$. $\mathcal{B}$'s goal is to solve the BDH Problem by computing $\hat{e}(P_0^{(1)}, P_0^{(1)})^{abc}$ in polynomial time. To achieve this goal, $\mathcal{B}$ initiates an *IND-thABE-CPA* game with $\mathcal{A}$, using an arbitrary security parameter k.

For simplicity, we shall assume that New-thABE is only used to encrypt for one set of attributes at a time (instead of two). The proof we construct could be easily be adapted to deal with the case $\delta = 2$.

$\mathcal{B}$ first sets $n = poly_1(k)$ and $\ell = poly_2(k)$, using polynomials $poly_1$ and $poly_2$. Then $\mathcal{B}$ picks $s_0 \in_R \mathbb{Z}_q^*$, $\eta \in_R \mathbb{Z}_q^*$, and sets $P_0^{(3)} = \eta P_0^{(1)}$, $P_{pub} = aP_0^{(1)}$, $X^* = bP_0^{(1)}$, and $P_0^{(2)} = cP_0^{(1)}$. $\mathcal{B}$ also sets $g = \hat{e}(P_{pub}, P_0^{(2)})$ and defines two hash functions $\mathcal{H}_1^{sim} : \{0,1\}^* \to \mathbb{G}_1$

and $\mathcal{H}_2^{sim} : \mathbb{G}_2 \rightarrow \{0,1\}^n$ over which it has full control. Then, $\mathcal{B}$ sets $\mathcal{M} = \{0,1\}^n$, $\mathcal{C} = \mathbb{N}_\ell \times \{0,1\}^\ell \times \mathbb{G}_1^{\ell+1} \times \{0,1\}^n$, $pubParams = (q, n, \hat{e}, P_0^{(1)}, g, \mathcal{H}_1, \mathcal{H}_2)$, $params = (pubParams, a, P_0^{(2)}, P_0^{(3)})$, where $s_0 = a$ is not known by $\mathcal{B}$. $\mathcal{B}$ also picks $\mu \in_R \mathbb{Z}_q^*$. Throughout the attack game, μ will be used to "personalize" the keys sent to $\mathcal{A}$. (Since $\mathcal{A}$ is the only users requesting keys, there is no need to personalize the keys for many users.)

The rest of the proof consists of three sections. The *Hash Simulation* section shows how $\mathcal{B}$ simulates $\mathcal{H}_1^{sim}$ and $\mathcal{H}_2^{sim}$. The *Attack Game* section explains how $\mathcal{B}$ handles the queries of the attack game. Finally, the *Complexity and Probability* section derives the complexity and probability results of the theorem.

Hash Simulation:
For $\mathcal{H}_1^{sim}$-queries, $\mathcal{B}$ maintains a list $L_{\mathcal{H}_1^{sim}}$ whose entries have the form $(\alpha, h_{1,\alpha})$, where $\alpha \in \{0,1\}^*$ and $h_{1,\alpha} \in \mathbb{G}_1$ is the simulated value of $\mathcal{H}_1(\alpha)$. Thus, on input $\alpha \in \{0,1\}^*$, $\mathcal{B}$ proceeds as follows:

- If $L_{\mathcal{H}_1^{sim}}$ already has an entry $(\alpha, h_{1,\alpha})$, then $\mathcal{B}$ returns $h_{1,\alpha}$ as an answer to the $\mathcal{H}_1^{sim}$-query;
- Otherwise, $\mathcal{B}$ picks $h_{1,\alpha}$ uniformly at random in $\mathbb{G}_1$, adds $(\alpha, h_{1,\alpha})$ to $L_{\mathcal{H}_1^{sim}}$, and returns $h_{1,\alpha}$ as an answer to the $\mathcal{H}_1^{sim}$ query.

$\mathcal{B}$ also maintains a list $L_{\mathcal{H}_2^{sim}}$ for $\mathcal{H}_2^{sim}$-queries, where the entries of $L_{\mathcal{H}_2^{sim}}$ have the form $(\gamma, h_{2,\gamma})$, where $\gamma \in \mathbb{G}_2$ and $h_{2,\gamma} \in \{0,1\}^n$ is the simulated value of $\mathcal{H}_2(\gamma)$. Thus, on input $\gamma \in \mathbb{G}_2$, $\mathcal{B}$ proceeds as follows:

- If $L_{\mathcal{H}_1^{sim}}$ already has an entry $(\gamma, h_{1,\gamma})$, then $\mathcal{B}$ returns $h_{1,\gamma}$ as an answer to the $\mathcal{H}_1^{sim}$-query;
- Otherwise, two cases are distinguished:
  1) If, for some existing entry $(\gamma', h_{2,\gamma'})$ of $L_{\mathcal{H}_2^{sim}}$, the discrete logarithm of γ' with respect to $\hat{e}(P_0^{(1)}, P_0^{(2)})$ is equal to the discrete logarithm of γ with respect to $\hat{e}(P_0^{(1)}, P_0^{(3)})$, then $\mathcal{B}$ returns $h_{2,\gamma'}$. (Note that the equality of discrete logarithm can be tested using standard techniques [1]. Moreover, note that the search of a potential entry with equal discrete logarithm takes at most $q_{\mathcal{H}_2}$ trials.)
  2) Otherwise, $\mathcal{B}$ picks $h_{1,\alpha}$ uniformly at random in $\{0,1\}^n$, adds $(\alpha, h_{1,\alpha})$ to $L_{\mathcal{H}_1^{sim}}$, and returns $h_{1,\alpha}$ as an answer to the $\mathcal{H}_1^{sim}$ query.

Attack Game:
$\mathcal{B}$ handles $\mathcal{A}$'s queries as follows:

- *Phase 1*: Given an identity $\omega_i = \{i_j\}_{j=1}^{t_i}$, $\mathcal{B}$ computes $D_{\omega_i} = (S_{\omega_i}, T_\mu, U_\mu)$, where $U_\mu = \mu P_0^{(1)}$, $T_\mu = \eta P_{pub} - \mu P_{pub} = s_0 P_0^{(3)} - \mu P_{pub}$, and $S_{\omega_i} = \{D_{i_j}\}_{j=1}^{t_i}$ with $D_{i_j} = \mu \cdot Q_{i_j}$ and $Q_{i_j} = \mathcal{H}_1(ID_{i_j})$ for $1 \leq j \leq t_i$. $\mathcal{B}$ then sends D_{ω_i} to $\mathcal{A}$.
- *Challenge Phase*: $\mathcal{A}$ issues a threshold parameter d^*, an identity $\omega^* = \{i_j^*\}_{j=1}^{t^*}$ of its choice, and a pair (m_0, m_1) of equal-length plaintexts, such that, in *Phase 1*, no *Key Extraction* queries were issued on an identity ω such that $|\omega \cap \omega^*| \geq d$. $\mathcal{B}$ then picks a random bit $\beta \in \{0,1\}$, $Z^* \in_R \{0,1\}^n$, and defines a polynomial $F(x) = P_{pub} + (\sum_{z=1}^{d^*-1} r_z x^z) P_0^{(1)}$, where $r_z \in_R \mathbb{Z}_q^*$ for $z = 1, \cdots, d^*$. $\mathcal{B}$ also computes $Y_{i_j^*} = F(i_j^*) + h_{1,ID_{i_j^*}} X^*$ for $j = 1, \cdots, t^*$, where $(ID_{i_j^*}, h_{1,ID_{i_j^*}})$ is the entry of $L_{\mathcal{H}_1^{sim}}$ used to simulate the output of $\mathcal{H}_1(ID_{i_j^*})$. Then $\mathcal{B}$ computes σ^* according to the method described in NewthABE's *Encryption* procedure, and returns $c^* = (\sigma^*, X^*, (Y_{i_j^*}^*)_{j=1}^{t^*}, Z^*)$ to $\mathcal{A}$.
- *Phase 2*: $\mathcal{B}$ proceeds as in *Phase 1*.
- *Guess*: $\mathcal{A}$ sends $\mathcal{B}$ his guess β' and wins if $\beta' = \beta$. Regardless of $\mathcal{A}$'s success, $\mathcal{B}$ picks (uniformly at random) an entry $(\gamma, h_{2,\gamma})$ of $L_{\mathcal{H}_2^{sim}}$, and submits $\gamma \cdot \hat{e}(\mu P_{pub}, X^*) \cdot \hat{e}(\mu P_0^{(1)}, P_{pub})^{-1}$ as a solution of the BDH problem.

Complexity and Probability:
Let $c = (d, \sigma, X, (Y_{u_v})_{v=1}^t, Z)$ be a valid ciphertext computed for an identity ω. Then $L_{\mathcal{H}_2^{sim}}$ contains an entry of the form $(\gamma, h_{2,\gamma})$, where $\gamma = g^r = \hat{e}(P_{pub}, P_0^{(2)})^r = \hat{e}(P_0^{(1)}, P_0^{(2)})^{s_0 r}$ for some $r \in \mathbb{Z}_q^*$ such that $X = rP_0^{(1)}$. Suppose that $\omega_i = \{i_j\}_{j=1}^{t_i}$ is some identity for which $\mathcal{A}$ receives $D_{\omega_i} = (S_{\omega_i}, T_\mu, U_\mu)$ from $\mathcal{A}$. Then $U_\mu = \mu P_0^{(1)}$, $T_\mu = \alpha P_{pub} - \mu P_{pub} = s_0 P_0^{(3)} - \mu P_{pub}$, and $S_{\omega_i} = \{D_{i_j}\}_{j=1}^{t_i}$ with $D_{i_j} = \mu \cdot Q_{i_j}$ and $Q_{i_j} = \mathcal{H}_1(ID_{i_j})$ for $1 \leq j \leq t_i$. Suppose that Φ is a d-element subset of $\omega \cap \omega_i$. Then $\lambda = \hat{e}(\sum_{j\in\Phi} \phi_j^\Phi D_j - T_\mu, X)^{-1} \cdot \hat{e}(U_\mu, \sum_{j\in\Phi} \phi_j^\Phi Y_j) = \hat{e}(s_0 P_0^{(3)} - \mu P_{pub}, rP_0^{(1)}) \cdot \hat{e}(U_\mu, rP_{pub}) = \hat{e}(P_0^{(3)}, P_0^{(1)})^{s_0 r}$. So the discrete logarithm (DL) of λ with respect to $\hat{e}(P_0^{(3)}, P_0^{(1)})$ is equal to the DL of g^r with respect to $\hat{e}(P_0^{(1)}, P_0^{(2)})$. Thus, if $\mathcal{A}$ encrypts a message m, and decrypts the corresponding ciphertext with the keys it receives from $\mathcal{B}$, then $\mathcal{A}$ recovers m.

Moreover, remark that the only way for $\mathcal{A}$ to win the attack game with non-negligible probability is to, somehow, obtain attribute keys associated with a d-element subset Φ^* of ω^*, and to use c^* to decrypt Z^*. Note that the attribute keys provided by $\mathcal{B}$ do not enable $\mathcal{A}$ to perform such a decryption. Thus, $\mathcal{A}$ must somehow obtain a d-element set of *valid* attribute keys. Moreover, note that the decryption of Z^* involves the computation of $\mathcal{H}_2(\zeta)$, where $\zeta = \hat{e}(\sum_{j\in\Phi^*} \phi_j^{\Phi^*} D_j - T_\mu, X^*)^{-1} \cdot \hat{e}(U_\mu, \sum_{j\in\Phi^*} \phi_j^{\Phi^*} Y_j^*) = \hat{e}(s_0 P_0^{(2)} - \mu P_{pub}, bP_0^{(1)}) \cdot \hat{e}(U_\mu, P_{pub}) = g^b \cdot \hat{e}(\mu P_{pub}, rP_0^{(1)})^{-1} \cdot \hat{e}(U_\mu, P_{pub})$. Since

$g^b = \hat{e}(P_0^{(1)}, P_0^{(1)})^{abc}$, the solution of the BDH can be obtained by computing $\zeta \cdot \hat{e}(\mu P_{pub}, X^*) \cdot \hat{e}(\mu P_0^{(1)}, P_{pub})^{-1}$. Thus, if $\mathcal{B}$ picks a random entry $(\gamma, h_{2,\gamma})$ of $L_{\mathcal{H}_2^{sim}}$ and if $\gamma = \zeta$, then $\mathcal{B}$ solves the BDH problem.

Let $\varepsilon(k)$ be $\mathcal{A}$'s advantage in the *IND-thABE-CPA* game. Then, $\mathcal{B}$ solves the BDH problem with probability at least $\frac{\varepsilon(k)}{q_{\mathcal{H}_2}}$, where $q_{\mathcal{H}_2}$ is the number of $\mathcal{H}_2$ queries issued by $\mathcal{A}$ in the attack game. Moreover, since each query of the *IND-thABE-CPA* game requires $\mathcal{B}$ to make a polynomial number of search operations and a polynomial number of operations in $\mathbb{Z}_q^*$, $\mathbb{G}_2$ and $\mathbb{G}_1$, it follows that $\mathcal{B}$ solves the BDH problem in time $O(\tau)$ where τ is the running time of $\mathcal{A}$ in the *IND-thABE-CPA* attack game. **Q.E.D.**

A Note on Chosen Ciphertext Security

In [6], it is shown how to prove that the FullIndent scheme is IND-ID-CCA secure using the fact that the BasicIdent scheme is IND-ID-CPA secure, where FullIdent is the obtained from BasicIdent by applying the so-called Fujisaki-Okamoto padding [10]. In the same way, New-thABE can be transformed into a scheme which is IND-thABE-CCA secure. This can be done as follows: (1) transform New-thABE into a scheme Basic-New-thABE-Pub (using the method described in [6] to transform BasicIdent into BasicPub); (2) show that Basic-New-thABE-Pub is IND-CPA if the BDH is intractable (using the method described in the proof of Theorem 1); (3) show that New-thABE is IND-ID-CPA if Basic-New-thABE-Pub is IND-CPA (using the method described in the proof of Lemma 4.2 of [6]); (4) transform Basic-New-thABE-Pub into a scheme Full-New-thABE-Pub (by applying Fujisaki-Okamoto padding [10]); transform Full-New-thABE-Pub into a scheme Full-New-thABE (as FullIdentPub is transform into FullIdent in [6]); show that Full-New-thABE is IND-ID-CCA if Full-New-thABE-Pub is IND-CCA (using the method described in the proof of Lemma 4.6 of [6]); apply Lemma 4.5 of [6] to prove that Full-New-thABE-Pub is IND-CCA if Basic-New-thABE-Pub is IND-CPA; conclude that Full-New-thABE is IND-ID-CCA secure if the BDH problem is intractable. (Note that the above line of reasoning has been considered standard in the literature [11, 21].)

6 Conclusion

The aim of this paper was to describe a provably-secure efficient collusion-resistant threshold attribute-based encryption (thABE) scheme which handles multiple attribute sets with dynamically-specifiable threshold values. This scheme can be used to support practical *biometric-based* cryptographic access control.

To the best of our knowledge, the proposed scheme is the most efficient one of its class. In particular, the new scheme is significantly more efficient and flexible than Sahai and Waters'scheme. Moreover, the proposed scheme was proven secure in the random oracle model, under a standard number theoretic assumption.

An optimization of our scheme allows to produce constant-size ciphertexts, when intended decryptors must have all the attributes of given attribute sets. However, it remains an open question whether it is possible to design a thABE scheme featuring constant-size ciphertexts regardless of both the number of target attributes and the specified threshold values.

References

[1] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Proceedings of CRYPTO'00 on Advances in cryptology*, LNCS 1880, pp. 255–270, Springer-Verlag, 2000.

[2] J. Baek and Y. Zheng, "Identity-based threshold decryption," in *Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography (PKC'04)*, LNCS 2947, pp. 262–276, Springer-Verlag, 2004.

[3] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for deisigning efficient protocols," in *Proceedings of the the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, ACM, 1993.

[4] D. Boneh, X. Ding, and G. Tsudik, "Identity-based mediated RSA," in *Proceedings of the third International Workshop on Information and Security Applications (WISA'02)*, Jeju Island, Korea, 2002.

[5] D. Boneh, X. Ding, G. Tsudik, and C. Wong, "A method for fast revocation of public key certificates and security capabilities," in *Proceedings of the 10th USENIX Security Symposium*, pp. 297–308. USENIX, 2001.

[6] D. Boneh and M.K. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of CRYPTO'01 on Advances in cryptology*, LNCS 2139, pp. 213–229, Springer-Verlag, 2001.

[7] X. Boyen, "Reusable cryptographic fuzzy extractors," in *ACM Conference on Computer and Communications Security (CCS'04)*, pp. 82–91, ACM Press, 2004. (Available at `http://www.cs.stanford.edu/~xb/ccs04/`)

[8] J. G. Daugman, "How Iris recognition works," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, 2004.

[9] Y. Dodis, L. Reyzin, and S. Smith, "Fuzzy extractor: how to generate string keys from biometrics and other noisy data," in *Proceedings of EUROCRYPT'04 on Advances in Cryptology*, LNCS 3027, pp. 523–540, Springer-Verlag, 2004.

[10] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," LNCS 1666, pp. 537–554, Springer-Verlag, 1999.

[11] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Proceedings of ASIACRYPT'02 on*

Advances in cryptology, LNCS 2501, pp. 548–566, Springer-Verlag, 2002.

[12] K. Harrison, D. Page, and N. P. Smart, "Software implementation of finite fields of characteristic three, for use in pairing based cryptosystems," *London Mathematical Society Journal of Computing Mathemetics*, vol. 5, pp. 181–193, 2002.

[13] A. Jain, A. Ross, and S. Prabhakar, "Fingerprint matching using minutiae and texture features," in *Proceedings of the International Conference on Image Processing (ICIP)*, pp. 282–285, 2001.

[14] A. Juels and M. Wattenberg, "A fuzzy commitment scheme,", in *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS'99)*, pp. 28–36, ACM Press, 1999.

[15] B. Libert and J.-J. Quisquater, "Efficient revocation and threshold pairing based cryptosystems," in *Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing*, pp. 163–171, ACM Press, 2003.

[16] F. Monrose, M. K. Reiter, Q. Li, D. Lopresti, and C. Shih, "Towards voice generated cryptographic keys on ressource constrained devices," in *Proceedings of the 11th USENIX Security Symposium*, pp. 283-296, 2002.

[17] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel, "Cryptographic key generation from voice," in *Proceedings of the IEEE Conference on Security and Privacy*, pp. 202–213, IEEE Press, 2001.

[18] F. Monrose, M. K. Reiter, and S. Wetzel, "Password hardening based on key-stroke dynamics," in *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS'99)*, pp. 73–82, ACM Press, 1999.

[19] D. Nali, C. Adams, and A. Miri, "Using mediated identity-based cryptography to support role-based access control," in *Proceedings of the 7th Information Security Conference (ISC'04)*, LNCS 3225, pp. 245–256, Springer-Verlag, 2004.

[20] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, 1978.

[21] A. Sahai and B. Waters, "Fuzzy identity based encryption," in *Proceedings of EUROCRYPT'05 on Advances in Cryptoglogy*, LNCS 3494, pp. 457–473, Springer-Verlag, 2005.

[22] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO'84 on Advances in Cryptology*, pp. 47–53. Springer-Verlag, New York, Inc., 1984.

[23] D. Yao, N. Fazio, Y. Dodis, and A. Lysyanskaya, "ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption," in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS'04)*, pp. 354–363, ACM Press, 2004.

Deholo Nali is a Ph.D. student in computer science, at the University of Ottawa. He holds a M.Sc. in mathematics. His research interests include the design and analysis of identity-based cryptographic protocols.

Carlisle Adams is an Associate Professor in the School of Information Technology and Engineering (SITE) at the University of Ottawa. Prior to his academic appointment in 2003, he worked for 13 years in industry in the design and standardization of a variety of cryptographic and security technologies for the Internet. His research and technical contributions include the CAST family of symmetric ciphers, protocols for authentication and management in PKI environments, and an architecture and policy language for access control in electronic networks. Dr. Adams is co-author of Understanding PKI: Concepts, Standards, and Deployment Considerations, Second Edition (Addison-Wesley, 2003).

Ali Miri received his BSc and MSc in Mathematics from the University of Toronto in 1991 and 1993 respectively, and his PhD in Electrical and Computer Engineering from the University of Waterloo in 1997. Having worked as an NSERC Postdoctoral Fellow at the University of Waterloo and the University of Toronto, he joined the School of Information Technology and Engineering (SITE) at the University of Ottawa in July of 2001, where he is currently working as an associate professor. His research interests include security and privacy technologies and their applications in e-business and e-commerce, such as network security and the role of Public Key Cryptography.