Outsourced CP-ABE with Whitebox Accountability in IoT Systems

Ambili K N, Jimmy Jose

Department of Computer Science and Engineering, National Institute of Technology Calicut, India
Email: ambili_p180002cs@nitc.ac.in

Cryptography based on identity and attributes enhances the chance of secure communication on a large scale. Several attribute-based encryption schemes achieve different objectives when used in various protocols. Most of these are suitable for large systems like cloud services. There are a few protocols which focus on reducing the computational overhead for lower end devices like Internet of Things sensors and actuators. It is desirable to have a mix of features in protocols for IoT security architecture. We first propose a scheme to ensure accountability in CPABE scheme FAME. The protocol is proven CPA-secure with full security in random oracle model. We also prove its accountability. We also propose a hybrid protocol that enforces user accountability and outsourced decryption in IoT systems and achieve full security in replayable chosen ciphertext attack (RCCA) under random oracle model.

Keywords: CPABE; Accountability; Outsourced decryption; IoT Systems; RCCA security

Received 09 August 2021; revised –00 Month 2009

1. INTRODUCTION

Attribute based encryption (ABE) was introduced in 2005 [1] broadening the scope to different types of systems. ABE schemes were introduced as an extension to identity-based encryption (IBE) [2] schemes which associated users with their identities. ABE schemes built upon this by associating users with a set of attributes and their identities. The cryptographic primitives in ABE are achieved using access policy.

ABE schemes make use of access policy embedded in access structures to map user attributes. The schemes wherein keys are based on attributes and access policy is transmitted together with the ciphertext are known as ciphertext-policy attribute-based encryption (CP-ABE). Other schemes make use of access policy to generate keys while using attributes for encryption. These are known as key-policy attribute-based encryption schemes (KP-ABE). The current work focuses on enhancing CP-ABE.

ABE has been exploited by researchers in various ways to provide fine-grained access control in cloud [3]. Several features have been introduced in various protocols. The paper hints at combining ABE schemes with different features to generate novel protocols that may be adapted to various Internet of Things (IoT) architectures.

A typical architecture for IoT makes use of cloud services in different ways. The innovative use of sensors and actuators in various real-life situations has made large amount of data available for storage and analysis. The secure storage of encrypted data is essential to facilitate quick and authentic analysis. The limited memory and computational power of IoT devices call for innovative methods of transformation of encrypted data for secure storage on cloud servers and also their decryption with less number of computations. The parties involved in communication need to be accountable for their operations. The decryption operation should be as simple as possible for easy computation on IoT devices. We discuss accountability and outsourced decryption as they are important features of protocol with ABE for IoT architecture.

1.1. Accountability

Public key cryptography (PKC) has the ability to ensure accountability of data items with digital signatures. The increase in the number of parameters or attributes that define a user make it difficult to achieve accountability in ABE schemes. Nevertheless, there have been various attempts to propose accountable ABE schemes. Several schemes achieve white-box traceability in which the malevolent users do not tweak the decryption algorithm or the secret key. The schemes which permit changes to decryption algorithm and the secret key by malevolent users during public auditing is more difficult to achieve and are known as blackbox traceable.

Accountable CP-ABE is essential to find the malevolent users who may delegate their keys to other users and malicious authorities which misuse the role of key management. A general solution has been to use multi-authority schemes wherein the authorities as well as the user are accountable. With the increasing number of IoT devices and data explosion, a simpler approach would facilitate safer storage of encrypted data. The domination of a single authority needs to be overcome while maintaining the accountability of user and authorities involved.

1.2. Outsourced Decryption

The increase in the number of miniature devices demand cryptographic systems with lesser number of computations. The applications like email on mobile phones is a typical example which can be made faster with lighter decryption key. This can be achieved by outsourcing complex operations of decryption. The extraction of data by IoT actuators from servers may also be made easier with outsourced decryption.

1.3. Our contribution

The high end servers in cloud architecture allows us to accommodate complicated methods to achieve confidentiality and accountability. The execution of complex operations involved in ABE system is time consuming on IoT devices and would hinder the performance of the overall system. Often, the servers in cloud misuse their role in key management. This leads to the key escrow problem [4]. Though most of the proposed ABE schemes prevent user collusion attacks, very few achieve accountability. These factors inspire us to think of designing a protocol which can make ABE practical on IoT devices and ensure end-to-end accountability. The user can establish a secure session with any authority that can successfully run pairing based operations and all other authorities will have only transformation key to analyse the encrypted data. We propose a CP-ABE protocol with following features:

- fast encryption
- white-box traceability
- secure outsourcing of decryption
- less computational overhead during decryption
- suitability for IoT deployment
- security analysis focusing on indistinguishability under chosen plaintext attack, full security in random oracle model, accountability and RCCA security for outsourced decryption

2. RELATED WORK

There are different approaches to public key cryptography. Identity based encryption [2, 5] is one among the new approaches.The scope of identity based encryption has gone beyond biometric identities by the addition of descriptive attributes. The concrete method proposed in [1] validates this. Two complementary forms of ABE were formalized in [6] as KP-ABE and CP-ABE. CP-ABE was designed earlier as a powerful public key primitive for access control by Bethencourt et.al.[7]. The recent survey paper on attribute based encryption for cloud computing [3] describes different features of ABE schemes.

One of the early research works on accountable ABE scheme is by Li et. al. [8]. The need for accountability arises due to the possibility of malicious behaviour of the parties involved. The key escrow problem reduces the trust in authorities. The malevolent behavior of users may cause the issue of key delegation. It becomes difficult to trace the malicious users in the system. The research on ABE takes into account two types of accountability. The schemes with white-box traceability assume that the parties involved in the scheme are faithful to the auditor. The focus of [9, 10] are on white-box traceability. The malicious user could manipulate or hide the tweaked decryption algorithm and the secret key that she uses from the auditor. The research work in [11, 12] achieve blackbox traceability. The blackbox traceable CP-ABE scheme of [11] is provably secure and suitable for cloud storage. The blackbox traceability in [12] can be used to detect malevolent decryption devices. The recent research also shows multi-authority schemes that have been proposed as a solution to key delegation issues [13, 14, 15, 16, 17].

Outsourced decryption was formalised by Dan Boneh in [18]. Due to the computational overhead, the practical implementation of ABE schemes on devices is challenging. As a solution to reduce computational effort, different methods for outsourced computation were proposed in [19, 20, 21]. The notion of proxy re-encryption is introduced in [22, 23]. The generation of transformation keys allows ciphertext to be shared across servers without loss of security. The functional encryption methods like homomorphism, allow certain computations to be done on ciphertext using specified inputs while keeping the underlying message undecipherable by unauthorised user.

There are few research papers that focus on making ABE schemes suitable for IoT deployment. Accountability in multi-authority environments is achieved in [24].

The proposal in [25] combines the outsourcing and accountability but its application is limited to cloud. A KP-ABE scheme with application to IoT is described in [26]. In general, there is a dearth of research works supporting efficient extension of attribute based cryptography to IoT deployment with various features. We propose a CP-ABE protocol with accountability and outsourced decryption. The relevant preliminaries are described in 3. CPA-secure protocol is described in 6.

In the next section, we provide the preliminaries needed to understand the rest of the paper. The primary actors involved are accountable authority and the user. Both are equally accountable for their behavior on the network. This is ensured by incorporating parameters from both of these into the keys generated in the system. Other authorities involved in this framework get a transformation key which has contributions from both the accountable authority and the user.

The accountable authority handles attribute set which are disjoint. Each user has a number of IoT devices she uses which map to a number of attributes. Among these, few identify the user uniquely. Each user generates a decryption key after appropriate communication with accountable authority. A ciphertext can be created by anyone using a policy. Decryption is possible only by a particular user. The apt combination of user attributes make it possible to use it on IoT devices owned by the user. The usage of transformation key makes decryption operation computationally easy. The mathematical preliminaries and definitions needed to describe this protocol are described in the next section.

3. PRELIMINARIES

Let G_1, G_2, and G_T, be multiplicative cyclic groups of prime order p. Let g be the generator of G_1, G_2 and $e : G_1 \times G_2 \Rightarrow G_T$ be a bilinear pairing with the following properties:

- Bilinearity: $e(u^a, v^b) = e(u, v)^{ab}$
- Non-degeneracy: $e(u, v) \neq 1_T$, where 1_T is the identity element of group G_T.
- Computability: there is an efficient algorithm for computing $e(u, v)$ for all $u \in G_1$ and $v \in G_2$

We consider prime order bilinear asymmetric groups. The group operations are assumed to be computable in polynomial time with respect to the security parameter λ. Here, G_1, G_2 are the source groups and G_T is the target group. The groups are chosen in such a way that their orthogonality under the bilinear map is preserved. The assumptions on subgroup decision problem for three prime numbers also hold with these groups [27]. Decisional Bilinear Diffie-Hellman also hold on these groups.

The ABE scheme begins by mapping attributes of universe to random elements. An access policy is framed based on the system requirements. These are mapped to the access structure, linear secret sharing scheme (LSSS). LSSS is itself represented as a matrix and is used encryption. More mathematical details of encryption method are available in [28]. The IoT system uses the encryption method enhanced with accountability and outsourced decryption.

4. SECURITY MODEL

We analyse the security of the protocol using three different games which are Indistinguishability under Chosen Plaintext Attack (IND-CPA) [29], Game 1 for accountability and Game 2 for unlinkability [14]. We consider these in the random oracle setting [30]. These games are defined in the following subsections suitable to verify the proposed protocol.

4.1. IND-CPA game

Selective Attack The adversary selects two access policies and submits them to the challenger.

Setup The challenger executes Setup for a security parameter λ and returns the public key to the adversary.

Phase 1 adversary is allowed to perform a polynomial number of queries on particular attribute set L.

Challenge The adversary outputs two equal length messages M_0, M_1 of equal length. The challenger tosses a coin and randomly picks $b \in 0, 1$ and the message M_b to be encrypted under the provided policy and returns the ciphertext to the adversary.

Phase 2 Adversary continues to issue queries as in phase 1.

Guess Adversary outputs a bit b' as a guess of b.

Here, we use proofs based on a sequence of games in which the keys and ciphertext generated are indistinguishable as in dual system encryption **??**. The attacker cannot distinguish one game from the next. In dual system encryption, the ciphertexts as well as the private keys can take on one of two indistinguishable forms. A private key or ciphertext is said to be normal if they are generated respectively from system's key generation or encryption algorithm. A semifunctional private key will be able to decrypt all normally generated ciphertexts. However, decryption will fail if decryption of a semifunctional ciphertext is attempted. The semifunctional ciphertexts are decryptable only by normal private keys.

4.2. Game 1 for accountability

The adversary is allowed to corrupt a few secret keys for attribute sets of its choice. Here, two users with the same attribute set are not considered. The adversary is supposed to generate a decryption box from which one can not extract any of the identities corresponding to the secret key that was corrupted by the adversary. Below we describe the game.

Setup. Given a security parameter 1^λ the challenger B runs Setup and gets $(params, (pk_k, sk_k)$ for $k \in 1, \ldots N)$, where pk, sk denote public key and secret key respectively. The params, $pk_{k_{k\in\{1,...,N\}}}$, and

$sk_{k k \in \{1, ..., N\}}$ are given to the adversary A.

Phase 1. A queries the key generation oracle on attribute set S of its choice using set of global identities GID. This can be done polynomial number of times. At the end of this phase, A determines a decryption box and submits it to B.

Trace. B executes the Trace protocol with the decryption box. B determines the identities using the decryption box and outputs an identity $GID^{'}$.

A wins the game if $GID^{'} \notin GID$.

4.3. Game 2 for unlinkability

The goal of the adversary is to get information on the attributes of a given identity. The game is described below.
Setup. Given a security parameter 1^{λ} the challenger B runs Setup and gets $(params, (pk_k, sk_k), k \in \{1, \ldots N\})$, where pk, sk denote public key and secret key respectively. These are given to A.

Phase 1. A queries key generation oracle for attribute set S using identity GID polynomial number of times.

Guess. Finally, the adversary outputs attribute set $S^{'}$ and the corresponding identity GID'.

A wins the game if $GID^{'}$ and $S^{'}$ are linkable and the success probability is greater than random guess with non-negligible probability.

Now, we provide the definitions of security in terms of the games defined above to validate the security of our protocols.

Definition : ABE scheme is said to be accountable if the probability that an adversary can win the games Game 1 and Game 2 is negligible.

Definition : ABE scheme is CPA-secure if the probability that an adversary win the IND-CPA game is negligible.

Definition : A CP-ABE scheme is said to be accountable with secure outsourced decryption if the probability that the adversary wins the IND-CPA, Game 1 and Game 2 simultaneously is negligible.

The above games and definitions will be used to verify CPA-secure construction.

4.4. RCCA Security

We define a slightly modified version of game called replayable chosen ciphertext attack such that it achieves full security. Here, the adaptive or full security implies that the access structure is not fixed before the game begins. This game is defined to verify security after outsourced decryption is included in the protocol.

Suppose there exists a PPT adversary A that can attack our scheme with accountability and outsourced decryption in the full security RCCA model with advantage ϵ.We build a simulator that can attack the underlying accountable scheme in adapative CPA model with advantage ϵ minus a negligible amount.

Init The simulator B runs A.

Setup The simulator B obtains the public parameters and a description of hash function H. It sends these to A as public parameters.

Phase 1 The simulator B initializes empty tables T and T_1, an empty set D and an integer $j = 0$. It answers adversaries queries as follows:
$H_1(R, M)$ where $R = (r1, r2)$: If there is an entry (R, M, s) in T_1 return s else choose a random $s \in Z_p$, record (R, M, s) in T_1 and return s.

Create(S, (M,Π)) : B sets $j = j + 1$. If S satisfies (M, Π), then the transformation key is chosen by running the key generation algorithm to obtain secret key $SK^{'}$. Set $TK = SK^{'}$ and $SK = (d, TK)$. If S does not satisfy (M, Π), pick a random access matrix that satisfies the set of attributes S and obtain $SK^{'}$. Pick $z \in Z_p$, set the transformation key by raising secret key to z and then the private key as (z, TK). Finally, store $(j, S, (M, \Pi), SK, TK))$ in table T and return TK to the adversary A.

Corrupt(i) : The adversary A asks to corrupt the i^{th} entry and sets $D := D \cup S$. B does so and returns the secret key SK to the adversary A.

Decrypt(i, CT) : If the access structure in i^{th} entry satisfies the attributes, decrypt the ciphertext using i^{th} entry. Otherwise, create new entries in T_1 and T and then decrypt.

Challenge : The adversary A submits message pairs (M_0, M_1). B generates the ciphertext and returns them to A.

Phase 2 : This phase is same as phase 1 query except that if response to decryption is the challenge messages, then a random answer is returned.

Guess : A guesses the message bit b. B searches through table T_1 for R. If found, it outputs the message bit else a random bit is output.

Ths advantage that A has in winning the game is same as advantage that A has in breaking B's assumption.

5. ASSUMPTIONS

The protocol is proven based on certain assumptions. These are restated below for clarity.

5.1. DLIN

An asymmetric bilinear pairing group satisifies the decisional bilinear assumption (DLIN) if for all PPT adversaries A, the advantage that the adversary has is negligible in the following:

$$Adv^{A}_{DLIN}\lambda = Pr[A(1^{\lambda}, par, D, T_0) = 1] - Pr[A(1^{\lambda}, par, D, T_1) = 1]$$

is negligible in λ where $par = (p, G, H, G_T, e, g, h)$, $a_1, a_2 \in Z_p^*$, $s_1, s_2, s \in Z$,
$D = (g^{a_1}, g^{a_2}, h^{a_1}, h^{a_2}, g^{a_1 s_1}, g^{a_2 s_2}, h^{a_1 s_1}, h^{a_2 s_2},)$,
$T_0 = (g^{s_1+s_2}, h^{s_1+s_2})$, $T_1 = (g^{s}, h^{s})$.

5.2. q-Strong Diffie-Hellman Inversion

Let (p, G_1, G_2, G_T, e) be an asymmetric bilinear pairing group generated with generators g_1, g_2 of G_1, G_2 respectively and let q be a polynomial in λ. For a randomly chosen element $z \in Z_p$, the q-DHI is

$$Pr[A(g_1, g_1^{z}, g_1^{z^2}, ..., g_1^{z^q}, g_2^{z^q}) = g_1^{1/z}] = negl(\lambda) \quad (1)$$

where λ is the security parameter.

5.3. q-Strong Diffie-Hellman

Let (p, G_1, G_2, G_T, e) be an asymmetric bilinear pairing group generated with generators g_1, g_2 of G_1, G_2 respectively. The q-Strong Diffie Hellman (q-SDH) holds, if for every PPT adversary A,

$$Pr[A(g_1, g_1^{x}, g_1^{x^2}, ..., g_1^{x^q}, g_2, g_2^{x^q}) = (g_1^{1/(x+c)}, c)] = negl(\lambda) \quad (2)$$

where λ is the security parameter.

6. CPA-SECURE CONSTRUCTION

The protocol is designed using bilinear pairing groups G and H. The encryption, transformation and decryption operations ensure confidentiality. Key sanity check, trace and audit operation help achieve accountability. We describe the design of the protocol in 7.1. The security analysis of the protocol is provided in 6.2.

6.1. Design

Setup(1^{λ}) The pair of asymmetric groups G, H and group needed for bilinear pairing G_T are generated using the group generation algorithm GroupGen(1^{λ}). The output is a tuple (p, G, H, G_T, g, h) where p is the prime order of the group G_T, g, h are the generators of the groups G, H respectively and G_T is the target group of the bilinear pairing.

Pick a_1, a_2, b_1, b_2 from Z_p^*.

Pick d_1, d_2, d_3, α, a from Z_p.

Output the public key as $h, H_1 = h^{a_1}, H_2 = h^{a_2}, T_1 = e(g, h)^{d_1 a_1 + d_3}, T_2 = e(g, h)^{d_2 a_2 + d_3}, g, u, x, v, w, g^{a}, e(g, h)^{\alpha}$.

Output master key as $g, h, a_1, a_2, b_1, b_2, g^{d_1}, g^{d_2}, g^{d_3}$.

KeyGen$(1^{\lambda}, pp, msk, S = A_1....A_k, id)$ The key generation algorithm computes user secret key using public parameters pp, set of attributes S and the master secret key msk. The identity id corresponding to a user is used in the process.

Pick r_1, r_2 from Z_p and compute

$$sk_0 = (h^{b_1 r_1}, h^{b_2 r_2}, h^{r_1 + r_2}) \quad (3)$$

For all $y \in S$, and $t = 1, 2$ compute

$$sk_{y,t} = H(y1t)^{b_1 r_1/a_t}.H(y2t)^{b_2 r_2/a_t}.H(y3t)^{r_1 + r_2/a_t}.g^{\sigma_y/a_t} \quad (4)$$

where σ_y is chosen from Z_p.

Set $sk_y = sk_{y,1}, sk_{y,2}, g^{1/\sigma_y}$.

Also compute

$sk_q = H(011t)^{b_1 r_1/a_t}.H(012t)^{b_2 r_2/a_t}.H(013t)^{r_1 + r_2/a_t}$

$sk_{t'} = g^{d_t}.sk_q.g^{\sigma^{'}/a_t}$

Set $sk_{'} = sk_1^{'}, sk_2^{'}, g^{d_3}.g^{1/\sigma^{'}}$.

We use the following mapping of values to new variables $K, K^{'}, L, L^{'}, K_{\tau,1} K_{\tau,2}$ to generate an accountability component sk_{acc}.

$\tau = y$ $a = g^{\sigma^{'}/a_1}$ $l = g^{\sigma^{'}/a_2}$ $c = g^{\sigma^{'}/a_3}$ $\alpha = g^{b_1 r_1/a_1} + g^{b_2 r_2/a_1} + g^{(r_1 + r_2)/a_1}$ $w = g^{b_1 r_1/a_2} + g^{b_2 r_2/a_2} + g^{(r_1 + r_2)/a_2}$ $u = g^{b_1 r_1/a_3} + g^{b_2 r_2/a_3} + g^{(r_1 + r_2)/a_3}$ $x = g^{d_1 \sigma^{'}}/a_1 + g^{d_2 \sigma^{'}}/a_2 + g^{d_3 \sigma^{'}}/a_3$ $v = g^{d_1 \sigma_{'}/a_1} + g^{d_2 \sigma_{'}/a_2} + g^{d_3 \sigma_{'}/a_3}$
$l_{\tau} = \sigma_y$

Choose random $c \in Z_p^*$ and $l, l_1, .., l_k \in Z_p$.

$K = g^{\alpha/(a+c).w^{l}}$
$K^{'} = c$
$L = g^{l}$
$L' = g^{al}$
$K_{\tau,1} = g^{l,\tau}, K_{\tau,2} = (u^{A_{\tau}} x)^{l_{\tau}} v^{1/(a+c)l}{}_{\tau \in [k]}$.

Set $sk_{acc} = K, K^{'}, L, L^{'}, \{K_{\tau,1}, K_{\tau,2}\}_{\tau \in [k]}$.

A table T is maintained by the key generating party which has two columns namely identity id and value c.

Set secret key $sk_{id,S}$ as

$$SK = sk_0, sk_y, sk^{'}, sk_{acc} \quad (5)$$

where there are k attributes.

The random numbers r_1, r_2 may be chosen using a zero knowledge protocol to ensure participation of user and the authority in key generation.

Encrypt The encryption uses public key pk, secret sharing scheme π with matrix M to encrypt message msg. This operation is done by any sender willing to send data to a user with public key pk. The sender picks s_1, s_2 from Z_p and computes
$ct_0 = (H_1^{s_1}), H_2^{s_2}, h^{s_1 + s_2}$. Suppose M has n_1 rows and

n_2 columns. Then, for $i = 1, 2, .., n$ and $l = 1, 2, 3$, compute
$ct_{i,l} = H(\pi(i)q_1)^{s_1}.H(\pi(i)q_2)^{s_2}.\Pi_{j=1}^{n_2}[H(0jq1)^{s_1}.H(ojq2)^{s_2}]^{(M)_{i,j}}$

where $(M)_{i,j}$ denotes (i, j)th element of matrix M.

Set $ct_i = (ct_{i,1}, ct_{i,2}, ct_{i,3})$.

Pick random $s \in Z_p^*$.

Compute $C = msg.e(g, h)^{\alpha s}, C_0 = g^s, C_0^{'} = g^{as}$.

Compute $ct^{'} = T_1^{s_1}.T_2^{s_2}.C$.

Output ciphertext as $ct_0, .., ct_n, C_0, C_0^{'}$.

Decryption If the set of attributes in the secret key satisfies the access structure associated with the ciphertext, then there exists constants $\gamma_{i_{i \in I}}$that satisfy the access structure.

$num := ct^{'} \times e(\Pi ct_{i,1}^{\gamma_i}, sk_{0,1}) \times e(\Pi ct_{i,2}^{\gamma_i}, sk_{0,2}) \times e(\Pi ct_{i,3}^{\gamma_i}, sk_{0,3})$

$den := e(sk_1^{'} \times \Pi sk_{\Pi(i),1}^{\gamma_i}, ct_{0,1}) \times e(sk_2^{'} \times \Pi sk_{\Pi(i),2}^{\gamma_i}, ct_{0,2}) \times e(sk_3^{'} \times \Pi sk_{\Pi(i),3}^{\gamma_i}, ct_{0,3})$

This step outputs the decrypted text by computing num/den.

Key Sanity Check and Trace The first step is to remove the user generated decryption power z. The integrity of the protocol is heavily dependent on the secret key z possessed by the user.

The algorithm takes as input the public parameters and a secret key sk from transformation key TK. The following checks are done on sk_{acc}. $e(L^{'}, g) = e(L, g^a)$ $e(K, g^a g^{K^{'}}) = e(g, g)^{\alpha} e(L^{'} L^{K^{'}}, w)$ $\exists \tau \in [k]$ such that $e(K_{\tau,2}, g)e(L^{K^{'}} L^{'}, v) = e(K_{\tau,1}, h)e(K_{\tau,1}, u)^{A_\tau}$.

Trace The algorithm for key sanity check is run to check if secret key is a well-formed decryption key. The Trace algorithm searches for $K^{'}$ in the identity table. If found, it outputs the corresponding identifier to identify the malicious user which could recover the secret of the access scheme being used.

6.2. Analysis

6.2.1. Correctness Analysis

The base algorithm used for the proposed scheme is [28] which is proven CPA-secure under random oracle model. The transformation key generation and usage is similar to [31]. The parameters for accountability added while generating the key do not change any of the parameters involved. The correctness of the proposed scheme hence follows from the correctness of the FAME scheme.

6.2.2. Security Proof

We consider the adversarial model and games described in Section 4 to analyse the security of our proposal. We refer to our proposal as $\sum_{accCPABE}$. It is based on FAME [28] and we refer to it as $\sum_{CPABE}$. FAME is proven to be CPA-secure and achieve full or adaptive security. We prove that our protocol is CPA-secure on the same direction as the proof of FAME. The proof is based on a sequence of games which are indistinguishable from one another.

Let *Samp* be an algorithm that on input a prime p outputs two matrices Z and z

$$\begin{bmatrix} \mathrm{u}_1 & 0 \\ 0 & \mathrm{u}_2 \\ 1 & 1 \end{bmatrix},$$

$$\begin{bmatrix} \mathrm{u}_2^{-1} \\ \mathrm{u}_2^{-1} \\ -1 \end{bmatrix} \text{ where } u_1, u_2 \leftarrow Z_p^*.$$

We describe a slightly modified version of IND-CPA game used in FAME, that is $EXPT_{XFAME,A}(i^\lambda, b)$ called $Hyb1a$.

To begin with the challenger *Chal* sets up the ABE scheme as follows:

Setup Run group generation algorithm to obtain (p, G, H, e, g, h). Pick A, a and $B, b$by running $Samp(p)$ and d_1, d_2, d_3 from Z_p. Let d denote the transpose of the column vector (d_1, d_2, d_3). Set the public key as $([A], [dA])$. Set the master secret key as $(g, h, A, B, [d])$. The challenger *Chal* simulates the random oracle using two lists P and Q. The list L has entries of the form (x, W_x) and (j, U_j) where x is an arbitrary binary string, j is a positive integer and W_x, U_j are 3×3 matrices over Z_p. The list Q ahs entries of the form (q, r) where q is either xlt or $0jlt$ for $l \in \{1, 2, 3\}$ and $t \in \{1, 2\}$ and r is an element in G.

The adversary A can make one of three types of oracle queries:

1. xlt : *Chal* checks if $(xlt, r) \in Q$ for some r or not. If found, then $r := [(W_x^T A)_{l,t}]$ is computed, (xlt, r) is added to Q and r is returned. Else, it picks a random $W_x \leftarrow Z_p^{3\times3}$, adds (x, W_x) to L, compute $r := [(W_x^T A)_{l,t}]$, add (xlt, r) to Q and return r.
2. $0jlt$: *Chal* checks if $(0jlt, r) \in Q$ for some r or not. If found, return r. Else check if $(j, U_j) \in L$ for some U_j or not. If found then compute $r := [(U_j^T A)_{l,t}]$, add $(0jlt, r)$ to Q and return r. Else it picks $U_j \leftarrow Z_p^{3\times3}$, adds (x, U_j) to L, compute $r := [(U_j^T A)_{l,t}]$, adds $(0jlt, r)$ to Q and return r.
3. Anything else say q: *Chal* checks if $(q, r) \in Q$ for some r or not. If found, return r. Otherwise a random element $R^{'}$ is picked from G, $(q, r^{'})$ added to Q and $r^{'}$ returned.

Key generation: When the adversary A makes a key query S, *Chal* retrieves W_y for every $y \in S$ and U from list L. If one of them is not available, then a random 3×3 matrix is generated and the list L updated accordingly.

Now, pick r_1, r_2, σ' from Z_p and σ_y from Z_p for $y \in S$. Let $r := (r_1, r_2)^T$ and compute $sk_0 = [Br]$
$sk_y = [W_y Br + \sigma_y a]$
$sk' = [d + U_1 Br + \sigma' a]$
$sk_{acc} = [\sigma' a + BrU_1 + d\sigma' a]$ for all $y \in S$. Then, return $(sk_0, \{sk_y\}_{y \in S}, sk', sk_{acc})$ as the key.

Encryption : When A sends messages msg_0, msg_1 and a policy (M, π), $Chal$ retrieves $[(W^T_{\pi(i)}A)_{l,t}]$ and $[(U^T_j A)_{l,t}]$, for all $i = 1, ..n_1$, $j = 1, ..n_2$, l, t from the list Q. Now pick $s_1, s_2 \leftarrow Z_p$ and set s to be $s_1, s_2$and compute
$ct_0 = [As]$
$ct_i = [W^T_{\pi(i)}As + \Sigma^{n_2}_{j=1}(M)_{i,j}U^T_j As]$
$ct' = [d^T As] \times msg_b$ for $i = 1, ..n_1$

Return ciphertext as $(ct_0, ct_1, ..ct_{n_1}, ct')$.

The structure of ciphertext is same as in FAME. But the structure of key is different. There is an extra component sk_{acc} in the secret key. There is a component in sk_{acc} part. A sequence of hybrids called Group 1 hybrids is described to get rid of this component. Group 2 hybrids defined in FAME remains the same. For clarity, we explain both the hybrids in our current work.

We describe the $3Q$ group 1 hybrids from $Hyb_{2,1,1}$ to $Hyb_{2,3,Q}$ where Q is the number of key queries an adversary makes. These hybrids modify the key components one by one. First, DLIN assumption is used to replace Br by $Br + r' a$ for a random r' becauase the linear independence of a from B makes $Br + r' a$ a random vector. Second, the W_x matrices have one unit of entropy even given $W_x A$ and $W_x B$ (same with U_j), which can be exploited to extract a component without affecting the challenge ciphertext and other parts of the keys. This step uses the parameter hiding method of dual system encryption based proofs. Lastly, DLIN is used to revert back to Br.

We restate the basis lemma below based on the $Samp(p)$ described above.

Basis Lemma

Let (Z_1, z_1) and Z_2, z_2 be two independent samples drawn from $Samp(p)$. Then, with probability $1 - 1/p$, it holds that $[Z_1||z_2]$ and $[Z_2||z_1]$ are full rank matrices as well as $< z_1, z_2 > \neq 0$.

Description of hybrids

We begin the proof by providing formal description of the rest of the hybrids. We use various forms of ciphertext and keys in the proof. It would be useful to describe the various forms of ciphertext and keys that will be used.

A key can be in one of the following forms:
Normal : Generated in Hyb_1.
P-Normal : Br replaced by $Br + r' A$ in a Normal key where $r' \leftarrow Z_p$.
*P-Normal** : $\sigma_y a$ for all $y \in S$ and $\sigma\prime a$ removed from a P-normal key.
*Normal** : $Br + r' A$ replaced by Br in a P-Normal* key.
*P-SF** : αa added to the last component of a P-normal* key where $\alpha \leftarrow Z_p$.
*SF** : $Br + r' A$ replaced by Br in a P-SF* key.

A ciphertext can be either:
*Normal** : Generated in Hyb_1.
*SF** : As replaced by $As + sb$ in a Nomral* ciphertext where $s \leftarrow Z_p$.
*Rnd** : msg_b replaced by msg^* where $msg^* \leftarrow G_T$.

Here, P and SF stand for pseudo and semifunctional respectively following the terminology in prior work.

The first objective of the proof is to remove the extra $\sigma_y a$ and $\sigma' a$ components from all the keys. To do this, we follow the same method as in FAME. We change the form of very first key from Normal to P-Normal in $Hyb_{2,1,1}$, then change it to P-Normal* in $Hyb_{2,2,1}$, and finally to Normal* in $Hyb_{2,3,1}$. The same steps are then carried out for second key, ,third key and so on until all keys are of type Normal*. Thus, we define the following hybrids for $q = 1, .., Q$ where Q is the total number of key queries the adversary A makes.

- $Hyb_{2,1,q}$: Same as Hyb_1 except first $i - 1$ keys are Normal-*, as the i-th key is P-Normal, and rest are Normal.
- $Hyb_{2,2,q}$: Same as $Hyb_{2,1,q}$ except i-th key is P-Normal*.
- $Hyb_{2,3,q}$: Same as $Hyb_{2,2,q}$ except i-th key is Normal*.

The next objective is to show that the challenge ciphertext is bale to hide the message encrypted if none of the keys issued can decrypt it individually. Here, the form of the ciphertext is first changed from Normal* to SF* in Hyb_3. Then one by one all the keys are changed from Normal* to P-Normal* then to P-SF* and finally to SF*. Thus, the hybrids are:

- Hyb_3 : Same as $Hyb_{2,3,q}$ except ciphertext is SF*.
- $Hyb_{4,1,q}$: Same as Hyb_3 except first $i - 1$ keys are SF$*$, i-th key is P-Normal* and rest are Normal*.
- $Hyb_{4,2,q}$: Same as $Hyb_{4,1,q}$ except i-th key is P-SF*.
- $Hyb_{4,3,q}$: Same as $Hyb_{4,2,q}$ except i-th key is SF*.
- Hyb_5 : Same as $Hyb_{4,2,q}$ except ciphertext is Rnd*.

The random oracle is simulated the same way as in Hyb_1 in all the above hybrids.

Indistinguishability of hybrids We denote the advantage that an adversary has in distinguishing Hyb_i from Hyb_j when the security parameter is λ with $Adv^{textitA}_{i,j}(\lambda)$.

LEMMA 1 : For any adversary A, $Adv_{0,1}^{textitA}(\lambda) = 0$.

Proof : The master secret keys and public keys are generated identically in both the hybrids. This is because the first output of $Samp(p)$ has exactly the same distribution as A from DLIN assumption. Further, the response of $Chal$ on an oracle query is the same as in Hyb_1. They are independent and uniform over G. This follows from the proof in FAME. For clarity, we describe the same here again.

The response of $Chal$ on an oracle query of the form xlt in Hyb_1 is $[(W_x^T A)_{l,t}]$ whose exponent is $a_t(W_x)_{t,l} + (W_x)_{3,l}$ for randomly chosen $(W_x)_{t,l}$ and $(W_x)_{3,l}$. Hence, $[(W_x^T A)_{l,t}]$ is uniformly and randomly distributed for every x, l, t. In the same way, we can argue that the response to queries of the form $0jlt$ are also independent and uniform over G. Thus, $Chal$ perfectly simulates a random oracle.

If we implicitly set the responses of random oracle in Hyb_0 to be the ones generated by $Chal$ in Hyb_1, then the $ct_{i,l}$ component of the challenge ciphertext in Hyb_0 is set to a sum of three terms invovling $W_{\pi(i)}$, s, U_j and M. The terms in ct_i is also a combination of these and s which is defined to be s_1, s_2. Hence, the ciphertext is identical to the one in Hyb_0.

Now, we consider the secret key components one by one. The components in $sk_{y,t}$ are terms involving W_y, Br, a^{-1} and σ_y. The sk_0 component has term Br.

Now we describe the contribution of sk_{acc} component added newly. It can be shown that $sk^{'}$ is identically distributed to terms involving d, Br, U_1, $\sigma^{'}$ and a. The sk_{acc} component can be reduced to terms involving Br, $d\sigma_{'}/a$ and $d/\sigma_{'}$. Thus, we obtain a key identical to the one in Hyb_1.

LEMMA 2 : For all $q = 1, ..., Q$ and PPT adversaries A, there exists a PPT adversary B such that $Adv^A_{(2,3,q-1),(2,1,q)}(\lambda) \leq Adv^B_{DLIN}(\lambda) + 1/p$

Proof : The only difference between $Hyb_{2,3,q-1}$ and $Hyb_{2,1,q}$ is in the form of the i-th key issued by the challenger. In the former case, this key is Normal while in the latter it is P-Normal. We design an adversary B that converts any advantage A has in distinguishing the two hybrids into an equal advantage in breaking the DLIN assumption.

B picks the DLIN challenge and simulates the challenge in the IND-CPA game that it plays with A. It simulates the random oracle in the same way as the challenger does in $Hyb_{2,3,q-1}$ or $Hyb_{2,1,q}$.

Since $[B||a]$ is a full rank matrix except with probability $1/p$, we can say that B receives new challenge with $r^{'}A$ in the DLIN tuple where $r^{'}$ is either zero or a randomly chosen value from Z_p. It is now straightforward to generate the challenge ciphertext using r, $\sigma_{'}$ and σ_y picked from Z_p. The i-th key is generated. It can be observed that if $r_{'}$ is zero, the view of A is identical to that in $Hyb_{2,3,q-1}$; otherwise, the view is identical to $Hyb_{2,1,q}$.

LEMMA 3 : For all $q = 1, .., Q$ and adversaries A, $Adv^A_{(2,1,q)(2,2,q)}(\lambda) \leq 2/p$

Proof : It is required to prove that the view of any adversary (even unbounded) in $Hyb_{2,1,q}$ is identically distributed to its view in $Hyb_{2,2,q}$. Consider a matrix V which is defined by the product of a with the transpose of b. Note that $V^T A = VB = 0$. Let β denote the inner product of a and b which is non-zero except with probability $1/p$. FAME describes the assumptions for W_x and U_j based on which it is shown that i-th key of $Hyb_{2,2,q}$ is distributed. The hybrids under consideration in this proof only differ on the i-th key.

LEMMA 4 : For all PPT adversaries A, there exists a PPT adversary B such that
$Adv^A_{(2,3,q),3}(\lambda) \leq Adv^B_{DLIN}(\lambda) + 1/p$.

Proof : The only difference between $Hyb_{2,3,Q}$ and Hyb_3 is in the form of the challenge ciphertext. All the keys are Normal*. B can be used to generate keys for any set of attributes. Since $[A||b]$ is a full rank matrix, B receives DLIN tuple with $s^{'}$ component which is either zero or a randomly chosen value from Z_p. Hence, the view of A is identical to that of $Hyb_{2,3,Q}$ if $s^{'} = 0$. Otherwise, the view is identical to that in Hyb_3.

LEMMA 5 : For all $q = 1, .., Q$ and PPT adversaries A there exists a PPT adversary B such that
$Adv^A_{(4,3,q-1),(4,1,q)}(\lambda) \leq Adv^B_{DLIN}(\lambda) + 1/p$.

Proof : B draws (A, a) from $Samp$ and d from Z_p and gives A the public key. It also uses A to simulate the random oracle queries. B generates the ciphertext using assumptions on s. An SF* key is also generated. Finally, B outputs i-th key which is Normal* if the random number chosen is zero else it is P-Normal*.

LEMMA 6 : For all $q = 1, .., Q$ and adversaries A, $Adv^A_{(4,1,q),(4,2,q)}(\lambda) \leq 2/p$.

Proof : The proof follows due to similarity between $Hyb_{4,1,q}$ and $Hyb_{4,2,q}$. The only difference is in the form of the i-th key. The key is P-Normal* in the former case and but P-SF* in the latter. The challenge ciphertext is SF* in both the cases. Here, we observe i-th key to be P-SF*.

LEMMA 7 : For all $q = 1, .., Q$ and adversaries A, $Adv^A_{(4,3,Q),5}(\lambda) \leq 2/p$.

Proof : The only difference between $Hyb_{4,3,q}$ and Hyb_5 is that the ciphertext in the former is an encryption of msg_b and in the latter it is an encryption of a random message. Suppose we implicitly set d chosen during the setup process of $Hyb_{4,3,Q}$ to $d - \delta a$ for $\delta \leftarrow Z_p$. There are only three places where d appears from the view of an adversary: in the public key, the last component of challenge ciphertext and the last and second last component of secret key. Among them, the public key and the SF* keys are not affected. Only the last component of challenge ciphertext is affected. The inner product of the orthogonal component is uniformly distributed. Its contribution in the challenge ciphertext is uniformly distributed over Z_p except with probability $1/p$. Thus, the ciphertext is now an encryption of a random message.

Main Theorem : Our scheme with accountability is fully secure under the DLIN assumption on asymmetric pairing groups in the random oracle model. Concretely, for any PPT adversary A making Q key queries in the IND_CPA security game, there exists a PPT adversary B such that
$Adv^{A}_{OurScheme}(\lambda) \leq (8Q+2)Adv^{B}_{DLIN}(\lambda) + (16Q+6)/p$
where $p = \Theta(\lambda)$ is the order of the pairing group.

Proof : We have shown using a sequence of lemmas (numbered 1 to 7) that games are indistinguishable from one another, irrespective of the bit b given to the challenger. That is, none of the proofs have anything to do with the value of bit b. Thus, Hyb_5 si indistinguishable from Hyb_0, proving the theorem.

Theorem: $\sum_{CPABE}$ is accountable.

Proof:

Setup The challenger B executes the setup, retrieves parameters and passes them on to the adversary A.

Query A runs queries using attribute set S and retrieves set of keys.

Guess A guesses the decryption box.

The probability of A arriving at a correct decryption box depends on his ability to determine the three random involved in the key generation step and also to break the underlying computationally hard problems which is DLIN 5.1. The probability that a decryption box will be correctly guessed by B becomes negligible. Hence, $\sum_{CPABE}$ is accountable.

Theorem: $\sum_{CPABE}$ is unlinkable.

Proof:

Setup The challenger B executes the setup, retrieves parameters and passes them on to the adversary A.

Query A runs queries using attribute set S and retrieves set of identities.

Guess A guesses the identity for a given attribute set.

The difficulty in correctly guessing the random numbers involved in key generation phase makes it difficult for adversary A to correctly guess the keys for a set of attributes. The chances of determining the global identifier given the set of attributes is negligible. Hence, $\sum_{CPABE}$ is unlinkable.

7. OUR OUTSOURCED SCHEME

We now enhance our CPA-secure scheme described in previous section with outsourcing. Achieving CCA security practically is difficult. Hence, we prove security replayable chosen ciphertext attack (RCCA) game.

7.1. Design

Setup(1^{λ}) The pair of asymmetric groups G, H and group needed for bilinear pairing G_T are generated using the group generation algorithm GroupGen(1^{λ}). The output is a tuple (p, G, H, G_T, g, h) where p is the prime order of the group G_T, g, h are the generators of the groups G, H respectively and G_T is the target group of the bilinear pairing.

Pick a_1, a_2, b_1, b_2 from Z_p^*.

Pick d_1, d_2, d_3, α, a from Z_p.

Output the public key as $h, H_1 = h^{a_1}, H_2 = h^{a_2}, T_1 = e(g,h)^{d_1 a_1 + d_3}, T_2 = e(g,h)^{d_2 a_2 + d_3}, g, u, x, v, w, g^{a}, e(g,h)^{\alpha}$.

Output master key as $g, h, a_1, a_2, b_1, b_2, g^{d_1}, g^{d_2}, g^{d_3}$.

Each user also has a secret z which is decided using a zero knowledge protocol with the attribute authority.

KeyGen$(1^{\lambda}, pp, msk, S = A_1....A_k, id)$ The key generation algorithm computes user secret key using public parameters pp, set of attributes S and the master secret key msk. The identity id corresponding to a user is used in the process.

Pick r_1, r_2 from Z_p and compute

$$sk_0 = (h^{b_1 r_1}, h^{b_2 r_2}, h^{r_1 + r_2}) \tag{6}$$

For all $y \in S$, and $t = 1, 2$ compute

$$sk_{y,t} = H(y1t)^{b_1 r_1/a_t}.H(y2t)^{b_2 r_2/a_t}.H(y3t)^{r_1 + r_2/a_t}.g^{\sigma_y/a_t} \tag{7}$$

where σ_y is chosen from Z_p.

Set $sk_y = sk_{y,1}, sk_{y,2}, g^{1/\sigma_y}$.

Also compute

$sk_q = H(011t)^{b_1 r_1/a_t}.H(012t)^{b_2 r_2/a_t}.H(013t)^{r_1 + r_2/a_t}$

$sk_{t'} = g^{d_t}.sk_q.g^{\sigma^{'}/a_t}$

Set $sk_{'} = sk_1^{'}, sk_2^{'}, g^{d_3}.g^{1/\sigma^{'}}$.

We use the following mapping of values to new variables $K, K^{'}, L, L^{'}, K_{\tau,1} K_{\tau,2}$ to generate an accountability component sk_{acc}.

$\tau = y$ $a = g^{\sigma^{'}/a_1}$ $l = g^{\sigma^{'}/a_2}$ $c = g^{\sigma^{'}/a_3}$ $\alpha = g^{b_1 r_1/a_1} + g^{b_2 r_2/a_1} + g^{(r_1+r_2)/a_1}$ $w = g^{b_1 r_1/a_2} + g^{b_2 r_2/a_2} + g^{(r_1+r_2)/a_2}$ $u = g^{b_1 r_1/a_3} + g^{b_2 r_2/a_3} + g^{(r_1+r_2)/a_3}$ $x = g^{d_1 \sigma^{'}}/a_1 + g^{d_2 \sigma^{'}}/a_2 + g^{d_3 \sigma^{'}}/a_3$ $v = g^{d_1 \sigma_{'}/a_1} + g^{d_2 \sigma_{'}/a_2} + g^{d_3 \sigma_{'}/a_3}$ $l_{\tau} = \sigma_y$

Choose random $c \in Z_p^*$ and $l, l_1, .., l_k \in Z_p$.

$K = g^{\alpha/(a+c).w^{l}}$

$K^{'} = c$

$L = g^{l}$

$L' = g^{al}$

$K_{\tau,1} = g^{l,\tau}, K_{\tau,2} = (u^{A_{\tau}} x)^{l_{\tau}} v^{1/(a+c)l}{}_{\tau \in [k]}$.

Set $sk_{acc} = K, K^{'}, L, L^{'}, \{K_{\tau,1}, K_{\tau,2}\}_{\tau \in [k]}$.

A table T is maintained by the key generating party which has two columns namely identity id and value c.

Set secret key $sk_{id,S}$ as

$$SK = sk_0, sk_y, sk^{'}, sk_{acc} \tag{8}$$

where there are k attributes.

The random numbers r_1, r_2 may be chosen using a zero knowledge protocol to ensure participation of user and the authority in key generation.

Key Transform The user picks $z \in Z_N$.
Generate transformation key $TK = SK^{1/z}$.
This is given to the proxy server which can perform the six pairing operations in the decryption phase of FAME **??**.

Encrypt The encryption uses public key pk, secret sharing scheme π with matrix M to encrypt message msg. This operation is done by any sender willing to send data to a user with public key pk. The sender picks s_1, s_2 from Z_p and computes
$ct_0 = (H_1^{s_1}), H_2^{s_2}, h^{s_1+s_2}$. Suppose M has n_1 rows and n_2 columns. Then, for $i = 1, 2, .., n$ and $l = 1, 2, 3$, compute
$ct_{i,l} = H(\pi(i)q_1)^{s_1}.H(\pi(i)q_2)^{s_2}.\Pi_{j=1}^{n_2}[H(0jq1)^{s_1}.H(ojq2)^{s_2}]^{(M)_{i,j}}$

where $(M)_{i,j}$ denotes (i, j)th element of matrix M.

Set $ct_i = (ct_{i,1}, ct_{i,2}, ct_{i,3})$.

Pick random $s \in Z_p^*$.

Compute $C = msg.e(g, h)^{\alpha s}, C_0 = g^s, C_0^{'} = g^{as}$.

Compute $ct^{'} = T_1^{s_1}.T_2^{s_2}.C$.

Output ciphertext as $ct_0, .., ct_n, C_0, C_0^{'}$.

Partial Decrypt The partial decryption is done using transformation key TK at the proxy server. Most of the complex operations involved in decryption are outsourced to the proxy server. The operations involved in decryption step are performed as in [28], except the transformation key is used for the purpose. Compute

$num := ct^{'} \times e(\Pi ct_{i,1}^{\gamma_i}, sk_{0,1}) \times e(\Pi ct_{i,2}^{\gamma_i}, sk_{0,2}) \times e(\Pi ct_{i,3}^{\gamma_i}, sk_{0,3})$
$den := e(sk_1^{'} \times \Pi sk_{\Pi(i),1}^{\gamma_i}, ct_{0,1}) \times e(sk_2^{'} \times \Pi sk_{\Pi(i),2}^{\gamma_i}, ct_{0,2}) \times e(sk_3^{'} \times \Pi sk_{\Pi(i),3}^{\gamma_i}, ct_{0,3})$
$X = num/den$.
Send X to the decryption device when requested.

Decryption The user two options depending on whether a partially decrypted ciphertext from a proxy server is received or encrypted ciphertext from a sender is received .The user can directly decrypt using six pairing operations as in σ_{acc} if a partially decrypted ciphertext is not received.

On the other hand, if an encrypted ciphertext is received, the user can raise it to $1/z$ and send it to the proxy server.The proxy server can then perform partial decryption of six pairing operations using the transformation key and store the transformed ciphertext for future use. This step outputs the decrypted text by computing $(num/den)^z$.

Key Sanity Check and Trace The first step is to remove the user generated decryption power z. The integrity of the protocol is heavily dependent on the secret key z possessed by the user.

The algorithm takes as input the public parameters and a secret key sk from transformation key TK. The following checks are done on sk_{acc}. $e(L^{'}, g) = e(L, g^a)$ $e(K, g^a g^{K^{'}}) = e(g, g)^{\alpha} e(L^{'} L^{K^{'}}, w)$ $\exists \tau \in [k]$ such that $e(K_{\tau,2}, g)e(L^{K^{'}} L^{'}, v) = e(K_{\tau,1}, h)e(K_{\tau,1}, u)^{A_\tau}$.

Trace The algorithm for key sanity check is run to check if secret key is a well-formed decryption key. The Trace algorithm searches for $K^{'}$ in the identity table. There is a master attribute authority that keeps the keys and corresponding identities safely. If found, it outputs the corresponding identifier to identify the malicious user which could recover the secret of the access scheme being used.

7.2. Security Analysis

We describe the RCCA security achieved using the game described below. The scheme with accountability and outsourcing is named σ_{new}. The adversary A attempts to break σ_{new}. The base scheme under consideration is called σ_{FAME}. The scheme modified to include accountability is named σ_{acc}.

Suppose there exists a PPT adversary A that can attack our scheme in the ful security model for outsourcing with advantage ϵ.We build a simulator B that can attack σ_{acc} in adaptive CPA-security model with advantage ϵ minus a negligible amount. We have proven σ_{acc} to be CPA-secure under full adaptive security. We have proven it to be accountable and IND-CPA seure under DLIN assumption.

Init The simulator A runs B.

Setup The simulator B obtains public parameters and a description of hash function H. It sends these to A as public parameters.

Phase 1 : The simulator B initializes empty tables T and T_1 and an integer $j = 0$. It answers adversary's query as follows :

Random Oracle Hash (R, M) where $R = (r_1, r_2)$: If there is an entry (R, M, s) in T_1, return s. Otherwise, choose a random $s \in Z_p$, record (R, M, s) in T_1 and return s.

Create$(S, (M, \pi))$: B sets $j = j + 1$. If S satisfies (M, π), choose a transformation key as follows:
Run the key generation algorithmto obtain $SK^{'}$.
Set $TK = SK^{'}$ and set $SK = (d, TK)$.
If S does not satisfy (M, π), pick a random (M, π) that is satisfied, obtain $SK^{'}$. Pick $z \in Z_p$ and set the transformation key and secret key.

Finally, store $(j, S, (M, \pi), SK, TK)$ in table T and return TK to the adversary A.

Corrupt(i) : A requests B to corrupt the i-th entry

$(i, S, (M, \pi), SK, TK)$ and sets the set $D := DS$. Return SK to A.

Decrypt (i, CT) : If the access structure $()M, \pi$ of i-th entry satisfies the attributes corresponding to the ciphertext to be decrypted, decrypt. Otherwise, create new entry in T_1 and T and then decrypt.

Challenge : A submits message pair (M_0, M_1). B generates the ciphertext and returns.

Phase 2 : This phase is same as the phase 1 except that if response to decrypt is M_0 or M_1, then a random answer is returned.

Guess : A must output a bit or abort.

B searches through table T_1 for R. If found, then it ouputs b. Else, it outputs a random b.

The advantage that A has in winning the game is the same as the advantage A has in breaking B's assumption which include discrete log when the transformation key is considered and DLIN assumption when CPA-security of σ_{acc} and σ_{FAME} are considered. This is negligible. Hence, our scheme σ_{new} is RCCA secure.

Theorem: $\sum_{CPABE}$ is accountable.

Proof:

Setup The challenger B executes the setup, retrieves parameters and passes them on to the adversary A.

Query A runs queries using attribute set S and retrieves set of keys.

Guess A guesses the decryption box.

The probability of A arriving at a correct decryption box depends on his ability to determine the three random involved in the key generation step and also to break the underlying computationally hard problems which are DLIN 5.1. The probability that a decryption box will be correctly guessed by B becomes negligible. Hence, $\sum_{CPABE}$ is accountable.

Theorem: $\sum_{CPABE}$ is unlinkable.

Proof:

Setup The challenger B executes the setup, retrieves parameters and passes them on to the adversary A.

Query A runs queries using attribute set S and retrieves set of identities.

Guess A guesses the identity for a given attribute set.

The difficulty in correctly guessing the user secret z involved in key generation phase using zero knowledge protocol makes it difficult for adversary A to correctly guess the keys for a set of attributes. The chances of determining the global identifier given the set of attributes is negligible. Hence, $\sum_{CPABE}$ is unlinkable.

8. CONCLUSION

We proposed CP-ABE schemes with outsourced decryption and accountability. The storage of transformed ciphertext on untrusted server paves way to the execution of functional encryption methods without disclosing details of the underlying message. This also ensures the security of the ever increasing data from billions of IoT devices. The schemes are useful to trace malevolent IoT devices which may be operated as bots also.

The transformation of ciphertext allows decryption on resource constrained devices with a simple exponentiation operation compared to more operations in [32, 33, 34]. Our method achieves outsourced decryption with public accountability using asymmetric prime groups. The protocol is built using [28] which is proven to be computationally efficient and CPA-secure. We have proved that the protocol is fully secure in random oracle model with accountability and unlinkability. It would be very helpful for IoT based implementations.

The user anonymity introduced by partial decryption makes it suitable to ensures safe storage on multiple different servers using the transformation key. The ultimate power lies with the user. A prudent choice of z for use in transformation key makes it impossible for the ciphertext to reveal any extra information about the underlying message.

The key escrow problem is dealt with in our protocol. The dishonest authority will not be able to generate the secret key because of the zero knowledge interaction involved. We have also proved that the dishonest user will not be able to duplicate the decryption key.

Future research may include practical implementation of a prototype based on the proposed scheme in a real-world IoT setting. The protocol may also be enhanced with new features like policy hiding.

ACKNOWLEDGEMENTS

This is an independent research.

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology – EUROCRYPT 2005* (R. Cramer, ed.), (Berlin, Heidelberg), pp. 457–473, Springer Berlin Heidelberg, 2005.

[2] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology — CRYPTO 2001* (J. Kilian, ed.), (Berlin, Heidelberg), pp. 213–229, Springer Berlin Heidelberg, 2001.

[3] Y. Zhang, R. H. Deng, S. Xu, J. Sun, Q. Li, and D. Zheng, "Attribute-based encryption for cloud computing access control: A survey," *ACM Comput. Surv.*, vol. 53, Aug. 2020.

[4] D. P. Maher, "Crypto backup and key escrow," *Commun. ACM*, vol. 39, p. 48–53, Mar. 1996.

[5] D. Boneh and X. Boyen, "Efficient selective-id secure identity-based encryption without random oracles," in *Advances in Cryptology - EUROCRYPT 2004* (C. Cachin and J. L. Camenisch, eds.), (Berlin, Heidelberg), pp. 223–238, Springer Berlin Heidelberg, 2004.

[6] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in

Automata, Languages and Programming (L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, eds.), (Berlin, Heidelberg), pp. 579–591, Springer Berlin Heidelberg, 2008.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, 2007.

[8] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in *Information Security* (P. Samarati, M. Yung, F. Martinelli, and C. A. Ardagna, eds.), (Berlin, Heidelberg), pp. 347–362, Springer Berlin Heidelberg, 2009.

[9] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1274–1288, 2015.

[10] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 76–88, 2013.

[11] H. Qiao, H. Ba, H. Zhou, Z. Wang, J. Ren, and Y. Hu, "Practical, provably secure, and black-box traceable cp-abe for cryptographic cloud storage," *Symmetry*, vol. 10, no. 10, 2018.

[12] Z. Liu, Z. Cao, and D. S. Wong, "Blackbox traceable cp-abe: How to catch people leaking their keys by selling decryption devices on ebay," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, CCS '13, (New York, NY, USA), p. 475–486, Association for Computing Machinery, 2013.

[13] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography* (S. P. Vadhan, ed.), (Berlin, Heidelberg), pp. 515–534, Springer Berlin Heidelberg, 2007.

[14] J. Li, X. Chen, S. S. Chow, Q. Huang, D. S. Wong, and Z. Liu, "Multi-authority fine-grained access control with accountability and its application in cloud," *Journal of Network and Computer Applications*, vol. 112, pp. 89–96, 2018.

[15] N. Doshi and D. Jinwala, "Constant ciphertext length in multi-authority ciphertext policy attribute based encryption," in *2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011)*, pp. 451–456, 2011.

[16] F. Yundong, W. Xiaoping, and W. Jiasheng, "Multi-authority attribute-based encryption access control scheme with hidden policy and constant length ciphertext for cloud storage," in *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp. 205–212, 2017.

[17] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, (New York, NY, USA), p. 121–130, Association for Computing Machinery, 2009.

[18] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography* (Y. Ishai, ed.), (Berlin, Heidelberg), pp. 253–273, Springer Berlin Heidelberg, 2011.

[19] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Advances in Cryptology – CRYPTO 2010* (T. Rabin, ed.), (Berlin, Heidelberg), pp. 483–501, Springer Berlin Heidelberg, 2010.

[20] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Advances in Cryptology – CRYPTO 2010* (T. Rabin, ed.), (Berlin, Heidelberg), pp. 465–482, Springer Berlin Heidelberg, 2010.

[21] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, (New York, NY, USA), p. 169–178, Association for Computing Machinery, 2009.

[22] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, p. 1–30, Feb. 2006.

[23] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology — EUROCRYPT'98* (K. Nyberg, ed.), (Berlin, Heidelberg), pp. 127–144, Springer Berlin Heidelberg, 1998.

[24] S. Banerjee, S. Roy, V. Odelu, A. K. Das, S. Chattopadhyay, J. J. Rodrigues, and Y. Park, "Multi-authority cp-abe-based user access control scheme with constant-size key and ciphertext for iot deployment," *Journal of Information Security and Applications*, vol. 53, p. 102503, 2020.

[25] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, "Auditable σ -time outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 94–105, 2018.

[26] L. Touati, "Grouping-proofs based access control using kp-abe for iot applications," in *2017 IEEE Trustcom/BigDataSE/ICESS*, pp. 301–308, 2017.

[27] S. Agrawal, B. Libert, and D. Stehlé, "Fully secure functional encryption for inner products, from standard assumptions," in *Advances in Cryptology – CRYPTO 2016* (M. Robshaw and J. Katz, eds.), (Berlin, Heidelberg), pp. 333–362, Springer Berlin Heidelberg, 2016.

[28] S. Agrawal and M. Chase, "Fame: Fast attribute-based message encryption," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, (New York, NY, USA), p. 665–682, Association for Computing Machinery, 2017.

[29] J. Katz and Y. Lindell, *Introduction to Modern Cryptography (Chapman; Hall/Crc Cryptography and Network Security Series)*. Chapman ; Hall/CRC, 2007.

[30] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, (New York, NY, USA), p. 62–73, Association for Computing Machinery, 1993.

[31] S. J. De and S. Ruj, "Efficient decentralized attribute based access control for mobile clouds," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 124–137, 2020.

[32] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[33] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2119–2130, 2015.

[34] H. Ma, R. Zhang, Z. Wan, Y. Lu, and S. Lin, "Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 6, pp. 679–692, 2017.

[35] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of Cryptography* (J. Kilian, ed.), (Berlin, Heidelberg), pp. 325–341, Springer Berlin Heidelberg, 2005.

[36] A. Beimel, "Secret-sharing schemes: A survey," in *Coding and Cryptology* (Y. M. Chee, Z. Guo, S. Ling, F. Shao, Y. Tang, H. Wang, and C. Xing, eds.), (Berlin, Heidelberg), pp. 11–46, Springer Berlin Heidelberg, 2011.

[37] R. Canetti, H. Krawczyk, and J. B. Nielsen, "Relaxing chosen-ciphertext security," in *Advances in Cryptology - CRYPTO 2003* (D. Boneh, ed.), (Berlin, Heidelberg), pp. 565–582, Springer Berlin Heidelberg, 2003.

[38] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," *Journal of Cryptology*, vol. 26, p. 80–101, 2013.

[39] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," in *Proceedings of the 20th USENIX Conference on Security*, SEC'11, (USA), p. 34, USENIX Association, 2011.

[40] J. Ning, X. Dong, Z. Cao, and L. Wei, "Accountable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud," in *Computer Security – ESORICS 2015* (G. Pernul, P. Y A Ryan, and E. Weippl, eds.), (Cham), pp. 270–289, Springer International Publishing, 2015.

[41] E. Bangerter, S. Barzan, S. Krenn, A. Sadeghi, T. Schneider, and J.-K. Tsay, "Bringing zero-knowledge proofs of knowledge to practice," in *Security Protocols Workshop*, 2009.

[42] B. Waters, "Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions," in *Advances in Cryptology - CRYPTO 2009* (S. Halevi, ed.), (Berlin, Heidelberg), pp. 619–636, Springer Berlin Heidelberg, 2009.