Expressive CP-ABE Scheme Satisfying Constant-Size Keys and Ciphertexts

Dhaval Khandla[1,2], Het Shahy[1,3], Manish Kumar Bz[1,4], Alwyn Roshan Pais[1,5] and Nishant Raj[1,6]

[1] Department of Computer Science and Engineering National Institute of Technology Karnataka, Surathkal, Karnataka, India 575025
[2] dhavaljkhandla26@gmail.com
[3] yhetshah247@gmail.com
[4] zbmanish15597@gmail.com
[5] alwyn@nitk.ac.in
[6] nishant2raj@gmail.com

Abstract. Ciphertext-policy attribute-based encryption (CP-ABE) is a desirable scheme to use in cloud-based applications, especially on IoT devices. As most of these devices are battery-limited and memory-limited, leading to a constraint in designing a robust and straightforward mechanism involving less computation and less memory. But none of the systems are secure and based on conventional cryptosystems. Here we propose a constant-size secret key and constant-size ciphertext scheme based on RSA cryptosystem, which performs encryption and decryption in $\mathcal{O}(1)$ time complexity. We also prove that the scheme is secure and compare it with already existing schemes.

Keywords: Ciphertext-policy attribute-based encryption, cloud computing, constant-size secret key, constant-size ciphertext, RSA-based cryptography

1 INTRODUCTION

As we plunge into the cloud-computing era, where most of the is being used on limited battery mobile or IoT devices, it becomes the need of the hour to design a mechanism that facilitates faster encryption and decryption [1] [25] [27]. One such scheme is CP-ABE, which is based on ABE proposed in [21]. CP-ABE allows the user to define an access policy associated with every message, thereby defining a set of users who can correctly decrypt the message. This makes CP-ABE a convenient mechanism to transfer messages in the cloud computing environment [29] [18] [2] [28] [26] [17]. Also, as most of the devices are battery constrained, this results in essential design criteria, i.e., CP-ABE should have cost efficient.

In traditional public-key cryptography, receiver specific message is encrypted using the receiver's public key. "Identity-based encryption (IBE) is a type of public-key encryption in which the public key of a user is some unique identifying information of the user (e.g., a user's email address), and there is a third party key server which computes the private key corresponding to the public key (e.g., a user's email address)" [2]. "Attribute-based encryption(ABE) is an extension of IBE, which defines the user's identity not in an atomic manner but as a set of attributes(e.g., occupation), and messages can be encrypted with the subset of policies or attributes defined over a fixed universal set of attributes" [9]. The main idea here is, if the attributes on which the cipher-text is created matches with the set of attributes of the user key, then only the user can decrypt the cipher-text. ABE is mainly

used as a key-text-policy attribute-based encryption (KP-ABE) and cipher-text-policy attribute-based encryption (CP-ABE). "In key-policy attribute-based encryption (KP-ABE), each cipher-text is associated with a set of attributes, and each user's private-key specifies an access policy over a defined universe of attributes. A user will be able to decrypt a cipher-text, if and only if attributes of cipher-text satisfy the policy of the respective user" [9] [22] [20] [3]. Whereas, "in cipher-text-policy attribute-based encryption (CP-ABE), a user's private key is associated with a set of attributes and a cipher-text satisfies an access policy over a defined universal set of attributes. If a user's set of attributes, satisfies the policy of the given cipher-text, then only he will be able to decrypt the associated cipher-text" [4] [12] [5] [14] [24] [15].

In recent times many CP-ABE schemes have been proposed, which are based on bi-linear maps. Among these, a few are with constant size cipher-texts [29] [8] [30] [7] and a few with constant-size secret keys [8] [10]. As these are based on bi-linear maps, they are costly than those based on conventional cryptosystems, such as [23] [16]. The various ABE schemes are summarised in Table 1. Hence, there is a need to design a cost-efficient and more expressive access structure CP-ABE cipher-texts using conventional public-key cryptosystems and to have constant size secret keys. One such attempt was made by [19].

A security flaw was shown in the scheme of [19] by [11]. It was proven that the scheme is not collusion resistant, by showing a scenario where users not having required attributes satisfying the policy can collude in order to decipher the ciphertext. It is observed that if the union of attributes of a set of colluding users satisfies the policy, then the attack is possible.

Here, we present a modification over the proposed scheme by [19] in order to avoid the attack. The proposed scheme is based on the RSA cryptosystem with an AND gate access structure and uses constant-size secret keys and ciphertexts. Also, our scheme performs encryption and decryption efficiently, i.e. in $\mathcal{O}(1)$ time complexity.

We divide the rest of the paper into different sections. First, we discuss the various mathematical definitions and preliminaries, which are a prerequisite to understanding the scheme in 2. Then, in 3, we explain the key management in the defined access structure. Following this, in 4, we propose our CP-ABE scheme. Then in 5, we discuss the security aspects of the scheme. After which we present the evaluation results of our scheme in 6. Finally, in 7, we provide a few concluding remarks.

2 MATHEMATICAL PRELIMINARIES AND DEFINITIONS

In this section, we explain the various definitions and preliminaries related to ciphertext-policy attribute-based encryption scheme.

2.1 Attribute Definition and Access Structure

We follow a similar definition for attributes and access policy, as provided in [10]. Firstly, let $\mathbb{U}$ be the set of all attributes in the universe. Also, assume that we have n attributes in $\mathbb{U}$, so we have $\mathbb{U} = \{A_1, A_2, A_3, \dots A_n\}$, where A_i represents the ith attribute in the universe. Secondly, let $\mathbb{A}$ be the attribute set associated with a user, so we have $\mathbb{A} \subseteq \mathbb{U}$. For convenience we represent $\mathbb{A}$ as a n-bit string $a_1 a_2 a_3 \dots a_n$, where

$$\begin{cases} a_i = 1, A_i \in \mathbb{A} \\ a_i = 0, A_i \notin \mathbb{A} \end{cases}$$

For example, if we have n = 5, then $\mathbb{U} = \{A_1, A_2, A_3, A_4, A_5\}$. Also, if the user has the following attributes $\{A_1, A_3, A_4\}$, then it's corresponding five-bit string takes the value 10110. Thirdly, let $\mathbb{P}$ be the access policy associated with a message, so we have $\mathbb{P} \subseteq \mathbb{U}$.

Table 1: Comparison of attribute-based encryption schemes

Scheme	**KP-ABE/CP-ABE**	**Access structure**	**Security model**	**LSK**	**LCT**
SW [21]	KP-ABE	Threshold	Selective security	nG	nG+G_t
GPSW [9]	KP-ABE	Tree	Selective security	$\|\mathbb{A}\|G$	$\|\mathbb{P}\|G$+G_t
OSW [20]	KP-ABE	Tree	Selective security	$2\|\mathbb{A}\|G$	$(\|\mathbb{P}\|+1)G$+G_t
BSW [4]	CP-ABE	Tree	Selective security	$(2\|\mathbb{A}\|+1)G$	$(2\|\mathbb{P}\|+1)G$+G_t
HLR [12]	CP-ABE	Threshold	Selective security	$(n+\|\mathbb{A}\|)G$	$2G$+G_t
CCLZFLW [5]	KP-ABE/CP-ABE	Threshold	Full security	$\mathcal{O}(n^2)$	$\mathcal{O}(1)$
EMNOS [8]	CP-ABE	(n,n)-Threshold	Selective security	$2G$	$2G$+G_t
LOSTW [14]	CP-ABE	Linear secret-sharing scheme	Full security	$(\|\mathbb{A}\|+1)G_c$	$(2\|\mathbb{P}\|+1)G_t$+G_{t_c}
Waters [24]	CP-ABE	Linear secret-sharing scheme	Selective security	$(\|\mathbb{A}\|+1)G$	$(2\|\mathbb{P}\|+1)G$+G_t
ALP [3]	KP-ABE	Linear secret-sharing scheme	Selective security	$3\|\mathbb{A}\|G$	$2G$+G_t
LW [15]	CP-ABE	Linear secret-sharing scheme	Full security	$(3+\|\mathbb{A}\|)G_c$	$(2\|\mathbb{P}\|+2)G_t$+G_{t_c}
DJ [7]	CP-ABE	AND gate-Multivalued	Full Security	$(n_{\mathbb{A}}\|\mathbb{A}\|+2)G_c$	$2G_c+G_{t_c}$
ZZCLL [29]	CP-ABE	AND gate-Multivalued with wild-cards	Selective security	$(n+1)G$	$2G+G_t$
CN [6]	CP-ABE	AND gates	Selective security	$2(\mathbb{A}+1)G$	$(\mathbb{P}+1)G+G_t$
ZH [30]	CP-ABE	AND gates	Selective security	$(\mathbb{A}+1)G$	$2G+G_t$
GSWV [10]	CP-ABE	AND gates	Selective security	$2G$	(n-$\mathbb{P}$+2)$G+G_t+L$
ODKCJ [19]	CP-ABE	AND gates	Selective security	$2G$	$3G+L$

Note: LSK: length of user secret key; LCT: length of cipher-text; L: length of plain-text M; $GandG_t$: Prime order pairing (In our scheme similar to [19], the group G is multiplicative group Z_N, where $N=pq$); G_c and G_{t_c}: composite order pairing; $n_{\mathbb{A}}$: average number of values assigned to each attribute in attribute set $\mathbb{A}$.

For convenience, we represent $\mathbb{P}$ as an n-bit string $b_1b_2b_3\ldots b_n$, where

$$\begin{cases} b_i = 1, A_i \in \mathbb{P} \\ b_i = 0, A_i \notin \mathbb{P} \end{cases}$$

For example, if we have n = 5, say a particular message has attributes $\{A_2, A_4\}$ associated with it, then the corresponding five-bit value string is 01010.

Now we shall define the AND gate access structure on a given set universal set of n attributes $\mathbb{U}$. Let, attribute set $\mathbb{A}$ be associated with a user, and let the string associated with $\mathbb{A}$ be $a_1a_2a_3 \dots a_n$. Also, let the access policy be $\mathbb{P}$ and the string associated with it be $b_1b_2b_3 \dots b_n$. Suppose that $a_i \geq b_i \forall i$, then we say that the attribute set $\mathbb{A}$ satisfies the access policy $\mathbb{P}$. We shall also use the notation $\mathbb{P} \subseteq \mathbb{A}$ to represent the same.

2.2 Definition of CP-ABE Scheme

A CP-ABE Scheme consists of four major algorithms. They are Setup, Encrypt, KeyGen, and Decrypt. These algorithms are defined below, in a similar fashion as in [10]:

1. Setup: Given a security parameter ρ and a set of universal attributes $\mathbb{U}$, this algorithm outputs master public key, denoted by MPK, and its corresponding master secret key, denoted by MSK.

2. KeyGen: This algorithm outputs the user secret key k_u, on inputs of the master public key MPK, the master secret key MSK and the user attribute set $\mathbb{A}$.

3. Encrypt: This algorithm converts the plain text message M to ciphertext C, using the given access policy $\mathbb{P}$ and the master public key MPK.

4. Decrypt: This algorithm takes the ciphertext C, the access policy $\mathbb{P}$ used to generate C, master public key MPK along with user secret key k_u and the corresponding user attributes $\mathbb{A}$, and outputs the corresponding plaintext message M or null($\perp$), based on whether $\mathbb{P} \subseteq \mathbb{A}$ or not respectively.

For any given (MPK, MSK), ciphertext is generated using Encrypt algorithm and the access policy $\mathbb{P}$ and the plain text message $\mathbb{M}$, and the user secret key k_u associated with attributes $\mathbb{A}$, and if $\mathbb{P} \subseteq \mathbb{A}$ then the Decrypt algorithm should always output the correct plain text message M. If this is not true then we cannot decrypt the message from C. Also, note that the above-mentioned property has to hold true for the correctness of the CP-ABE scheme.

2.3 Selective Game for CP-ABE Scheme

In this subsection, we are going to show our scheme being secure under the chosen cipher-text attack by using the selective game for CP-ABE as defined in [6]. The CP-ABE game shows the messages being indistinguishable and the collision-resistance to user secret keys if the attacker is unauthorized to get the message, the attackers by combining their secret keys should not be able to generate a new secret key which satisfies the cipher-text access poilicy i.e., collusion-resistant. In the game, after the challenge phase, the multiple secret key queries are issued by an adversary $\mathbb{A}$. The game is described as follows between the challenger $\mathbb{B}$ and an adversary $\mathbb{A}$.

1. Initialization: The adversary $\mathbb{A}$ sends to the challenger $\mathbb{B}$ an n-bit access policy $\mathbb{P}$.

2. Setup: The challenger $\mathbb{B}$ gives MPK to the adversary $\mathbb{A}$, after generating the key pair (MPK, MSK) with the security parameter ρ by running Setup and KeyGen algorithms.

3. Query: The adversary $\mathbb{A}$ generates the following queries for challenger $\mathbb{B}$.
   (a) The adversary $\mathbb{A}$, only queries for those secret key k_{u_i} whose attribute set A_i, does not satisfy the access policy $\mathbb{P}$ (Initially chosen by adversary $\mathbb{A}$ in Initialization phase).
   (b) The decryption query on cipher-text $E[P_i, M_i]$.
4. Challenge: In this phase, the adversary $\mathbb{A}$ outputs two messages (M_0, M_1) for the challenger $\mathbb{B}$. It requires that the adversary $\mathbb{A}$ generates queries only for a secret key on an attribute set A not satisfying $\mathbb{P} \subseteq \mathbb{A}$. The challenger $\mathbb{B}$ outputs computed cipher-text $E[\mathbb{P}, M_c]$, where $c \in \{0, 1\}$ randomly, as challenge to the adversary $\mathbb{A}$.
5. Guess: The adversary $\mathbb{A}$ outputs a guess c'.
   (a) If (c' = c): The adversary $\mathbb{A}$ wins the game.
   (b) else: The adversary $\mathbb{A}$ loses the game.

3 KEY MANAGEMENT IN DEFINED ACCESS STRUCTURE

The key management in defined access structure is based on the scheme given in [13]. It is proven to be robust against key recovery attacks.

"Suppose Z_n is a set of equivalence classes modulo $N = pq$, where p,q are RSA primes and $p \neq q$. For any non-zero a $\in Z_n$, $\gcd(a, N) = 1$ iff there exists a multiplicative inverse b for $a \pmod N$.

$$ab \equiv 1 \pmod N$$

b can be calculated efficiently using the extended Euclidean algorithm.

For each attribute $A_i \in \mathbb{U}$, select a prime number p_i such that $\gcd(p_i, \phi(N)) = 1$. Then for each p_i, calculate its inverse q_i such that $p_i q_i \equiv 1 \pmod{\phi(N)}$, where $p_i \neq q_i$ iff $i \neq j$. Let the secret parameters be $phi(N)$, $q_1, \ldots, q_n$ and the public parameters be $N, p_1, \ldots, p_n$. As integer factorization is a computationally hard problem, factoring the product $N = pq$ is also hard. So, without knowing secure primes p and q, calculating $\phi(N) = (p-1)(q-1)$ is also computationally infeasible. Hence, finding a prime q_i such that $p_i q_i \equiv 1 \pmod{\phi(N)}$ is computationally hard, as it is dependent on integer factorization problem.

Select a random number g such that $2 < g < N - 1$, and $\gcd(g, N) = 1$. Calculate the secret keys $K_{\mathbb{A}}$ and $K_{\mathbb{P}}$ corresponding to attribute set $\mathbb{A}$ and access policy $\mathbb{P}$ respectively, as under:

$$K_{\mathbb{A}} = g^{d_{\mathbb{A}}} \pmod N,$$
$$K_{\mathbb{P}} = g^{d_{\mathbb{P}}} \pmod N,$$

where $d_{\mathbb{A}} = \prod_{i=1}^{n} q_i^{a_i}$, $a_i \in \mathbb{A}$ and $d_{\mathbb{P}} = \prod_{i=1}^{n} q_i^{b_i}$, $b_i \in \mathbb{P}$.

Theorem 1. *The attribute set $\mathbb{A}$ fulfills access policy $\mathbb{P}$ (i.e. $\mathbb{P} \subseteq \mathbb{A}$) if and only if $\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}$ is an integer, where $e_{\mathbb{A}} = \prod_{i=1}^{n} p_i^{a_i}$, $e_{\mathbb{P}} = \prod_{i=1}^{n} p_i^{b_i}$, and $K_{\mathbb{P}} = K_{\mathbb{A}}^{\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}}$ (mod N).*

Proof. Suppose that $\mathbb{A}$ does not satisfy $\mathbb{P}$ (i.e. $\mathbb{P} \not\subseteq \mathbb{A}$). As we know that $a_i, b_i \in \{0, 1\}$, $a_i - b_i \in \{-1, 0, 1\}$. Therefore, we can say that in the fraction $\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}} = \prod_{i=1}^{n} p_i^{a_i - b_i}$, at least one inverse term p_j^{-1} exists, and computing p_j^{-1} without finding factors of $N = pq$ is computationally hard. So, $\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}$ is not an integer when $\mathbb{P} \not\subseteq \mathbb{A}$.

Another way around, if $\mathbb{P} \subseteq \mathbb{A}$, we can calculate $K_{\mathbb{P}}$ as follows:

$$\begin{aligned}
K_{\mathbb{P}} &= K_{\mathbb{A}}^{\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}} (\text{mod } N) \\
&= (g^{d_{\mathbb{A}}} (\text{mod } N))^{\frac{\prod_{i=1}^{n} p_i^{a_i}}{\prod_{i=1}^{n} p_i^{b_i}}} (\text{mod } N) \\
&= g^{d_{\mathbb{A}} (\prod_{i=1}^{n} p_i^{a_i - b_i})} (\text{mod } N) \\
&= g^{(\prod_{i=1}^{n} q_i^{a_i})(\prod_{i=1}^{n} p_i^{a_i - b_i})} (\text{mod } N) \\
&= g^{(\prod_{i=1}^{n} q_i^{a_i - b_i + b_i})(\prod_{i=1}^{n} p_i^{a_i - b_i})} (\text{mod } N) \\
&= g^{(\prod_{i=1}^{n} q_i^{b_i})(\prod_{i=1}^{n} q_i^{a_i - b_i} p_i^{a_i - b_i})} (\text{mod } N) \\
&= g^{(\prod_{i=1}^{n} q_i^{b_i})(\prod_{i=1}^{n} (q_i p_i)^{a_i - b_i})} (\text{mod } N) \\
&= g^{(\prod_{i=1}^{n} q_i^{b_i})} (\text{mod } N) \\
&= g^{d_{\mathbb{P}}} (\text{mod } N)
\end{aligned}$$

□

Example 1: Consider the following example related to key management discussed above. Suppose 101 and 001 are the 3-bit strings associated with the attribute set $\mathbb{A}$ and access policy $\mathbb{P}$, respectively. Let the chosen RSA pairs corresponding to the attributes A_i's be (p_i, q_i), where i = 1, 2, 3. Thus, $\mathbb{A} = \{A_1, A_3\}$ and $\mathbb{P} = \{A_3\}$. It is clearly that $\mathbb{P} \subseteq \mathbb{A}$. So we have $K_{\mathbb{A}} = g^{q_1 q_3}$, $K_{\mathbb{P}} = g^{q_3}$, $e_{\mathbb{A}} = p_1 p_3$ and $e_{\mathbb{P}} = p_3$. We can calculate $K_{\mathbb{P}}$ using $K_{\mathbb{A}}$ as follows:

$$\begin{aligned}
K_{\mathbb{P}} &= K_{\mathbb{A}}^{\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}} (\text{mod } N) \\
&= (g^{q_1 q_3})^{\frac{p_1 p_3}{p_3}} (\text{mod } N) \\
&= (g^{q_1 q_3})^{p_1} (\text{mod } N) \\
&= g^{(q_3)(q_1 p_1)} (\text{mod } N) \\
&= g^{q_3} (\text{mod } N)
\end{aligned}$$

" [13]

4 PROPOSED CP-ABE-CSKC SCHEME

Here, we propose CP-ABE scheme with constant-size secret keys and ciphertexts, which will be referred to as CP-ABE-CSKC from this section. Other notations we use are enlisted in Table 2. For the sake of simplicity, (mod N) will be omitted from $g^z (\text{mod } N)$ for the remaining part of this paper.

The scheme consists of five phases, as follows:

4.1 SETUP PHASE

In this phase, the security parameter ρ and the universe of attributes $\mathbb{U} = \{A_1, A_2, \ldots A_n\}$ are taken as inputs. Here, we add one extra attribute A_{n+1}, which is 1 for every user and 0 for every policy. The Setup algorithm consists of the following steps:

1. Select two RSA primes p and q with p ≠ q, and compute $N = pq$. Then, choose the RSA public exponent p_i randomly such that $\gcd(p_i, \phi(N)) = 1$, and calculate q_i such that $p_i q_i \equiv 1 (\text{mod } \phi(N))$ corresponding to each attribute $A_i \in \mathbb{U}$,

Table 2: Notations

Notation	Description
(k, x)	System private key pair
$N = pq$	RSA modulus with large primes p and q, where $p \neq q$
$\phi(x)$	Euler's totient function of x, $\phi(N) = (p-1)(q-1)$
H_1, H_2, H_3	Three one-way collision-resistant hash functions
$\mathbb{U}$	Universe of $(n+1)$ attributes $\{A_1, A_2, A_3, \dots, A_n, A_{n+1}\}$
$\mathbb{A}$	Set of user attributes, $\mathbb{A} \subseteq \mathbb{U}$
$\mathbb{P}$	Access policy, $\mathbb{P} \subseteq (\mathbb{U} \setminus A_{n+1})$

$\forall$ i $= 1, 2, \dots, n, n+1$. Then, select two system private keys k and x such that $\gcd(k, \phi(N)) = 1$, $\gcd(x, \phi(N)) = 1$, $\gcd(k, q_i) = 1$ and $\gcd(x, q_i) = 1 \; \forall$ i = 1, 2, ..., n, n+1. Now pick a random number g such that $2 < g < N - 1$ and $\gcd(g, N) = 1$.

2. Select three one-way collision-resistance hash functions H_1, H_2, and H_3 as follows:

$$\begin{aligned} H_1 &: \{0,1\}^* \rightarrow \{0,1\}^{\rho}, \\ H_2 &: \{0,1\}^* \rightarrow \{0,1\}^{l_\sigma}, \\ H_3 &: \{0,1\}^* \rightarrow \{0,1\}^{l_m}, \end{aligned}$$

where l_σ is the length of a random string under the security parameter, and l_m is the length of plaintext message M.

3. Calculate the public parameters $D_u = g^{d_u}$, $Y = g^x$, and $R = g^k$, where $d_u = \prod_{A_i \in U} q_i$.

4. Produce the master secret key MSK and master public key MPK as follows:

$$\begin{aligned} MSK &= \{k, x, p, q, q_1, \dots, q_n, q_{n+1}\}, \\ MPK &= \{N, D_{\mathbb{U}}, Y, R, H_1, H_2, H_3, p_1, \cdots, p_n, p_{n+1}\}. \end{aligned}$$

4.2 ENCRYPT PHASE

Encryption mechanism is based on the approach given in [23], to achieve security against chosen-ciphertext attack.

This algorithm takes an access policy $\mathbb{P}$, the master public key MPK, and plaintext M as inputs. The encryption algorithm gives ciphertext C as output.

$$E(\sigma_m, H_1(\mathbb{P}, M, \sigma_m)), H_3(\sigma_m) \oplus M, S_m = H_1(\sigma_m, M)$$

Let σ_m be random secret using the hash output $r_m = H_1(\mathbb{P}, M, \sigma_m)$, and let $E(\sigma_m, H_1(\mathbb{P}, M, \sigma_m))$ denote attribute-based encryption on σ_m. The random secret σ_m is encrypted with the key $g^{r_m d_P}$, and the plaintext M is encrypted with random secret σ_m, and they are denoted by C_{σ_m} and C_m respectively in C. We also calculate the signature $S_m = H_1(\sigma_m, M)$ on the plaintext M using the random secret σ_m to verify the validity of the derived plaintext M. The remaining components of the ciphertext C are Y_m and R_m.

Our encryption algorithm is similar to the one used in [19]. It takes an access policy $\mathbb{P} \subseteq \mathbb{U}$ where $|\mathbb{P}| \neq 0$, the master public key MPK and a plaintext message M as inputs, and outputs the ciphertext $C = \{Y_m, R_m, C_{\sigma_m}, C_m, S_m\}$ using the following steps:

1. Pick a random number $\sigma_m \in \{0,1\}^{l_\sigma}$, and compute $r_m = H_1(P, M, \sigma_m)$.
2. Compute K_m as

$$\begin{aligned} K_m &= D_{\mathbb{U}}^{r_m \frac{e_{\mathbb{U}}}{e_{\mathbb{P}}}} \\ &= \left(g^{d_{\mathbb{U}}}\right)^{r_m \frac{e_{\mathbb{U}}}{e_{\mathbb{P}}}} \\ &= g^{r_m d_{\mathbb{P}}}, \end{aligned}$$

   where $d_{\mathbb{P}} = \prod_{A_i \in \mathbb{P}} q_i$. $e_{\mathbb{P}} = \prod_{A_i \in \mathbb{P}} p_i$ and $e_{\mathbb{U}} = \prod_{A_i \in \mathbb{U}} p_i$.
3. Calculate $Y_m = g^{xr_m}$, $R_m = g^{kr_m}$, $C_{\sigma_m} = H_2(K_m) \oplus \sigma_m$, $C_m = H_3(\sigma_m) \oplus M$, and $S_m = H_1(\sigma_m, M)$.

This algorithm outputs the ciphertext C as $C = \{\mathbb{P}, Y_m, R_m, C_{\sigma_m}, C_m, S_m\}$. Now, the ciphertext C is sent to a centralized server to check if the policy $\mathbb{P}$ contains the attribute A_{n+1} or not.

4.3 VALIDATE PHASE

In this phase, the ciphertext C is sent to a centralized server for validation after the encrypt phase. The steps are as follows:

1. First, we check if the attribute A_{n+1} is in the policy $\mathbb{P}$ of the ciphertext C or not. If the attribute is present, then the security of the system is compromised as the attack shown in [11] is possible. So, we discard the message and notify the user to do the encryption again.
2. If the attribute A_{n+1} is not present in the ciphertext C, then it is transmitted to all the users.

4.4 KEY-GEN PHASE

Here we generate the user secret key k_u corresponding to the user attributes $\mathbb{A}$, using the master secret key MSK and the master public key MPK. This is done using the following steps:

1. Calculate $d_{\mathbb{A}} = \prod_{i=1}^{n+1} q_i^{a_i}$, such that $a_i = 1$ if $A_i \in \mathbb{A}$ and $a_i = 0$ if $A_i \notin \mathbb{A}$
2. Choose two random number r_u and t_u and then calculate s_u, such that the following condition is satisfied, $d_{\mathbb{A}} = ks_u + xr_u (\text{mod } \phi(N))$. Next, calculate $k_1 = s_u + xt_u$ and $k_2 = r_u - kt_u (\text{mod } \phi(N))$.

This algorithm finally outputs the user secret key $k_u = (k_1, k_2)$.

4.5 DECRYPT PHASE

In this phase, we discuss the steps performed for decryption. This algorithm takes the ciphertext $C = \{\mathbb{P}, Y_m, R_m, C_{\sigma_m}, C_m, S_m\}$ corresponding to a given access policy $\mathbb{P}$ along with the user secret key k_u corresponding to the user attribute $\mathbb{A}$. The steps are as follows:

1. From Theorem 1, we have $\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}$ is an integer if $\mathbb{P} \subseteq \mathbb{A}$ and vice versa. If we have $\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}$, then we compute K_m,

$$\begin{aligned} K_m &= (Y_m^{k_2} R_m^{k_1})^{\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}} \\ &= (g^{xr_m(r_u - kt_u)} g^{kr_m(s_u + xt_u)})^{\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}} \\ &= (g^{r_m(xr_u + ks_u)} g^{xr_m(-kt_u) + kr_m(xt_u)})^{\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}} \\ &= (g^{r_m d_{\mathbb{A}}})^{\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}} \\ &= g^{r_m d_{\mathbb{P}}}. \end{aligned}$$

   Else, $\frac{e_{\mathbb{A}}}{e_{\mathbb{P}}}$ is not an integer. Thereby rendering the computation of K_m infeasible.

2. Next calculate $\sigma_m^{'} = H_2(K_m) \oplus C_{\sigma_m}$ and $M^{'} = C_m \oplus H_3(\sigma_m^{'})$.

3. Finally, to check if the signature matches, check if the condition $S_m = H_1(\sigma_m^{'}, M^{'})$ is true or not. If it is true then output then $M^{'}$, else output $\perp$.

5 SECURITY ANALYSIS

In this section, first, we explain the attack possible on [19] as shown in [11]. Then, we provide intuition regarding why we choose the extra attribute to prevent the attack. Finally, we prove mathematically that the attack presented in [11] is not possible in our scheme.

We explain the attack using two users. Let the universe of attributes be $\mathbb{U} = \{A_1, A_2\}$. From, the Setup Phase we have $MSK = \{k, x, p, q, q_1, q_2\}$, $MPK = \{N, D_{\mathbb{U}}, Y, R, H_1, H_2, H_3, p_1, p_2\}$. Also, let the first user have attribute $\mathbb{U}_{\not\vDash} = \{A_1\}$ and the secret key be $K^{(1)} = \{k_1^{(1)}, k_2^{(1)}\}$ and the second user have attribute $\mathbb{U}_{\not\vDash} = \{A_2\}$ and the secret key be $K^{(2)} = \{k_1^{(2)}, k_2^{(2)}\}$. Suppose that we want to send a message M having policy $\mathbb{P} = \{A_1, A_2\}$.

The cipher-text produced using the encrypt phase be $C = \{\mathbb{P}, Y_m, R_m, C_{\sigma_m}, C_m, S_m\}$. Now, both the user can calculate $T_1 = Y_m^{k_2^{(1)}} R_m^{k_1^{(1)}}$ and $T_2 = Y_m^{k_2^{(2)}} R_m^{k_1^{(2)}}$ respectively. Observe that we have $T_1^{p_1} = T_2^{p_2} = g^{r_m}$. As p_1 and p_2 are prime numbers, we have gcd(p_1, p_2) = 1. Then using Bezout's Identity we know for two numbers a and b with gcd g, we can find the Bezout Coefficients x and y, such that ax + by = g. Using this here we get, the coefficients a_1, a_2, such that $a_1 p_1 + a_2 p_2 = 1$. Now we can have

$$\begin{aligned} T_1 =& T_1^{a_1 p_1 + a_2 p_2}, \\ =& T_1^{p_1 a_1} T_1^{a_2 p_2}, \\ =& T_2^{a_1 p_2} T_1^{a_2 p_2}, \\ =& (T_2^{a_1} T_1^{a_2})^{p_2}\}. \end{aligned}$$

Thus, we can easily get K_m, as we have $K_m^{p_2} = T_1$. So raising the last equation by q_2 we get K_m. Thus, both the attackers can collude and decrypt the message without having the necessary attributes to decrypt the message individually.

The attack on [19] was possible because of a relationship between the public prime numbers, $p_1, p_2, \ldots p_n$, can be obtained using Bezout's Identity. This relationship can be used to collude and thereby attack. Now, if we include an extra attribute to all the users and not include it in the policy, then a relationship cannot be found among the public prime numbers, $p_1, p_2, \ldots p_n, p_{n+1},$. This prevents the attack. Now, we present the proof which supports our claim.

Proof. As shown in [11],

$$\begin{aligned} T_1 &= Y_m^{k_2^{(1)}} R_m^{k_1^{(1)}}, \\ &= g^{r_m q_1 q_3} \end{aligned}$$

$$\begin{aligned} T_2 &= Y_m^{k_2^{(2)}} R_m^{k_1^{(2)}}, \\ &= g^{r_m q_2 q_3} \end{aligned}$$

$$T_1^{p_1 p_3} = T_2^{p_2 p_3}$$

Using Bezout's identity, we can compute integer values a_1, a_2 such that

$$a_1 p_1 p_3 + a_2 p_2 = 1$$

Now we can write

$$\begin{aligned} T_1 &= T_1^{a_1 p_1 p_3 + a_2 p_2} \\ &= T_1^{a_1 p_1 p_3} T_1^{a_2 p_2} \\ &= T_2^{a_1 p_2 p_3} T_1^{a_2 p_2} \\ &= (T_2^{a_1 p_3} T_1^{a_2})^{p_2} \end{aligned}$$

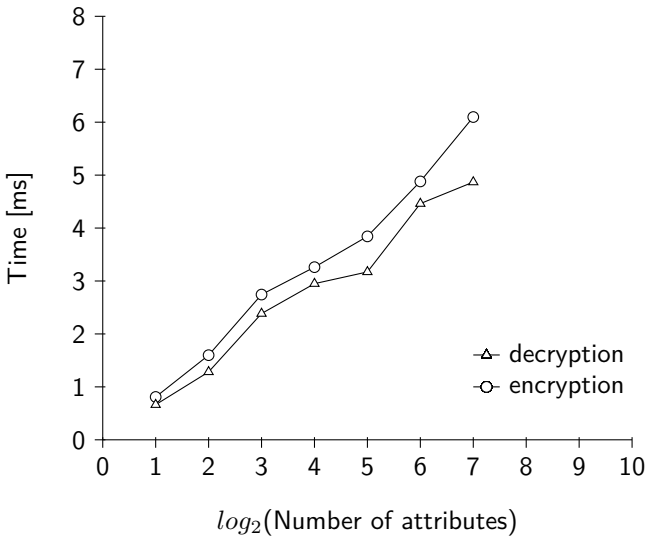
There is no way to remove p_3 from this term. So we won't get

$$K_m = T_1^{q_2} = g^{r_m q_1 q_2}$$

Hence, we prove that the attack shown in [11] is not possible.
Other proves of security analysis is the same as shown in [19]. □

6 PERFORMANCE COMPARISON

The following table shows the number of attributes in the universal set and the corresponding time taken for encrypting and decrypting a message of size 256 bytes. Note that the number of attributes in the ciphertext policy and the user was half the total number of attributes. The value of security parameter ρ and length of security parameter l_σ is 32. The execution was done on a system with Intel Core i5-7200U(2.5 GHz) CPU and 8 GB RAM, running on Ubuntu 18.04 operating system. Also, note that we have not factored the extra time taken in transmitting the message between the device and the centralized server.



7 CONCLUSION

As more and more cloud-based applications and IoT devices are introduced, it becomes necessary to have an efficient encryption and decryption system to facilitate. We have proposed a secure RSA based CP-ABE scheme with constant-size secret keys and ciphertexts. Further, we have also provided the security analysis and the intuition for the same. Currently, the scheme requires the message to be sent to a centralized server in order to perform the Validate Phase. This, however, might cause a bottleneck and extra overhead. For future work, we can look into removing this phase and thereby making the scheme more robust.

References

[1] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya. Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges. *IEEE Communications Surveys Tutorials*, 16(1):337–368, First 2014.

[2] Moreno Ambrosin, Christoph Busold, Mauro Conti, Ahmad-Reza Sadeghi, and Matthias Schunter. Updaticator: Updating billions of devices by an efficient, scalable and secure software update distribution over untrusted cache-enabled networks. In Mirosław Kutyłowski and Jaideep Vaidya, editors, *Computer Security - ESORICS 2014*, pages 76–93, Cham, 2014. Springer International Publishing.

[3] Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography – PKC 2011*, pages 90–108, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

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

[5] Cheng Chen, Jie Chen, Hoon Wei Lim, Zhenfeng Zhang, Dengguo Feng, San Ling, and Huaxiong Wang. Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures. In Ed Dawson, editor, *Topics*

in Cryptology – CT-RSA 2013, pages 50–67, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[6] Ling Cheung and Calvin Newport. Provably secure ciphertext policy abe. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 456–465, New York, NY, USA, 2007. ACM.

[7] Nishant Doshi and Devesh C. Jinwala. Fully secure ciphertext policy attribute-based encryption with constant length ciphertext and faster decryption. *Sec. and Commun. Netw.*, 7(11):1988–2002, November 2014.

[8] Keita Emura, Atsuko Miyaji, Akito Nomura, Kazumasa Omote, and Masakazu Soshi. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In Feng Bao, Hui Li, and Guilin Wang, editors, *Information Security Practice and Experience*, pages 13–23, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[9] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pages 89–98, New York, NY, USA, 2006. ACM.

[10] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan. Cp-abe with constant-size keys for lightweight devices. *IEEE Transactions on Information Forensics and Security*, 9(5):763–771, May 2014.

[11] J. Herranz. Attribute-based encryption implies identity-based encryption. *IET Information Security*, 11(6):332–337, 2017.

[12] Javier Herranz, Fabien Laguillaumie, and Carla Ràfols. Constant size ciphertexts in threshold attribute-based encryption. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography – PKC 2010*, pages 19–34, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[13] Min-Shiang Hwang. A new dynamic key generation scheme for access control in a hierarchy. *Nordic J. of Computing*, 6(4):363–371, December 1999.

[14] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 62–91, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[15] Allison Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, pages 180–198, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[16] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen. Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid. *IEEE Transactions on Parallel and Distributed Systems*, 25(8):2053–2064, Aug 2014.

[17] Jing Li, Xiong Li, Licheng Wang, Debiao He, Haseeb Ahmad, and Xinxin Niu. Fuzzy encryption in cloud computation: efficient verifiable outsourced attribute-based encryption. *Soft Computing*, 22(3):707–714, Feb 2018.

[18] J. K. Liu, M. H. Au, W. Susilo, K. Liang, R. Lu, and B. Srinivasan. Secure sharing and searching for real-time video data in mobile cloud. *IEEE Network*, 29(2):46–50, March 2015.

[19] V. Odelu, A. K. Das, M. Khurram Khan, K. R. Choo, and M. Jo. Expressive cp-abe scheme for mobile devices in iot satisfying constant-size keys and ciphertexts. *IEEE Access*, 5:3273–3283, 2017.

[20] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 195–203, New York, NY, USA, 2007. ACM.

[21] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'05, pages 457–473, Berlin, Heidelberg, 2005. Springer-Verlag.

[22] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 457–473, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[23] Damien Vergnaud. Comment on "a strong provably secure ibe scheme without bilinear map" by m. zheng, y. xiang and h. zhou j. comput. syst. sci. 81 (2015) 125-131. *J. Comput. Syst. Sci.*, 82(5):756–757, August 2016.

[24] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography Conference on Public Key Cryptography*, PKC'11, pages 53–70, Berlin, Heidelberg, 2011. Springer-Verlag.

[25] Yanjiang Yang, Haibin Cai, Zhuo Wei, Haibing Lu, and Kim-Kwang Raymond Choo. Towards lightweight anonymous entity authentication for iot applications. In Joseph K. Liu and Ron Steinfeld, editors, *Information Security and Privacy*, pages 265–280, Cham, 2016. Springer International Publishing.

[26] Yanjiang Yang, Joseph K. Liu, Kaitai Liang, Kim-Kwang Raymond Choo, and Jianying Zhou. Extended proxy-assisted approach: Achieving revocable fine-grained encryption of cloud data. In Günther Pernul, Peter Y A Ryan, and Edgar Weippl, editors, *Computer Security – ESORICS 2015*, pages 146–166, Cham, 2015. Springer International Publishing.

[27] Yanjiang Yang, Jiqiang Lu, Kim-Kwang Raymond Choo, and Joseph K. Liu. On lightweight security enforcement in cyber-physical systems. In *Revised Selected Papers of the 4th International Workshop on Lightweight Cryptography for Security and Privacy - Volume 9542*, LightSec 2015, pages 97–112, Berlin, Heidelberg, 2016. Springer-Verlag.

[28] Yanjiang Yang, Haiyan Zhu, Haibing Lu, Jian Weng, Youcheng Zhang, and Kim-Kwang Raymond Choo. Cloud based data sharing with fine-grained proxy re-encryption. *Pervasive Mob. Comput.*, 28(C):122–134, June 2016.

[29] Yinghui Zhang, Dong Zheng, Xiaofeng Chen, Jin Li, and Hui Li. Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts. In Sherman S. M. Chow, Joseph K. Liu, Lucas C. K. Hui, and Siu Ming Yiu, editors, *Provable Security*, pages 259–273, Cham, 2014. Springer International Publishing.

[30] Zhibin Zhou and Dijiang Huang. On efficient ciphertext-policy attribute based encryption and broadcast encryption: Extended abstract. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 753–755, New York, NY, USA, 2010. ACM.