CP-ABE Scheme Satisfying Constant-size Keys based on ECC

Nishant Raj and Alwyn Roshan Pais
Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Karnataka, India

Keywords: Ciphertext-policy Attribute-based Encryption, Elliptic Curve based Cryptography, Cloud Computing, Security, Constant-size Secret Key.

Abstract: Cloud-based applications, especially on IoT devices, is one of the desired fields to apply Ciphertext-Policy Attribute-Based Encryption (CP-ABE). Most of the IoT devices are with the low-end configuration; hence, they need better time and computation efficient algorithms. There are existing algorithms, but none of the systems are based on conventional cryptosystems as well as secure at the same time. Here, we propose a CP-ABE scheme based on the elliptic curve cryptosystem with a constant-size secret key, which is capable of addressing the collusion attack security issue.

1 INTRODUCTION

With the emergence of cloud computing, it becomes the need of the hour to design a mechanism that facilitates faster encryption and decryption (D.Pharkkavi and Maruthanayagam, 2018). One such scheme is CP-ABE, which is based on ABE proposed in (Sahai and Waters, 2005). 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(Zhang et al., 2014)(Li et al., 2014). Also, CP-ABE should be cost-efficient to work on battery constrained devices.

In recent times many CP-ABE schemes have been proposed, which are based on bi-linear maps, which are computationally intensive than those based on conventional cryptosystems, such as (Vergnaud, 2016)(Li et al., 2014). Hence, there is a need to design a cost-efficient access structure CP-ABE ciphertexts using conventional public-key cryptosystems, with constant-size secret keys. One such attempt was made by (Odelu et al., 2016).

A security flaw was shown in the scheme of(Odelu et al., 2016) by (Herranz, 2017). It was proven that the scheme could be broken using collusion of non-policy satisfying users. This is based on the observation that the attack is possible if the union of users attribute set satisfies access policy.

Here, we have proposed a modified scheme, originally proposed by (Odelu et al., 2016). The proposed scheme is based in ECC, so cost-efficient in both encryption and decryption, and follows the AND-gate access structure, with a constant-size secret key.

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. Following this, in 3, we propose our CP-ABE scheme. Then in 4, we discuss the security aspects of the scheme. Then, we discuss the implementation detail of the scheme in 5. Finally, in 6, we provide a few concluding remarks.

2 MATHEMATICAL PRELIMINARIES AND DEFINITIONS

In this section, we explain the various definitions and preliminaries related to CP-ABE scheme.

2.1 Attribute Definition and Access Structure

We follow a similar definition for attributes and access policy, as provided in (Guo et al., 2014). Assume that we have n attributes in $\mathbb{U}$, set of all attributes in the universe, so we have $\mathbb{U} = \{A_1, A_2, A_3, \ldots A_n\}$, where A_i represents the ith attribute in the universe. Also, for convenience we represent $\mathbb{A}$, attribute set associ-

Raj, N. and Pais, A.
CP-ABE Scheme Satisfying Constant-size Keys based on ECC.
DOI: 10.5220/0009590905350540
In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications (ICETE 2020) - SECRYPT*, pages 535-540
ISBN: 978-989-758-446-6

ated with a user, so we have $\mathbb{A} \subseteq \mathbb{U}$, as a n-bit string $a_1a_2a_3 \ldots a_n$, 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. Similarly, we represent $\mathbb{P}$ be the access policy associated with a message($\mathbb{P} \subseteq \mathbb{U}$) as an n-bit string $b_1b_2b_3 \ldots b_n$, same assignment as $\mathbb{A}$.

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 in the bit string, it is $a_1a_2a_3 \ldots a_n$. Similarly, for access policy $\mathbb{P}$ be $b_1b_2b_3 \ldots b_n$. If $a_i \geq b_i \forall i$, then we say $\mathbb{A}$ satisfies $\mathbb{P}$, in other words, $\mathbb{P} \subseteq \mathbb{A}$. From here, $\mathbb{A}$ and $\mathbb{P}$ are represented as n-bit-string.

2.2 Compuational Hard Problem

This section describes the computationally hard problems.

q-Generalized Diffie-Hellman (q-GDH) Assumption. It is hard to compute $(a_1a_2a_3 \ldots a_q)P \in G$, given $a_1P, a_2P, a_3P \ldots a_qP \in \mathbb{G}$ where P is a base point in $E_p(a,b)$ and $(\prod_{i \in S} a_i)P \in \mathbb{G}$ for all proper subsets $S \subset \{1, \ldots, q\}$. The access to all the above-mentioned subset products, which are exponential in q, is provided by an oracle. For a vector $a = (a_1, a_2 \ldots a_q) \in (Z_p)q$, define $O_{P,a}$ to be an oracle that for any proper subset $S \subset \{1, 2, \ldots, q\}$ responds with $O_{P,a} = (\prod_{i \in S} a_i)P$.

Definition 2.2.1. q-GDH Assumption: The (t, q, ε)-GDH assumption is satisfied by $\mathbb{G}$ if the advantage of all the t-time algorithms $\mathcal{A}$ is given by $ADV^{GDH}_{\mathcal{A},q} = Pr[\mathcal{A}^{O_{P,a}} = (a_1 \ldots a_q)P] < \varepsilon$, where $a = (a_1, a_2, a_3 \ldots a_q) \leftarrow (\mathbb{Z}_p)^q$ for any sufficiently small $\varepsilon > 0$.

q-DHI Problem. The q-Diffie-Hellman Inversion problem states that given a $(q+1)$-tuple $(P, xP, x^2P, \ldots, x^qP) \in \mathbb{G}^{q+1}$ as input, output $(1/x)P \in \mathbb{G}$ where $x \in Z^*_p$.

Definition 2.2.2. q-DHI Assumption (Boneh and Boyen, 2004): The (t, q, ε)-GDI assumption is satisfied by $\mathbb{G}$ if the advantage of all the t-time algorithms $\mathcal{A}$ is given by

$$ADV^{DHI}_{\mathcal{A},q} = Pr[\mathcal{A}(P, xP, x^2P, ..., x^qP) = (1/x)P] < \varepsilon$$

for any sufficiently small $\varepsilon > 0$. where the probability is considered over the random choices of x in Z^*_p and random bits of $\mathcal{A}$.

2.3 Definition of CP-ABE Scheme

There are four major algorithms in a CP-ABE scheme. They are Setup, Encrypt, KeyGen, and Decrypt. These algorithms are defined in Table 1, similarly as in (Guo et al., 2014). We have introduced Validate Phase for our scheme.

For any given (MPK, MSK), the ciphertext generated using Encrypt algorithm and the access policy $\mathbb{P}$, the plain text message M, and the user secret key k_u associated with attributes $\mathbb{A}$. If $\mathbb{P} \subseteq \mathbb{A}$ then the Decrypt algorithm should always output the correct plain text message M, otherwise, user cannot decrypt M.

2.4 Security Model - Selective Game for CP-ABE Scheme

In this section, we have defined a selective game to show restiveness against the chosen cipher-text attack. In the game, after the challenge phase, an adversary $\mathbb{R}$. issues multiple secret key queries. The game is described as follows between the challenger $\mathbb{B}$ and an adversary $\mathbb{R}$.

Game is the same as mentioned in (Cheung and Newport, 2007).

In the game, the advantage ε of $\mathbb{R}$, adversary, is given by,

$$\varepsilon = \left| Pr[c' = c] - \frac{1}{2} \right|$$

For the above scheme to be secure ε must be negligible function of ρ(security parameter).

3 PROPOSED CP-ABE-CSSK SCHEME

Here, we propose a CP-ABE scheme with constant-size secret keys based on ECC, i.e., CP-ABE-CSSK-ECC. Other notations we use are enlisted in Table 2.

The scheme consists of five phases, as follows:

3.1 Setup Phase

In this phase,

Input: Security parameter ρ and the universe of attributes $\mathbb{U} = \{A_1, A_2, \ldots A_n, A_{n+1}\}$.

Output: Master Secret Key MSK and Master Public Key MPK.

Note: Here, we add one extra attribute A_{n+1}, which is 1 for every user and 0 for every access policy in bit strings.

The algorithm works the same as in (Odelu et al., 2016).

Table 1: Inputs and Outputs for various phases in Defination Scheme.

Phase	Input	Output
Setup Phase	$\mathbb{U}, \rho$	MSK, MPK
KeyGen Phase	$\mathbb{A}, MSK, MPK$	$k_u = (u_1, u_2)$
Encrypt Phase	$\mathbb{P}, MPK, M$ message	$C = \{\mathbb{P}, P_{m,i}, K_{1m}, K_{2m}, C_{\sigma_m}, C_m\}$
Validate Phase (Only our scheme)	$C = \{\mathbb{P}, P_{m,i}, K_{1m}, K_{2m}, C_{\sigma_m}, C_m\}$	C or (fail/no transmit)
Decrpyt Phase	$C, k_u, \mathbb{A}, MPK$	message M or $\perp$

Table 2: Notations.

Notation	Description
$"(\alpha, k_1, k_2)$	System private keys
p	A large prime number
$E_p(a,b)$	An elliptic curve $y_2 = x_3 + ax + b(mod p)$ defined over the finite field Z_p ;
Z_p	$\{0, 1, \ldots, p-1\}$
P	A base point in $E_p(a,b)$ whose order is a 160-bit number in Z_p
xP	$P + P + \ldots P(x$ times), scalar multiplication, $P \in E_p(a,b)$
$\mathbb{G}$	Elliptic curve group $\{p, E_p(a,b), P\}$ generated by P
$\mathbb{U}$	Universe of $(n+1)$ attributes
	$\{A_1, A_2, A_3, \ldots, A_n, A_{n+1}\}$
$\mathbb{A}$	User set attributes, $\mathbb{A} \subseteq \mathbb{U}$
$\mathbb{P}$	Access policy, $\mathbb{P} \subseteq (\mathbb{U} \setminus A_{n+1})$" (Odelu et al., 2016)

3.2 Encrypt Phase

The Encrypt phase is the same as in (Odelu et al., 2016), with taking into changes of value of A_{n+1}.

In encryption algorithm,

Input: An access policy $\mathbb{P} \subseteq \mathbb{U}$ where $|\mathbb{P}| = 0$, the Master Public Key MPK and a plaintext message M.

Output: The ciphertext C as $C = \{\mathbb{P}, P_{m,i}, K_{1m}, K_{2m}, C_{\sigma_m}, C_m\}$.

The algorithm works the same as in (Odelu et al., 2016).

3.3 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 (Herranz, 2017) 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.

3.4 Key-gen Phase

In this phase,

Input: The Master Secret Key MSK and the Master Public Key MPK.

Output: The user secret key k_u corresponding to the user attributes $\mathbb{A}$

The algorithm works the same as in (Odelu et al., 2016).

3.5 Decrypt Phase

In this phase,

Input: The ciphertext $C = \{\mathbb{P}, P_{m,i}, K_{1m}, K_{2m}, C_{\sigma_m}, C_m\}$ corresponding to a given access policy $\mathbb{P}$, the user secret key k_u corresponding to the user attribute $\mathbb{A}$.

Output: Original message M, if successful, else $\perp$.

The algorithm works the same as in (Odelu et al., 2016).

Assumption: *All transmissions of data between phases are completely secure.*

4 SECURITY ANALYSIS

Here, we provide proof of security against some possible known attacks. Then, prove our intuition regarding why we choose the extra attribute to prevent the attack(Herranz, 2017) with proof.

4.1 Security against Collusion Attack

We analyze the situation where multiple adversaries, having valid secret keys corresponding to their attributes, collaborate, and try to generate the system's private keys (k_1, k_2).

Theorem 4.1. The scheme is secure against collusion attacks by adversaries who aim at deriving the system's private key pair (k_1, k_2).

Proof as mentioned in (Odelu et al., 2016) and includes proof in section 4.6

4.2 Security against Key Recovery Attack

This section provides an analysis of the proposed scheme where an adversary attempts to obtain a valid user secret key corresponding to the attribute set $\mathbb{A}$.

Theorem 4.2. This scheme is secure against an adversary who tries to derive valid user secret key $k_u = (k_1, k_2)$ corresponding to the attribute set $\mathbb{A}$.

Proof as mentioned in (Odelu et al., 2016) and includes proof in section 4.6

4.3 Security against Adversary Deriving Original Message without Secret User Keys

In this subsection, we discuss what adversary can derive from the ciphertext without knowing a valid user secret key.

Theorem 2.4.3. Our proposed scheme is secure against an adversary who executes a message recovery attack and tries to derive the original message without knowing the secret user key $k_u = (u_1, u_2)$.

The proof is the same as mentioned in (Odelu et al., 2016).

4.4 Description of the CP-ABE Scheme (Odelu et al., 2016)

Our CP-ABE scheme differs from the CP-ABE Scheme (Odelu et al., 2016), as we have added one extra attribute as A_{n+1}, where $A_{n+1} \in \mathbb{A}_i$ and $\mathbb{A}_i$ is the attribute set of i^{th} user.

KeyGen ($\mathbb{A}$, MSK, PMS): Step to compute generate key (Key-Gen Phase):

1. Select two random numbers, $r_u, t_u \in \mathbb{Z}_p$
2. Choose s_u such that $\frac{1}{f(\alpha,\mathbb{A})} = (k_1 s_u + k_2 r_u) \bmod p$. That is:

$$s_u = \frac{1}{k_1}.\left(\frac{1}{f(\alpha,\mathbb{A})} - k_2 r_u\right) \bmod p \qquad (1)$$

3. Compute $u_1 = r_u + k_1 t_u \bmod p$ and $u_2 = s_u - k_2 t_u \bmod p$. We have the secret key as $sk_{\mathbb{A}} = (u_1, u_2)$.

Note (1): The attack was possible from the fact that entropy of the secret key, $sk_{\mathbb{A}} = (u_1, u_2)$, is not enough. Even though they used two random and independent values r_u, t_u, the final secret key $sk_{\mathbb{A}} = (u_1, u_2)$ is not independent.

Proof: Let us write the relation between the pairs (r_u, t_u) and (u_1, u_2), we get

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 & k_1 \\ \frac{k_2}{k_1} & -k_2 \end{bmatrix} . \begin{bmatrix} r_u \\ t_u \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{k_1 f(\alpha,\mathbb{A})} \end{bmatrix} \bmod p \qquad (2)$$

The matrix is not invertible: the first one multiplied with $-\frac{k_2}{k_1}$ equals the second row. Thus, we have

$$u_2 = -\frac{k_2}{k_1} u_1 + \frac{1}{k_1 f(\alpha,\mathbb{A})} \bmod p. \qquad (3)$$

4.5 Attack on CP-ABE Scheme (Odelu et al., 2016) by (Herranz, 2017)

According to the paper (Herranz, 2017), the attack is based on three simple observations, of which the first and third are not relevant to our solution. Here is the second:

(b). Pair information (X, Y_B) (derived as in (Herranz, 2017)) is enough for each subset of attributes B to create a valid secret key sk_B for subset B. Again, using the Note (1) in previous subsection i.e., equation (1), we choose a random $u_1 \in \mathbb{Z}_p$ and compute:

$$u_2 = -\frac{k_2}{k_1} u_1 + \frac{1}{f(\alpha,\mathbb{B})} = X u_1 + Y_B \bmod p \qquad (4)$$

Proof using Example. Let us take n = 4 and the subsets of attributes defined by bit strings $A_1 = 0011, A_2 = 1100, A_3 = 0100, B = 1011$, where, A_i is i^{th} user's attribute set and B is access policy. We can easily see the equality holds:

$$f(\alpha, B) = \frac{f(\alpha, A_1).f(\alpha, A_2)}{f(\alpha, A_3)} \bmod p. \qquad (5)$$

Now, for above subsets of attributes, we get:

$$\begin{aligned} Y_B &= \frac{1}{k_1 f(\alpha,B)} = \frac{1}{k_1 \frac{f(\alpha,A_1).f(\alpha,A_2)}{f(\alpha,A_3)}} \\ &= \frac{\frac{1}{k_1 f(\alpha,A_1)} \cdot \frac{1}{k_1 f(\alpha,A_2)}}{\frac{1}{k_1 f(\alpha,A_3)}} = \frac{Y_{A_1}.Y_{A_2}}{Y_{A_3}} \bmod p. \end{aligned} \qquad (6)$$

Using all three observation (Herranz, 2017) have shown a possible attack.

Note. The above attack is a key recovery attack that is even stronger than an attack against the IND-CPA property.

4.6 Security Analysis of Our Solution

As in equation (6) we have established:

$$Y_B = \frac{1}{k_1 f(\alpha, B)} = \frac{Y_{A_1}.Y_{A_2}}{Y_{A_3}} \bmod p.$$

where, $A_1 = 1100, A_2 = 0011, A_3 = 0100$ and $B = 1011$.

To Proof. The above equation is not possible in our scheme, as it is essential to perform attack as in (Herranz, 2017).

As mentioned before, we have introduced one more attribute, which is included in every user's attribute set and excluded from the access policy attribute set. So, new $A_1 = 11001, A_2 = 00111, A_3 = 01001$ and $B = 10110$, now, the bit string length is $n+1$.

As mentioned earlier,

$$f(x, \mathbb{A}) = \prod_{i=1}^{n+1} (x + H_4(i))^{1-a_i},$$

with degree of $n+1-|\mathbb{A}|$

Let $a_1 a_2 \ldots a_n a_{n+1}$ be the bit string of attribute set $\mathbb{A}$, for values of $a_i = 0 : 1 <= i <= n+1$, we have $\alpha + H_4(i)$ in $f(\alpha, \mathbb{A})$, otherwise it is 1. Now, $f(\alpha, \mathbb{A})$ can be redefined as:

$$f(\alpha, \mathbb{A}) = \prod_{i=1}^{n+1} \begin{cases} (\alpha + H_4(i)), & \text{if } a_i = 0 \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

Folowing equation (7), we can say, $f(\alpha, A_1)$, $f(\alpha, A_2)$ and $f(\alpha, A_3)$ doesn't contains $(\alpha + H_4(n+1))$, whereas, $f(\alpha, B)$ contains $(\alpha + H_4(n+1))$.

Proposition: In our CP-ABE scheme with any algebraic combination of Y_{A_i} we cannot generate Y_B.

Proof to the proposition: We know,

$$Y_{A_i} = \frac{1}{k_1 f(\alpha, A_i)} \text{ and } Y_B = \frac{1}{k_1 f(\alpha, B)}.$$

Hence, $Y_{A_i} \propto \frac{1}{f(\alpha, A_i)}$ and $Y_B \propto \frac{1}{f(\alpha, B)}$.

Considering equation,

$$Y_B = (\text{Product and Division combination of any } Y_{A_i}) \bmod p. \quad (8)$$

In the left-hand side of equation (6) contains $(\alpha + H_4(n+1))$, whereas, any term in right-hand side doesn't contain $(\alpha + H_4(n+1))$ term. So, the generation of the left-hand side from the right-hand side is highly improbable. So, we have,

$$Y_B \neq (\text{Product and Division combination of any } Y_{A_i}) \bmod p.$$

with overwhelming probability.

Hence, it proved the proposition.

So, equation (8) doesn't hold good for our CP-ABE scheme; hence, the attack mentioned in (Herranz, 2017), also in section 4.5, is not possible on our CP-ABE scheme, as 4.5.(b) does not give value Y_B, and the rest of the attack would be unsuccessful.

5 IMPLEMENTATION DETAILS

We have implemented the proposed scheme in C++, using Crypto++, an open-source, the cryptographic library. System configuration that we used: Intel Core i5-4210U CPU, 8 GB RAM, processor speed 2.7GHz $\times$ 4, and 64-bit Linux-based OS.

The bottleneck will be in calculating the i^{th} coefficient of polynomial $f(x, \mathbb{P})$. To compute the coefficient of x^i of these polynomials, we can have the following methods with time complexity:

- Brute Force: $O(2^n)$
- Dynamic Programming approach: $O(n^2)$
- Divide and conquer approach: $O(n^{log_2(3)})$
  - Karatsuba Method (modified)

5.1 Dynamic Programming Approach

We have defined the sub problem in co-efficent calculation of polynomial, $f(x) = \prod_{k=1}^{n'} (x + c_k)$, of degree n'.

$Table(i, j) =$ coefficient of x^{i-j} in the polynomial

$$\left(\prod_{k=1}^{i} (x + c_k) \right)$$

for $i, j = 0, 1, 2, \ldots n'$. Where $\left(\prod_{k=1}^{i} (x + c_k)\right)$ is the product of first i terms of $f(x)$.

The time and space complexity of the above approach is $O(n'^2)$ and $O(n')$, respectively.

5.2 Divide and Conquer Approach

Karatsuba Algorithm. We extend this approach to find coefficients of polynomial function $f(x) = \prod_{k=1}^{n'} (x + c_k)$. Let polynomials two $A(x)$ and $B(x)$ be defined as:

$$A(x) = A_0 + A_1 x + A_2 x^2 + \cdots + A_{n'} x^{n'}$$

$$B(x) = B_0 + B_1 x + B_2 x^2 + \cdots + B_{n'} x^{n'}$$

Let $A(x)$ be expressed in terms of two polynomial $A'(x)$ and $A''(x)$ as described below:

$$A(x) = A'(x) + x^{n'/2} A''(x) \text{ and}$$

$$B(x) = B'(x) + x^{n'/2} B''(x)$$

Now, the Karatsuba algorithm recursively computes the following three products:

- $X(x) = A'(x) \times B'(x)$
- $Y(x) = A''(x) \times B''(x)$
- $Z(x) = (A'(x) + A''(x)) \times (B'(x) + B''(x))$

The product of $A(x)$ and $B(x)$ will be given by,

$$A(x) \times B(x) = X(x) + x^{n'/2}$$
$$(Z(x) - X(x) - Y(x)) + x^{n'} Y(x)$$

The recurrence relation formulated to be:

$$T(n') = 3T(n'/2) + O(n')$$

So the time complexity for polynomial multiplication by Karatsuba algorithm is $O(n'^{(log_2(3))})$ where n' is the degree of polynomials.

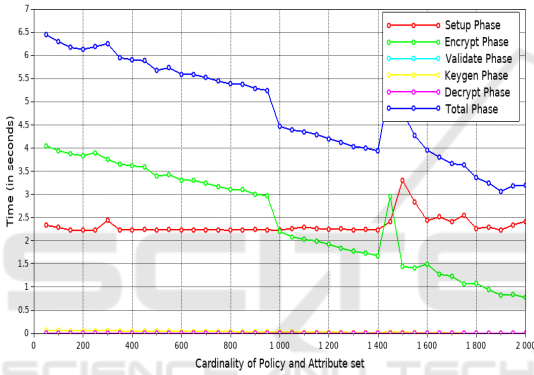

Figure 1: Execution Time of different phases in proposed scheme - Calculating coefficients of $f(x) = \prod_{k=1}^{n'} (x + c_k)$ using Divide and conquer approach (Karatsuba Method) for $|\mathbb{U}| = n' = 2000$.

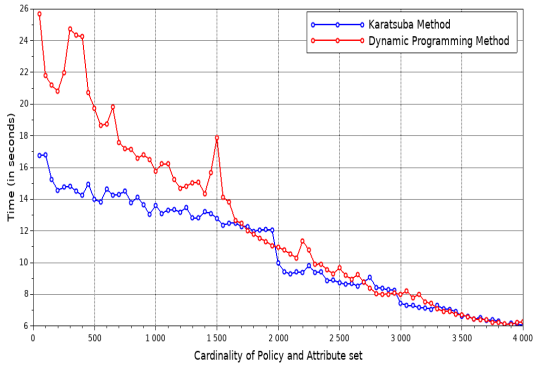

Figure 2: Execution time comparison of Dynamic Programming algorithm and Karatsuba algorithm for all 5 phase together for $|\mathbb{U}| = n' = 4000$.

6 CONCLUSION

With the boom in cloud-based applications and IoT devices in the market, and an efficient CP-ABE scheme is a necessity. We have proposed a secure ECC based CP-ABE scheme with constant-size secret keys. Further, we have also provided the security analysis and the intuition for the same.

In this paper, we require a centralized server to perform the Validate Phase. This, however, may be the cause for a bottleneck or an extra overhead. For future work, we can look into removing this validation phase and thereby making the scheme more robust.

REFERENCES

Boneh, D. and Boyen, X. (2004). Efficient selective-id secure identity-based encryption without random oracles.

Cheung, L. and Newport, C. (2007). 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. ACM.

D.Pharkkavi and Maruthanayagam, D. D. (2018). Time complexity analysis of rsa and ecc based security algorithms in cloud data. *International Journal of Advanced Research in Computer Science*, 9(3).

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

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

Li, H., Lin, X., Yang, H., Liang, X., Lu, R., and Shen, X. (2014). 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.

Odelu, V., Das, A. K., and Goswami, A. (2016). An efficient cp-abe with constant size secret keys using ecc for lightweight devices. *ACM*, 9(17):4048–4059.

Sahai, A. and Waters, B. (2005). 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. Springer-Verlag.

Vergnaud, D. (2016). 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.

Zhang, Y., Zheng, D., Chen, X., Li, J., and Li, H. (2014). Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts. In Chow, S. S. M., Liu, J. K., Hui, L. C. K., and Yiu, S. M., editors, *Provable Security*, pages 259–273, Cham. Springer International Publishing.