Pro-SRCC: Proxy-based Scalable Revocation for Constant Ciphertext Length

Zeya Umayya*, Divyashikha Sethia†
Department of Computer Science and Engineering, Delhi Technological University

New Delhi, India
Email: *zeyaumayya@gmail.com, †sethiadivya@gmail.com

***Abstract*—Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a fine-grained encryption technique, which can provide selective access control. Although it is computationally expensive, it has been proved feasible on resource-constrained devices, such as mobile devices and Internet of Things (IoT) devices. We look into the use case of storing important information, such as health records or sensor information from such devices by the user locally or through direct selective access by various users based on their roles. It must protect the information from malicious users with the support of an efficient revocation scheme. It must provide uninterrupted access to the unrevoked users without re-encryption or redistribution of keys. In this paper, we review the Emura's constant ciphertext CP-ABE scheme, which offers the advantage of retaining constant-sized ciphertext on resource-constrained devices. We propose a novel scheme called Proxy-based Scalable Revocation for Constant Ciphertext Length (ProSRCC) to improve it for scalable revocation without re-encryption and re-distribution of keys. It uses a trusted proxy server for partial decryption and revocation of users. The paper presents ProSRCC's design and implementation on the Pairing-based cryptography (PBC) library and compares it with the Proxy based Immediate Revocation of ATTribute-based Encryption (PIRATTE) and Emura's constant length CP-ABE schemes. The results indicate that computation time for ProSRCC is least as compared to the other schemes. Hence, it is beneficial to encrypt information with ProSRCC and get constant-sized ciphertext, as well as support for scalable revocation especially on static and resource-constrained devices.**

***Keywords*:—PIRATTE; CP-ABE; ProSRCC.**

I. INTRODUCTION

An Attribute-based Encryption (ABE) is an encryption scheme, where different users have specific attributes and can decrypt a given ciphertext, which is associated with an access policy of these attributes. Characteristics of a user, e.g., his name or date of birth can be used for access control of important resources and information. Schemes, such as [1] [2], are an example of the Identity-Based Encryption (IBE) scheme, which does not disclose the identity of the decryptor in any case. Canetti et al. [3] proposed the first ABE scheme inspired by IBE. In the IBE schemes, there is a one-to-one relationship between an encryptor and a decryptor and the schemes assign only one decryptor for an encryptor. Whereas the ABE schemes assign many decryptors to a single encryptor by assigning some common attributes to the decryptors, such as mail ID, gender, age and so on. The ABE schemes have two variants namely Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE). The KP-ABE [1] [3] is a scheme such that it associates each user's private key with an access structure. However, in the CP-ABE schemes, an access-structure is defined for each ciphertext, which means that an encrypting party can decide who should be allowed to access the ciphertext. However, in the earlier ABE schemes [7] [8], the ciphertext length was dependent on the number of attributes present in the access structure. Also, the number of pairing computations increased with an increase in the number of attributes. Boneh et al. [4] and Katz et al. [5] presented the idea of the Predicate Encryption Scheme (PES) in which the predicates and attributes are associated with the users and ciphertexts respectively. According to Boneh et al. [4] and Katz et al. [5], PES is another variant of the CP-ABE scheme. However, both the schemes [4] [5] suffered from the problems of increase in the number of pairing computations and the length of the ciphertext with the increase in the number of attributes.

According to the survey of the existing techniques presented by Hwang et al. [6], an ideal ABE scheme must have the following capabilities:

- *Data confidentiality:* Any unauthorized participant cannot find out any information about the encrypted data.
- *Fine-grained access control:* For access control to be flexible, the access rights, even for the users of the same group, are different.
- *Scalability:* The overall performance of an ABE scheme will not go down with the total number of approved participants. Thus, we can say that an ABE scheme can deal with the case where the number of the authorized users increases dynamically.
- *Attribute or user-based revocation:* If any participant leaves the system, then his access rights will be revoked by the ABE scheme. Similarly, attribute revocation is inevitable.
- *Accountability:* In all previous the ABE schemes, the dishonest/illegal users were able to directly distribute some part of the transformed or original keys such that nobody will know the real distributor of these keys. Accountability should prevent the above problem, which is called key abuse.
- *Collusion resistance:* The unauthorized users cannot de-

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-661-3

crypt the secure data by combining their attributes to match the access policy.

The length of ciphertext plays an important role in any CP-ABE system. Cloud storage systems are capable of storing long ciphertexts, but for those devices where space is limited, an increase in the length of ciphertext can become a problem. Emura et al. [10] provided a solution of constant length CP-ABE scheme. The number of pairing computations also affects the time taken to either encrypt or decrypt. In the Emura et al.'s [10] scheme, the number of pairing computations is also constant for both encryption and decryption.

Revocation is an essential feature for CP-ABE schemes. According to Jiang et al. [16], revocation can be done using direct and indirect methods. The indirect methods require re-encryption of the ciphertext after revocation. Re-encryption involves the regeneration of ciphertext and secret keys. However, in the direct method, re-encryption is not necessary. There are different revocation techniques proposed to date. For resource constrained devices, re-encryption is costly and time-consuming and can interrupt the service for unrevoked users. Li et al. [18] have proposed a revocation scheme based on Emura et al.'s [10] CP-ABE scheme for both user and attributes. However, it requires re-encryption and key re-generation. Jahid et al. [19] proposed another such scheme for revocation, named Proxy based Immediate Revocation of ATTribute-based Encryption (PIRATTE). Their scheme uses a trusted proxy server and enhances the Bethencourt et al.'s [7] CP-ABE scheme. However, both the schemes suffer from the increasing ciphertext size problem. Proxy-based solutions have been proposed based on a proxy server, a third party which should be online all the time, to ensure malicious user revocation. Such schemes divide the user secret-key into two parts. The proxy server keeps a revocation list, and one part of the user secret-key to itself and the user keeps the other part. Whenever the Trusted Computing Authority (TCA) discovers a malicious user or some attributes to be revoked, it lists them in the revocation list held by the proxy server. Decryption involves two steps: First, the proxy does partial decryption using part of the key held by it. Then, the user receives this part and continues with the rest of the decryption process. The proxy causes the partial decryption to fail for revoked users and hence, they cannot decrypt the ciphertext successfully [20].

A. Contribution

- We propose a Proxy-based Scalable Revocation for Constant Ciphertext Length (ProSRCC) scheme for improving the Emura et al.'s [10] scheme for scalable revocation. A trusted proxy server calculates a partial decryption element and passes it to all users such that users in the revocation list get revoked, and the unrevoked users can decrypt without interruption. Based on this element, only the legitimate users can obtain access to the ciphertext. The ProSRCC does not require re-encryption of the ciphertext or re-distribution of the keys. The proxy server and the revocation list are enough to handle the access control.
- Experimental results and comparison of ProSRCC with the existing techniques indicate that it is an efficient and scalable revocation scheme.
- We present a Case Study using ProSRCC for resource-constrained devices with scalable revocation, such as accessing a food vending machine using the user's mobile device for allowing access to selective food items based on the user's role.

B. Organization

The rest of the paper is organized as follows. Section II discusses the related work for the previous CP-ABE schemes and revocation schemes. Section III presents the preliminary construction, some definitions and notations used in the paper. We also describe the CP-ABE scheme with constant ciphertext length in Section III. Section IV explains the proposed revocation scheme ProSRCC followed by its implementation in Section V. We present the experimental results in Section VI, which is followed by a case study on a smart food vending machine in Section VII. We finally conclude the paper in Section VIII.

II. Related Work

A. Basic CP-ABE

There are several CP-ABE schemes introduced to date. They require access policies using attributes within the encryption procedure. Sahai and Waters (SW) [7] first presented the idea of access policies over attributes. They suggest that there must be an association of both the secret keys and ciphertexts with some sets of attributes. Decryption is possible only if the secret key and ciphertext attribute set overlap each other.

Goyal et al. [1] suggested the possibility of a CP-ABE scheme, but they did not provide any constructions. In a CP-ABE scheme, every user's secret key is associated with an arbitrary number of attributes expressed as strings and the ciphertext is associated with an access structure. A user can decrypt a ciphertext, only if his attributes satisfy the access structure related to the ciphertext. Goyal et al. [11] and Liang et al. [12] use a bounded tree as access structure. Goyal et al. [11] presented a bounded CP-ABE scheme and gave an idea of generalizing the approach to show how to transform a KP-ABE scheme into an equivalent CP-ABE scheme. Ibraimi et al. [7] [13] have used the tree access structure to remove the boundary constraints presented in [11] [12] and proposed a new CP-ABE scheme without using Shamir's threshold secret sharing. Bethencourt et al. [7] provided an implementation of CP-ABE scheme and has an open source CP-ABE-toolkit.

B. CP-ABE Schemes Supporting AND Gate Access Policy

Cheung et al. [8] introduced a new CP-ABE scheme, which supports AND gate access policy with two types of attributes, positive and negative attributes. It terms the attributes, which participate in the access policy as positive terms. The scheme is secure under the standard model. For those attributes, which are not be a part of the access structure, it uses a wildcard (do not care) element. The scheme is Chosen Ciphertext Attack

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-661-3

(CPA) secure under the Decisional Bilinear Diffie-Hellman (DBDH) assumption. Moreover, it improves the security proof in Bethencourt et al. [7]. Unfortunately, Cheung et al.'s [8] scheme has two drawbacks. Firstly, it is not flexible enough since it supports only policies with the logical conjunction. Secondly, the size of the ciphertext and the secret key linearly increase as the number of attributes increase in this scheme. Hence, this scheme is less proficient as compared to Bethencourt et al.'s CP-ABE scheme [7].

Based on Cheung et al.'s [8] scheme, Nishide et al. [9] and Emura et al. [10] further improved the efficiency and provided hidden access policies. Nishide et al. [9] also proposed another scheme, which supported the AND gate access policy on multi-valued attributes. Emura et al. [10] have used the same access policy and further improved the scheme to achieve a constant number of bilinear pairing operations along with a constant length of ciphertext.

C. CP-ABE with Revocation

The revocation feature is essential for encryption systems to deal with the malicious behavior of users. However, addition of the revocation feature in ABE schemes is much more complicated than any public key cryptosystem or IBE schemes. The design of revocation mechanisms in previous CP-ABE schemes was difficult as users with same attributes might have been holding same user secret key.

There are two methods to realize revocation: indirect revocation method and direct revocation method. In an indirect revocation method, the owner delegates authority to execute the revocation function, which releases a key-update material after every delegation, in such a way that only non-revoked users will be able to update their keys. An advantage of the indirect revocation method is that the data owner does not need to know the revocation list. However, the disadvantage of the indirect revocation method is that all non-revoked users need communication from the respective authority at all time slots in the key-update phase. Some related attribute revocable ABE schemes, which used the indirect method, have been proposed. In the direct revocation method, the data owner performs direct revocation, which specifies the revocation list while encrypting the ciphertext. The benefit of the direct revocation method over the indirect revocation one is that there is no requirement for a key-update phase for all non-revoked users who are interacting with the authority.

Attrapadug et al. [21] first proposed a hybrid ABE (HR-ABE) scheme, which utilized the advantage of both indirect and direct methods. Jahid et al. [19] proposed a proxy-based solution for revocation scheme called Proxy-based Immediate Revocation of ATTribute-based Encryption (PIRATTE). In their scheme, the proxy is trusted minimally and also it is not able to decrypt ciphertexts on its own. The proxy has a part of the key, so each time before decryption proxy calculates a proxy data, which assists in decryption. The PIRATTE scheme provides both user and attribute-level revocation. It involves two additional costs before decryption: re-generation of the elements held by the proxy server and reconstruction of the ciphertext elements specific to the leaves in the tree access policy. The PIRATTE scheme uses the idea of re-encryption by the proxy server. However, it can revoke only a limited number of users.

In the PIRATTE scheme, the key authority generates a polynomial P of degree t over Z_p. Here, t is the maximum number of users, which can be revoked at a time. The user's secret keys are blinded with P(0). All users get a share of the polynomial P. For a revoked user, the proxy share takes its share and adds it into the proxy-key. Thus, any revoked user will not be able to get the plaintext from a ciphertext as it does not have enough points to unblind their secret key.

Sethia et al. [23] presented another novel scheme Scalable Proxy-based Immediate Revocation For CP-ABE Scheme (SPIRC) for user revocation. It improves the PIRATTE scheme for scalable user revocation. However, since it is based on Bethencourt's CP-ABE scheme [7], the length of the ciphertext is not constant.

Zhang et al. [22] have proposed a revocation technique using the subset difference scheme, which supports the attribute level revocation. In this scheme, the authors have changed the access structure completely. Instead of taking the attribute set, they have taken the set of users satisfying a subset of attributes. Their scheme ensures forward and backward secrecy. Li et al. [18] have proposed an efficient and attribute revocable scheme for cloud-based systems. They have used the same access policy as Emura et al.'s [10] scheme, which is AND-gates on multi-value attributes. Their scheme sends a key-update message to users for updating their keys. In case of user revocation, the non-revoked users must again update the authorization key, which interrupts the access.

In this paper, we propose a novel revocation scheme, which is based on Emura et al.'s [10] CP-ABE scheme. It improves it for scalable user revocation and allows uninterrupted access to non-revoked users. Hence, it can be used for direct selective access for information on resource-constrained static devices, such as a mobile-based health card or a static food vending machines.

III. PRELIMINARY CONSTRUCTION

A. Bilinear Group

Bilinear groups make the CP-ABE scheme secure against various attacks. The algebraic groups are called bilinear groups, which are groups with bilinear map.

Definition (Bilinear map). Assume $G1, G2$, and $G3$ are three multiplicative cyclic groups of prime order p. A bilinear mapping is done as follows

$e : G1 \times G2 \rightarrow G3$

e is a deterministic function; it takes one element from each group *G1* and *G2* as input, and then produces an element of group *G3*, which satisfies the following criteria:

1) *Bilinearity :* For all

$$x \in G1, y \in G2, a, b, \quad e(x^a, y^b) = e(x, y)^{ab}.$$

2) *Non degeneracy:* $e(g_1, g_2) \neq 1$ where g_1 and g_2 are generators of group G1 and G2 respectively.

TABLE I. LIST OF NOTATIONS

Notations	*Their Meaning*
PK, MK, SK_L	User's Public, Master and Secret Keys
M, C, RL	Message, Ciphertext, Revocation List
L/L_u	User attribute list associated with a user, also called user access structure
W/W_c	Access structure associated with ciphertext
$G1, G2, G3, GT$	Multiplicative cyclic groups of order p
e, g	Pairing, Generator of multiplicative cyclic group
C_{user_i}	An element computed by proxy server for the ith user to be used in decryption.
C_{attr_i}	An element computed by proxy server for the ith user to be used only in decryption.
K_{attr_i}	An element computed by key authority (KA) for the ith user to be used by proxy server.

3) *e* must be computed efficiently.

Table I defines the different notations used throughout the paper.

B. Emura et. al's Constant Ciphertext Length CP-ABE Scheme [10]

In this section we describe the basic algorithms for the different phases of the Emura et. al's [10] scheme.

- **Setup:** It takes the security parameter *K* as an input and produces two keys, a public key *PK*, and a master key *MK*.
- **KeyGen:** It takes the keys *PK*, *MK*, and a set of user attributes *L* as input and produces a user secret key SK_L associated with user's attribute list L_u.
- **Encrypt:** It takes the key *PK*, a message *M* and an access structure *W* as input. It produces a ciphertext *C* such that a user with secret key SK_L can decrypt the ciphertext *C* if $L_u \models W_c$, i.e the attribute list L_u satisfies the access structure W_c.
- **Decrypt:** It takes *PK*, ciphertext *C*, which is encrypted by W_c, and SK_L as inputs. It returns *M* if user attribute list L_u, which is associated with SK_L satisfies W_c.

1) Definition of Access Structures

Previous ABE schemes have used different variants of access structures, such as tree-based, threshold structure, linear, AND-gates with positive and negative attributes along with wild-cards and AND-gates on multi-valued attributes. This scheme uses the sum of master keys to achieve the constant ciphertext length. Hence, it uses AND-gates on multi-valued attributes. They are defined as follows:

Definition 1. Let $Univ = att_1, .., att_n$ be a set of all possible attributes. For $att_i \in Univ$, $Si = v_{i,1}, v_{i,2}, .., v_{i,n_i}$ is a set of all possible values, where n_i is the total number of possible values for att_i. Let $L_u = [L_{u1}, L_{u2}, .., L_{un}], L_{ui} \in S_i$ be an attribute list for a user, and $W_c = [W_{c1}, W_{c2}, .., W_{cn}], W_{ci} \in S_i$ be an access structure defined on a ciphertext. The notation $L_u \models W_u$ expresses that an attribute list L_u satisfies an access structure W_c , namely, $L_{ui} = W_{ci} (i = 1, 2, ..., n)$.

The number of access structures are $\prod_{i=1}^{n} ni$. For each att_i, an encryptor has to explicitly indicate a status $v_{i,*}$ from $Si = v_{i,1}, v_{i,2}, .., v_{i,n_i}$.

The access structure of our scheme ProSRCC is based on AND-gate access structure. It does not include wild-cards as it has been used in [7] [12]. In [12], an access structure W_c is defined as $W_c = [W_{c1}, W_{c2}, .., W_{cn}]$ for $W_{ci} \subseteq S_i$, and $L_u \models W_c$ is defined as $L_{ui} \in W_{ci} (i = 1, 2, ..., n)$. ProSRCC access structure is a subset of the access structures used in [7] [12]. However, even if previous CP-ABE schemes [7] [12] use AND-gate access structure with multivalued attributes, then the length of their ciphertext still depends on the number of attributes.

2) Details of the Algorithms

The details of the algorithms for the Emura et al.'s [10] scheme are:

- **Setup Algorithm**
  A Trusted Certified Authority (*TCA*) selects a prime number *p*, a bilinear group *(G1,GT)* with order *p*, a generator $g \in G1, h \in G1, y \in Z_p$ and $t_{i,j} \in_R Zp (i \in [1, n], j \in [1, n_i])$. *TCA* computes $Y = e(g, h)^y$, and $T_{i,j} = g^{t_{i,j}} (i \in [1, n], j \in [1, n_i])$. *TCA outputs* $PK = (e, g, h, Y, T_{i,j} i \in [1, n], j \in [1, n_i])$ *and* $MK = (y, t_{i,j} i \in [1, n], j \in [1, n_i])$.
  Note that we assume
  $\forall L_u, L'_u (L_u \neq L'_u), \sum_{v_{i,j} \in L_u} t_{i,j} \neq \sum_{v_{i,j} \in L'_u} t_{i,j}$.
- **Keygen Algorithm**
  KeyGen (PK, MK, L_u): The TA chooses $r \in_R Zp$, and outputs the secret key $SK_L = (h^y (g^{\sum_{v_{i,j} \in L_u} t_{i,j}})^r, g^r)$, and sends it to a user with access structure L_u.
- **Encrypt Algorithm**
  Encrypt (PK, M, W_c): An encryptor chooses $s \in_R Zp$ and computes $C1 = M.Y^s, C2 = g^s$ and $C3 = (\prod_{v_{i,j} \in W_c} T_{i,j})^s$. The encryptor outputs the ciphertext $C = (W_c, C1, C2, C3)$.
- **Decrypt Algorithm**
  Decrypt (PK, C, SK_L)*:* Before decryption, it checks if the access structure of the user and access structure related to the ciphertext are equal or not. If they are not same, it means that particular user can not access the ciphertext. However, if they are the same then decryption is done as follows:

$$= \frac{C1.e(C3, g^r)}{e(C2, h^y.(g^{\sum_{v_{i,j} \in L_u} t_{i,j}})^r)}$$
$$= \frac{M.e(g, h)^{sy}.e(g, g)^{s.r.\sum_{v_{i,j} \in W_c} t_{i,j}}}{e(g, h)^{sy}.e(g, g)^{s.r.(\sum_{v_{i,j} \in L_u} t_{i,j})}}$$
$$= M$$

This way, the decryption of the ciphertext is successful.

IV. PROXY-BASED SCALABLE REVOCATION FOR CONSTANT CIPHERTEXT LENGTH (PROSRCC) SCHEME

In this paper, we propose a novel proxy-based scalable revocation scheme called Proxy-based Scalable Revocation for Constant Ciphertext Length (ProSRCC) scheme. It improves the Emura et al.'s [10] scheme with scalable revocation. It accomplishes revocation with the help of a trusted proxy server, which computes a proxy element to complete the decryption process. It modifies the proxy term only for a revoked unauthorized users. The ProSRCC scheme supports two types of revocation schemes attribute-based and user-based revocation.

Role of Proxy Server: In our scheme the proxy server assists in partial decryption by providing two proxy terms required to complete decryption process. The proxy server contains a list of revoked users, a list of revoked attributes and corresponding users from whom attributes have been revoked. This list is called the revocation list *RL*. The proxy server uses the list *RL* and the user's secret key to compute two components named as C_{user_i} and C_{attr_i}. It modifies the two components for revocation for a revoked user so that decryption fails. The non-revoked users can continue to access the ciphertext uninterruptedly without re-encryption or re-distribution of the keys.

The Key Authority (*KA*) handles all the attributes for a user. In case of attribute level revocation, the proxy server contacts *KA* to calculate K_{attr_i} value and uses it to compute C_{attr_i}. The proxy server does not need K_{attr_i} in case of user revocation or simple decryption for a non-revoked user. The proxy server calculates C_{user_i} and C_{attr_i} and uses it to complete the decryption process.

The setup(), keygen() and encrypt() phases are the same in all cases as similar to the Emura et al's [10] phases are discussed in the previous section.

The proxy and decrypt algorithms are different in all cases whether it is user-based revocation, attribute-based revocation or no revocation and are described in the following subsections.

A. CASE I: No Revocation

- **Proxy**
  Proxy(SK_L, RL): The proxy server computes the components C_{user_i} and C_{attr_i}.

$$C_{user_i} = (g^{\lambda}), \lambda \in RandomNumber$$
$$C_{attr_i} = h^{y}.(g^{\sum_{v_{i,j} \in L_u} t_{i,j}})^{r}.g^{\lambda}$$

  The proxy server forwards C_{user_i} and C_{attr_i} to the user for further decryption.
- **Decrypt Algorithm**
  Decrypt $(PK, C, SK_L, C_{user}, C_{attr})$: Decryption proceeds as follows:

$$= \frac{C1.e(C3, g^{r})}{e(C2, C_{attr_i}/C_{user_i})}$$
$$= \frac{M.e(g,h)^{sy}.e(g,g)^{s.r.\sum_{v_{i,j} \in W_c} t_{i,j}}}{e(g^{s}, h^{y}.(g^{(r.\sum_{v_{i,j} \in L_u} t_{i,j})+\lambda}).g^{-\lambda})}$$
$$= \frac{M.e(g,h)^{sy}.e(g,g)^{s.r.\sum_{v_{i,j} \in W_c} t_{i,j}}}{e(g,h)^{sy}.e(g,g)^{s.r.(\sum_{v_{i,j} \in L_u} t_{i,j})+s(\lambda-\lambda)}}$$
$$= M$$

  Thus, the decryption of a non-revoked user is done successfully.

B. CASE II: Attribute-based Revocation

- **Proxy**
  Proxy(SK_L, RL): If attributes have been revoked for a user i then the proxy server calculates C_{user_i} and C_{attr_i} as follows: The proxy server will call the Key Authority (*KA*) to calculate the value K_{attr_i} and send it back to the proxy server. $K_{attr_i} = (g^{-r.\sum_{v_{i,j} \in RL} t_{i,j}})$
  After receiving K_{attr_i} from KA, the proxy server calculates C_{user_i} and C_{attr_i}.

$$C_{user_i} = (g^{\lambda}), \lambda \in RandomNumber$$
$$C_{attr_i} = h^{y}.(g^{\sum_{v_{i,j} \in L_u} t_{i,j}})^{r}.K_{attr_i}.g^{\lambda}$$
$$= h^{y}.(g^{\sum_{v_{i,j} \in L_u} t_{i,j} - \sum_{v_{i,j} \in RL} t_{i,j}})^{r}.g^{\lambda}$$

  After calculating the components C_{user_i} and C_{attr_i}, the proxy server sends these values to the user for further decryption.
- **Decrypt Algorithm**
  Decrypt $(PK, C, SK_L, C_{user}, C_{attr})$: Decryption is performed as follows:

$$= \frac{C1.e(C3, g^{r})}{e(C2, C_{attr_i}/C_{user_i})}$$
$$= \frac{M.e(g,h)^{sy}.e(g,g)^{s.r.\sum_{v_{i,j} \in W_c} t_{i,j}}}{e(g^{s}, h^{y}.(g^{(r.\sum_{v_{i,j} \in L_u} t_{i,j} - \sum_{v_{i,j} \in RL} t_{i,j})+\lambda}).g^{-\lambda})}$$
$$= \frac{M.e(g,h)^{sy}.e(g,g)^{s.r.\sum_{v_{i,j} \in W_c} t_{i,j}}}{e(g,h)^{sy}.e(g,g)^{s.r.(\sum_{v_{i,j} \in L_u} t_{i,j} - \sum_{v_{i,j} \in RL} t_{i,j})}}$$
$$\neq M$$

  It is clear from the above expression that in the denominator part, all the revoked attributes cancel out and thus numerator is not nullified by the denominator. In this way, decryption of the ciphertext fails.

C. CASE III: User-based Revocation

- **Proxy**
  Proxy(SK_L, RL): The proxy server computes the components C_{user_i} and C_{attr_i}. Suppose any user i is revoked completely then the proxy server computes the values of C_{user_i} and C_{attr_i} as follows:

$$C_{user_i} = (g^{\lambda_1}), \lambda_1 \in RandomNumber$$
$$C_{attr_i} = h^{y}.(g^{\sum_{v_{i,j} \in L_u} t_{i,j}})^{r}.g^{\lambda_2}, \lambda_2 \in RandomNumber$$

  The proxy server passes C_{user_i} and C_{attr_i} to the user i to complete the decryption process.

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-661-3

TABLE II. SYSTEM SETUP

Hardware Requirements	1.Disk space of 2 GB or more 2.RAM of 2048 MB or more 3.Intel Dual Core Processor of 1.7 GHz or faster
Software Requirements	1.32/64-bit Windows XP/2008/7/8 2..PBC Library [14] 3.GMP Library [25] 4.CP-ABE Toolkit

- **Decrypt Algorithm**
  Decrypt $(PK, C, SK_L, C_{user}, C_{attr})$: The decryption proceeds as follows:

$$= \frac{C1.e(C3, g^r)}{e(C2, C_{attr_i}/C_{user_i})}$$
$$= \frac{M.e(g,h)^{sy}.e(g,g)^{s.r.\sum_{v_{i,j}\in W_c} t_{i,j}}}{e(g^s, h^y.(g^{(r.\sum_{v_{i,j}\in L_u} t_{i,j})+\lambda_1}).g^{-\lambda_2})}$$
$$= \frac{M.e(g,h)^{sy}.e(g,g)^{s.r.\sum_{v_{i,j}\in W_c} t_{i,j}}}{e(g,h)^{sy}.e(g,g)^{s.r.(\sum_{v_{i,j}\in L_u} t_{i,j})+s(\lambda_1-\lambda_2)}}$$
$$\neq M$$

  A revoked user cannot access the ciphertext since λ_1 and λ_2 do not cancel each other and this causes the decryption to fail.

V. IMPLEMENTATION

We have implemented the ProSRCC algorithm with the setup given in Table II.

We have first implemented the CP-ABE scheme with AND-gate access policy and revocation scheme using the CP-ABE toolkit. All pairing based operations have been implemented using the PBC library [14] and the GMP library [25]. The PBC library is the backbone of all pairing based crypto-systems. The PBC library uses the GMP library internally for performing on signed integers and floating-point numbers. We have implemented both the Emura et. al's scheme [10] and our proposed scheme ProSRCC using the PBC library. Secion VI discusses the evaluation of their performance.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

We compare our scheme with Jahid et al.'s [19] PIRATTE scheme and Li et al.'s [18] schemes. We compare our proposed revocation scheme with the other revocation schemes for the access policy used, the time taken in encryption and decryption, scalability and security features. The scheme given by Jahid et al. [19] is a proxy-based scheme. The scheme proposed by Li et al. [18] is a multi-authority scheme for cloud servers and enhances Emura et al.'s [10] CP-ABE scheme with scalable revocation.

A. Access policy

Access policy is the combination of attributes, which allows decryption of a document. Jahid et al.'s [19] scheme use the same tree access structure as used in the Bethencourt et al.'s [7] scheme. Table III shows the comparison of the access policies.

TABLE III. ACCESS POLICY

Scheme	*Access policy used*
Bethencourt et al. [7]	Tree-based Access Structure
Jahid et al. [19]	Tree-based Access Structure
ProSRCC	AND-gates on multi-valued attributes
Li et al. [18]	AND-gates on multi-valued attributes

B. Size of Each Entity

We compare the sizes for various entities such as PK, MK, SK, and ciphertext in terms of the elements of a bilinear group. Table IV illustrates the comparison between the different schemes. Here n is the number of attributes. The ciphertext for Jahid et al.'s [19] scheme depends on the total number of attributes present in the access policy, whereas in the case of the ProSRCC and Li et al.'s [18] schemes the size of the ciphertext is constant. If number of attributes = 9, then size of each value will be as given in Table V.

C. Computational Overhead

Computational overhead is shown in the form of group operation and pairing operation in Table VI. Jahid et al. [19] does more number of group operations and pairing computations as compared to ProSRCC and Li et al. [18] in encryption and decryption.

D. Running Time

We have implemented the schemes and measured the actual time taken by the encryption and decryption processes as given in Table VII. The ProSRCC scheme provides scalable revocation with a constant-sized ciphertext. The encryption times are much less as compared to Jahid et al.'s [19] scheme for the same number of attributes.

E. Comparison of Features Provided by Different Schemes

Different features of the attribute based encryption schemes like- revocation, scalability and size of ciphertext have been compared in Table VIII.

Our scheme is efficient from Jahid et al.'s [19] scheme in that the length of the ciphertext and the costs for decryption does not depend on the number of attributes. Especially, the number of pairing computations is constant. AND-gates on multi-valued attributes makes the access structure, a subset of the access structures presented in [9]. Our scheme is better than the scheme provided by Li et al. [18] because, in this scheme, the user secret key is updated each time attribute-based revocation occurs. However, in our scheme whenever any number of attributes are revoked from any user, it is added to the revoked list and is maintained and taken care by the proxy server.

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-661-3

TABLE IV. SIZE OF EACH ENTITY

Scheme	*PK*	*MK*	*SK*	*Cipher-text*
Jahid et al. [19]	3G1 + GT	Zp + G	(2n+1)G1	(2n+1)G1 + GT
ProS-RCC	(2n+1)G1 + GT	(n+1)Zp	2 G1	2G1 + GT
Li et al. [18]	(2n+1)G1 + GT	(n+1)Zp	(n+1)G1	2G1 + GT

TABLE V. SIZE OF EACH ENTITY WITH NUMBER OF ATTRIBUTES=9

Scheme	*PK*	*MK*	*SK*	*Ciphertext*
Jahid et al. [19]	3G1 + GT	Zp+G	19G1	19G1 + GT
ProSRCC	19G1 + GT	10Zp	2G1	2G1 + GT
Li et al. [18]	19G1 + GT	10Zp	10G1	2G1 + GT

F. Performance graph

The performance graphs in Figures 1, 2 and 3 illustrate the time required by the different schemes by Jahid et. al. [19], Emura et al. [10] CP-ABE and ProSRCC our proposed revocation scheme for key generation, encryption, and decryption respectively. Key-generation time and encryption time for Jahid et al. [19] and the original CP-ABE scheme [7] are almost same. Only decryption time differs from the original CP-ABE scheme. Hence, we compare the performances of Jahid et al.'s [19] PIRATTE scheme, Emura et al.'s [10] scheme without revocation and our revocation scheme ProSRCC.

It is clear from figure 1 that the time taken by Jahid et al. [19] scheme to generate the private key is high as compared to the Emura et al.'s [10] and our proposed schemes. Initially, our scheme is taking less time to generate the private keys as compared to the Emura et al.'s scheme [10]. However, after 6-7 attributes time taken to generate keys is increased. Figure 2 shows that Jahid et al.'s [19] scheme takes more time for encryption as compared to Emura et al.'s [10] and our scheme. In case of decryption initially, Jahid et al. [19] scheme is taking less time as compared to Emura et al. [10] and our scheme. However, after 3-4 attributes time is increasing almost linearly, and it is more as compared to Emura et al.'s [10] and our scheme.

G. Security Features of Our Scheme

Our scheme is secure against following attacks

1) *Collusion resistant:* For every user, their secret key is blinded by a secret number r, so two users can never collude to decrypt a ciphertext.
2) *Chosen ciphertext attack (CCA):* According to the selective security game for CP-ABE, as explained by Emura et al. [10], adversary sends the challenge access structure *W* to the challenger. As a result, the challenger replies with *PK*. Then adversary submits an attribute list *L* to the challenger, where $L \neq W$. The challenger gives the corresponding secret key. Adversary further submits an encrypted text *C*, for which access structure is *W*. The challenger replies with the decrypted plaintext *M*. After the completion of this phase, adversary now gives *M0* and *M1*, two equal length messages to the challenger. The challenger is free to choose either *M0* or *M1* and then runs the encryption algorithm on the chosen plaintext and gives it to the adversary. Now the adversary can submit multiple keygen queries to get the secret keys related to the various set of the attributes list. Each time, it generates the secret key with a different random number *r*, which blinds the key, so adversary will not be able to guess the secret key even in a brute-force manner. In case of revocation, the problem is still the same, so the adversary will not be able to guess or compute the secret key.
3) *Chosen plaintext attack (CPA):* It is CPA secure because it links each ciphertext with a different secret key *s*. The selective game for CPA security eliminates the decryption queries; rest is same as in the Chosen-ciphertext attack(CCA) secure selective game. The adversary submits the keygen queries and gives the plaintext to encrypt. It repeats the process several times. However, each time it encrypts the ciphertext with a different random number *s*; it blinds the new ciphertext *s*. Moreover, it also blinds each secret key *SK* by a new random number *r*, so the secret key can also not be guessed. As explained in Section IV that finding the value of x in g^x is a computationally hard problem.
4) *Forward secrecy:* It is secure because for each different ciphertext secret key is different, this means that compromise of one message cannot jeopardize others as well, and there is no one secret value for encryption whose acquisition would compromise multiple messages.

TABLE VI. COMPUTATIONAL OVERHEAD

Scheme	*Encryption time*	*Decryption time*
Jahid et al. [19]	(n+1)G1 + nG2 + GT	2GT + nG2
ProSRCC	(n+1)G1 + 2G2	2GT + 2G2
Li et al. [18]	3G1	3GT

TABLE VII. RUNNING TIME

Scheme	*Encryption time*	*Decryption time*
Jahid et al. [19]	0.36sec	0.08sec
ProSRCC	0.0605sec	0.042sec

VII. CASE STUDY: SELECTIVE FOOD TOKEN VENDING MACHINE

As traditional mobile phones have evolved into smart mobile phones, vending machines have also developed into smart vending machines, though at a much slower pace. Newer technologies, such as the Internet connectivity, different types

TABLE VIII. FEATURE COMPARISON

Scheme	*Revocation*	*Scalability*	*Constant Length Ciphertext*
CP-ABE [7]	✗	✗	✗
EMURA [10]	✗	✓	✓
PIRATTE [19]	✓	✗	✗
SPIRC [23]	✓	✓	✗
ProSRCC	✓	✓	✓

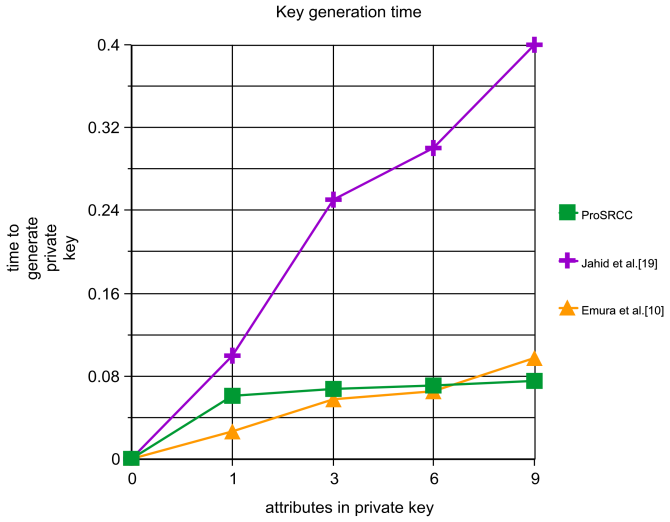

Figure. 1. Key generation time

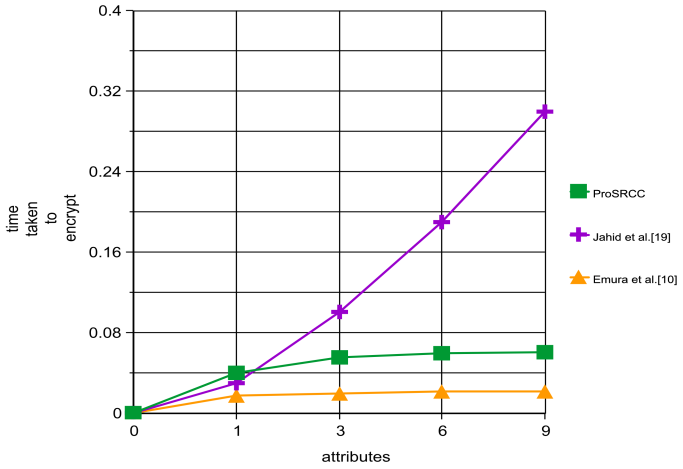

Figure. 2. Encryption time

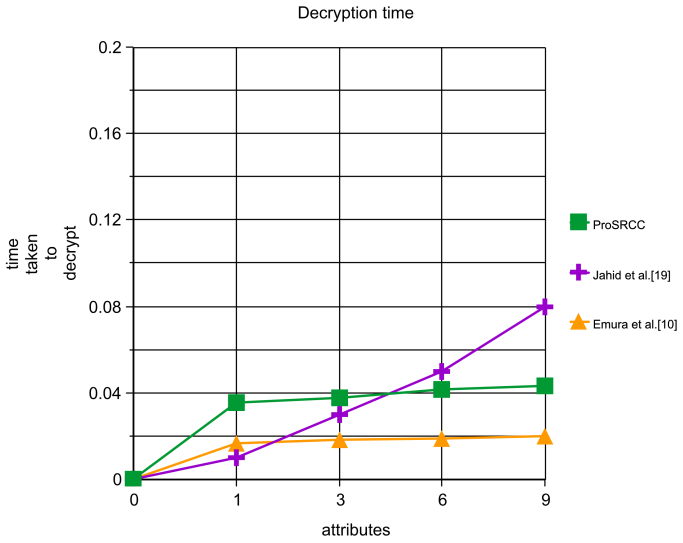

Figure. 3. Decryption time

of cameras and sensors, advanced payment systems, and a wide range of identification technology, such as Near-Field Communication (NFC) and Radio-frequency identification(RFID) [23] [24] have been an important part in this development. Such smart vending machines provide a more user-friendly experience and further reduce the operating costs thus improving the performance of the vending operations using remote manageability and intelligent back-end algorithms. These smart vending machines can be used easily as a selective access control systems.

Consider a token vending machine installed in a company's office as shown in Figure 4. A food token vending machine is such type of machine, which provides the tokens based on the level of an employee. The mode of payment can be coins or smart cards. It provides many types of tokens, e.g., T1, T2, and T3. However, it provides a different type of token for a different level of employee, and there can be a large number of tokens. The food vending machine accepts a smart card/ID card of an employee. Based on their work-level, each employee's card has different attributes, which make their secret key. Once an employee inserts his card to the token vending machine, it reads the secret key. Based on their secret key, a certain menu is shown on the screen. The menu can be reading by partial decryption process on the machine and the proxy server. The proxy then checks its revocation list and computes C_{user_i} and C_{attr_i} elements and passes to the in-built decryption process. Then the decryption process finds the value of *M* (here type of *M* is the type of food token, e.g., *T1*, *T2*, and *T3*). If M matches to any token type value *M*, it provides that type of token. Otherwise, the machine prints an appropriate message on the screen. The vending machine is shared by number of people and provides beverages on a selective basis. The ProSRCC scheme is suitable to encrypt the menu. It is based on Emura et. al's CP-ABE scheme [10] and hence the ciphertext will be constant in size so that minimal storage is required on the vending machine. Also, the ProSRCC scheme provides scalable revocation of users so that the vending machine can be used uninterrupted by other valid users.

The values of *T1, T2*, and *T3* are pre-calculated as an encrypted ciphertext. The proxy can communicate with the server having information about the employee and their work-level. Suppose an employee leaves the company, another employee

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-661-3

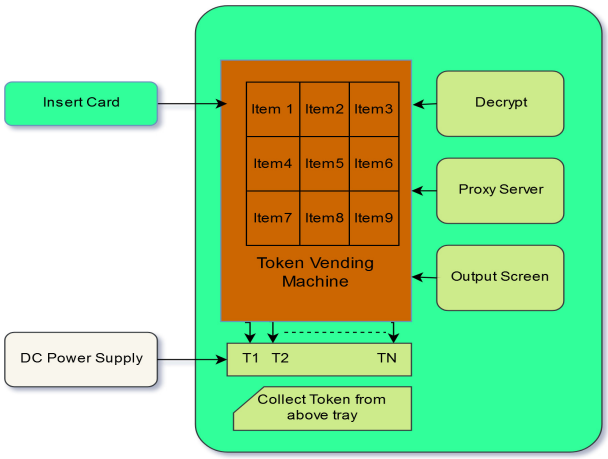

Figure. 4. Selective Food Token Vending Machine

wants to use his card. In this case, the proxy server will deny access because when an employee leaves the company, his id is then added to the revoked user list, resulting in the denial of service. The other non-revoked users can access the vending machine uninterruptedly without any requirement of re-encryption of re-distribution of keys.

Whenever the food items are updated, the food token vending machine also updates itself. Each time an employee is promoted or demoted from his work-level, an update is made in the revocation list by the proxy server. The changes made by the proxy server are reflected while providing the food token to an promoted/demoted employee of the company.

VIII. Conclusion

Revocation mechanism is an important feature of any encryption system to administer the malicious behavior of its users and to provide the selective access to its users based on their attributes. For such a system to work in a resource-constrained device, our scheme ProSRCC provides scalable revocation feature with constant ciphertext length. It is an improvement over Emura et al.'s [10] scheme as their scheme does not provide revocation feature. Li et al. [18] propose a revocation scheme for Emura et al.'s [10] scheme. However, it lacks scalable revocation. The ProSRCC scheme is secure as compared to the other schemes. Our scheme is secure against CPA and CCA attacks, and it is also collusion resistant. It is scalable as compared to Jahid et al.'s [19] PIRATTE scheme because the number of attributes revoked in our scheme is not limited. In the the PIRATTE scheme it is limited to t users(t represents the polynomial's degree used in the scheme). It provides the revocation feature, but ciphertext length is not constant. Thus, ProSRCC can provide selective access from a stationary device used for sharing selective data to multiple users by supporting optimized ciphertext length and scalable revocation feature.

References

[1] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," SIAM J. Comput., vol. 32, no. 3, pp. 586-615, 2003.

[2] X. Boyen and B. Waters. "Anonymous hierarchical identity-based encryption (without random oracles),"CRYPTO, pages 290-307, 2006.

[3] R. Canetti, S. Halevi, and J. Katz. "Chosen-ciphertext security from identity-based encryption,"EUROCRYPT, pp. 207-222, 2004.

[4] D. Boneh and B. Waters. "Conjunctive, subset, and range queries on encrypted data,"TCC, pp. 535-554, 2007.

[5] J. Katz, A. Sahai, and B. Waters. "Predicate encryption supporting disjunctions, polynomial equations, and inner products,"EUROCRYPT, pp. 146-162, 2008.

[6] C. Lee, P. Chung, and M. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments,"International Journal of Network Security, vol-15, no. 4, pp. 231-240, 2013.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption,"IEEE Symposium on Security and Privacy, pp. 321-334,2007.

[8] L. Cheung and C. Newport, "Provably secure ciphertext-policy ABE,"ACM Conference on Computer and Communications Security, pp. 456-465,2007.

[9] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures,"Springer Applied Cryptography and Network Security, pp. 111-129, 2008.

[10] K. Emura, A. Miyaji, K. Omote, A. Nomura, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length,"International Journal of Applied Cryptography, vol. 2, no. 1, pp. 46-59, 2010.

[11] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute-based encryption,"Springer Automata, Languages, and Programming Lecture Notes in Computer Science, vol. 5126, pp. 579-591, 2008.

[12] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute-based encryption,"ACM conference of International Symposium on ACM Symposium on Information, Computer and Communications Security, pp. 343-352, 2009.

[13] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes,"Springer Information Security Practice and Experience, pp. 1-12, 2009.

[14] PBC Library: https://crypto.stanford.edu/pbc/, Last accessed July 29,2018

[15] CP-ABE Toolkit- http://acsc.cs.utexas.edu/cpabe/,Last accessed July 29,2018

[16] L. Pang, J. Yang, and Z. Jiang, "A Survey of Research Progress and Development Tendency of Attribute-Based Encryption," The Scientific World Journal, vol. 2014, 13 pages, 2014.

[17] Software used to create graph https://nces.ed.gov/nceskids/createagraph/default.aspx?ID=6499c61b86e443359920ad4fa9c65166 Last accessed July 29,2018

[18] X. Li, S. Tang, L. Xu, H. Wang, and J. Chen, "Two-Factor Data Access Control With Efficient Revocation for Multi-Authority Cloud Storage Systems,"Journal of IEEE Access, pp. 1-1. 10, 2016.

[19] S. Jahid and N. Borisov. "Piratte: Proxy-based immediate revocation of attribute-based encryption." arXiv preprint arXiv:1208.4877, 2012.

[20] Y. Imine, A. Lounis, and A. Bouabdallah, "Immediate attribute revocation in decentralized attribute-based access control,"IEEE conference of Trustcom/BigDataSE/ICESS, pp. 33-40. IEEE, 2017.

[21] N. Attrapadung and H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes,"Springer International Conference on Cryptography and Coding, pp. 278-300, 2009.

[22] R. Zhang, L. Hui, S. Yiu, X. Yu, Z. Liu, Z. L. Jiang, "A Traceable Outsourcing CP-ABE Scheme with Attribute Revocation,"IEEE conference Trustcom/BigDataSE/ICESS, pp. 363-370, 2017

[23] D. Sethia, H. Saran, and D. Gupta, "CP-ABE for Selective Access with Scalable Revocation: A case study for Mobile-based Healthfolder,"International Journal of Network Security, Vol.20, No.4, pp.689-771, 2018.

[24] V. Coskun, B. Ozdenizci, and K. Ok, "A Survey on Near Field Communication (NFC) Technology," Springer Wireless Personal Communications, Vol.71, No.0, pp.2259-2294, 2013.

[25] The GNU Multiple Precision Arithmetic Library: https://gmplib.org/, Last accessed July 29,2018

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-661-3