Revocable ABE with Bounded Ciphertext in Cloud Computing

Mohamed Ali Hamza[1,2], Jianfei Sun[1], Xuyun Nie[1], Zhiquan Qin[1], and Hu Xiong[1,3]
(Corresponding author: Mohamed Ali Hamza)

School of Software Engineering, University of Electronic Science and Technology of China[1]
4 Jianshe North Rd 2nd Section, Chenghua Qu, Chengdu Shi, Sichuan Sheng 610051, China
(Email: mody231279@yahoo.com)
Department of Electronic Engineering, Karary University, Omdurman, Sudan[2]
State Key Laboratory of Cryptology, P. O. Box 5159, Beijing, 100878, China[3]

(Received June 24, 2016; revised and accepted Sept. 3 & Oct. 25, 2016)

Abstract

Revocable Attribute-Based Encryption (R-ABE) has received much concern recently due to its characteristic of capability on encrypting the Data, according to some attributes, whereas users can decrypt the ciphertexts if they own the credential of those attributes with ability to revoke the expired users. We propose a new practical Revocable Attribute Based Encryption which has a short ciphertext O(1) and private keys O(1) with efficient running time. In this scheme the users can effectively be revoked and added with backward and forward secrecy in the indirect mode, which can controlled by Key Authority Party without reseting the system parameter's or updating and redistributing the attributes private keys which has expense. Assuming the cloud provider is semi-honest and has been delegated by KA in order to apply dynamic processing on the data and controlling users. This scheme is secured against Chosen Plaintext Adversary (CPA), assuming the (Decision) Bilinear Diffie-Hellman Exponent assumption (n-BDHE) is being held.

Keywords: Access Control; Attributes Based Encryption; Bounded CipherText; Key Policy; Revocation; Revocable Storage Attribute-based Encryption

1 Introduction

Outsourcing is a movement has been influencing the global revolution of the information technology which gives effective solutions for data managing of the organizations, such as installations, data analysis, networks and data protection. It offers wonderful benefits such as better operating, reducing employment cost, delegating responsibilities to external agencies, as well as mitigating risk and resource's scalability.

Moreover, delegating responsibilities to other party as Cloud Service Party (CSP), there are data owner who still worrying about privacy preserving of their data and how they controlling the accessibility, in order to guarantee secure offshoring.

ABE is one of popular accessing control techniques and has been appeared firstly with Sahai and Waters [2] where they aim to encrypt Ciphertexts one-to-many. However, the users can decrypt if they have certain requirements, although ABE algorithms suffer from two significant drawbacks. For instance, growing of the Ciphertext impractically, and the revoking mechanism of expired or dishonest user.

In fact there are two types of revocations, direct and indirect models [13]. the first scenario revocation is enforced directly by the sender who determines the revoked list during encryption stage, whilst indirect revocation are controlled by the key authority KA which issues an updated key, such that only non-revoked users can update their keys.

We present a novel way of an Indirect R-ABE technique with bounded Ciphertext that overcomes the revocation challenges, such as revoking users without resetting credentials of others users and preventing revoked users from accessing the data or collude with dishonest users.

The challenging areas which have been handled in this work are dynamic controlling of the users and shortening ciphertext, the scheme relays on broadcast encryption technique that proposed in [5] which has collusion resistant and short ciphertext features.

1.1 Related Works

Many Revocable-ABE [1, 4, 9, 17, 18] were introduced recently. However, most of them suffered from the growth of ciphertext's size proportionally, with number of users and attributes. Updating periodically the attributes private keys which is unaccepted for practicable applications, particularly when users have limited resources.

Revocable Storage is a challenging task where the third party can modify the existing ciphertext to block the repealed users from accessing the stored data in outsourcing storage without intermediating of the data owner while the other user can keep accessing it. Sahai and Waters innovated et al. [3] the first revocable storage when they used only publicly available information with periodically updating the ciphertexts and private keys. However, the size of ciphertext increased linearly with number of associated attributes, also needs to re-distribute periodically the private keys for all non-revoked users.

Nuttapong and Hideki proposed a Conjunctive Broadcast and Attributed Based Encryption, where the private key Conjuncted with a user index and the ciphertext associated also with a user index set S, the decryption can achieve if the condition on attributes of the ABE hold and, in addition, $ID \in S$ and KeyGen used ID with Linear Secret Sharing Schemes, but the size of CT and private keys were large [14].

Junbeom and Dong proposed et al. [11] designed revocable CP-ABE Schemes with periodic or timed revocation with the help of the semi-trusted proxy deployed in the Cloud Services Provider (CSP). The main drawback of these schemes is relayed on other part for re-encryption.

David and Thomas presented et al. [8] a broadcast encryption scheme, with attribute-based mechanisms that lets the Data Owner to add/revoke groups of users were defined by their attributes, also the size of private keys is grown with the number of attributes that are related to the user and size of Ciphertext is also increased linearly with the number of attributes used in the access policy whereas the public key is somewhat large.

1.2 Our Result

This section gives a comparison between state-of-art schemes and our novel approach which realizes shrinking of the private keys's and ciphertext size without influences with number of users or associated attributes, also the performance is enhanced by applying precomputed algorithms and cached the computation in secure memory. Table 1 shows the comparison.

We denote for the parameters of table1 as follows: **U** is universe attributes or all possible attributes in the scheme, **S** is set of attributes that have assigned to the user, **Y** is set of possible attributes have associated to CT, **r** is number of revoked user, **Nmax** the number of leaf nodes in **I** where The total number of all nodes in the circuit is $2Nmax?1$, **SD** is Subset Difference Method, (**RSABE**) Revocable Storage Attribute-based Encryption, **DO**, **nx** is the number of rows that selected by map function p(x) for all x in Y, **aMSE-DDH** augmented multi-sequence of exponents decisional Diffie Hellman problem, **IBBE** identity-based broadcast encryption, M number of total users in the schemes, **COBG** Composite-Order Bilinear Groups.

1.3 Contributions

We proposed a concrete R-ABE with following achievements (I) Short ciphertext and independent from number of users or attributes. (II) The key authority KA has ability to revoke or add users efficiently (III) The revocation processes did not need reseting user credentials or redistributing of the private keys or the public key (only the updated key) (IV) There a proxy re-encryption to prevent the existing data, however, each ciphertext will re-encrypt once before storing it in semi-trust third party TTP (V) The scheme prevents a repealed user from accessing the old ciphertext by modify tiny part of ciphertext (about **25%** of the original ciphertext).

1.4 Organization

Section 2 will present preliminaries and definitions of some security notions, Section 3 describes the scheme's constructions and correctness of the scheme, Section 4 introduces the security game and proving of the system's security, Section 5 presents some enhancement technique and applying precomputed algorithm to improve the cost of system's computations, Section 6 presents implementation and result, and in Section 7 concludes and shows the open problems and future work.

2 Preliminaries and Definitions

This section shows the preliminaries and definitions of some security tools which will use to construct the scheme.

2.1 Bilinear Mapping

We review some facts associated to bilinear map cycle groups which are efficient and computable, introduced by Boneh and Frankin [7], both groups have the same prime order group p, the map function must satisfy the following properties:

Computability: There exist polynomial time algorithm when given $g_1, g_2 \in G$ that can compute $e(g_1, g_2) \in G_T$.

Bilinear: For any $a, b \in Z_p$ the bilinear function is such that $e(g_1^a, g_2^b) = e(g_1, g_2)^{\mathrm{a.b}} \in G_T$ are Non-Degenerate where g is generator of G and $e(g, g)$ generator of G_T where $e(g, g) \neq 1$.

Access structure: Suppose $\{P_1, P_2, \ldots, P_n\}$ is a set of attributes, we say a selection of attribution $S \in 2^{\{P_1, P_2, \ldots, P_n\}}$ is monotone if $\forall B, C$: $B \in A$ and $B \subseteq C$ then $C \in S$, a monotone access structure is a group collection of non-empty subsets $S \in 2^{\{P_1, P_2, \ldots, P_n\}} \backslash \{\emptyset\}$, the authorized sets is in S or qualified set, and the sets are not in S called the unauthorized sets. We emphasize on restriction that using monotone access structures in our system.

Table 1: Comparison between other ABE schemes

Scheme	**[3]**	**[20]**	[16]	**[10]**	[15]	Our
PK	$O(\|U\|)$	6	112	$O(2\|U\|)$	$O(2\|U\|)$	$O(2\|M\|)$
Pr	$O(2\|S\|)$	$2\|S\|+2$	$5+16L+16[$ $log2Nmax]$ $+logNmax]$	$O(\|S\|)$	$O(2\|U\|)$ $+O(\|S\|)$	$O(\|S\|)$
CT	$O(\|Y\|)$	$2\|Y\|+2$	$O(\|RL\|)$	$O(3.nx*L)$	$O(1)$	$O(1)$
Updated	$(Pr+CT)$	$(PK+Pr$ $+MSK)$	$(St+RL)$	Pr	Pr	CT
Security Assumption	COBG	(DBDH)	(DLIN)	COBG	aMSE-DDH	Decisional qBDHE
Security Game	Oracles CPA	Selective CPA	Full CPA	Selective Oracle-CPA	Selective-CPA Non-interactive	Selective CPA
Access Structure	LSSS $L\times n$	LSSS $L\times n$	(SD) LSSS	LSSS $L\times n$	LSSS $L\times n+IBBE$	Fine Grained
Policy	KP-ABE	CP-ABE	KP-ABE	CP-ABE	CP-ABE KP-ABE	KP-ABE
Revocation Delegation	KA to TTP	KA to TTP	KA to DO	KA to AAs	KA only	KA to TTP
Revocation Methodology	Periodically	On Demands	On Demands	Periodically	On Demands	On Demands
Supporting RSABE	Yes	NO	NO	NO	NO	Yes

Access Circuit: Let C be a circuit represents accessing control of attributes holders, which contains mainly from (AND-Gate, OR-Gate) nodes, we denote to $\{att_i\}_{i\in k}$ as the set of attributes which are given to the user k, $Nmax$ is total number of attributes which input to the circuit (leaf nodes), d is a depth of the circuit and equals generally to the number of circuit's layers, $node_x$ is an indexed node which starts from initial node (the root) $node_1$ or the output of circuit down to the last node $node_l$ notice that $l \leq 2Nmax - 1$ is total number of nodes in the circuit, a non-leaf nodes are the attribute nodes nod_x where $(l - Nmax) \leq x \geq l$, an input to the $node_x$ are $input(node_x) = (A, B)$ where A and B are the direct inputs to the node, an output of $node_x$ is denoted by $output(node_x)$, namely if $\{att_i\}_{i\in k}$ is set of attributes which assigned to user k so we say $C(\{att_i\}_{i\in k})$ =true obviously if $\{att_i\}_{i\in k}$ satisfied the access circuit C, also any $node_x$ is satisfied if its output is true $C(\{att_i\}_{i\in k}) = true|\forall input(node_x) \subseteq \{att_i\}_{i\in k}$.

3 Revocable ABE with Bounded Ciphertext Scheme

The proposed system is contained of six probabilistic algorithms which are setup, keygen, encrypt, re-encrypt, decrypt and updatekey as described in next paragraph assuming that there exist semi Trusted Third Party TTP who is responsible of re-encrypts the data and proceed the revocability tasks which are delegated from Key Authority KA who is accountable for keys managing, figure 1 shows the interactions between parties.

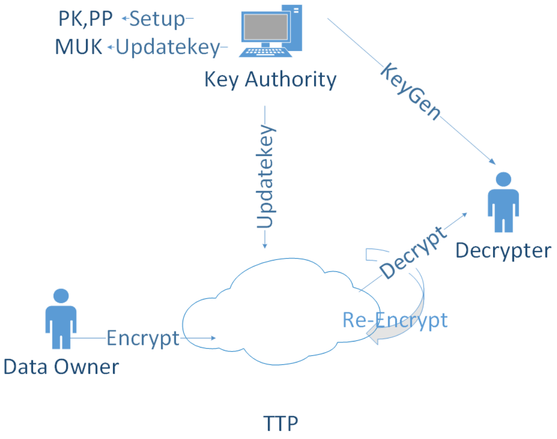


Figure 1: System model

3.1 Scheme Definition

Setup(n, λ)**:** This algorithm runs by KA after inputs the number of total users n with the security parameter λ and publishes out the public key PK, public parameters PP and keeps master secret key MSK secret.

KeyGen $(k, \{Attr\}_{\forall i\in S_k}, MSK)$**:** KA takes the user index $k \in [n]$, set of user's attributes $\{Attr_i\}_{\forall i \in S_k}$ and the master secret key MSK and outputs the private key for each attribute $\{pr_{k,i}\}$.

UpdateKey$(k, \{Attr_i\}_{\forall i \in S_k}, MSK)$**:** The KA uses k the current user index, $\{Attr_i\}_{i \in S_k}$ user's attributes that were assigned to user k and MSK master secret key, this algorithm will output refreshed master updated key MUK and submits the delegation keyDK that can depute TTP to handle either adding or revoking users, an UpDateKey algorithm might run if one of the four actions happened:

1) Adds new user with new attributes and submits the new attributes private key$\{pr_{k',i}\}$ and outs a new index k'.
2) Adds new attributes for existing user k.
3) Revokes existing user k permanently from the scheme.
4) Revokes some Attributes?$\{pr_{k,i}\}$ from user k.

Encrypt(M, PP, PK, MUK)**:** This a probabilistic algorithm works in very straightforward ways by taken the message M, public parameter PP, public key PK and master updated key MUK, the algorithm outputs succinct ciphertext CT_0, we emphasis on the size of ciphertext is not impacted neither number of legitimate user nor valid attributes, size of CT_0 precisely $O(CT_0) = 1$ which offered efficient property.

Re-encrypt(CT_0, PP, PK, MUK)**:** The TTP is allowed to modify the existing ciphertext either for preventing expired users from accessing it or allowing new user to permit accessing these encrypted data, this algorithm runs after TTP received the CT_0 directly or on demand of KA after receiving delegation keys, and outs the CT.

Decrypt$(CT, k, \{pr_{k,i}\}, PP, MUK)$: The decrypter uses this algorithm and inputs ciphertext CT, user index k, attributes private keys $\{pr_{k,i}\}$, public parameter PP and master updated key MUK, then it decrypts out the message M.

Correctness: Required that the system to be correct, specifically as follows:

$$\begin{aligned} \Pr[Dec(CT, S, k, \{pr_{k,i}\}, PP, MUK) = M | \forall S, k, \\ (PP, PK, MSK) \leftarrow Setup(n, \lambda), \\ \{pr_{k,i}\} \leftarrow KeyGen(k, \{Attr_i\}, MSK) \\ MUK \leftarrow UpDatekey(k, \{Attr_i\}, MSK) \\ CT_0 \leftarrow Encrypt\,(M, PP, PK, MUK) \\ CT \leftarrow ReEncrypt\,(CT_0, PP, PK, MUK)\,, \\ \forall i \in S_k] = 1. \end{aligned} \tag{1}$$

3.2 Security Game

Revocable ABE with Bounded Ciphertext is secure against selective chosen plaintext adversary CPA, where the security game is made up between an adversary A and a challenger B as follows.

Setup: B selects at the beginning the authorized set S_0, also selects the revoked set S_r such that $S_r \subset S$, then runs setup and UpDatekey algorithms and submits public key PK, public parameter PP and master updated key MUK to A whereas keeps master secret key MSK hidden from A, the adversary selects set of users and submits them to B as challenged set S.

Phase 1: Adversary A is asking adaptively the challenger B queries about attribute private keys for number of users $S_q = \{q_1, q_2, ..., q_r\}$ with one of the two restrictions:

1) **Case 1:** In this case the adversary A chose an user $k \in S_q$, that must not belong to authorized sets $k \notin S_0$ and each attributes $\{att_i\}$ of user k satisfied the access circuit C commonly $C(\{att_i\}_{i \in k}) = true$.
2) **Case 2:** A asks for the user $k \in S_q$ belonged to authorized sets $k \in S_0$ and he/she has been revoked $k \in S_r$. Also user k satisfied the access circuit $C(\{att_i\}_{i \in k}) = true$.

Then challenger obtains attributes private keys by running KeyGen algorithm and responds to adversary A with attributes private keys.

Challenge: After adversary A satisfied from asking queries then will pick up two random messages m_0, m_1 where $|m_0| = |m_1|$ and submits two message to challenger who will toe coin $b \in \{0, 1\}$ and applies encryption algorithm on $CT_b = Encrypt(m_b, PK, PP, MUK)$ and sends CT_b to adversary as challenge.

Phase 2: The adversary A is continuing adaptively queries the challenger β in similar way of Phase1 by sending request for other attributes private keys $S_2 = \{q_{r+1}, q_{r+2}, \ldots, q_m\}$ and we recall same phase1's restrictions.

Guess: Eventually adversary A outs the guessing of b' and wins iff $b = b'$.

3.3 Security Assumption

Our system's security is based on the complexity of (Decisional) Bilinear Diffie-Hellman Exponent Assumption (n–$BDHE$) [6, 19] relays on choosing a symmetric pairing $e\colon G \times G \Rightarrow G_T$ where G is a bilinear multiplicative group of prime order P, G_T is target group of prime order P. The (decisional) $n - BDHE$ problem described when given to an algorithm B this tuples

$(h, g, g_1, g_2, \ldots, g_n, g_{(n+2)}, \ldots, g_{2n}) \in G^{(2n+1)}$ then the algorithm B can output $b \in \{0,1\}$ with advantage ζ, in breaking decisional $n\text{–}BDHE$ in G_T if

$$|\Pr[B(h, g, g_1, \ldots, g_n, g_{(n+2)}, \ldots, g_{2n}, e(g_{n+1}, h)) = 0] \\ -Pr[B(h, g, g_1, \cdots, g_n, g_{(n+2)}, \ldots, g_{2n}, T) = 0]| \geq \zeta.$$

With probability over the random choice of generator $g, h \in G$, $\alpha \in Z_p$, $T \in G_T$ and the random bits used by B, the left part of above equation is valid distribution and is denoted $V\text{–}BDHE$ and the right part invalid random distribution and denoted $R\text{–}BDHE$.

4 Construction

We describe in this section the constructing of revocable ABE, as far we assume there exist Key Authority (**KA**) that in charges for creating users attributes private keys and revokes or adds users, Semi Trusted Third Party (**TTP**) that will re-encrypt the ciphertext and applies revocation or addition of users, Data Owner (DO) and decrypter, all of the above parties are participating as follows.

Setup(n, λ)**:** Setup algorithm is running by KA to generate the public parameters PP, public key PK and master secret key MSK, n is the input for this algorithm which is a number of expected users and λ is security parameter, the algorithm chooses $g \Leftarrow G$ uniformly as generator of source group G and $\alpha, \gamma \Leftarrow Z_p$, we denote $g_k = g^{(\alpha^k)}$ the public parameter is PP to compute public PK, the KA picks random $\beta \Leftarrow Z_p$ and computes the following tuple:

$$\begin{aligned} PP &= (g, g_1, g_2, \ldots, g_n, g_{(n+2)}, \ldots, g_{2n}, v = g^{\gamma}) \\ &\qquad \in G^{(2n+1)} \\ PK &= \left(g' = g^{\beta}, w = e(g_n, g_1)^{\beta}, S_0)\right), \\ MSK &= (\alpha, \gamma, \beta) \end{aligned} \quad (2)$$

where S_0, S are initial authorized and current authorized set respectively, then KA publishes PP and PK while Master Secret key MSK are kept secret.

KeyGen $(k, \{att_i\}_{(i \in S_k)}, MSK)$**:** For each user k the TTP computes $d_k = g^{(\alpha^{k^{\gamma}})} = v^{(\alpha^k)}$ and sets $Y = d_k$ as final output of the circuit and assumes $Y = d_k = g^{(\alpha^{k^{\gamma}})} = g^{(\alpha^y)}$ where $y = k^{\gamma}$ is the final output of the root gate. Now to compute the attributes keys of user k the KeyGen algorithm is inspired from fine-grained structure so if the next gate is OR-Gate it just pass same value to the two next fans and if the next gate is AND-Gate it chooses random $r_{(l,A)} \in Z_p$ uniformly for A's input and sets $r_{(l,B)} = y - r_{(l,A)} \in Z_p$ where l is root gate index, for B's input again it works same as above if the gate is OR-Gate $r_{(l-1,B)} = r_{(l-1,A)} = r_l$ and if the gate is AND-Gate then chooses random $r_{(l-1,A)} \in Z_p$ and sets $r_{(l-1,B)} = r_l - r_{(l-1,A)} \in Z_p$ and it continues in the same way until reaches the inputs of the circuit (leaves) which are the attributes of this circuit $\{r_{(i,j)}\}_{i \in [m], j \in \{A,B\}}$. KeyGen algorithm computes private key of each attribute as $\{g^{(\alpha^{r(i,j)})}\}_{i \in [m], j \in \{A,B\}}$ and sends attributes private keys $S_k \subseteq S$ of user k and sends to k via secured channel the values $\{g^{(\alpha^{r(i,j)})}\}_{i \in ([m] \cap S_k), j \in \{A,B\}}$. We denote the attribute private of user k as $pr_{(k,i)} = g^{(\alpha^{r(i,j)})}$.

UpDatekey $(k, \{att_i\}_{(i \in S_k)}, PK, MSK)$**:** This algorithm runs by Key Authority when decides to revoke certain user u or adding new user k' or after setup algorithm the output of this is master updated key (MUK) and delegated key DK which will be submit to TTP as follows.

$$MUK = \left(S, v' = (v. \prod_{(\forall j \in S)} g_{(n+1-j)})^{\beta}\right) \quad (3)$$

$$DK = (\{g^{-\beta}_{n+1-u_i}\}_{i \in S_r}, \{g^{\beta}_{n+1-k'_i}\}_{i \in S_a}) \quad (4)$$

where S_r, S_a are set of revoked and added users list respectively, then DK will send to TTP as delegated key to run Re-encrypt algorithm and updates the existing ciphertext for modification and publishes MUK. Our scheme is flexible for efficient key management processing in the following way:

1) Removing all user's attributes (revoke an user u), in this case the key authority refreshes MUK and updates v' in particularly

$$v' \leftarrow \left(\frac{v'}{g_{n+1-u}}\right)^{\beta} \quad (5)$$

simultaneously in other side the TTP will update (small part only C_1) the existing ciphertext as follows:

$$C_1 \Leftarrow \left(C_1 \cdot (DK_u)^{t'}\right) = \left(C_1 \cdot (g^{-\beta}_{n+1-u})^{t'}\right) \\ = \left(\frac{C_1}{\left(g^{\beta}_{n+1-u}\right)^{t'}}\right)$$

Here $DK_u \in Dk$ is assigned to user u. TTP will update some part of the ciphertext for existing data, however for future encryption the TTP is preventing from adding revoked user with two safe guards MUK and DK, also re-encryption algorithm guaranties forward and backward secrecy.

2) Removing some part of user's attributes in this case the key authority first applies the above step (remove the user u) then run KeyGen algorithm again to create keys as a new user with new index k' and this index must be unique

$k' \notin S$ and for allowing the new user k' to decrypt previous ciphertext update some part of ciphertext as follows:

$$\begin{aligned} C_1 \Leftarrow \left(C_1 \cdot (DK)^{t'}\right) &= C_1 \cdot \left(g^{\beta}_{n+1-k'} \cdot g^{-\beta}_{n+1-u}\right)^{t'} \\ &= C_1 \cdot \left(\frac{g^{\beta}_{n+1-k'}}{g^{\beta}_{n+1-u}}\right)^{t'} \end{aligned}$$

3) For adding a new user the key authority runs KeyGen algorithm for obtaining attributes private keys and submits the keys through secure channel.

Encrypt(M, PK, MUK)**:** Data owner intends to encrypt the data before outsourced in the TTP environment, starts by chooses random $t \in Z_p$ uniformly, and inputs the plaintext $M \in G_T$ then computes

$$CT = (C_0, C_1, C_2) = \left(g'^{t}, (v')^{t}, M.(w)^{t}\right) \quad (6)$$

Re-encrypt(CT, PK, PP, MUK)**:** This algorithm runs by TTP after receiving the ciphertext this step can assist to revoke the expired user u or adds new user k' to the scheme where u, k' are the index of revoke and added user respectively. Then the TTP will select randomly $t' \in Z_p$ and recomputes the ciphertext:

$$\begin{aligned} C_1 &\leftarrow \left(C_1.v'^{t'}\right) = v'^{t}.v'^{t'} = v'^{(t+t')} \quad (7) \\ C_2 &\leftarrow \left(C_2.w^{t'}\right) \end{aligned}$$

Note that there are difference between the authorized current set S which were chosen to compute v' and the authorized initial set S_0, so S is used to re-encryption whereas S_0 is used for encryption, is obviously to notice that the exponent of ciphertext t is shifted to $t + t'$ with this algorithm as follows.

$$C_0 = g'^{t}, C_3 = g'^{t'}. \quad (8)$$

From above we realize no changing in C_0 and TTP added new part the ciphertext C_3

$$\begin{aligned} C_1 &= \left(v. \prod_{(\forall j \in S_0)} g_{(n+1-j)}\right)^{\beta.t} \\ &\quad . \left(v. \prod_{(\forall j \in S)} g_{(n+1-j)}\right)^{\beta.t'} \\ C_2 &= \left(C_2.w^{t'}\right) = \left(M.w^{t}.w^{t'}\right) = \left(M.w^{t+t'}\right). \end{aligned}$$

We emphasize this algorithm is run once so will not effect the performance of scheme.

Decrypt$(CT, k, \{pr_{k,i}\}, PP, MUK)$**:** If the decrypter k has enough attribute's private keys that can fulfill the circuit's requirement, then user k is capable to compose d_k and decrypts the ciphertext $CT = (C_0, C_1, C_2)$ according to the ABE circuit, the decrypter starts in reverse way beginning from the circuit's input (leaves) until final gate(root), at first k inputs the attributes private keys $pr_{(k,i)} = g^{(\alpha^{r(i,j)})}$ then if the gate is OR-Gate then chooses any one of A or B as input of the gates (leaves)

$$\{y_{i+1} = g^{(\alpha^{r(i,A)})} or = g^{(\alpha^{r(i,B)})}\}_{i \in ([m] \cap S_k)}$$

where S_k is set of attributes belong to user k. Or the gate might be AND-Gate, then multiply the two inputs as

$$\begin{aligned} \{y_{i+1} &= g^{(\alpha^{r(i,A)})} \times g^{(\alpha^{r(i,B)})} \\ &= g^{(\alpha^{r(i,A)}) + (\alpha^{r(i,B)})}\}_{i \in ([m] \cap S_k)} \end{aligned}$$

for remaining gates acts in same way namely for OR-Gate:

$$\begin{aligned} \{y_{i+1} &= y_{(i,A)} = g^{(\alpha^{r(i,A)})} \text{ or} \\ &= y_{(i,B)} = g^{(\alpha^{r(i,B)})}\}_{\forall i \in \{(m+1),\ldots,(l-1)\}} \end{aligned}$$

and for AND-Gate the decrypter follows the circuit rules and computes:

$$\begin{aligned} \{y_{i+1} &= y_{(i,A)} \times y_{(i,B)} \\ &= g^{(\alpha^{r(i,A)})} \times g^{(\alpha^{r(i,B)})} \\ &= g^{(\alpha^{r(i,A)+r(i,B)})}\}_{\forall i \in \{(m+1),\ldots,(l-1)\}} \end{aligned}$$

until reaches final gate $Y = g^{\alpha^{y}} = g^{(\alpha^{(k^{\gamma})})} = d_k$. Hence decrypter gets d_k.

This above steps run once and not in each decryption processing and decrypter will store d_k in secure place, then to decrypt CT which has been re-encrypted with TTP:

$$\begin{aligned} T &= \frac{e(g_k, C_1)}{e(d_k \cdot \prod_{j \in S_0, j \neq k} g_{n+1-j+k}, C_0)} \\ &\quad \times \frac{1}{e(d_k \cdot \prod_{j \in S, j \neq k} g_{n+1-j+k}, C_3)} \end{aligned}$$

Recall the ciphertext CT is composed from:

$$\begin{aligned} C_0 &= g^{\beta(t)}, \\ C_1 &= \left(v. \prod_{(\forall j \in S_0)} g_{(n+1-j)}\right)^{\beta.t} \cdot \left(v. \prod_{(\forall j \in S)} g_{(n+1-j)}\right)^{\beta.t'} \\ C_2 &= M \cdot e\left(g_{n+1}, g\right)^{\beta \cdot t} \cdot e\left(g_{n+1}, g\right)^{\beta \cdot t'}, C_3 = g^{t'} \end{aligned}$$

First we reduce the numerator of decryption equation:

$$\begin{aligned} T &= \frac{e\left(g_k, \left(v. \prod_{(\forall j \in S_0)} g_{(n+1-j)}\right)^{\beta.t}\right)}{e\left(d_k \cdot \prod_{j \in S_0, j \neq k} g_{n+1-j+k}, C_0\right)} \\ &\quad \times \frac{e\left(g_k, \left(v. \prod_{(\forall j \in S)} g_{(n+1-j)}\right)^{\beta.t'}\right)}{e\left(d_k \cdot \prod_{j \in S, j \neq k} g_{n+1-j+k}, C_3\right)} \end{aligned}$$

$$
\begin{aligned}
&= \frac{e\left(g_k, g_{n+1-k}^{\beta\cdot t}\right)\cdot e\left(g_k, \left(v.\prod_{(\forall j\in S_0, j\neq k)} g_{(n+1-j)}\right)^{\beta.t}\right)}{e\left(d_k\cdot \prod_{j\in S_0, j\neq k} g_{n+1-j+k}, C_0\right)} \\
&\quad \times \frac{e\left(g_k, \left(v.\prod_{(\forall j\in S)} g_{(n+1-j)}\right)^{\beta.t'}\right)}{e\left(d_k\cdot \prod_{j\in S, j\neq k} g_{n+1-j+k}, C_3\right)} \\
&= \frac{e\left(g, g_{n+1}\right)^{\beta\cdot t}\cdot e\left(g_k, \left(v.\prod_{(\forall j\in S_0, j\neq k)} g_{(n+1-j)}\right)^{\beta.t}\right)}{e\left(d_k\cdot \prod_{j\in S_0, j\neq k} g_{n+1-j+k}, g^{\beta\cdot t}\right)} \\
&\quad \times \frac{e\left(g_k, \left(v.\prod_{(\forall j\in S)} g_{(n+1-j)}\right)^{\beta.t'}\right)}{e\left(d_k\cdot \prod_{j\in S, j\neq k} g_{n+1-j+k}, g^{\beta\cdot t'}\right)} \\
&= \frac{e\left(g, g_{n+1}\right)^{\beta\cdot t}\cdot e\left(g_k, \left(v.\prod_{(\forall j\in S_0, j\neq k)} g_{(n+1-j)}\right)\right)^{\beta.t}}{e\left(g^{\gamma}\cdot \prod_{j\in S_0, j\neq k} g_{n+1-j}, g_k\right)^{\beta\cdot t}} \\
&\quad \times \frac{e\left(g_k, \left(v.\prod_{(\forall j\in S)} g_{(n+1-j)}\right)\right)^{\beta.t'}}{e\left(g^{\gamma}\cdot \prod_{j\in S, j\neq k} g_{n+1-j}, g_k\right)^{\beta\cdot t'}} \\
&= \frac{e\left(g, g_{n+1}\right)^{\beta\cdot t}\cdot e\left(g_k, \left(v.\prod_{(\forall j\in S)} g_{(n+1-j)}\right)\right)^{\beta.t'}}{e\left(v\cdot \prod_{j\in S, j\neq k} g_{n+1-j}, g_k\right)^{\beta\cdot t'}}
\end{aligned}
$$

The right part of equation will be reduce as same manner:

$$T = (g, g_{n+1})^{\beta\cdot t}\cdot (g, g_{n+1})^{\beta\cdot t'}$$

then the decrypter can obtain the plaintext by computes $M = \frac{C_2}{T}$ in blow equation.

$$M = \frac{C_2}{T} = \frac{M.e\left(g_{n+1}, g\right)^{\beta.t}.e\left(g_{n+1}, g\right)^{\beta.t'}}{e\left(g_{n+1}, g\right)^{\beta.t}.e\left(g_{n+1}, g\right)^{\beta.t'}}$$

5 Security

In the following theorem we prove semantic security of the Bounded R-ABE scheme assuming the hardness of the (Decisional) n-BDHE assumption holds, which are:

Theorem 1. *let G be bilinear group of order p where p is prime and $n > 1$ our proposed Bounded R-ABE scheme is (n) semantically secure if the decision n-BDHE assumption holds in G_T.*

Proof. Assuming there exists PPT adversary algorithm A that can breakdown our scheme with advantage $AdvBRABE_(A, n) > \zeta$ in time t, also there exist algorithm B has advantage ζ to break n-BDHE problem in G_T, B calls algorithm A which selects the set S of users that A wishes to be challenged on.

Setup: B selects the initial authorized set S_0, current set S_c, chooses randomly $g, \alpha, \gamma, \beta \in Z_p$, then computes PP and publishes the public keys PK as $PP = (g, g_1, g_2, \ldots, g_n, g_{(n+2)}, \ldots, g_{2n}, v = g\cdot \left(\prod_{j\in S_c} g_{n+1-j}\right)^{-1}$, where $PK = (g' = g^{\beta}, w = e(g_n, g_1)^{\beta}, S_0))$ hence g, α, β, γ were chosen randomly then PK and PP have uniform distribution same as original scheme. $MUK = (S_c, v')$

where $v' = \left(v\cdot \prod_{j\in S} g_{n+1-j}\right)^{\beta}$ as in Equation (3).

Phase 1: The adversary A asks the algorithm B in this phase for attribute's private keys of users $S_q = \{q_1, q_2, \ldots, q_r\}$ recall that there are two possible scenarios:

1) Recall case1 when the user's index is not in authorized set S_0 such that $\forall i\ q_i \in S_q$ and $q_i \notin S_0$, and each attributes $\{att_i\}$ of user k satisfied the access circuit $C(\{att_i\}_{i\in k}) = true$. In this case the algorithm B computes:

$$
\begin{aligned}
d_k &= g_k\cdot \left(\prod_{j\in S_c} g_{n+1-j+k}\right)^{-1} \\
&= \left(g\cdot \left(\prod_{j\in S_c} g_{n+1-j+k}\right)^{-1}\right)^{\alpha^k} \\
&= v^{\alpha^k}
\end{aligned}
$$

Challenger B sets $y = k^{\gamma}$ then follows fine-grained tree to compute $\{r_{(i,j)})\}_{i\in Att_k, k\notin S, j\in\{A,B\}}$ similar to original scheme and then responds to A With attributes private keys $\left\{pr_{(k,i)} = g^{(\alpha^{r(i,j)})}\right\}_{i\in Att_k, j\in\{A,B\}}$ such that:

$$C\left(\left\{pr_{(k,i)} = g^{(\alpha^{r(i,j)})}\right\}_{i\in Att_k, j\in\{A,B\}}\right) = true$$

Note that the output for root node is $output(node_1) = d_k$.

2) In other hand for Case 2 when A is asking for the user $k \in S_q$ belonged to authorized sets $k \in S_0$ and he/she has been revoked $k \in S_r$. Also user k satisfied the access circuit $C(\{att_i\}_{i\in k}) = true$. Then B will update MUK.

Then B continues in computing the attribute private keys as in case1.

Challenge: After adversary A finished from the query phase then will submit to B two equal random messages m_0, m_1 where $|m_0| = |m_1|$, so B choses $\beta \in Z_p$ and toes fair coin $b \in \{0, 1\}$ to select one message, then B will simulate the running of encrypt and re-encrypt algorithms sequentially on the message $CT = Re - Encrypt(encrypt(m_b))$, while picks random value for $CT_{1-b} \in G_T$ and computes $C_{2,b} =$

$m_b \cdot e\left(g_{n+1}, h\right)$ WOLOG:

$$\begin{cases} C_{2,1-b} \in G_T \\ \quad \text{if ciphertext is chosen randomly(invalid))} \\ C_{2,b} = m_b \cdot e\left(g_{n+1}, h\right) \ \text{is valid n-BDHE.} \end{cases}$$

where $h = g^{\beta(t+t')}$ for the remaining ciphertext, B computes C_1 similar to the real scheme as in Equation (7) recall $C_1 = v'^{(t'+t)}$ and from the simulated value v' as in Equation (7) then C_1 is computed as follows:

$$\begin{aligned} C_1 &= (v')^{(t+t')} \\ &= \left(v \cdot \prod_{j \in S_c} g_{n+1-j}\right)^{\beta(t+t')} \\ &= \left(g \cdot \left(\prod_{j \in S_c} g_{n+1-j}\right)^{-1} \cdot \prod_{j \in S_c} g_{n+1-j}\right)^{\beta(t+t')} \\ &= g^{\beta(t+t')} \\ &= h. \end{aligned}$$

For other part of the ciphertext $C_0 = g'^t = g^{\beta \cdot t}, C_3 = g^{\beta \cdot t'}$ from Equation (8), then the algorithm B will submit the challenging ciphertext CT to A where $CT = (C_0, C_1, C_{2,0}, C_{2,1}, C_3)$.

Phase 2: The game between A and B will play identically as in phase1 with same restrictions.

Guess: Eventually the Algorithm A submits out b' guessing of the challenging ciphertext if $b' = b$ then B outs 0 showing that $T = C_{2,b} \div m_b = m_b \cdot e(g_{n+1}, h) \div (m_b) = e\left(g_{n+1}, h\right)$, else B outs 1 and that refers T is chosen randomly in G_T, Note that $|\Pr[B(h, g, g_1, \ldots, g_n, g_{(n+2)}, \ldots, g_{2n}, e(g_{(n+1)}, h)) = 0] - \Pr[B(h, g, g_1, ., g_n, g_{(n+2)}, \ldots, g_{2n}, T) = 0]| \geq \zeta$, is same to (Decisional)n-BDHE assumption and that is the proof of Theorem 1.

□

6 Implementation and Result

This section examines the system performance and tests the computations complexities, also we resolve the problem of complexity increasing significantly with number of users by applying powerful tool of pre-computation method.

6.1 Enhancing Performance

The proposed scheme has a dramatic increase of computation when KA is updating the keys or during user is decrypting the ciphertext which they need to multiply about $|S_0| + |S|$ times for every process and that consumes the resources especially on the users's side which are limited resources, so to overcome this problem we implement cached algorithm in both side without impacts the security, following equations shows the caching steps as:

1) When MUK is updated KA runs UpDateKey algorithm obtaining $MUK = \left(S, v' = (v.\prod_{(\forall j \in S)} g_{(n+1-j)})^{\beta}\right)$ to reduce the overhead multiplication KA can pre-compute $v' = \left(v.\prod_{(\forall j \in S)} g_{(n+1-j)}\right)^{\beta}$ and stores v' in cache memory so the MUK will reconstruct from v' and if the kA intends to add new user k' in this case will update only $v' = v'.g^{\beta}_{(n+1-k')}$, and in case of revoking existing user u then $v' = v'/g^{\beta}_{(n+1-k)}$, this will lead efficient calculation for MUK and reduces the computation cost from $O(|S|)$ to $O(1))$ for each time we run UpDateKey.

2) When decrypter user aims to decrypt some ciphertext according to decryption algorithm.

$$\begin{aligned} T &= \frac{e\left(g_k, C_1\right)}{e\left(d_k \cdot \prod_{j \in S_0, j \neq k} g_{n+1-j+k}, C_0\right)} \\ &\quad \times \frac{1}{e\left(d_k \cdot \prod_{j \in S, j \neq k} g_{n+1-j+k}, C_3\right)} \end{aligned}$$

There exit overhead computation on the client side who also has limited resource so this calculation can consume huge part from decrypter resources, again to handle this problem we apply caching algorithm by the client k as pre-computing parameter.

$$\begin{aligned} z_1 &= \prod_{j \in S_0, j \neq k} g_{n+1-j+k}, \\ z_2 &= \prod_{j \in S, j \neq k} g_{n+1-j+k} \end{aligned}$$

and stores z in fast cache memory, then for each decryption process the client k computes:

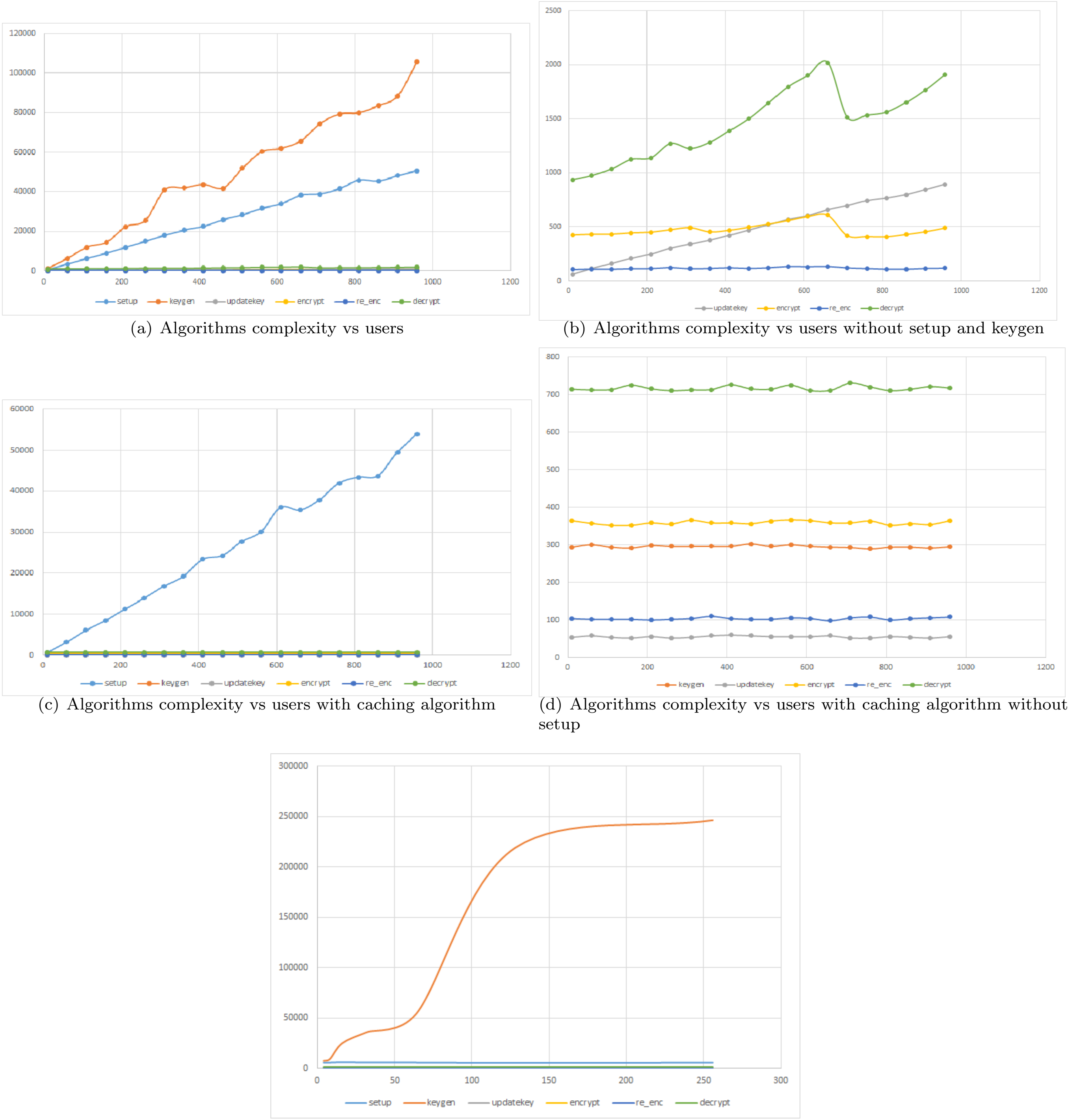
$$T = \frac{e\left(g_k, C_1\right)}{e\left(d_k \cdot z_1, C_0\right)} \times \frac{1}{\cdot e\left(d_k \cdot z_2, C_3\right)}$$

which again minimizes the overhead computing from $O(|S_0| + |S|)$ to $O(1))$.

3) There also overhead computation and communication between the KA and TTP when is sending the delegated keys DK, suppose there are many users were wanted to revoke and adds so each time KA sends:

$$DK = (\{g^{-\beta}_{n+1-u_i}\}_{i \in S_r}, \{g^{\beta}_{n+1-k'_i}\}_{i \in S_a}),$$

so we need about $O(|S_a| + S_r)$ iterations for computation and communication in both side (KA and



(a) Algorithms complexity vs users

(b) Algorithms complexity vs users without setup and keygen

(c) Algorithms complexity vs users with caching algorithm

(d) Algorithms complexity vs users with caching algorithm without setup

(e) Algorithms complexity vs users with caching algorithm

Figure 2: Comparison of computational efficiency

TTP) and that can be reduced by mixing all in one:

$$DK = \left(\prod_{i \in S_r} g_{n+1-u_i} \cdot \prod_{i \in S_a} g_{n+1-k'_i} \right)^{\beta}$$

6.2 Implementation

We demonstrate the proposed scheme and analyze the performance, we uses the useful MIRACLE library and runs under visual stdio 2012 C++ platform [12].

Figure a shows the growing of users affections only with setup, keygen, updatekey and decrypt (small affections). In figure b setup and keygen were hidden to presents there small correlation in decryption process.

Figure c shows the enhancing of system and powerful reduction of algorithms's complexity when we apply caching algorithm and that leads most of algorithms are running in few computation cost except setup algorithm, the setup algorithm is committed in figure d and is clear that all algorithms are running independently from number of users with low cost.

Setup algorithm is effected only with increasing exponentially with number of attributes as in figure e, whereas the remained algorithms are not effected.

7 Conclusion

We could be concluded that R-ABE with bounded ciphertext has short ciphertext and private keys in addition low computations complexity in both sides client users (encrypter and decrypter) and could be operated in limited resources environment, also we overcome the updating private keys problem, and we avoid the obstacle of stateless problem. The open problem to reduce the large size of the public keys and in our future work also we will intend to design multi key authorities R-ABE.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant No 61370026 and the National High Technology Research and Development Program of China (863) under Grant 2015AA016007, the Major International (Regional) Joint Research Project of China National Science Foundation under grant No.61520106007 and the science and technology foundation of Sichuan Province under Grant 2014GZ0109.

References

[1] B. Alexandra, G. Vipul, and K. Virendra, "Identity-based encryption with efficient revocation," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pp. 417–426, 2008.

[2] S. Amit and W. Brent, "Fuzzy identity-based encryption," in *Advances in Cryptology (EUROCRYPT'05)*, pp. 457–473, 2005.

[3] S. Amit, S. Hakan, and W. Brent, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology (CRYPTO'12)*, pp. 199–217, 2012.

[4] N. Dalit, N. Oni, and L. Jeff, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology (CRYPTO'01)*, pp. 41–62. Springer, 2001.

[5] B. Dan, G. Craig, and W. Brent, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Advances in Cryptology (CRYPTO'05)*, pp. 258–275, 2005.

[6] B. Dan and K. Jonathan, "Improved efficiency for cca-secure cryptosystems built using identity-based encryption," in *Topics in Cryptology (CT-RSA'05)*, pp. 87–103, 2005.

[7] B. Dan and F. Matt, "Identity-based encryption from the weil pairing," in *Advances in Cryptology (CRYPTO'01)*, pp. 213–229, 2001.

[8] L. David and S. Thomas, "Attribute-based broadcast encryption scheme made efficient," in *Progress in Cryptology (AFRICACRYPT'08)*, pp. 325–342, 2008.

[9] Z. Fengli, L. Qinyi, and X. Hu, "Efficient revocable key-policy attribute based encryption with full security," in *IEEE Eighth International Conference on Computational Intelligence and Security (CIS'12)*, pp. 477–481, 2012.

[10] C. Hui and D. Robert, "Revocable and decentralized attribute-based encryption," *The Computer Journal*, vol. 59, no. 8, pp. 1220–1235, 2016.

[11] H. Junbeom and N. Dong, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[12] MIRACL, *Users Manual Shamus Software Ltd*, No. 4 Foster Place North, Aug. 2006. (http://docs.miracl.com)

[13] A. Nuttapong and I. Hideki, "Attribute-based encryption supporting direct/indirect revocation modes," in *Cryptography and Coding*, pp. 278–300, 2009.

[14] A. Nuttapong and I. Hideki, "Conjunctive broadcast and attribute-based encryption," in *Pairing-Based Cryptography (Pairing'09)*, pp. 248–265, 2009.

[15] A. Nuttapong, H. Javier, L. Abien, L. Benoît, D. Panafieu, and R. Carla, "Attribute-based encryption schemes with constant-size ciphertexts," *Theoretical Computer Science*, vol. 422, pp. 15–38, 2012.

[16] D. Pratish, D. Ratna, and M. Sourav, "Adaptively secure unrestricted attribute-based encryption with subset difference revocation in bilinear groups of prime order," in *International Conference on Cryptology in Africa*, pp. 325–345, 2016.

[17] M. Silvio, "Efficient certificate revocation," US Patent 6,487,658, Nov. 26 2002.

[18] A. William, L. Sachin, and O. Rafail, "Fast digital identity revocation," in *Advances in Cryptology (CRYPTO'98)*, pp. 137–152, 1998.

[19] D. Yevgeniy and Y. Aleksandr, "A verifiable random function with short proofs and keys," in *Public Key Cryptography (PKC'05)*, pp. 416–431, 2005.

[20] X. Zhiqian and M. Keith, "Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage," in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'12)*, pp. 844–849, 2012.

Biography

Mohamed Ali Hamza received the master degree in the School of Computer Science and Engineering from the University of Electronic Science and Technology of China (UESTC) in Dec 2013, Now he is Ph.D candidate, His research areas in cryptography and information security.

Jianfei Sun is pursuing his Master degree from the Department of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC). His current research interests include cryptographic protocols and network security.

Xuyun Nie received the Ph.D. degree in Information security from Graduate university of Chinese Academy of Sciences, Beijing, China, in 2007. He is presently an Associate Professor at University of Electronic Science and Technology of China (UESTC). His research interests include cryptography and information security.

Zhiguang Qin is a professor in the School of Information and Software Engineering, UESTC. He received his PH.D. degree from UESTC in 1996. His research interests include: information security and computer network.

Hu Xiong received his PhD degree in the School of Computer Science and Engineering from the University of Electronic Science and Technology of China (UESTC) in Dec 2009. He is currently State Key Laboratory of Cryptology, P. O. Box 5159, Beijing, 100878, China and an associate professor in the School of Information and Software Engineering and School of Computer Science and Engineering, UESTC. His research interests include cryptographic protocols and network security.