International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119 Vol. 3 No. 1 June 2016, pp. 87-98

http://ijpsat.ijsht-journals.org

Characteristics Based Encoding Data Key Update CSP with Outsourced Revocation in Cloud Computing

Sai Rajesh Nalamati,Sriram Srujan

Department of Computer Science,
VIT University, Vellore

ABSTRACT: Characteristics-Based Encryption (CBE) which simplifies the public key and permit managing at Public Key Infrastructure (PKI) is an imperative different to public key encode. However, one of the major effectiveness problems of CBE is the overhead division at Private Key Generator (PKG) through user revocation. Proficient revocation has been glowing intentional in traditional PKI setting, but the burdensome managing of certificates is accurately the burden that CBE strives to alleviate. In this paper, planed at tackle the important issue of identity revocation, we intro outsourcing computation into CBE for the first time and suggest a revocable CBE system in the server-aided setting. Our system offloads most of the key generation allied operations through key-issuing and key-update procedure to a Key Update Cloud Service Provider (CSP), departure only a constant number of easy operations for PKG and users to make local. This goal is reached by utilizing a novel collusion-resistant method: we utilize a hybrid private key for every user, in which an AND gate is concerned to connect and jump the identity constituent and the time component. Moreover, we propose another building which is provable locked under the newly formulized Referred Delegation of Computation replica. Finally, we provide widespread experimental results to exhibit the efficiency of our proposed structure.

KEYWORDS: Characteristics-based encryption (CBE), revocation, outsourcing, cloud computing.

1. INTRODUCTION

Characteristics -Based Encryption (CBE) is an exciting different to public key encryption, which is planned to abridge key management in a verification-based Public Key Infrastructure (PKI) through using human- comprehensible identities (e.g., IP address, unique name, email address, etc) as public keys. Therefore, dispatcher using CBE does not need to appear happy public key and verification, but straightly encrypts message with receiver identity. Consequently, receiver achieving the private key related with the corresponding identity from Private Key Generator (PKG) is capable to decrypt such ciphertext. Though CBE allows an subjective twine as the public key which is measured as an attractive advantages over PKI, it demands an capable revocation mechanism. Specifically, if the private keys of some users get concessioner, we must provide a signify to revoke such users from system. In PKI setting, revocation mechanism is realized by append validity periods to certificates or using involved combination of technique [1]–[3]. Even so, the cumbersome management of permits is precisely the weigh down that IBE strives to assuage. As far as we know, though revocation has been methodically studied in PKI, few revocation mechanisms are known in CBE setting. In [4], Boneh and Franklin optional that users renew their private keys uncommonly and senders use the receivers' identities concatenated with current point period. But this instrument would result in an in the clouds load at PKG. In an additional word, all the customers despite of whether their keys have been withdraw or not, have to contact through PKG periodically to prove their distinctiveness and update new private keys. It necessitate that PKG is online and the secure channel must be maintained for all business, which will happen to a bottleneck for CBE system as the number of users grows. Presented a revocable CBE scheme. Their system is built on the idea of fuzzy IBE primitive [6] but utilizing binary tree data structure to evidence users' identities at leaf nodes. Therefore, key-update competence at PKG is able to be appreciably compact from linear to the elevation of such binary tree (i.e. logarithmic in the number of users).

Corresponding Author: Sai Rajesh Nalamati

Nonetheless, we point out that although the binary tree introduction is talented to achieve a relation high performance, it will answer in other problems: 1) PKG have to generate a key couple for all the lumps on the path from the individuality leaf node to the root lump, which outcome in complexity logarithmic in the no. of users in system for issue a single private key . 2) The bulk of private key produce s in logarithmic in the integer of users in system, which makes it tricky in private key storage space for users. 3) As the no. of users in scheme grows, PKG has to preserve a double tree with a huge amount of nodes, which initiates another bottleneck for the global organization. In tandem with the enlargement of cloud computing, there has emerge the capability for addicts to buy on-demand enlarge from cloud-based soldierly such as Amazon's EC2 and Microsoft's Windows Azure. Thus it requests a new working paradigm for commence such cloud services into CBE revocation to fix the question of efficiency and storage in the clouds portray above. A naive draw near would be to minimally hand over the PKG's master key to the Cloud Service Providers (CSPs). The CSPs could then merely update all the confidential keys by using the conventional key update technique [4] and convey the private keys back to unrevoked users. However, the inexperienced approach is based on an impractical supposition that the CSPs are fully faith and is allowed to access the master key for CBE system. On the opposing, in practice the public clouds are expected outside of the similar trusted domain of users and are inquisitive for users' individual solitude. For this reason, a confront on how to design a safe revocable CBE method to decrease the overhead totaling at PKG with an un trusted CSP is raised.

In this paper, we launch outsourcing computation into CBE revocation, and party the defense portrayal of outsourced revocable CBE for the first time to the best of our familiarity. We proposition a scheme to relieve of all the key generation correlated function during key-issuing and key update, departure only an invariable number of effortless procedures for PKG and at liberty users to achieve locally. In our scheme, as with the proposal in [4], we appreciate revocation during inform the private keys of the unrevoked users. But unlike that work [4] which unimportantly concatenates time affair with individuality for key generation/update and involve to re-issue the entire private key for unrevoked users, we proposition a novel collusion-resistant key issuing method: we utilize a jumble confidential key for every user, in which an AND gate is worried to connect and spring two sub-components, namely the independence ingredient and the time component. At first, user is able to gain the identity constituent and a defaulting time constituent (i.e., for current time period) from PKG as his/her private key in key-issuing. Afterwards, in order to uphold decrypt ability, unrevoked users wants to infrequently request on key update for time part to a newly introduced entity named Key Update Cloud Service Provider (KU-CSP).

Compared with the former work [4], our system does not have to re-issue the whole private keys, but just require informing a lightweight constituent of it at a particular entity KU-CSP. We also state that 1) with the aid of KU-CSP, user wants not to contact with PKG in key-update, in additional words, PKG is allowable to be offline following sending the revocation list to KU-CSP. 2) No protected channel or user verification is required during key-update connecting user and KU-CSP. Furthermore, we consider realizing revocable CBE with a semi-honest KU-CSP. To achieve this goal, we nearby a security improved structure under the recently dignified Refereed Delegation of Computation (RDoC) representation [7]. Finally, we provide extensive investigational results to show the effectiveness of our proposed structure.

2. PRELIMINARY

In this section, we give a succinct review on some cryptographic background and characteristic based encryption.

A. CRYPTOGRAPHIC BACKGROUND

Definition 1. (Bilinear map) Let G G be cyclic groups of prime order , writing the group action multiplicatively. g is a generator of G. Let G × G→ GT be a map with the following properties:

- Bilinearity: $e(g_1^a, g_2^b)=e(g_1,g_2)^{ab}$ for all g_1,g_2 £ G, and a,b £ Z_q ;
- Non-degeneracy: There exists g_1,g_2 £ G with $e(g_1,g_2)\neq 1$, in other words, the does not map send all pairs in G× G to the identity in G_T ;
- Computability: There is an efficient algorithm to compute for $e(g_1,g_2)$ all g_1,g_2 £ G.

Bilinear Diffie-Hellman Problem and Bilinear Diffie-Hellman Assumption:

The Bilinear Diffie-Hellman (BDH) problem in (G,G1, ˆe) is described as follows, given random g ∈ G, and ga, gb, gc for some a, b, c ∈ Z∗p, compute ˆe(g, g)abc ∈ G1. The BDH assumption is presented as follows, given (G,G1, ˆe), g ∈ G, and ga, gb, gc for some a, b, c ∈ Z∗p, an adversary A has advantage ϵ in solving BDH when Pr[A(ga, gb, gc) = ˆe(g, g) abc] ≥ ϵ. The BDH assumption tells that the advantage ϵ is negligible for any polynomial time A.

B. CHARACTERISTIC-BASED ENCRYPTION

An CBE scheme which typically absorb two entities, PKG and consumers (including sender and receiver) is consisted of the subsequent four algorithms.

Setup(λ) : The setup algorithm capture as input a security limit and outputs the public key PK and the master key MK. Note that the master key is reserved covert at PKG.

KeyGen(MK,ID) : The private key creation or generator algorithm is run by PKG, which achieve as input the master key MK and user's individuality ID £ {0,1}* . It returns a private key SKID equivalent to the identity ID.

Encrypt (M,ID'): The encryption algorithm is jog by sender, which gets as enter input the receiver's identity ID' and a message M to be encrypted. It outputs the ciphertext CT.

Decrypt (CT,SKID'): The decryption algorithm is run by receiver, which gets as enter input the ciphertext CT and his/her private key SKID' . It returns a message or an error ⊥.

A CBE scheme must satisfy the definition of steadiness. Specifically, when the private key generate by algorithm KeyGen when it is given as the contribution, then Decrypt where Encrypt. The motivation of CBE is to simplify diploma organization. For ex, when Alice sends an email to Dob at dob@company.com, she only encrypts her message by means of Bob's email address "bob@company com", but does not need to obtain Bob's public key credential. When Bob receives the encrypted email he authenticates himself at PKG to find his private key, and examine his email with such a private key.

3. PROBLEM STATEMENTS

A. SYSTEM MODEL

We present system replica for outsourced revocable CBE in Fig. 1. Compared with that for archetypal CBE scheme, a KU-CSP is concerned to become conscious revocation for cooperation users. Actually, the KU-CSP can be imagining as a public cloud run by a third party to transport basic work out capabilities to PKG as standardized services over the network. Typically, KU-CSP is hosted gone from either users or PKG, but provides a way to reduce PKG totaling and storage cost by given that a flexible, even impermanent additional room to infrastructure. When revocation is triggered, as an alternative of re-requesting personal keys from PKG in [4], unrevoked users have to ask the KU-CSP for updating a insubstantial component of their personal keys. Though many information are involved in KU-CSP's operation, in this paper we just logically foresee it as a computing service source, and fear how to design sheltered scheme with an untrust KU-CSP. Bottom on the system model proposed, we are intelligent to describe the outsourced revocable CBE scheme. Evaluate with the traditional CBE definition, the KeyGen Encrypt and Decrypt algorithms are redefined as follows to amalgamate time component. Note that two lists RL and TL are make use of in our definition, where RL records the identity of revoked users and TL is a linked list for past and present time period. Setup(λ) : The setup algorithm takes as input a security stricture and outputs the public key PK and the master key MK. Note that the master key is kept secret at PKG. KeyGen(MK,ID,RL,TL) : The private key generation algorithm is run by PKG, which obtain as input the master key MK and user's identity ID £ {0,1}* . It returns a private key SKID = (IK[ID],TK[ID]Ti) equivalent to the identity ID. In this paper, we argue user revocation that is how to deny users of decrypt ability even if they contain been issued their confidential keys. To this end, we insert a time epoch into private key in a clever style for revocation. Specifically, in the matching example illustrated in Section 2.2, Alice in our setting not only encrypts message with Bob's email address "bob@company com" except also with present time period (e.g., "Thu Jul 18 2015"). When receives the encrypted email, Bob then obtains his private key consisting of an individuality component and a time period constituent from PKG. With the both right components, the email can be read. Presume Bob is compromised. Then, the time machinery of all the other users are updated by KU-CSP with a new time period (e.g., "Fri Jul 19 2015"). From then on, the memorandum sent to Bob should be encrypted with Bob's email address and the updated time period. Since Bob does not surround the time constituent correspondent to the updated time period, the following encrypted messages cannot be decrypted by Bob smooth if they are wished-for for him. The challenge in designing the outsourced revocable CBE scheme is how to thwart collusion between Bob and other unrevoked deceiving users. intentionally, a untruthful user (named Eve) can share her updated time element (i.e., "Fri Jul 19 2015") with Bob, and help Bob decrypt ciphertext even if Bob just has the preceding one (i.e., "Thu Jul 18 2015"). We will illustrate how to avoid such collusion later.

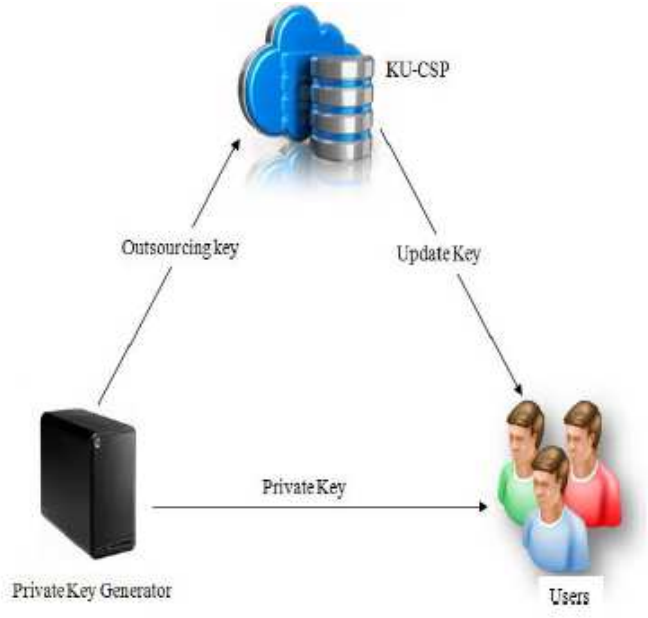


Fig. 1. System model for CBE with outsourced revocation.

B. SECURITY DEFINITION

We suppose that KU-CSP in the planned system representation is semi-trusted. Specifically, it will pursue our practice but try to find out as to a large extent secret information as possible based on its possession. Therefore, two types of adversaries are to be painstaking as follows. Type-I adversary. It is defined as a curious user with identity but repeal before time period. Such challenger tries to obtain useful information from ciphertext anticipated for him/her at or after (e.g. time period) throughout colluding with other users even if they are unrevoked. Therefore, it is allowed to ask for private key counting identity component and updated time constituent for cooperative users. We specify that beneath the supposition that KU-CSP is semi-trusted, type-I adversary cannot get outsourcing key for any users. Type-II adversary. It is defined as a snooping KU-CSP which aims to obtain useful information from ciphertext wished-for for some target identity at time period. Such opposition not only possess of outsourcing keys for all users in the system, but also is clever to get user's private key through colluding with any other user with identity. It is noted that to make such attack reasonable, we must restrict ID≠ID.

Having the intuition exceeding, we are able to describe CCA security game for type-I and type-II adversary correspondingly for our setting in Fig. 2. Suppose Ai is the type- adversary for i=I,II . Then, its advantage in attacking the CBE with outsourced revocation system E is defined as Adυε,Ai (λ)=|Pr[bi=bi']-1/2|.

Definition 3. A CBE with outsourced revocation method is semantically secure touching adaptive chose-ciphertext attack (IND-ID-CCA) if veto polynomially bounded adversary have a non-negligible advantage touching challenger in security game for both type-I and type-II adversary. Finally, away from the CCA security, we also specify that 1) An CBE with outsourced revocation method is INDID- CPA protected (or semantically secure against chosen plaintext attack) if no polynomial time challenger has non-negligible improvement in modified games for both type-I and type-II adversary, in which the decryption revelation in both phase 1 and phase 2 is removed; 2) An CBE with outsourced revocation method is secure in discerning model if no polynomial time adversary has non negligible advantage in modified games for both type-I and type-II adversary, in which the challenge distinctiveness and time period is submitted before setup.

CCA Security Game for Type-I Adversary

Setup: Challenger runs Setup(λ) to obtain the key pair (PK, MK) and output PK.

Phase 1: Challenger initializes an empty table list L and an empty set S. Adversary is provided the following oracles.

- *Private key extraction oracle*. Upon receiving ID, run KeyGen to obtain $SK_{\mathrm{ID}} = (IK[\mathrm{ID}], TK[\mathrm{ID}]_{T_i})$ and OK_{ID}. After adding the entry $(\mathrm{ID}, SK_{\mathrm{ID}}, OK_{\mathrm{ID}})$ into L, output $IK[\mathrm{ID}]$.
- *Updated key extraction oracle*. Upon receiving ID and T_j, if there exists an entry of $(\mathrm{ID}, SK_{\mathrm{ID}}, OK_{\mathrm{ID}})$ in L, set $S = S \cup \{(\mathrm{ID}, T_j)\}$. Accordingly, run KeyUpdate and output the updated key $TK[\mathrm{ID}]_{T_j}$.
- *Decryption oracle*. Upon receiving ID, T_j and CT, run Decrypt and output the resulting plaintext M.

Challenge: Adversary outputs two equal-length plaintexts M_0, M_1, T^* and ID^* with the restriction that $(\mathrm{ID}^*, T^*) \notin S$. Challenger picks a random bit $b_{\mathrm{I}} \in \{0,1\}$ and sets $CT^* = \mathsf{Encrypt}(M_{b_{\mathrm{I}}}, \mathrm{ID}^*, T^*, PK)$.

Phase 2: Adversary adaptively issues more queries as in phase 1 with the restriction that (ID^*, T^*) cannot be queried in updated key extraction oracle.

Guess: Finally, adversary outputs a guess $b'_{\mathrm{I}} \in \{0,1\}$ and wins the game if $b'_{\mathrm{I}} = b_{\mathrm{I}}$.

CCA Security Game for Type-II Adversary

Setup: It is identical to the setup phase in the CCA security game for type-I adversary.

Phase 1: Challenger initializes an empty table list L, and two empty sets U and I. Then, adversary is provided the following oracles.

- *Outsourcing key extraction oracle*. Upon receiving ID, challenger runs KeyGen to obtain $SK_{\mathrm{ID}} = (IK[\mathrm{ID}], TK[\mathrm{ID}]_{T_i})$ and OK_{ID}. After adding the entry $(\mathrm{ID}, SK_{\mathrm{ID}}, OK_{\mathrm{ID}})$ into L, output OK_{ID}.
- *Private key extraction oracle*. Upon receiving ID, if there exists an entry $(\mathrm{ID}, SK_{\mathrm{ID}}, OK_{\mathrm{ID}})$ in L, set $U = U \cup \{\mathrm{ID}\}$ and return $IK[\mathrm{ID}]$ back to adversary.
- *Updated key extraction oracle*. Upon receiving ID and T_j, if there exists an entry $(\mathrm{ID}, SK_{\mathrm{ID}}, OK_{\mathrm{ID}})$ in L, check on whether $\mathrm{ID} \in U$, if not set $I = I \cup \{T_j\}$. Then, run KeyUpdate and output $TK[\mathrm{ID}]_{T_j}$.
- *Decryption oracle*. It is identical to the decryption oracle in the CCA security game for type-I adversary.

Challenge: Adversary outputs two equal-length plaintexts M_0, M_1, T^* and ID^* with the restriction that $T^* \notin I$ and $\mathrm{ID}^* \notin U$. Challenger picks a random bit $b_{\mathrm{II}} \in \{0,1\}$ and sets $CT^* = \mathsf{Encrypt}(M_{b_{\mathrm{II}}}, \mathrm{ID}^*, T^*, PK)$.

Phase 2: Adversary adaptively issues more queries as in phase 1 with the restrictions that i) ID^* cannot be queried in private key extraction oracle; ii) (ID^*, T^*) cannot be queried in updated key extraction oracle.

Guess: Finally, adversary outputs a guess $b'_{\mathrm{II}} \in \{0,1\}$ and wins the game if $b'_{\mathrm{II}} = b_{\mathrm{II}}$.

Fig. 2. CCA security game for type-I and type-II adversary

4. EFFICIENT IBE WITH OUTSOURCED REVOCATION

A. INTUITION

In order to accomplish efficient revocation, we introduce the proposal of "partial private key update" into the proposed building, which functions on two sides: 1) we exploit a "hybrid private key" for each user in our scheme, which employs an AND gate concerning two sub-components specifically the identity constituent IK and the time component TK respectively. IK is make by PKG in key-issuing but is restructured by the newly introduced KU-CSP in key update; 2) In encryption, we get as input user's identity ID as well as the time period T to confine decryption, more accurately, a user is allowed to achieve successful decryption if and only if the personality and time stage entrenched in his/her private key are identical to that connected with the ciphertext. Using such ability, we are talented to rescind user's decrypt ability through updating the time constituent for private key by KU-CSP.

Furthermore, we observation that it cannot irrelevantly operate an matching updated moment component for all users since revoked user is able to re-construct his/her aptitude during colluding with unrevoked users. To eradicate such conspiracy, we randomly produce an outsourcing key for each characteristic, which essentially decides a "matching relationship" for the two sub-components. In addition, we let KU-CSP maintain a list UL to evidence user's identity and its equivalent outsourcing key. In key-update, we be able to use OKID to update the time constituent TK[ID']T for identity ID'. Suppose a user with identity ID is rescind at Ti. Even if he/she is able to acquire TK[ID']Ti+1 for identity ID', the revoked consumer still cannot decrypt ciphertext encrypted underneath Ti+1.

B. PROPOSED CONSTRUCTION

We present our construction based on [6] as follows. Setup(λ) : The setup algorithm is run by PKG. It selects a random generator gεRG as well as a random integer xεR Z , and sets g1= gx . Then, PKG picks a random element gεR G and two hash functions H1,H2{0,1}*→ GT. Finally, output the public key PK=(g,g1,g2,H1,H2) and the master key MK=x.

KeyGen(MK,ID,RL,TL,PK): For each user's private key request on identity ID, PKG firstly checks whether the request identity exists in RL , if so the key generation algorithm is aborted. Next,PKGrandomly selects x1εR Zq and sets x2=x-x1 mod q . It randomly chooses rID εRZq , and computes IK[ID]=(gx12.(H1(ID))rID,grID) . Then, PKGreads the current time period Ti from TL (we require that PKG should create current time period firstly if TL is empty). Accordingly, it randomly selects rTi εR Zq and computes TK[ID]Ti=(dTi0,dTi1) , where dTi0=gx22.(H2(Ti))rTi and dTi1=grTi. Finally, output SKID=(IK[ID],TK[ID])and OKID. Encrypt(M,ID,Ti,PK): Suppose a user wishes to encrypt a message M under identity and time period Ti . He/She selects a random value and computes C0=Me(g1,g2)s,C1=gs,EID=(H1(ID))S ETi=(H2(Ti))S.

Finally, publish the ciphertext as CT=(C0,C1,EID,ETi). Decrypt(CT,SKID,PK): Suppose that the ciphertext CT is encrypted under ID and , and the user has a private key SKID=(IK[ID],TK[ID]Ti), where IK[ID]=(d0,d1) and TK[ID]Ti=(dT0,dT1). He/She computes

$$\begin{aligned} M &= \frac{C_0 e(d_1, E_{\mathrm{ID}}) e(d_{T_i 1}, E_{T_i})}{e(C_1, d_0) e(C_1, d_{T_i 0})} \\ &= \frac{Me(g_1, g_2)^s}{e(g, g_2)^{x_2 s} e(g, g_2)^{x_1 s}} \\ &= M. \end{aligned}$$

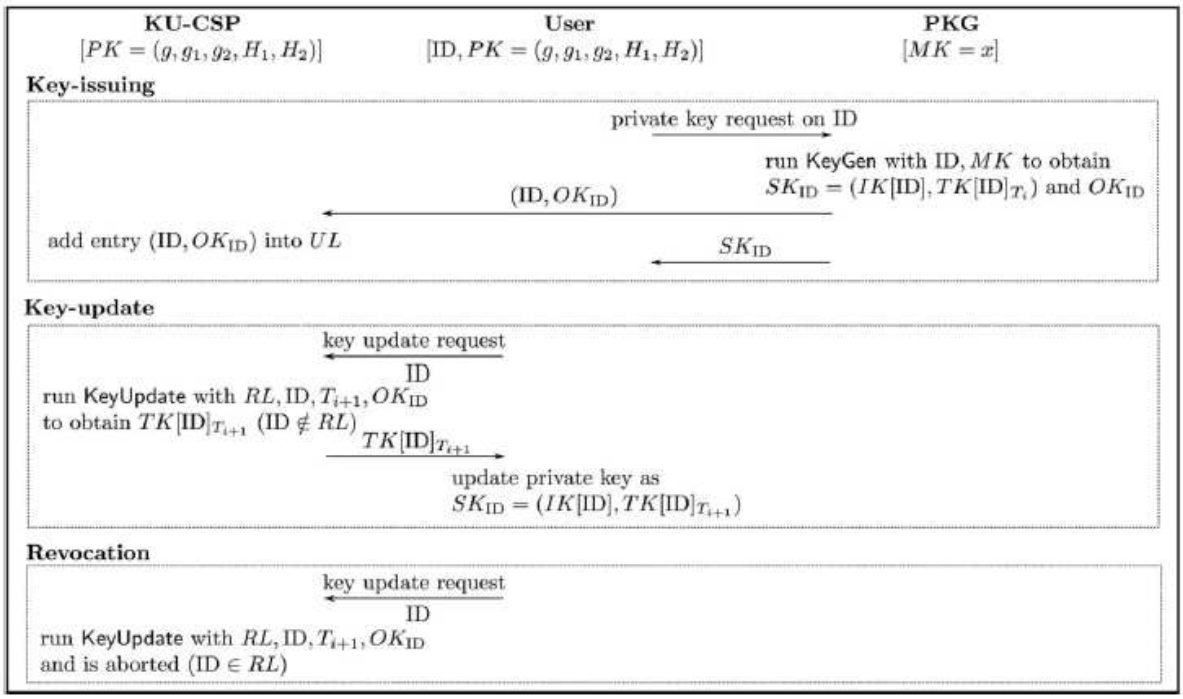


Fig. 4. Protocol for key-issuing, key update and revocation.

Revoke(RL,TL,{IDi1,IDi2,….,IDik}): If users with identities in the set are to be revoked at time period , PKG updatesthe revocation list as as well as the time list through linking the newly created time period onto original list . Finally send a copy for the updated revocation list as well as the new time period to KU-CSP. KeyUpdate : Upon receiving a keyupdate request on , KU-CSP firstly checks whether exists in the revocation list , if so KU-CSP returns and key-update is aborted. Otherwise, KU-CSP fetches the corresponding entry ( ) in the user list . Then, it randomly selects Z , and computes

Finally, we emphasize that the idea behind our construction is to realize revocation through updating the time component

$$d_{T_{i+1}0} = g_2^{x_2} \cdot (H_2(T_{i+1}))^{r_{T_{i+1}}} \quad \text{and} \quad d_{T_{i+1}1} = g^{r_{T_{i+1}}}$$

Final output is TK[ID]Ti+1=(dTi+10,dTi+11)1

output .1 Finally, we emphasize that the idea behind our construction is to realize revocation through updating the time component in private key.

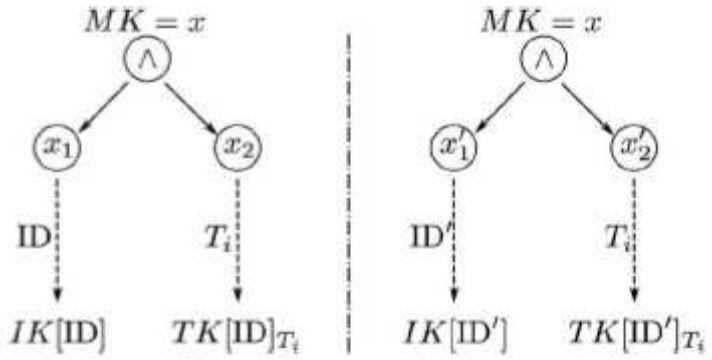


Fig. 3. A comparison on generating private key for two different users.

Therefore, the key point is to avoid revoked user from conspire with other users to re-construct his/her private key. As announce in intuition, such scheme assault is resistant in our designed building due to the random split on for each user. Specifically, as shown in Fig. 3 in which is an AND gate connecting two sub-components, if two different users call for their private keys, PKG will obtain two randomly splits (x1,x2) and (x1',x2') with the complementary that x1+x2=x mod q and x'1+x'2=x mod q.x1 and x'1 are used to fabricate the identity component for and respectively, while the time element is individually produce. By the reason that the complementary exists between x1 and x2 as well as x1' and x2', the identity element and time component should consequently have a "verification" in private key. With such "verification", even if a curious user achieve time constituent of other users, he/she cannot counterfeit a valid private key for himself to absolute decryption successfully.

C. KEY SERVICE PROCEDURES

Based on our algorithm building, as shown in Fig. 4, the key overhaul procedures including key-issuing, key-update and revocation in proposed CBE scheme with outsourced revocation work as follows.

Key-issuing. We oblige that PKG maintains a revocation list RL and a time list TL locally. Upon getting a private key req request on ID, PKG runs KeyGen(MK,ID,RL,TL,PK) to obtain private key and outsourcing key . Finally, it sends SKID to user and (ID,OKID) to KUCSP respectively. As illustrate in intuition, for each entry (ID,OKID) sent from PKG, KU-CSP ought to add it into a constrained maintained user list UL.

Key-update. If some users have been rescind at time period, each unrevoked user needs to send key-update demand to KU-CSP to uphold decrypt ability. Upon receiving the appeal on identity, KU-CSP runs KeyUpdate (RL,ID,Ti+1,OKID) to obtain TK[ID]Ti+1. Finally, it sends such time section back to user who is able to update his/her private key as SKID=(IK[ID], IK[ID]Ti+1) .

Revocation. Similar to key-update, if a retract user sends a key-update demand on identity ID, KU-CSP runs.

5. ADVANCED CONSTRUCTION UNDER REFEREED DELEGATION OF COMPUTATION MODEL

In this division, we will challenge to propose a security improved construction under the in recent times formalized RDoC structure.

A. ADVANCED CONSTRUCTION

RDoC structure creates from the model of refereed games in [8], and is later formalized in [9] and [10]. In RDoC representation, the customer is able to join up with several servers and it has a correct output as extended as there exists one server that pursues the proposed protocol. One of the most advantages of RDoC over time-honored model with single server is that the defense risk on the single server is packed in to multiple servers implicated in. As the result of both the common sense and efficacy, RDoC model freshly has been extensively utilized in the journalism of outsourced computation [9], [10], [11], [7], [12]. In order to relate RDoC to our setting, we introduce another independent KU-CSPs. For simplicity, in the rest of paper, we only paying attention on the case that as shown in Fig. 5. In addition, we have three requirements in such form: 1) At least one of the KU-CSPs is honest. 2) Computational difficulty at the truthful KU-CSP is not greatly more than the other necessary performing revocation.

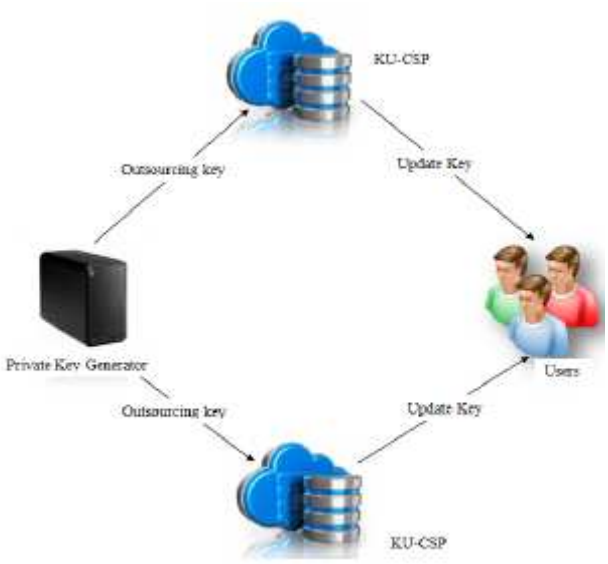


Fig. 5. System model with two KU-CSPs.

PKG's running time would be a lot smaller than necessary to directly achieve revocation. We figure out that the challenge to realize such advanced construction is to demand that and cannot be leaked at the same time. To achieve this goal, we randomly split into and which will be separately used by the two KU-CSPs to produce partial time component. After receiving the two partial time components, user performs a production to make a combination and obtains the final updated key (i.e. time component for private key). Since the setup, encryption and decryption phases operate exactly as before, we will introduce the KeyCombine algorithm and only provide the key generation and revocation for the advanced construction as follows. KeyGen(MK,ID,RL,TL,PK) : The algorithm is presented similar to that in our proposed construction in Section 4. The only difference is that PKG does not directly send OKID=x2 to KU-CSP, but makes a further random split on x2 to obtain x21 and x22 with x2=x21+x22 . Finally, PKG sends to ⌊-th KU-CSP for ⌊=1,2. KeyUpdate (RL,ID,Ti+1,OK(⌋)ID) : Upon receiving the keyupdate request on , the -th KU-CSP checks whether exists in the revocation list , if so the key update is aborted. Otherwise, it fetches the corresponding entry (ID, OKID=x21) in the user list and computes $TK[\mathrm{ID}]_{T_{i+1}} = (d_{T_{i+1}0}, d_{T_{i+1}1});$ as shown in Fig. 6. Finally send the updated partial time component back to in Fig. 6 to obtain . Finally update SKID as (IK[ID],TK[ID]Ti+1).

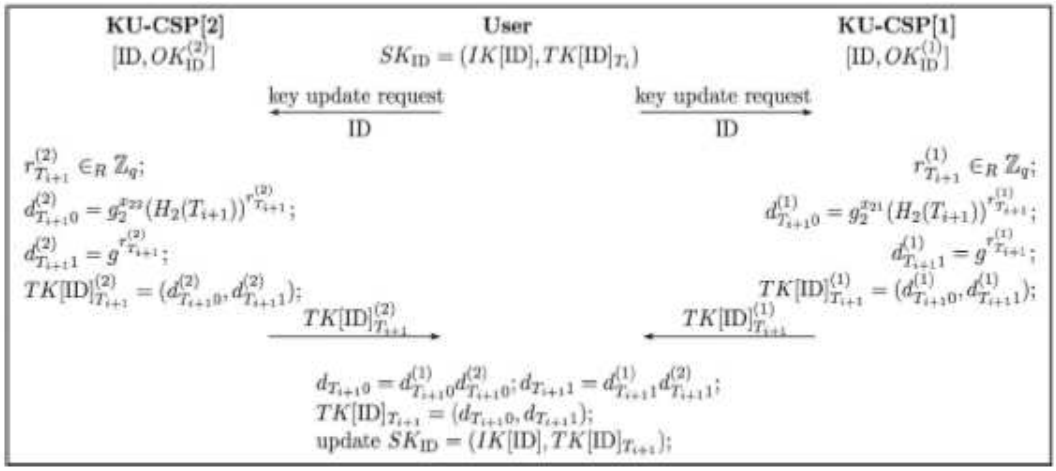


Fig. 6. KeyUpdate and KeyCombine in advanced construction.

B. Security Analysis

As a stronger opponent model, RDoC imprison much more connotation ahead of the "honest-but-curious" sense, that is inquisitive user is allowable to cooperate with at most servers if servers are concerned. To accommodate to this case, we

amend the private key oracle somewhat to adapt to a pair of outsourcing keys and commence another outsourcing key drawing out oracle for Type-I adversary as follows. It is noted that the challenger is required to continue a pour set to restrict adversary admittance the whole outsourcing key for a quantity of identity. These correspond with the hypothesis that at least one of the KU-CSPs is truthful.

TABLE 1 Efficiency Comparison for Stages in Revocable CBE

	Our Scheme	IBE without Revocation [4]
Setup	83.764 ms	80.233 ms
Key-Issuing	40.369 ms	20.121 ms
Encryption	39.840 ms	24.595 ms
Decryption	21.278 ms	10.285 ms
Key-Update	10.300 ms[1]	—-

This time cost is evaluated at KU-CSP.

6. PERFORMANCE EVALUATION

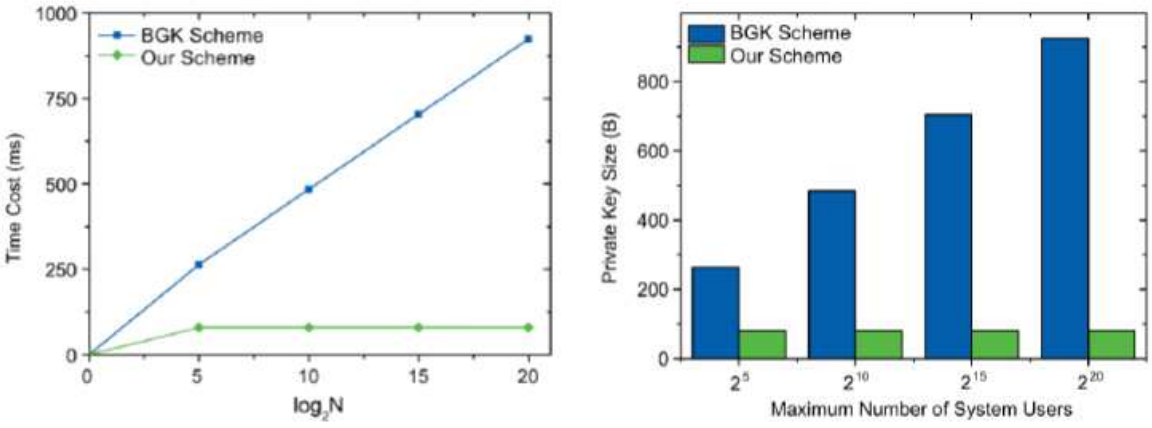
In this fragment, we will give a methodical experimental evaluation of the construction planned in Section 4. We manufacture our testbed by using 64-bit M2 high-memory user. KeyCombine : Upon receiving and , user performs a key combination by computing and as the paradigm shown quadruple additional large Linux servers in Amazon EC2 podium as KU-CSP, and a Linux machine with Intel(R) Core(TM)2 Duo CPU clocked at 2.40 GHz and 2 GB of system memory as the user and PKG. Note that in all the evaluations, the groups G and G are selected in 160-bit and 512-bit length respectively.

A. PERFORMANCE EVALUATION FOR OVERALL SCHEME

Firstly, we aim to appraise the competence of our outsourced revocable system by comparing the total time full during each phase with the original CBE [4] which do not believe revocation. In Table 1, we examine the time cost of perform someone stage by the both schemes. It is not astounding to observe that our scheme obtains more time since we judge the revocability issue. Note that our format shares the matching setup algorithm with the CBE system in [4]. Our key-issuing phase is qualified longer than that in the CBE proposal [4]. This is because we embed a time constituent into each user's confidential key to allow sometimes update for revocation, resulting that some additional multiplication are needed in our proposal to initialize this constituent. Our encryption and decryption is faintly longer than the CBE scheme [4], which is also outstanding to the existence of the time module. The user desires to complete an additional encryption/decryption for this component, relatively than just encrypt/decrypt the individuality component. To sum up, our revocable proposal achieves characteristic based encryption/decryption and revocability without introducing important in the clouds compared to the original CBE system [4] (our implementation time is still within millisecond).

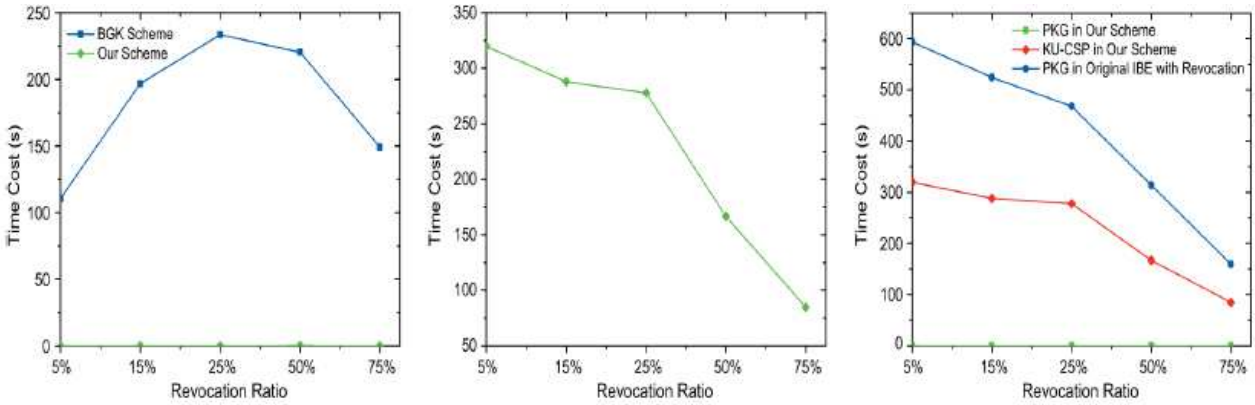
B. PERFORMANCE EVALUATION FOR REVOCATION

Secondly, we challenge to replicate the situation of multi-user revocation, and show an broad judgment between our experimentation, we use a 32-bit numeral to identify each node in binary tree which is developed in BGK scheme [5] for organization users. Our judgment is in conditions of the key-issuing period and the key-update phase.



(a) Responding Time for Single Key Generation Request (b) Private Key Size

Fig. 7. Comparisons in key-issuing (N is the maximum number of users in the system).



(a) Key Update at PKG (b) Key Update at KU-CSP (c) Key Update Comparison with [4]

Fig. 8. Comparisons in key update (Case: 215 system users).

C. KEY-ISSUING STAGE

We differ the maximum number of consumers in the organization and show the respond time for a single key production request in Fig. 7(a). It is not tough to see that the respond time in BGK method [5] is in proportion of wherever is the maximum number of consumers in system. This is since a binary tree is utilized to supervise all the users, each leaf node of which is disperse to a solitary user in outsourced revocation scheme and a further revocable CBE scheme–BGK scheme [5]. Note that in this set of structure. During key-issuing, PKG has to perform multiplication on all the nodes in the path starting the consequent leaf node to root node. Compared to the logarithmically emergent efficiency in [5], our schemes achieve constant effectiveness (nearly six modular exponentiation in G) in particular key-issuing. Likewise, we demonstrate the comparison on confidential key size in Fig. 7(b). Due to the same motive of difficult for computation on all the nodes in conduit from leaf node to root node, the preceding approach [5] has an mounting private key size, whereas ours accomplish constant key size (nearly four facet in group G). In addition the better presentation in competence and private key size, a different benefit of our method over the preceding work [5] is that it chains dynamic number of users. Specifically, the previous works [5] necessitate fixing the maximum number of consumers in system originally to facilitate building the binary tree. Once the maximum number is permanent, it is difficult to add users beyond this bound. Ours do not contain such a drawback, and flexibly chains energetic management of users.

D. KEY UPDATE STAGE

In this conduct experiment, we haphazardly pick 5% to 75% users and contrast the total time of inform private keys for the rest users. For effortlessness, we just demonstrate an example and contrast the key-update time at PKG in revocation in the case of organization users in Fig. 8(a). It can be see that the good organization curve of BGK scheme [5] shows a parabolic

shape, and at the 25% revocation ratio, the effectiveness achieves the lowly position in our evaluation. This is for the reason that it is the gap that the leaf nodes to be revoke has a large amount but low aggregation degree, which entail that we have to modernize a lot of domestic nodes for key-update. However, in our design, such a performance is avoided, and just a unimportant unvarying time is taken at PKG. More in general, this constant key-update competence is actually achieve by our system with not in view of to the number of system consumers while we tender over the revocation to KU-CSP, but BGK scheme [5] require an growing time cost with the no. of system users (Table 2 shows the underlying tendency in more cases). Accordingly, we also show the time cost at KU-CSP in our scheme for update private keys for all the unrevoked users in the revocation ratio assortment from 5% to 75%. It must be piercing out, though, such a time cost is increasing with the number of users in each case as with the presentation of PKG in [5], this multiplication is conducted at cloud, which usually has abundant resources. Moreover, we estimate the announcement cost for each user's key-update application in Amazon EC2 cloud surroundings, which is 87ms. Note that such an above your head includes the time strong for transmission and confirmation at Amazon EC2 cloud platform.

TABLE 2 Key-Update in BGK Scheme in Varying System Users

Revocation Ratio	2^5	2^{10}	2^{15}	2^{20}
5%	132.646 ms	3.66 s	1.845 min	0.998 h
15%	212.162 ms	6.607 s	3.283 min	1.879 h
25%	194.135 ms	7.232 s	3.894 min	2.15 h
50%	189.851 ms	9.63 s	3.674 min	1.99 h
75%	133.072 ms	4.186 s	2.488 min	1.207 h

Therefore, in key-update, the totaling demand for updating time constituent and re-issuing clandestine key are nearly the same. As we highlight before, such multiplication is conduct by KU-CSP typically with plentiful resources, which will not gravely affect the effectiveness of our system.

7. CONCLUSION

In this paper, focused on the significant issue of characteristic revocation, we launch outsourcing computation into CBE and proposition a revocable method in which the revocation operations are entrusted to CSP. With the assist of KU-CSP, the proposed method is full-featured: 1) It achieve unvarying efficiency for both totaling at PKG and private key size at consumer 2) consumer needs not to make contact with PKG during key update, in additional words, PKG is acceptable to be offline after allocation the revocation catalog to KU-CSP; 3) No secure channel or user authentication is necessary during key-update between user and KU-CSP. In addition, we judge to realize revocable CBE under a stronger opponent model. We there an superior construction and demonstrate it is secure beneath RDoC model, in which at smallest amount one of the KU-CSPs is unspecified to be honest. Consequently, even if a revoked consumer and either of the KU-CSPs collude, it is powerless to help such user re-obtain his/her decrypt aptitude. Finally, we provide widespread experimental results to display the competence of our proposed structure.

REFERENCES

[1] S. Lodha, W. Aiello, and R. Ostrovsky, "Fast digital identity revocation," in Advances in Cryptology (CRYPTO'98). New York, NY, USA: Springer, 1998, pp. 137–152.

[2] V. Goyal, "Certificate revocation using fine grained certificate space partitioning," in Financial Cryptography and Data Security, R. Dhamija and S. Dietrich, Eds. Berlin, Germany: Springer, 2007, vol. 4886, pp. 247–259.

[3] Gentry , F. Elwailly, and Z. Ramzan, "Quasimodo: Efficient certificate validation and revocation," in Public Key Cryptography (PKC'04), F. Bao, J. Zhou , R. Deng, and Eds. Germany, Berlin,: Springer, 2004, vol. 2947, pp. 375–388.

[4] Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in Advances in Cryptology (CRYPTO '01), J. Kilian, Ed. Berlin, Germany: Springer, 2001, vol. 2139, pp. 213– 229.

[5] A.Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in Proc. 15thACMConf. Comput. Commun. Security (CCS'08), 2008, pp. 417–426.
[6] Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology (EUROCRYPT'05), R. Cramer, Germany, Ed. Berlin: Springer, 2005, vol. 3494, pp. 557–557. R. Canetti, B. Riva, and G. N. Rothblum, "Two 1-round protocols for delegation of computation," ePrint Archive, Rep. 2011/ 518, 2011 Available: http://eprint.iacr.org/2011/518.
[7] U. Feige and J. Kilian, "Making games short (extended abstract)," in Proc. 29th Annu. ACM Symp. Theory Comput. (STOC'97), 1997, pp. 506–516.
[8] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in Proc. 2nd Int. Conf. Theory Cryptography (TCC'05), 2005, pp. 264–282.
[9] R. Canetti, B. Riva, and G. Rothblum, "Two protocols for delegation of computation," in Information Theoretic Security, A. Smith, Ed. Berlin, Germany: Springer, 2012, vol. 7412, pp. 37–61.
[10] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New and secure outsourcing algorithms of modular exponentiations," in Proc. 17th Eur. Symp. Res. Comput. Security (ESORICS), 2012, pp.31–45.
[11] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations," in Proc. 5th ACM Symp. Inf. Comput. Commun. Security (ASIACCS'10), 2010, pp. 48–59.
[12] A.Shamir, "Identity-based cryptosystems and signature schemes," in Advances in Cryptology (CRYPTO), G. Blakley and D. Chaum, Eds. Berlin, Germany: Springer, 1985, vol. 196, pp. 47–53.
[13] A.Cocks, "An identity based encryption scheme based on quadratic residues," in Cryptography and Coding, B. Honary, Ed. Berlin/ Heidelberg: Springer, 2001, vol. 2260, pp. 360–363.
[14] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," in Advances in Cryptology (EUROCRYPT'03), E. Biham, Ed. Berlin, Germany: Springer, 2003, vol. 2656, pp. 646–646.
[15] A.Boneh and X. Boyen, "Efficient selective-id secure identity-based encryption without random oracles," in Advances in Cryptology (EUROCRYPT'04), C. Cachin and J. Camenisch, Eds. Berlin, Germany: Springer, 2004, vol. 3027, pp. 223–238
[16] D. Boneh and X. Boyen, "Secure identity based encryption without random oracles," in Advances in Cryptology (CRYPTO'04), M. Franklin, Ed. Berlin, Germany: Springer, 2004, vol. 3152, pp. 197–206.
[17] Waters, "Efficient identity-based encryption without random oracles," in Advances in Cryptology (EUROCRYPT'05), R. Cramer, Ed. Berlin, Germany: Springer, 2005, vol. 3494, pp. 114–127.
[18] Gentry, "Practical identity-based encryption without random oracles," in Advances in Cryptology (EUROCRYPT'06), S. Vaudenay, Ed. Berlin, Germany: Springer, 2006, vol. 4004, pp. 445–464.
[19] Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in Proc. 40th Annu. ACM Symp. Theory Comput. (STOC'08), 2008, pp. 197– 206.
[20] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (h)ibe in the standard model," in Advances in Cryptology (EUROCRYPT'10), H. Gilbert, Ed. Berlin, Germany: Springer, 2010, vol. 6110, pp. 553–572.