A Novel Dynamic One-Way Accumulator based on Strong RSA Assumption

[*]Aina Sui, Yongbin Wang, Wenlong Fu
[*]*Communication University of China, aina@cuc.edu.cn*

Abstract

Authentication is the core problem in the information communication fields such as cloud computing system, internet of things and other large-scale distributed networks. Currently, the most applied authentication scheme is PKI (Public Key Infrastructure). However, that method is limited in key distribution, identity revocation and cross authentication, thus not quite suitable for the networks with large number of users may participate and leave at any time.

We propose a novel dynamic one-way accumulator that can be used to construct more efficient authentication scheme. The new accumulator not only can verify whether a number belongs to a specific set，but also provides efficient algorithms to add and delete the value on behalf of the user, which is very important to the identity authentication and revocation of network nodes. We formalize its definition and present its implementation scheme, then prove that under the strong RSA assumption, the new scheme can avoid forgery attack, nodes internal cooperation attack and internal node active forgery attack. Furthermore, the new accumulator defines accumulated value as the product of two prime numbers, which can guarantee that the accumulator is collision-free. The new scheme solves the problem of key distribution during constructing the authentication protocol, so it is more attractive for the large-scale dynamic networks with randomly joining and leaving nodes.

Keywords: *Dynamic One-Way Accumulator, Strong RSA Assumption, Zero-Knowledge Proof, Authentication*

1. Introduction

One-way accumulator is an emerging technology closely related to Cryptography and has a wide range of applications in information communication field such as cloud computing system, internet of things and other large-scale distributed networks. At present, there are some research works in combining accumulator with authentication and signature areas [1, 2, 3], but it is still in the early stage. So it is one of the essential researches how to apply one-way accumulator to information security field.

The One-way accumulator is constructed on the base of the quasi-commutative one-way function, which can be used to verify whether the number belongs to a specific set. One-way function is the necessary condition to make one-way accumulator possible. The existence of one-way function and the efforts of looking for a new one-way function has been a hot research of cryptography field.

Currently, the most popular authentication scheme in information communication system is PKI (Public Key Infrastructure). However, this kind of technology is limited in the aspect of key distribution, identity revocation and cross-authentication, thus not quite suitable for the networks with large number of users who may participate and leave at any time.

Benaloh and de Mare [4] constructed the first RSA one-way accumulator, and by using the features that the accumulation value Z is irrelevant to number of the values to be accumulated, they proposed a time-stamping protocol based on one-way accumulator. On the basis of lecture [4], Barić [5] presented the concept of collision-free one-way accumulator through restricting on the accumulated value. Adopting the collision-free one-way accumulator in lecture [4], Sander [6] constructed a new one-way accumulator that does not limit the accumulated value to be prime number, but the accumulator is still collision-free, which is the big difference from the scheme in [5]. Camenisch [7] put forward a definition of dynamic accumulator that provides an algorithm to delete the accumulated number and then update the partial accumulation value. Goodrich, Shi and Ma [8, 9, 10] also presented a kind of dynamic accumulator that can implement the same functions as [7].

[*] This work was supported by Beijing Excellent Talents Training Project and CUC “382” Talents Project.

Combining with the properties of one-way accumulators mentioned above, we construct a novel dynamic one-way accumulator based on the strong RSA assumption, then prove that it is secure against some attacks and analyze that it can be applied to efficient authentication scheme.

In section 2 and 3, we firstly introduce the relative concept of one-way accumulator and strong RSA assumption to prepare for the construction of novel one-way accumulator. In section 4, we propose a new dynamic one-way accumulator and formalize its definition, and then its implementation scheme is presented. In section 5 we prove that the new scheme is safe against some attacks. Finally, in section 6, we give the conclusion.

2. One-Way Accumulator Preliminaries

Definition 2.1 One-way hash function H is the finite set formed by the function $h_l : X_l \times Y_l \to Z_l$. Its components have the following properties:

(1) There is a polynomial P that makes $h_l(x,y)$ is computable within the polynomial time $P(l,|x|,|y|)$ with regard to every integer l and $x \in X_l$, $y \in Y_l$.

(2) There is no probabilistic polynomial-time algorithm like that: To the big enough integer l, when l is given and the integer pair $(x,y) \in X_l \times Y_l$ and $y' \in Y_l$ are randomly chosen from $X_l \times Y_l$ and Y_l, $x' \in X_l$ could be found to make $h_l(x,y) = h_l(x',y')$ with the probability of more than $1/P(l)$.

The above definition shows that if x and y are given, computing $z = h(x,y)$ can be completed within polynomial time. If given only x, y and y', it is hard to find x' within polynomial time to make $h(x,y) = h(x',y')$. This definition does not deny the existence of x that meets the requirements, but indicates that it is very difficult to find that value in whole set X, which means that, to different integer pair (x,y), the collision of computing $h(x,y)$ is very small.

Definition 2.2 If to all $x \in X$ and $y_1, y_2 \in Y$, function $f : X \times Y \to X$ satisfies the equation as follows, the function f is quasi-commutative.

$$f(f(x,y_1),y_2) = f(f(x,y_2),y_1)$$

By using the quasi-commutative one-way hash function, we can verify whether y_i is in the specified set $Y = \{y_i\}$. Specifically, the accumulation value Z, which is the accumulation result of Y, can be computed by function $h \in H$ as follows:

$$Z = h(h(h(h(x,y_1),y_2),y_3),\rightleftharpoons,y_{N-1}),y_N)$$

Here, x is called **seed or initial value**, y_i is called **accumulated value** that is the value to be accumulated, Z is called **accumulation value**. The **partial accumulation value** $Y_i = \{y \in Y \wedge y \neq y_i\}$, i.e. other values except y_i, also can be computed by one-way function as follows:

$$Z_i = h(h(h(h(x,y_1),y_2),y_3),\rightleftharpoons y_{i-1}, y_{i+1},\rightleftharpoons, y_{N-1}),y_N)$$

We can authenticate $y_i \in Y$ through computing Z' by using the following equation:

$$Z' = h(Z_i, y_i)$$

If $Z = Z'$, then $y_i \in Y$.

The above conclusion is established because that if the adversary does not know y_i, he will face the computing difficulty to find y' to make $Z = h(Z_i, y')$. Similarly, if the adversary does not know Z_i, it is hard to find Z_i' to make $Z = h(Z_i', y)$. Therefore, (Z_i, y_i) is the witness of $y_i \in Y$.

The Lecture [11] presented the formal definition of one-way accumulator.

Definition 2.3 One-way accumulator is the function with the following properties:

(1) A set $K=\{k_i \mid i=1,2,\cdots,N\}$ with N keywords and relating with a security parameter s as well as a probabilistic polynomial-time algorithm satisfies following conditions:

$$DOWA\_GenKey_s(K): K \to k, \text{ and}$$

$$P\{DOWA\_GenKey_s(K): K \to k\} = 1/N$$

This algorithm randomly produces a security keyword according to the security parameter, which can be implemented by general random function (denoted as $Random(X)$).

(2) $\forall k \in K$, the suitable input values Y about k can be generated with a probabilistic polynomial-time algorithm, i.e.,

$$DOWA\_GenRep_k(Z_N): Z_N \to Y,\ Y=\{y_i \mid i=1,2,\cdots,N\}$$

(3) The following probabilistic polynomial-time algorithms can be constructed according to the quasi-commutative one-way hash function $h \in H$, $H: X \times Y \to X$:

a. Accumulation Value Algorithm: $Z(Y)=DOWA\_AccTot(Y)$;

b. Partial Accumulation Value Algorithm: $Z_i(Y-\{y_i\})=DOWA\_AccPar(Y-\{y_i\})$;

c. Verification Algorithm: $Z'(Z_i,y_i)=DOWA\_AccAut(Z_i,y_i)$.

Algorithms a and b show that it is easy to construct the input set and output set of this special one-way function. Algorithm c shows that this one-way function has to be quasi-commutative and it should be easy to generate an algorithm for authentication purpose.

3. Strong RSA Assumption Preliminaries

The relative concepts of strong RSA assumption are as follows.

Definition 3.1 If prime number p can be expressed as $p=2p'+1$, where p' is prime number too, and then p is called safe prime number.

Definition 3.2 If integer n can be expressed as $n=p\cdot q$, where p and q are different safe prime numbers, and $|p|=|q|$, then n is called strict integer.

Definition 3.3 Normal RSA Problem is that $\forall y,z \in Z_n, n \in Z_s, \exists x \in Z_n: z=x^y \bmod n$, where Z_n is a set of the integers less than n, Z_s is a set of strict integers.

Definition 3.4 Normal RSA Assumption is that Normal RSA Problem is incomputable with probabilistic polynomial-time algorithm A, i.e.,

$$P\{y,z,n \in Z_n :: \exists x: z=x^y \bmod n\} \leqslant 1/A(n)$$

Definition 3.5 Strong RSA Problem is that $\forall z \in Z_n, n \in Z_s, \exists y>1, x: z=x^y \bmod n$.

Comparing with normal RSA problem, strong RSA problem allows to choose x and y freely, in another word, the adversary can choose not only the root but also the exponent of the function. Therefore, strong RSA assumption also is called flexibility RSA problem. Obviously, strong RSA problem is easier to solve than normal RSA problem where the exponent has been specified. However, strong RSA problem is still hard to solve. And the security of many encryption algorithms and protocols is based on that difficulty condition.

Definition 3.6 Strong RSA Assumption is that Strong RSA Problem is incomputable with any probabilistic polynomial-time algorithm A, i.e.,

$$P\{\forall z \in Z_n, n \in Z_s, \exists y,x: z=x^y \bmod n\} \leqslant 1/A(n)$$

Theorem 3.1 To a strict integer $n=(2p'+1)\cdot(2q'+1)$, when y is relatively prime to $n'=p'\cdot q'$, the function $e_{n,y}(x)=x^y$ is a transformation on the quadratic residue set modulo n.

This theorem guarantees that, to the same root *x*, RSA function will not produce collision even after many times of repeat.

4. A Novel Dynamic One-Way Accumulator based on Strong RSA Assumption

Combining with the dynamic accumulator proposed in lecture [7], we construct a novel one-way accumulator based on strong RSA assumption.

4.1. Formal definition of new dynamic one-way accumulator

According to the formal definition given in [11], we formalize our new accumulator as follows:

(1) $DOWA\_GenKey_s = Random(Z_n)$

Here, $Random(Z_n)$ is a function to randomly choose big integers.

(2) $DOWA\_GenRep(Z_N) := Random(Z_p) \times Random(Z_p) \wedge \gcd(Random(Z_p), \varphi(n)) = 1$

$\wedge Random(Z_p) \neq Random(Z_p) \bmod 8 \wedge Random(Z_p) \neq 1 \bmod 8$

Here, $\varphi(n)$ is Euler function of *n*, $Random(Z_p) \neq Random(Z_p) \bmod 8$ means that any $Random(Z_p)$ that is generated each time cannot be congruent modulo 8, if so, it needs to be regenerated.

(3) $h(x, y) = x^y \bmod n$

Here, *n* is strict integer.

The algorithms of new accumulator are as follows:

(1) Accumulation value algorithm

$$DOWA\_AccTot(Y) := x^{\prod^{y \in Y}} \bmod n$$

To improve the computing efficiency, we can firstly compute $y = \prod_i y_i \bmod \varphi(n)$ for all value y_i to be accumulated, and after only one time of modular exponentiation computing, we can get accumulation value *Z*.

(2) Partial accumulation value algorithm

$$DOWA\_AccPar(Y - \{y_i\}) := x^{\prod^{Y/y_i \in Y}} \bmod n$$

To compute partial accumulation value, we can firstly compute $1/y_i \bmod \varphi(n)$, and after one time of modular exponentiation computing to *Z*, we can get Z_i.

(3) Verification algorithm

$$DOWA\_AccAut(Z_i, y') := Z_i^{y'} \bmod n$$

(4) Adding accumulated value algorithm

$$DOWA\_AccAdd(y) := Z^y \bmod n$$

(5) Updating partial accumulation value algorithm after adding an accumulated value

$$DOWA\_PaccAdd(Z_i, y_j) := Z_i^{y_j} \bmod n$$

If need to add multiple values, the adding operation can be carried out at one time. Specifically, we can firstly compute $y = \prod_j y_j \bmod \varphi(n)$ for all values y_j that need to be added, and then compute a modular exponentiation to *Z* or Z_i, we will get updated *Z* or Z_i.

(6) Deleting accumulated value algorithm

$$DOWA\_AccDel(y) := Z^{1/y \bmod \varphi(n)} \bmod n$$

(7) Partial accumulation value updating algorithm after deleting a value

$$DOWA\_PaccDel(Z_i, y_j) := Z_i^b \cdot Z^a \bmod n$$

Here, *a* and *b* are the parameters that enable $a \cdot y_i + b \cdot y_j = 1$, *Z* is the final accumulation value after deleting some values. This algorithm is correct because:

$$Z_i^b \cdot Z^a \bmod n = Z_i^b \cdot Z_i^{y_i \cdot (1/y_j \bmod \varphi(n)) \cdot a} \bmod n = Z_i^{(1/y_j \bmod \varphi(n)) \cdot (by_j + ay_i)} \bmod n = Z_i^{(1/y_j \bmod \varphi(n))}$$

If need to delete multiple values, the deleting operation also can be carried out at one time. Specifically, we can firstly compute $y = \prod_j y_j \bmod \varphi(n)$ for all values y_j, and then compute a modular exponentiation to *Z* or Z_i, we will get updated *Z* or Z_i.

4.2. Implementation scheme of the novel dynamic one-way accumulator

In order to further analyze the security and efficiency of the application of new one-way accumulator, we will describe the implementation scheme in the form of protocol as follows.

We define a reliable authority node to collect all the values and carry out some relative algorithms and then publish the computing result to those corresponding nodes. We denote this authority node as *CP*.

The system initialization protocol is described as follows:

(1) *CP* chooses a big strict integer $n = p_{cp} \cdot q_{cp}$ where $p_{cp} = 2p'_{cp} + 1$, $q_{cp} = 2q'_{cp} + 1$, p'_{cp} and q'_{cp} are primes. Moreover, *CP* chooses *x* that satisfies $x \in QR_n$ and $x \neq 1$, then chooses a group $G = \langle g \rangle$.

(2) The node *i* chooses a value $y_i \in \{y = p \cdot q \wedge p,q \in primes \wedge p,q \neq 1 \bmod 8 \wedge p \neq q \bmod 8\}$, $i \in \{1,2,\cdots,m\}$ where *m* is the number of nodes. The reason why y_i is restricted like that is described in [12] where a zero-knowledge proof protocol is given, in which a big integer is proved to be the product of only two prime factors. Then node *i* sends $y_i = p_i \cdot q_i$, g^{p_i} and g^{q_i} to *CP*.

(3) *CP* firstly verifies if g^{p_i} and g^{q_i} received from different nodes are repeated, if not, it shows that y_i is relatively prime to another one, if so, *CP* will demand those nodes whose g^{p_i} and g^{q_i} are repeated to choose and send them again. Then *CP* verifies whether $\gcd(y_i, \varphi(n)) = 1$ is established to y_i, if not, *CP* will demand those nodes to choose and send again, if so, *CP* randomly chooses 20 $w_j \in Z_{y_i}^*$, $j = \{1,2,\cdots,20\}$ and publishes them to the node *i*.

(4) After receiving $w_j \in Z_{y_i}^*$, $j = \{1,2,\cdots,20\}$, the node *i* computes the square root r_j of any one of $\pm w_j$ and $\pm 2w_j$, then sends r_j to *CP*.

(5) After receiving r_j, *CP* verifies whether r_j^2 is congruent with one of $\pm w_j$ and $\pm 2w_j$, even if only one is not congruent, *CP* will demand those nodes to choose and send y_i, g^{p_i}, g^{q_i} again, then return to step (3); if all are congruent, *CP* computes the accumulation value $Z = x^{\prod_{i=1}^m y_i} \bmod n$ and the partial accumulation value $Z_i = x^{\prod_{j \neq i, j=1}^m y_j} \bmod n$ for each node. After that, *CP* publishes Z_i respectively to corresponding nodes, and broadcasts *Z* to all the nodes.

The steps (3) to (5) refer to a zero-knowledge proof protocol, which certifies a big integer is product of only two safe primes in [12].

The adding nodes protocol is described as follows:

(1) The node *m*+1 chooses $y_{m+1} \in \{y = p \cdot q \wedge p,q \in primes \wedge p,q \neq 1 \bmod 8 \wedge p \neq q \bmod 8\}$, then sends $y_{m+1} = p_{m+1} \cdot q_{m+1}$, $g^{p_{m+1}}$ and $g^{q_{m+1}}$ to *CP*.

(2) *CP* carries out the steps (3) to (5) of the system initialization protocol, if successful, then continues.

(3) *CP* re-computes the accumulation value $Z' = Z^{y_{m+1}}$, and then broadcasts Z' and y_{m+1} to all nodes.

(4) After receiving Z' and y_{m+1}, each node computes the new partial accumulation value $Z_i' = Z_i^{y_{m+1}}$ respectively. The partial accumulation value of the node *m*+1 is *Z*.

The deleting nodes protocol is described as follows:

(1) Suppose the node to be deleted is *j*. *CP* updates the accumulation value $Z'' = Z^{1/y_j \bmod \varphi(n)}$, and then broadcasts Z'' and y_j to all nodes.

(2) The node *i* firstly uses the extended Euclidean algorithm to compute *a* and *b* to enable $a \cdot y_i + b \cdot y_j = 1$, then updates the partial accumulation value $Z_i'' = Z_i^b \cdot Z''^a \bmod n$. So the equation $Z_i''^{y_i} = Z''$ is established.

5. Security Proof of New Accumulator

5.1. Resistance to forgery attack

Theorem 5.1 Under strong RSA assumption, the new one-way accumulator is secure against the forgery attack.

Proof. Suppose that there is an attacker A, its inputs are *n* and $x \in QR_n$ where $x \neq 1$, its outputs are $y_i \in \{y = p \cdot q \wedge p,q \in primes\}$, $i \in \{1,2,\cdots,m\}$ and x', y', here, $\gcd(y', \prod_{i=0}^{m} y_i) \neq y'$ (To $\gcd(y', \prod_{i=0}^{m} y_i) = y'$, we will discuss it in Theorem 5.2). So $(x')^{y'} = x^{\prod_{i=0}^{m} y_i}$ is established, then A will crack strong RSA assumption.

Let *n* be a strict integer, i.e., $n = (2p_{cp}' + 1) \cdot (2q_{cp}' + 1)$. To crack strong RSA assumption, the values $e \succ 1$ and *m* must be input to make $m^e = x \bmod n$.

The inputs of attacker A are *n* and *x*. Suppose the forgery outputs of attacker A are $(x', y', (y_1, y_2, \cdots, y_m))$ where $y' = p' \cdot q'$. Let $y = \prod_{i=0}^{m} y_i$, then $x'^{y'} = x^y$.

Now compute $d = \gcd(y, y')$:

a. Suppose $d \neq 1$ and *d* is not relatively prime to $\varphi(n)$.

Because *y* is the product of prime numbers and $\varphi(n) = 4p_{cp}' \cdot q_{cp}'$, $\gcd(d, \varphi(n)) = p_{cp}'$ or $\gcd(d, \varphi(n)) = q_{cp}'$. The attacker A knows the factorization of *y*, so p_{cp}' or q_{cp}' can be derived with non-negligible probability. Furthermore, because of $n = (2p_{cp}' + 1) \cdot (2q_{cp}' + 1)$, the attacker A can get the factorization of *n* with non-negligible probability.

b. Suppose $d \neq 1$ and *d* is relatively prime to $\varphi(n)$.

$x^{(y/d)d} = ((x')^{(y'/d)d})$, therefore $x^{y/d} = (x')^{y'/d}$. Set $\alpha = y/d$, $\beta = y'/d$, then $\gcd(\alpha, \beta) = 1$. The attacker can get *a* and *b* through computing $a \cdot \alpha + b \cdot \beta = 1$ by using extended Euclidean algorithm, then inputs them to $(m = x'^a \cdot x^b, e = \beta)$ where $m^e = x \bmod n$ ①, which cracks strong RSA assumption.

The correctness of equation ① is guaranteed by following equation:

$$m^e = (x'^a \cdot x^b)^\beta = x'^{a \cdot (y'/d)} \cdot x^{b \cdot (y'/d)} = x^{(y/d) \cdot a} \cdot x^{b \cdot (y'/d)} = x$$

5.2. Resistance to nodes internal attacks

Theorem 5.2 The protocol based on the new one-way accumulator is secure against nodes internal attacks.

We firstly define two kinds of nodes internal attacks.

Definition 5.1 Suppose there are two authorized nodes. According to the system initialization protocol mentioned above, they get $< y_1, accu_1 >$, $< y_2, accu_2 >$. Here, $y_1 = p_1 \cdot q_1$, $y_2 = p_2 \cdot q_2$, $accu_1^{y_1} = accu_2^{y_2} = Z$ where $accu_1$ and $accu_2$ are their partial accumulation values. These two nodes can construct a new number pair $< y_3, accu_3 >$ through sharing the factorization of y_1 and y_2. Here, $y_3 = p_1 \cdot q_2$, which makes $accu_3^{y_3} = Z$. We call such attack nodes as internal cooperation attack.

Now we will prove that the new one-way accumulator can resist the nodes internal cooperation attack.

Proof. In nodes internal cooperation attack, $\gcd(y', \prod_{i=0}^{m} y_i) = y'$ mentioned in Theorem 5.1 should be considered, which means that y_3 counterfeited by the attacker A is the product of legal prime numbers. If the partial accumulation value $accu_3$ corresponding to y_3 could be derived, the attacker could get new $< y_3, accu_3 >$. Because $accu_3^{y_3} = Z \bmod n$, then $accu_3 = accu_1^{-p_2 q_1}$ or $accu_2^{-q_1 p_2}$. In the case of without $\varphi(n)$, it is impossible to compute $accu_2^{-p_2}$ or $accu_2^{-q_1}$. In another word, the attacker must face RSA problem. Therefore, our scheme is secure against nodes internal cooperation attack.

Definition 5.2 In the system initialization phrase, if a node sends a value like $y = p_1 \cdot p_2 \cdot p_3$ and then receives corresponding partial accumulation value *accu* to enable $accu^y = Z$, the attacker could construct a new number pair $< accu', y' >$ where $accu' = accu^{p_1}$ and $y' = p_2 \cdot p_3$, which makes $accu'^{y'} = Z$ be established. We call such attack as internal node active forgery attack.

Now we will prove that the new one-way accumulator can resist the internal node active forgery attack.

Proof. In system initialization phrase, we refer a zero-knowledge proof protocol to steps (3) to (5). According to the conclusion of [12], if y_i includes more than two safe prime factors, to each $w_j \in Z_{y_i}^*$ given by *CP*, the protocol will be failure with the probability of higher than 1/2. Therefore, when *CP* publishes 20 w_j, the probability of failure will be higher than $1-(1/2)^{20}$.

In addition, the adversary node might submit a value like $y = p_1^i \cdot p_2^j$. We can simply restrict the length of *y* to avoid this attack.

6. Conclusion

We proposed a novel dynamic one-way accumulator and proved its security under strong RSA assumption.

Comparing with the accumulators given in [4] and [5], the new accumulator can efficiently add and delete the value of nodes, which is very important to the participation and revocation of node's identity. While the main difference from the scheme in [7] is that the value chosen by node is the product of two primes, which does not violate the collision-free of one-way accumulator. Furthermore, our new scheme can resist the attack from internal nodes.

During the initialization and adding-nodes phrases, our scheme needs more computation. But these computing processes are mainly concentrated on *CP*.

Moreover, our new scheme can solve the key distribution problem when applying to construct the authentication protocol, therefore is more attractive to applying to a dynamic network with large number of users who may participate and leave at any time.

7. References

[1] Xiaobiao Li, Qiaoyan Wen, "A revocation scheme for the cloud computing environment", In Proceeding of IEEE CCIS (Cloud Computing and Intelligence Systems), pp. 254-258, 2011.
[2] Zhimin Xu, Hao Tian, DongSheng Liu, Jianming Lin, "A ring-signature anonymous authentication method based on one-way accumulator", Second International Conference on Communication Systems, Networks and Applications (ICCSNA 2010), pp. 56-59, 2010.
[3] Dongsheng Liu, Hao Tian, "A New Anonymous Authentication Method Based on One-way Accumulator", JCIT: Journal of Convergence Information Technology, vol. 5, no. 6, pp. 33-39, 2010.
[4] Josh Benaloh, Michael de Mare, "One-Way Accumulators: A Decentralized Alternative to Digital Signatures", Advances in Cryptology - EUROCRYPT'93, vol. 765, pp. 274-285, 1993.
[5] Niko Barić, Birgit Pfitzmann, "Collision-free accumulators and fail-stop signature schemes without trees", Advances in Cryptology - EUROCRYPT'97, LNCS 1233, pp.480-494, 1997.
[6] Tomas Sander, Amnon Ta-Shma, and Moti Yung, "Blind, Auditable Membership Proofs", Financial Cryptography, LNCS 1962, pp. 53-71, 2001.
[7] Jan Camenisch, Anna Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials", Advance in Cryptology CRYPTO 2002, LNCS 2442, pp.61-76, 2002.
[8] Michael T. Goodrich, Roberto Tamassia, Jasminka Hasić, "An efficient dynamic and distributed cryptographic accumulator", Information Security, LNCS 2433, pp.372-388, 2002.
[9] Shi Yuanying, "Secure P2PSIP-based conference system with dynamic scalability", International Conference on Computer Science and Information Technology (ICCSIT 2011), IPCSIT vol. 51, pp. 487-493, 2012.
[10] Chunguang Ma, Jiuru Wang, Peng Wu, Hua Zhang, "Identity Authentication and Key Agreement Integrated Key Management Protocol for Heterogeneous Sensor Networks", Journal of Computers, vol. 7, no. 8, pp. 1847-1852, 2012.
[11] Wan Guogen, Zhou Shijie, Qin Zhiguang, "An Overview of the One-Way Accumulator Technology", Computer Science, vol. 32, no. 8, pp. 57-59, 2005.
[12] Rosario Gennaro, Daniele Micciancio, and Tal Rabin, "An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products", The 5th ACM Conference on Computer and Communications Security, pp.67-72, 1998.