A Survey to Design Privacy Preserving Protocol Using Chaos Cryptography

Hongfeng Zhu, Rui Wang
(Corresponding author: Hongfeng Zhu)

Software College, Shenyang Normal University
No. 253, HuangHe Bei Street, HuangGu District, Shenyang 110034, China
(Email:zhuhongfeng1978@163.com; 670322496@qq.com)

(Received Apr. 13, 2016; revised and accepted Sept. 25, 2016)

Abstract

Chaos theory has been widely studied and adapted in cryptography for achieving some security mechanisms, such as encryption/decryption, key agreement and hash function. The privacy of using chaos cryptography mostly relies on one of or the combination of three mechanisms: (1) Universal construction symmetric cryptography; (2) Efficient type multiplication in finite field; (3) Prudent operation XORed. This paper introduces four efficient generic methods based on three mechanisms for protecting privacy. Our four methods firstly achieve encrypted messages with mutual authentication in one-way flow. In addition, we discuss some methods about using more than two of the methods to form hybrid cases. Finally, implementation analysis, formal proof and efficiency comparison are provided to show that these mechanisms are practical, secure, and privacy preserving.

Keywords: Chaotic Maps; Privacy; Symmetric Cryptography; XORed Operation

1 Introduction

The need of mutual authentication with privacy protection is a fundamental security requirement in computer society. With wide-spread of distributed computer networks, due to most of the applications are client-server architecture, the problem of only legal users have access to use the various remote services has attracted much attention. (see [9, 10, 23, 25]). Combined with the recent trend, chaos theory has widely used to cryptography. Chaotic system has numerous advantages, such as extremely sensitive to initial parameters, unpredictability, boundness, etc. Meanwhile, chaotic sequence generated by chaotic system has the properties of non-periodicity and pseudorandomness.

In 1998, Baptista [1] firstly connects cryptography with chaos theory. As a fundamental cryptographic primitive, key agreement protocol allows two or more parties to agree on shared keys which will be used to protect their later communication. Then, combining chaos theory and key agreement primitive, many authenticated key exchange (AKE) protocols [19] have been proposed. The literature [34] firstly proposed a new one-way authenticated key agreement scheme (OWAKE) based on chaotic maps with multi-server architecture [21, 36]. The OWAKE scheme is widely used to no need for mutual authentication environment on Internet, such as readers-to-journalists model and patient-to-expert model. Using the chaotic maps, the literature [32] firstly proposed a new multiple servers to server architecture (MSTSA) to solve the problems caused by centralized architecture, such as multi-server architecture with the registration center (RC). The core ideas of the proposed scheme are the symmetry (or called peer to peer) in the servers side and the transparency for the clients side. In brief, based on chaotic maps, there were many AKE protocols from functionality aspect, or from efficiency aspect, or from security aspect, or from architecture aspect to improve the AKE protocols. For capturing more functionality, identity-based MSAKA protocols, based on bilinear pairings or elliptic curve cryptosystem (ECC) MSAKA protocols, dynamic identity-based MSAKA protocols and other MSAKA protocols came up recently [6, 7, 19, 27].

Recently, many schemes based on chaos theory are proposed [8, 16, 24, 26, 27, 35]. Compared with the related other schemes, these schemes avoid numerous complex operations. One direction is about static/dynamic Identity (ID) authentication schemes [24, 26] which are based on chaotic maps. But the literature [35] pointed out that Lin's scheme [24] cannot resist dictionary attack, user spoofing attack, denial of service attack and exclusive-or operation with pad operation leaking attack. In 2013, Guo *et al.* [8] proposed a chaotic maps-based key agreement protocol which avoided modular exponential computing and scalar multiplication on elliptic curve. Nowadays, with the fast development of Internet, privacy protection of users is a hot issue. In 2014, Liu *et al.* [16] proposed a multi-function password mutual authentication key agreement scheme with privacy preserving. In 2015, Zhu *et al.* [29, 33] proposed an even more efficient scheme which is only used chaotic maps for mutual authentication instead of encrypting/decrypting messages transferred between user and server, and the users' privacy information is also protected.

There are so many AKE, dynamic ID and others' schemes using chaotic maps so that we cannot introduction by one. In our opinions, these schemes can be refined and further to form some methods or primitive units of protocol. Our goals are to sum up the core contents to serve for constructing diversified schemes with chaotic maps rather than designing a concrete and nonadaptive scheme.

In this paper, we sum up four methods to capturing privacy attribute based on chaotic maps and some expandable forms which can construct many security protocol with privacy preserving. The main contributions of this paper are shown below:

- In Symmetric Encryption Method, we propose a method to design a privacy preserving protocol with mutual authentication in one-way flow using chaotic maps, secure symmetric encryption/decryption and a secure hash function. This kind of method will provide all sensitive information, such as identity, timestamp, value of hash and so on. That will give the attacker the maximum limit of attacks.
- Multiplication in Finite Field method. For achieving both efficiency and privacy preserving attribute, we only use chaotic maps and secure hash function and eliminate secure symmetric encryption/decryption. Our new idea is first to construct a ciphertext based on opposite side's public key and own chosen random number, and then combine both side's public keys with one-time hash value to compute an authenticator for achieving mutual authentication and one-way flow.
- XORed method. For improving efficiency further, we adopt XORed operation instead of Multiplication in Finite Field. This method must pay attention to leak any bits. Aimming at this method, we sum up two rules to resist the potential risk: make the same length about the two sides of $\oplus$ and keep strict secret for the two sides of $\oplus$.
- We sum up many hybrid modes which can design many new protocols with privacy preserving for adapting to changeable environments (see Section 4). We also sum up the security proofs methods and give many literatures to refer (see Section 5).

The rest of the paper is organized as follows: Some preliminaries are given in Section 2. In Section 3, we describe three technologies for privacy with chaotic maps. In Section 4, we discuss some evolved methods. The efficiency analysis of our proposed protocol and methods of provable security are given in Section 5. This paper is finally concluded in Section 6.

2 Chebyshev Chaotic Maps

Let n be an integer and let x be a variable with the interval $[-1,1]$. The Chebyshev polynomial [24] $T_n(x):[-1,1]\rightarrow[-1,1]$ is defined as $T_n(x)=\cos(ncos^{-1}(x))$. Chebyshev polynomial map $T_n:R\rightarrow R$ of degree n is defined using the following recurrent relation:

$$T_n(x)=2xT_{n-1}(x)-T_{n-2}(x),$$

$$\text{where } n\geq 2, T_0(x)=1, \text{ and } T_1(x)=x.$$

The first few Chebyshev polynomials are:

$$T_2(x)=2x^2-1, T_3(x)=4x^3-3x, T_4(x)=8x^4-8x^2+1,...$$

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that

$$T_r(T_s(x))=T_{rs}(x).$$

An immediate consequence of this property is that Chebyshev polynomials commute under composition

$$T_r(T_s(x))=T_s(T_r(x)).$$

Because it is actually proven insecure in literature [2] that Chebyshev polynomials are running the polynomial on decimal number, we adopts the enhanced Chebyshev polynomials to design our protocols. In order to enhance the security, Zhang [28] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty,+\infty)$.

Definition 1. *(Enhanced Chebyshev polynomials)The enhanced Chebyshev maps of degree $n(n\in N)$ are defined as: $T_n(x)=(2xT_{n-1}(x)-T_{n-2}(x))(\mathrm{mod}p)$, where $n\geq 2, x\in(-\infty,+\infty)$, and p is a large prime number. Obviously, $T_{rs}(x)=T_r(T_s(x))=T_s(T_r(x))$.*

Definition 2. *(DLP, Discrete Logarithm Problem)Given an integer a,find the integer r, such that $T_r(x)=a$.*

Definition 3. *(CDH, Computational* $\mathrm{Diffie}--\mathrm{Hellman}$ *Problem)Given an integer x, and the values of $T_r(x),T_s(x)$, what is the value of $T_{rs}(x)=?$.*

It is widely believed that there is no polynomial time algorithm to solve DLP, CDH with a non-negligible probability.

3 Three Technologies for Privacy with Chaotic Maps

In this section, we present three general technologies for privacy with chaotic maps, including the methods, extended methods and some deductive ways. Simply speaking, for all the nodes $node_i(1\leq i\leq n)$, their public keys are $(x,T_{K_i}(x))(1\leq i\leq n)$ and the corresponding secret keys are $K_i(1\leq i\leq n)$. And without loss of generality, we assume a user or a server $node_1$ is the sender, and the others $node_i (2 \leq i \leq n)$ are the receivers. Due to space limitation in this paper, we are not able to discuss the details about how to distribute the public-private key pairs of the users. Some notations hereafter are shown in Table 1.

Table 1: Notations

Symbol	Definition
ID_i	The identity of nodes
a, b, r_i	Nonces
$(x, T_{K_i}(x))$	Public key of $node_i$ based on Chebyshev chaotic maps
K_i	Secret key of $user_i$ based on Chebyshev chaotic maps
$E_K()/D_K()$	A pair of secure symmetric encryption/decryption functions with the key K
M, H, T, F	Plaintext, a secure chaotic maps-based hash function, Timestamp and Pseudo-random function
$\|\|$	Concatenation operation
...	Some other information

3.1 Privacy Protection Mechanism I: Symmetric Encryption

The method of Privacy Protection Mechanism I is shown in Figure 1.

Step 1. When $node_1$ wants to send an encrypted messages with mutual authenticator in one-way flow, it chooses a random a and the public key of the peer to compute:$T_a(x)$, $K_{1i} = T_a T_{K_i}(x)$, $H_i(M_i||T||...)$, $C_i = E_{K_{1i}}(M_i||H_i||T||T_{K_1}T_{K_i}(x)...)$. Then $node_1$ sends $\{T_a(x), C_i\}$ to the others peers.

Step 2. Upon receiving $\{T_a(x), C_i\}$ from the $node_1$, the $node_i$ computes $K_{1i} = T_{K_i}T_a(x)$ and uses K_{1i} to decrypt C_i. Then, $node_i$ can get the $M_i||H_i||T||T_{K_1}T_{K_i}(x)...$ and do the following tasks:

1) To verify the timestamp T;

2) If timestamp authentication passed, $node_i$ computes hash value and compare it with the H_i;

3) If the hash value authentication passed, $node_i$ computes $T_{K_i}T_{K_1}(x)$ and compare with $T_{K_1}T_{K_i}(x)$. If they are equals, that means $node_1$ is the real $node_1$.

Finally, $node_i$ will accept the messages in this flow.

Remark 1. *1) The timestamp T is encrypted during all the communication process which can resist the common interrupt attack. If the timestamp T is plaintext on the public channel, the adversary only makes the timestamp smaller simply so that let the other peer authentication fail.*

2) The information $T_{K_1}T_{K_i}(x)$ is the authenticator which can let the receiver authenticate the sender while no need for another exchange, in other words, mutual authentication can be achieved in one communication which is an efficient method. Although the $T_{K_1}T_{K_i}(x)$ is invariant, it is encrypted and the C_i is always changing. So our Privacy Protection Mechanism I is efficient and secure.

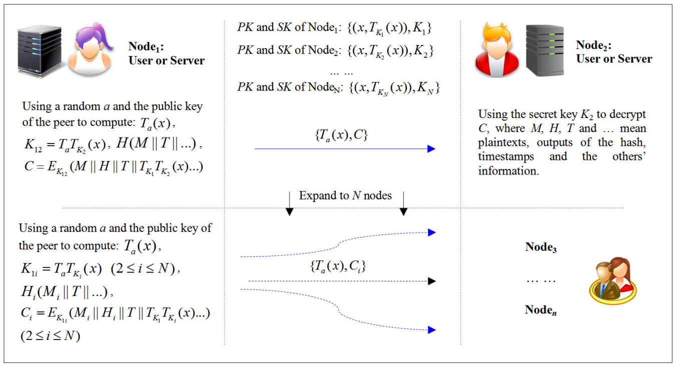

Node₁:
User or Server
PK and SK of Node₁: {(x,T_{K₁}(x)),K₁}
PK and SK of Node₂: {(x,T_{K₂}(x)),K₂}
… …
PK and SK of Node_N: {(x,T_{K_N}(x)),K_N}
Node₂:
User or Server
Using a random a and the public key of the peer to compute: T_a(x),
K₁₂ = T_aT_{K₂}(x), H(M || T || ...),
C = E_{K₁₂}(M || H || T || T_{K₁}T_{K₂}(x)...)
{T_a(x),C}
Using the secret key K₂ to decrypt C, where M, H, T and … mean plaintexts, outputs of the hash, timestamps and the others' information.
Expand to N nodes
Using a random a and the public key of the peer to compute: T_a(x),
K_{1i} = T_aT_{K_i}(x) (2 ≤ i ≤ N),
H_i(M_i || T || ...),
C_i = E_{K_{1i}}(M_i || H_i || T || T_{K₁}T_{K_i}(x)...)
(2 ≤ i ≤ N)
{T_a(x),C_i}
Node₃
… …
Node_n

Figure 1: The method of privacy protection mechanism I: Symmetric encryption

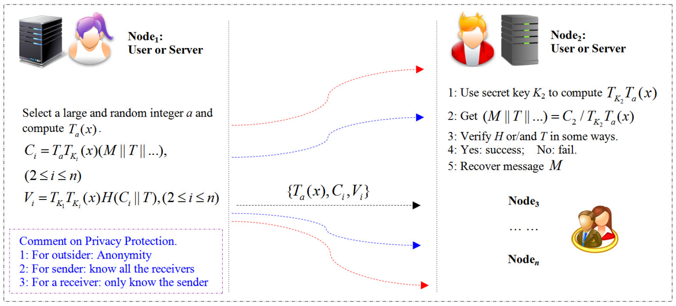

Node₁:
User or Server
Node₂:
User or Server
Select a large and random integer a and compute T_a(x).
C_i = T_aT_{K_i}(x)(M || T || ...),
(2 ≤ i ≤ n)
V_i = T_{K₁}T_{K_i}(x)H(C_i || T),(2 ≤ i ≤ n)
Comment on Privacy Protection.
1: For outsider: Anonymity
2: For sender: know all the receivers
3: For a receiver: only know the sender
{T_a(x),C_i,V_i}
1: Use secret key K₂ to compute T_{K₂}T_a(x)
2: Get (M || T || ...) = C₂ / T_{K₂}T_a(x)
3: Verify H or/and T in some ways.
4: Yes: success; No: fail.
5: Recover message M
Node₃
… …
Node_n

Figure 2: The 1-to-N method of privacy protection mechanism II: Multiplication in finite field

3.2 Privacy Protection Mechanism II: Multiplication in Finite Field

The 1-to-N method of Privacy Protection Mechanism II is shown in Figure 2.

Step 1. When $node_1$ wants to send an encrypted messages with mutual authenticator in one-way flow, it chooses a random a and the public key of the peer to compute:$T_a(x)$, $C_i = T_a T_{K_i}(x)(M||T||...)$, $V_i = T_{K_1}T_{K_i}(x)H(C_i||T), (2 \leq i \leq n)$. Then $node_1$ sends $\{T_a(x), C_i, V_i\}$ to the others peers.

Step 2. Upon receiving $\{T_a(x), C_i, V_i\}$ from the $node_1$, the $node_i$ computes $T_{K_i}T_a(x)$ and uses it to decrypt C_i. Then, $node_i$ can get the $(M||T||...) = C_2/T_{K_2}T_a(x)$ and do the following tasks:

1) To verify the timestamp T;
2) If timestamp authentication passed, $node_i$ computes hash value $V_i^{'} = T_{K_i}T_{K_1}(x)H(C_i||T)$ and compare it with the V_i;
3) If the authentication passed, that means $node_1$ is the real $node_1$.

Finally, $node_i$ will accept the messages in this flow.

Remark 2:

1) The timestamp T is encrypted during all the communication process which can resist the common interrupt attack. If the timestamp T is plaintext on the public channel, the adversary only makes the timestamp smaller simply so that let the other peer authentication fail.
2) The authenticator $V_i = T_{K_1}T_{K_i}(x)H(C_i||T)$ is always changing because T and C_i are different for each interaction and each node. So the 1-to-N Method of Privacy Protection Mechanism II can also achieve mutual authentication in one communication.

Step 1. When the $node_1$ wants to send the same message m to the $node_i (2 \leq i \leq n)$, it chooses two large and random integers a and b. Next, the $node_1$ computes $T_a(x)$,$T_b(x)$,$C_i = T_bT_{K_i}(x)ID_1, (2 \leq i \leq n)$, $V_i = T_a(x)T_{K_1}T_{K_i}(x), (2 \leq i \leq n)$, $W = T_a(x)m$ and $F_i = F_{T_a(x)}(C_i||V_i||W), (2 \leq i \leq n)$. Finally, U_1 sends $\{T_b(x), C_i, V_i, W, F_i\}$ to the users $U_i (2 \leq i \leq n)$.

Step 2.

1) Upon receiving $\{T_b(x), C_i, V_i, W, F_i\}$ from the sender, firstly, any node can recover the identity of the sender by using secret key K_i to compute $T_{K_i}T_b(x)$ and get $ID_1 = C_i/T_{K_i}T_b(x)$.
2) Based the sender's identity ID_1, $node_i$ can get the public key $T_{K_1}(x)$ and compute $T_{K_i}T_{K_1}(x)$ for getting $T_a(x) = V_i/T_{K_i}T_{K_1}(x)$. This step is also authenticating the sender, if the sender is the sender In the last step, any user can recover the right message, if not, the recovered message will not be the plaintext.
3) U_i authenticates the message integrity $F_{T_a(x)}(C_2||V_2||W) = F_2$?. If yes, the ciphertext is valid. Otherwise, the ciphertext is invalid or has been damaged during transmission.
4) Finally, based on their secret key Ki, any node in the group can recover the message $m = \frac{W}{V_i/T_{K_i}T_{K_1}(x)}$.

Remark 3: In this method, we use pseudo-random function instead of hash function for achieving in the standard model. You can also use hash function for getting high-efficiency in the random oracle.

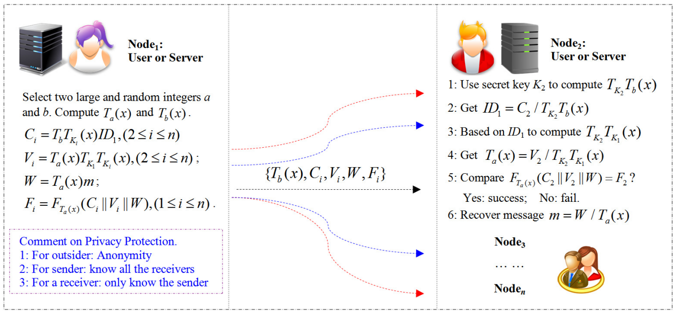

Node$_1$:
User or Server
Select two large and random integers a and b. Compute $T_a(x)$ and $T_b(x)$.
$C_i = T_bT_{K_i}(x)ID_1, (2 \leq i \leq n)$
$V_i = T_a(x)T_{K_1}T_{K_i}(x), (2 \leq i \leq n)$;
$W = T_a(x)m$;
$F_i = F_{T_a(x)}(C_i \| V_i \| W), (1 \leq i \leq n)$.
Comment on Privacy Protection.
1: For outsider: Anonymity
2: For sender: know all the receivers
3: For a receiver: only know the sender
$\{T_b(x), C_i, V_i, W, F_i\}$
Node$_2$:
User or Server
1: Use secret key K_2 to compute $T_{K_2}T_b(x)$
2: Get $ID_1 = C_2 / T_{K_2}T_b(x)$
3: Based on ID_1 to compute $T_{K_2}T_{K_1}(x)$
4: Get $T_a(x) = V_2 / T_{K_2}T_{K_1}(x)$
5: Compare $F_{T_a(x)}(C_2 \| V_2 \| W) = F_2$?
Yes: success; No: fail.
6: Recover message $m = W / T_a(x)$
Node$_3$
… …
Node$_n$

Figure 3: The chained method of privacy protection mechanism II: Multiplication in finite field

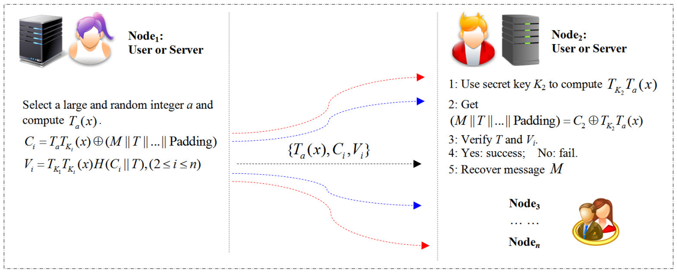

Node$_1$:
User or Server
Select a large and random integer a and compute $T_a(x)$.
$C_i = T_aT_{K_i}(x) \oplus (M \| T \| ... \| \text{Padding})$
$V_i = T_{K_1}T_{K_i}(x)H(C_i \| T), (2 \leq i \leq n)$
$\{T_a(x), C_i, V_i\}$
Node$_2$:
User or Server
1: Use secret key K_2 to compute $T_{K_2}T_a(x)$
2: Get
$(M \| T \| ... \| \text{Padding}) = C_2 \oplus T_{K_2}T_a(x)$
3: Verify T and V_i.
4: Yes: success; No: fail.
5: Recover message M
Node$_3$
… …
Node$_n$

Figure 4: The method of privacy protection mechanism III: XORed operation

3.3 Privacy Protection Mechanism III: XORed Operation

The method of Privacy Protection Mechanism III is shown in Figure 4.

Step 1. When $node_1$ wants to send an encrypted messages with mutual authenticator in one-way flow, it chooses a random a and the public key of the peer to compute:$T_a(x)$, $C_i = T_aT_{K_i}(x) \oplus (M||T||...||\text{Padding})$,$V_i = T_{K_1}T_{K_i}(x)H(C_i||T), (2 \leq i \leq n)$. Then $node_1$ sends $\{T_a(x), C_i, V_i\}$ to the others peers.

Step 2. Upon receiving $\{T_a(x), C_i, V_i\}$ from the $node_1$, the $node_i$ computes $T_{K_i}T_a(x)$ and uses it to decrypt C_i. Then, $node_i$ can get the $(M||T||...||\text{Padding}) = C_2 \oplus T_{K_2}T_a(x)$ and do the following tasks:

1) To verify the timestamp T;
2) If timestamp authentication passed, $node_i$ computes hash value $V_i^{'} = T_{K_i}T_{K_1}(x)H(C_i||T)$ and compare it with the V_i;
3) If the authentication passed, that means node1 is the real $node_1$.

Finally, $node_i$ will accept the messages in this flow.

Remark 4:

1) In this method, XORed operation may be lead some potential risk. The messages $(T_aT_{K_i}(x)$

and $(M||T||...||\text{Padding})$) of both sides for the XORed operation cannot leak any bits. Otherwise, the $T_aT_{K_i}(x)$ may leak the same bits. For example, in $C_i = T_aT_{K_i}(x) \oplus (M||T||...||\text{Padding})$, the M may include the identity of a node. An adversary get the identity in some way, so the correspondent bits of the $T_aT_{K_i}(x)$ will be leak. The concrete details can be found in [12].

2) In some way, the padding may also leak some bits of the end of the $T_aT_{K_i}(x)$.Because the padding mode is usually public. The best method to make Privacy Protection Mechanism III secure are no padding (cut out part of $T_aT_{K_i}(x)$ to make the same length with $(M||T||...)$), and at the same time don't leak any message of the $(M||T||...)$.

4 The Discussions about Hybrid or Evolved Schemes

For simplicity, we make four methods (Symmetric Encryption, 1-to-N Method of Multiplication in Finite Field, Chained Method of Multiplication in Finite Field and XORed Operation) expressed as Algorithm 1, Algorithm 2, Algorithm 3, Algorithm 4.

4.1 Composable Mode

The four methods can be combined to achieve diversified security protocol which may be more efficient or more functions. We set some examples as follows:

Algorithm 1 + Algorithm 2: $C_i = E_{T_aT_{K_i}(x)}(M_i||T...)$, $V_i = T_{K_1}T_{K_i}(x)H(C_i||T)$, $(2 \leq i \leq n)$. This is a kind of composable mode uses Symmetric Encryption to protect messages and privacy and adopts multiplication to provide the mutual authentication.

Algorithm 1 + Algorithm 4: $C_i = E_{T_aT_{K_i}(x)}(M_i||T...) \oplus T_{K_1}T_{K_i}(x), V_i = H(C_i||T), (2 \leq i \leq n)$. This is a kind of composable mode can improve the security level, and the authenticator is a simple value of hash function.

4.2 Modified Mode

The four methods can be modified to diversified security protocol which may be more efficient or more functions. We set some examples as follows:

1) Modified the authenticator: For Algorithm 4, we can make the authenticator $V_i = T_{K_1}T_{K_i}(x)H(C_i||T)$ become $V_i = H(T_{K_1}T_{K_i}(x)) \oplus H(C_i||T)$. This modification can get more efficient. Because the modified edition uses one XORed and one hash function instead of one multiplication, which is more efficient than before (see Section 5.1).

2) Modified the communication round number: For Algorithm 1, Algorithm 2, Algorithm 4, you can use multiple communication to improve computational efficiency and eliminate the $T_{K_1}T_{K_i}(x)$ or others information.

3) For Algorithm 3, the encrypted message $W = T_a(x)m$ can be modified into $W = T_a(x) \oplus m$. This will be more efficient because compared with Multiplication, XORed operation can be ignored. Both sides of XORed, $T_a(x)$ and m cannot be leak any bits (see Remark 4).

4.3 Extended Mode (Three Nodes and Key Agreement)

The four methods can be extended to three-party or N-party schemes. We can use Algorithm 1 or Algorithm 2 or Algorithm 4 repeatedly can achieve three-party key exchange/key distribution or encrypted messages with mutual authentication.

About N-party group key agreement schemes, we can use Algorithm 1 or Algorithm 2 or Algorithm 4 repeatedly and refer the literature [30] to divide N-party schemes into two phases: the first phase is two-party exchange phase, the second phase is group key generated phase. From Table 2, we can see the general structure about N-party group key agreement schemes based on Algorithm 1. And the examples for Algorithm 2 or Algorithm 4 we just omit for simplicity.

4.4 Functional Mode (Multiple Keys Agreement)

In this mode, we only provide multiple Keys agreement for two-party. Just like the literature [31], we can also use Algorithm 1 or Algorithm 2 or Algorithm 4 repeatedly to achieve two-party get multiple keys in an agreement instance. The multiple keys scheme can be divided into two phases: the first phase is authenticated and transmit secret shadows phase, the second phase is Non-interactive key establishment with privacy preserving phase. From Table 3, we can see the general structure about multiple keys agreement schemes based on Algorithm 2. And the examples for Algorithm 1 or Algorithm 4 we just omit for simplicity.

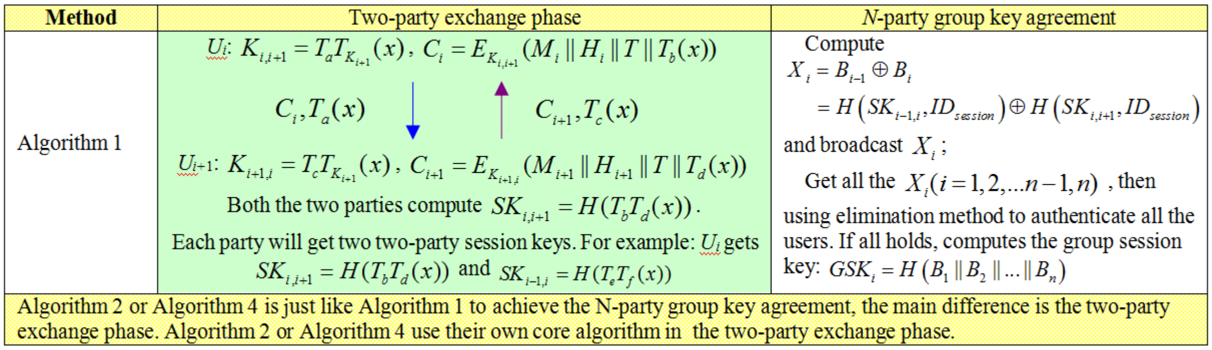
In brief, we can design many hybrid or evolved schemes from our four basic methods. We can't list all the schemes, even the listed schemes, we just describe the main structure and omit the details.

5 Features Comparison

5.1 The Sum up about Our Four Methods Efficiency

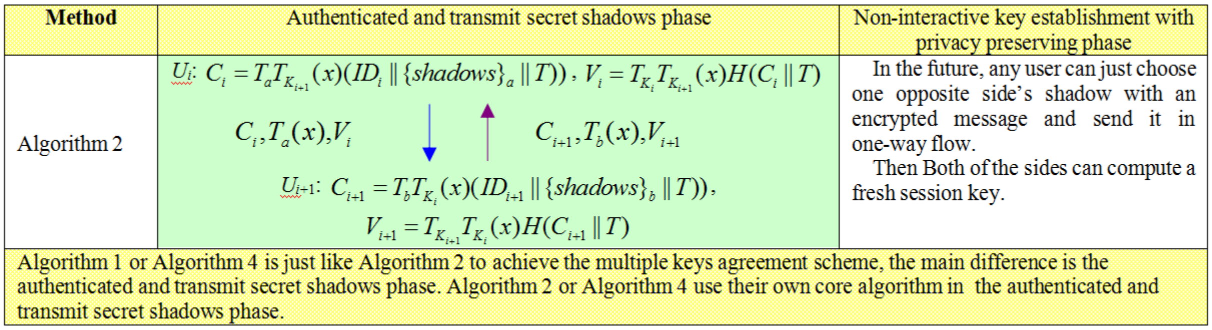
Because the paper given the methods which are the universal formulations can be used directly by any specific

Table 2: N-party group key agreement scheme based on Algorithm 1



Method	Two-party exchange phase	*N*-party group key agreement
Algorithm 1	U_i: $K_{i,i+1}=T_aT_{K_{i+1}}(x)$, $C_i=E_{K_{i,i+1}}(M_i \parallel H_i \parallel T \parallel T_b(x))$ $C_i, T_a(x)$ ↓ ↑ $C_{i+1}, T_c(x)$ U_{i+1}: $K_{i+1,i}=T_cT_{K_{i+1}}(x)$, $C_{i+1}=E_{K_{i+1,i}}(M_{i+1} \parallel H_{i+1} \parallel T \parallel T_d(x))$ Both the two parties compute $SK_{i,i+1}=H(T_bT_d(x))$. Each party will get two two-party session keys. For example: U_i gets $SK_{i,i+1}=H(T_bT_d(x))$ and $SK_{i-1,i}=H(T_eT_f(x))$	Compute $X_i=B_{i-1}\oplus B_i$ $=H\left(SK_{i-1,i},ID_{session}\right)\oplus H\left(SK_{i,i+1},ID_{session}\right)$ and broadcast X_i; Get all the $X_i(i=1,2,\ldots n-1,n)$, then using elimination method to authenticate all the users. If all holds, computes the group session key: $GSK_i=H\left(B_1 \parallel B_2 \parallel \ldots \parallel B_n\right)$
Algorithm 2 or Algorithm 4 is just like Algorithm 1 to achieve the N-party group key agreement, the main difference is the two-party exchange phase. Algorithm 2 or Algorithm 4 use their own core algorithm in the two-party exchange phase.		

Table 3: Multiple keys agreement scheme based on Algorithm 2



Method	Authenticated and transmit secret shadows phase	Non-interactive key establishment with privacy preserving phase
Algorithm 2	U_i: $C_i=T_aT_{K_{i+1}}(x)(ID_i \parallel \{shadows\}_a \parallel T))$, $V_i=T_{K_i}T_{K_{i+1}}(x)H(C_i \parallel T)$ $C_i, T_a(x), V_i$ ↓ ↑ $C_{i+1}, T_b(x), V_{i+1}$ U_{i+1}: $C_{i+1}=T_bT_{K_i}(x)(ID_{i+1} \parallel \{shadows\}_b \parallel T))$, $V_{i+1}=T_{K_{i+1}}T_{K_i}(x)H(C_{i+1} \parallel T)$	In the future, any user can just choose one opposite side's shadow with an encrypted message and send it in one-way flow. Then Both of the sides can compute a fresh session key.
Algorithm 1 or Algorithm 4 is just like Algorithm 2 to achieve the multiple keys agreement scheme, the main difference is the authenticated and transmit secret shadows phase. Algorithm 2 or Algorithm 4 use their own core algorithm in the authenticated and transmit secret shadows phase.		

scheme, not the concrete schemes, we can not give any comparisons about efficiency with some related literatures. From Table 4, we can conclude that our four methods have high-efficient property.

All the computational time of some common algorithms can be summarized as follows [11, 12]:

T_{Mac}: The time for executing a strongly unforgeable MAC algorithm computation;

T_F: The time for executing a secure pseudorandom function computation$T_F \approx 4T_{MUL}$;

$T_{MUL/EXP/INV}$: The time for computing a modular multiplication/exponentiation/inversion $T_{EXP} \approx 240T_{MUL}$, $T_{INV} \approx 10T_{MUL}$);

T_H: The time for computing a one-way hash function computation ($T_H \approx 4T_{MUL}$);

$T_{EM/EA}$: The time for computing a point multiplication/addition operation over an elliptic curve($T_{EM} \approx 29T_{MUL}$, $T_{EA} \approx 0.12T_{MUL}$);

$T_{SE/SD}$: The time for performing a symmetric encryption/decryption algorithm computation ($T_{SE} \approx T_H \approx 4T_{MUL}$, $T_{SD} \approx T_H \approx 4T_{MUL}$);

T_C: The time for executing enhanced Chebyshev polynomial ($T_C \approx 60T_H$); T_{XOR}: The computational cost of XOR operation could be ignored when compared with other operations.

Based on the Table 4, we can conclude that the efficiency ranking is: $\mathbf{III > II : ChainedMethod >> II : 1-to-NMethod > I}$.

5.2 The Sum up about Our Four Methods Security

For simplicity, we make four methods (Symmetric Encryption, 1-to-N Method of Multiplication in Finite Field, Chained Method of Multiplication in Finite Field and XORed Operation) expressed as Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4.

1) The standard model: This model must pay attention to not use hash function. The Algorithm 1 can be proved just like the literature [17]. The Algorithm 2 and Algorithm 3 can be proved just like the literature [14]. The literature [14] is also as the reference for Algorithm 4, but what calls for special attention is that the both sides for the XORed operation must not leak any bit information.

2) The random oracle model: This model can use hash function for achieving high efficiency. In this model, you can firstly define some roles, such as the users, the servers and the adversary. Next, this model should let the adversary make some following oracle queries, such as send, reveal, corrupt, test and so on. Finally, this model should define some secure goals,

Table 4: Our proposed scheme efficiency

Privacy Protection Method	Main features	Party	Efficiency(for one node)
I: Symmetric Encryption	Only one-way Flow can achieve Encrypted message Delivered with mutual Authentication	Sender	$(n-1)(T_H+2T_C+T_{SE/SD}) \approx (n-1)488T_{MUL}$
		Receivers(n-1)	$T_H+2T_C+T_{SE/SD} \approx 488T_{MUL}$
II: Multiplication in Finite Field 1-to-N Method		Sender	$(n-1)(T_H+2T_C+2T_{MUL}) \approx (n-1)486T_{MUL}$
		Receivers(n-1)	$T_H+2T_C+2T_{MUL} \approx 486T_{MUL}$
II: Multiplication in Finite FieldChained Method		Sender	$nT_F+2nT_C+(2n+1)T_{MUL} \approx (126n+1)T_{MUL}$
		Receivers(n-1)	$T_F+2T_C+3T_{MUL} \approx 127T_{MUL}$
III: XORed Operation		Sender	$(n-1)(T_H+2T_C+T_{MUL}+T_{XOR}) \approx (n-1)125T_{MUL}$
		Receivers(n-1)	$T_H+2T_C+T_{MUL}+T_{XOR} \approx 125T_{MUL}$

T_H: Time for Hash operation.
T_F: Time for Pseudo-random function.
$T_{SE/SD}$: Time for Symmetric parametric function.
T_{MUL}: Time for Integer multiplication operation in the field.
T_{XOR}: Time for XOR operation.The computational cost of XOR operation could be ignored.
T_C: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial using the algorithm in literature [18].

for example, matching conversations, secure mutual authentication and secure key exchange. The literature [3, 4] can be as a reference for Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4.

3) The logical analysis method: BAN-like logics [5] is one of the main tools for analysis cryptographic protocols in recent years, the limitations of BAN logic are analyzed and illustrated with examples, and then the features of the extended BAN-like logics and their common defects are studied.

6 Conclusion

In this paper, we propose MRCM, a novel scheme towards building a PKC-based scheme for a sender sending only one encrypted message with some authentication information to multi-receiver, and at the same time, achieving privacy protection. The core idea we have followed is that the most existing multi-receiver schemes are bilinear pairing-based, for improving the efficiency, should be exploited to securely change another efficient cryptosystem, such as, chaotic maps in this paper. Since the hash function is not used, and chaotic maps is adopted to a new encrypted algorithm without using symmetrical encryption, the proposed solution offers significant advantages (the standard model and high-efficiency) with respect to a traditional multi-receiver protocols. Compared with the related works, our MRCM scheme is not the trade off between security and efficiency, but is comprehensively improved scheme.

Acknowledgments

This work is supported by the Liaoning Provincial Natural Science Foundation of China (Grant No. 201602680).

References

[1] M. S. Baptista, "Cryptography with chaos," *Physics Letters A*, vol. 240, no. 1, pp. 50–54, 1998.

[2] P. Bergamo, P. DArco, A. D. Santis, L. Kocarev, "Security of public-key cryptosystems based on chebyshev polynomials," *IEEE Transactions on Circuits and Systems. Part I: Regular Papers*, vol. 52, no. 7, pp. 1382–1393, 2005.

[3] M. Bellare, P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology (CRYPTO'93)*, pp. 232–249, 1993.

[4] M. Bellare, D. Pointcheval, P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology (EUROCRYPT'00)*, pp. 139–155, 2000.

[5] M. Burrows, M. Abadi, R. Needham, "A logic of authentication,"*ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[6] R. Canetti, H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology (EUROCRYPT'01)*, pp. 453–474, 2001.

[7] K. Charif, A. Drissi, Z. El A. Guennoun, "A pseudo random number generator based on chaotic billiards," *International Journal of Network Security*, vol. 19, No. 3, pp. 479-486, 2017.

[8] C. Guo, C. C. Chang, "Chaotic maps-based password-authenticated key agreement using smart

Table 5: Descriptions the model of Canetti and Krawczyk

Symbol	Definition
parties $P_1, ...P_n$	Modeled by probabilistic Turing machines.
Adversary Λ	A probabilistic Turing machine which controls all communication, with the exception that the adversary cannot inject or modify messages (except for messages from corrupted parties or sessions), and any message may be delivered at most once.
Send query	The adversary can control over Parties outgoing messages via the Send query. Parties can be activated by the adversary launching Send queries.
Corrupt(P_i)	This query allows the adversary to take over the party P_i, including long-lived keys and any session-specific information in P_i memory. A corrupted party produces no further output.
Test(s)	This query allows the adversary to be issued at any stage to a completed, fresh, unexpired session s. A bit b is then picked randomly. If b = 0, the test oracle reveals the session key, and if b = 1, it generates a random value in the key space. The adversary can then continue to issue queries as desired, with the exception that it cannot expose the test session. At any point, the adversary can try to guess b. Let $GoodGuess^{\Lambda}(k)$ be the event that the adversary Λ correctly guesses b, and we define the advantage of adversary Λ as $Advantage^{\Lambda}(k) = \max\{0, \lvert \Pr[GoodGuess^{\Lambda}(k)] - \frac{1}{2} \rvert\}$, where k is a security parameter.

cards,"*Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 6, pp. 1433–1440, 2013.

[9] X. Y. Huang, Y. Xiang, A. Chonka, J. Y. Zhou, R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems,"*IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1390–1397, 2011.

[10] X. Y. Huang, Y. Xiang, E. Bertino, J. Y. Zhou, L.Xu, "Robust multi-factor authentication for fragile communications,"*IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2014.

[11] N. Koblitz, A. Menezes, S. Vanstone, "The state of elliptic curve cryptography,"*Designs, Codes and Cryptography*, vol. 19, no. 2–3, pp. 173–193, 2000.

[12] L. Kocarev, S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*, Springer, 2011.

[13] C. C. Lee, C. T. Li, C. W. Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps,"*Nonlinear Dynamics*, vol. 73, no. 1–2, pp. 125–132. 2013.

[14] H. Lai, M. A. Orgun, J. H. Xiao, J. Pieprzyk, L. Y. Xue, Y. X. Yang, "Provably secure three-party key agreement protocol using Chebyshev chaotic maps in the standard model,"*Nonlinear Dynamics*, vol. 77, no. 4, pp. 1427–1439, 2014.

[15] H. Y. Lin, "Chaotic map based mobile dynamic ID authenticated key agreement scheme,"*Wireless Personal Communications*, vol. 78, no. 2, pp. 1487–1494, 2014.

[16] T. H. Liu, Q. Wang, H. F. Zhu, "A multifunction password mutual authentication key agreement scheme with privacy preserving,"*Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 2, pp. 165–178, 2014.

[17] C. C. Lee, C. T. Li, C. W. Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Nonlinear Dynamics*, vol. 73, no. 1–2, pp. 125–132, 2013.

[18] P. R. Newswire, *Ticketmaster Launches New, Innovative CAPTCHA Solutions, Making The Fan Experience Better*, Jan, 2013. (`http://www.prnewswire.com/news-releases/ticketmaster-launches-new-innovative-captcha-solutions-making-the-fan-experience-better-189000181.htm`)

[19] R. S. Pippal, C. D. Jaidhar, S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[20] A. Stolbunov, "Reductionist security arguments for public-key cryptographic schemes based on group action," in *The Norwegian Information Security Conference (NISK'09)*, pp. 97–109, 2009.

[21] Y. Sun, H. Zhu, and X. Feng, "A novel and concise multi-receiver protocol based on chaotic maps with privacy protection," *International Journal of Network Security*, vol. 19, No. 3, pp. 371-382, 2017.

[22] B. Wang, M. Ma, "A smart card based efficient and secured multi-server authentication scheme," *Wireless Personal Communications*, vol. 68, no. 2, pp. 361–378, 2013.

[23] G. Wang, J. Yu, Q. Xie, "Security analysis of a single sign on mechanism for distributed computer networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 294–302, 2013.

[24] X. Wang, J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlin-*

ear Science Numerical Simulation, vol. 15, no. 12, pp. 4052–4057, 2010.

[25] H. Wang, H. Zhang, J. Li and C. Xu, "A(3,3) visual cryptography scheme for authentication", *Journal of Shenyang Normal University (Natural Science Edition)*, vol. 31, no. 101, pp. 397–400, 2013.

[26] Q. Xie, J. M. Zhao, X. Y. Yu, "Chaotic maps-based three-party password-authenticated key agreement scheme," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1021–1027, 2013.

[27] E. J. Yoon, K. Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *The Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2013.

[28] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos Solitons Fractals*, vol. 37, no. 3, pp. 669–674, 2008.

[29] H. Zhu, Y. Zhang, and Y. Zhang, " A provably password authenticated key exchange scheme based on chaotic maps in different realm," *International Journal of Network Security*, vol. 18, No. 4, pp. 688-698, 2016.

[30] H. Zhu, "Secure chaotic maps-based group key agreement scheme with Privacy Preserving," *International Journal of Network Security*, vol. 18, No. 6, pp. 1001-1009, 2016.

[31] H. F. Zhu, "Sustained and authenticated of a universal construction for multiple key agreement based on chaotic maps with privacy preserving," *Journal of Internet Technology*, vol. 17, no. 5, pp. 1–10, 2015.

[32] H. F. Zhu, M. Jiang, X. Hao, Y. Zhang, "Robust biometrics-based key agreement scheme with smart cards towards a new architecture," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 81–98, 2015.

[33] H. Zhu, Y. Zhang, "An improved two-party password-authenticated key agreement protocol with privacy protection based on chaotic maps," *International Journal of Network Security*, vol. 19, No. 4, pp. 487-497, 2017.

[34] H. Zhu, Y. Zhang, Y. Xia, and H. Li, "Password-authenticated key exchange scheme using chaotic maps towards a new architecture in standard model," *International Journal of Network Security*, vol. 18, No. 2, pp. 326-334, 2016.

[35] H. Zhu, Y. Zhang, Y. Zhang and H. Li, "A novel and provable authenticated key agreement protocol with privacy protection based on chaotic maps towards mobile network," *International Journal of Network Security*, vol. 18, No. 1, pp. 116-123, 2016.

[36] H. Zhu, Y. Zhang, and Y. Sun, "Provably secure multi-server privacy-protection system based," *International Journal of Network Security*, vol. 18, No. 5, pp. 803-815, 2016.

A Appendix: A case proof for Symmetric Encryption

There are two points of this section should be explained firstly:

1) In order to simplify, we just give the provable security of the first method–Algorithm 1 Symmetric Encryption. And we only consider the one-way secret delivery with privacy preserving between node1 and $node_2$.

2) We only use parts of the adversarial model of Canetti and Krawczyk [6]. The basic descriptions are shown in Table 5.

Definition 4. *A secret messages transfered with privacy preserving protocol Π_1 in security parameter k is said to be secure in the adversarial model of Canetti and Krawczyk if for any polynomial-time adversary Λ,*

Algorithm 1 Symmetric Encryption method simulator

```
Input: $H, E_K()/D_K(), (x, T_{K_2}(x))$
Output: $d$
$r \overset{R}{\longleftarrow} \{1, ..., k\}$ where $k$ is an upper bound on the number of sessions activated by $\Lambda$ in any interaction.
Invoke $\Lambda$ and simulate the protocol to $\Lambda$, except for the $r-th$ activated protocol session.
for the $r-th$ session, let $node_1$ send $\{i, T_a(x), C\}$ to $node_2$, where $i$ is the session identifier. The $node_2$ can get the secret messages after he authenticate the timestamp and the value of hash in the ciphertext using his own secret key by one-round. do
    if the $r-th$ session is chosen by $\Lambda$ as the test session then
        Provide $\Lambda$ as the answer to the test query.
        $d \overset{R}{\longleftarrow} \Lambda'$s output.
    else
        $d \leftarrow \{0, 1\}$.
    end if
end for
```

1) If two uncorrupted parties have completed transferring secret messages with privacy preserving by pre-distributed parameter, these sessions produce the same messages as node1s input;

2) $Advantage^{\Lambda}(k)$ is negligible.

Theorem 1. *Under the CDH assumption, using the Algorithm 1 to transfer messages is message-secure in the adversarial model of Canetti and Krawczyk [22].*

Proof. The proof is based on the proof given by Refs. [19, 22]. There are two uncorrupted parties in matching sessions output the same session key, and thus the first part of Definition 4. is satisfied. To show that the second part of the definition is satisfied, assume that there is a polynomial-time adversary Λ with a non-negligible advantage ε in standard model. We claim that Algorithm 1 forms a polynomial-time distinguisher for CDH having non-negligible advantage. □

Probability analysis. It is clear that Algorithm 1 runs in polynomial time and has non-negligible advantage. There are two cases where the r-th session is chosen by Λ as the test session:

1) If the r-th session is not the test session, then Algorithm 1 outputs a random bit, and thus its advantage in solving the CDH is 0.

2) If the r-th session is the test session, then Λ will succeed with advantage ε, since the simulated protocol provided to Λ is indistinguishable from the real protocol. The latter case occurs with probability $1/k$, so the overall advantage of the CDH distinguisher is ε/k, which is non-negligible.

Biography

Hongfeng Zhu obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. He is a full professor of the Kexin software college at Shenyang Normal University. He is also a master's supervisor. He has research interests in wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Dr. Zhu had published more than 50 international journal papers (SCI or EI journals) on the above research fields.

Rui Wang graduated with a Bachelor of Engineering from Shenyang Normal University in 2016. In her college, after completing the learning task, She interests in exploring her professional knowledge. During graduate, under the guidance of his master instructor, she researches information security theory and technology.