Probabilistic RSA with Homomorphism and Its Applications

Yaling Geng, Shundong Li, and Sufang Zhou
(Corresponding author: Shundong Li)

School of Computer Science, Shaanxi Normal University, Xi'an 710062, China
(Email: gengtheory@snnu.edu.cn)

(Received Mar. 15, 2018; Revised and Accepted July 13, 2018; First Online Jan. 26, 2019)

Abstract

RSA is the most famous and the most efficient public key encryption algorithm. However it is deterministic and this seriously restricts its applications in designing general cryptographic protocols. In this paper, we modify the RSA scheme to obtain two probabilistic encryption algorithms that are semantically secure and multiplicatively homomorphic. The two probabilistic RSA variants with higher security can be used either to encrypt or to sign messages, and they can also resist homomorphic attacks. Furthermore, the improved RSA variant algorithms can be extensively applied in designing various cryptographic protocols, such as digital commitment protocol, zero-knowledge proof protocol, oblivious transfer protocol and secure multi-party computation protocol. They provide new efficient tools for cryptographic protocol designing. Theoretical analysis and implementations show that the improved RSA variants are secure and efficient.

Keywords: Homomorphism; Homomorphism Attack; Public Key Encryption; Probabilistic Encryption; Probabilistic Signature

1 Introduction

RSA is the first public key encryption algorithm [44]. It can be used not only for encryption but also for digital signature, and it is simple and easy to implement. For a long time, the researchers continue to improve the efficiency of RSA by using a variety of hardware and software technologies [5, 27, 28], which makes RSA become one of the most efficient public key encryption algorithms and the international standard of asymmetric encryption system. In addition, RSA is also widely used in key distribution, public key encryption, digital signature, authentication and other fields, and it becomes the standard in these fields. Moreover, RSA occupies an extremely important position in many aspects of cryptography and information security.

However, RSA is a deterministic public key encryption algorithm, which means it could not resist chosen plaintext attack, that is, attackers can randomly select a certain number of plaintexts, and encrypt the plaintexts to obtain the corresponding ciphertexts. After comparing the corresponding ciphertexts with intercepted ciphertexts, the attackers determine whether the corresponding paintexts are the chosen plaintexts. Chosen plaintext attack is very effective if plaintext space is small. RSA cannot resist chosen plaintext attack because it is deterministic, which largely limits its application to other aspects of cryptography.

In order to make public key encryption algorithm resist chosen plaintext attack, Goldwasser and Micali proposed a solution, namely probabilistic encryption scheme. The scheme proposed the concept of probabilistic encryption for the first time [18, 19], and it has been widely accepted later. The new cryptosystem is defined as probabilistic encryption cryptosystem (PEC). PEC is a kind of non-deterministic public key cryptosystem, and the essence of which is to add random parameters in the encryption, such that the corresponding ciphertexts of two same plaintexts are completely different even using the same encryption key. In brief, there is one-to-many correspondence between plaintexts and the corresponding ciphertexts in PEC.

Probabilistic encryption algorithms, such as the ElGamal [10], the Pailliar [39], the Okamoto-Uchiyama [38], elliptic curve cryptography [25, 36] and NTRU [24] can be widely used in new fields of cryptography such as oblivious transfer [42], zero-knowledge proof [20], bit commitment [37], secret sharing [47], or secure multi-party computation [17]. Deterministic RSA encryption algorithm cannot be used to construct bit commitment protocols, oblivious transfer protocols and zero-knowledge proof protocols.

In the basic public key encryption systems, RSA and the Rabin [43] are deterministic encryption algorithms with homomorphism, and the ElGamal, the Paillier, the Okamoto-Uchiyama, NTRU and elliptic curve are probabilistic encryption algorithms with homomorphism. Uncertainty makes that public key encryption algorithm has higher security and can resist chosen plaintext attack. A more important reason that above probabilistic encryption algorithms are widely applied is that they are homomorphic. Homomorphism makes these algorithms be widely used in secure multi-party computation [7,23,29–31,33,41,49,50]. Because of the advantage of the importance and efficiency of RSA, it is of very important theoretical significances and practical significances to modify RSA to probabilistic encryption algorithm with homomorphism, which will greatly extend the range of applications of RSA, such that RSA will play a bigger role in cryptography and information security practice.

So far, the improvement schemes of RSA algorithm are divided into three categories as follows.

1) The improved algorithms are probabilistic without homomorphism [2, 14, 48, 52];

2) The improved algorithms are neither probabilistic nor homomorphic [11, 34];

3) The improved algorithms are probabilistic with homomorphism [8].

Bellare *et al.* [2] first proposed randomized filling technique and the corresponding optimal asymmetric encryption filling scheme, namely RSA-OAEP. Shoup *et al.* [48] and Fujisaki *et al.* [14] both presented the improvement schemes of RSA-OAEP. These randomized filling solutions guarantee that the probabilistic RSA that are modified from deterministic RSA is more secure in the random oracle model. The probabilistic RSA not only improves efficiency of encryption but also extends application range. However, it is not homomorphic, which limits its applications in the fields of secure multi-party computation and so on, and the length of plaintexts have decreased. In addition, the security of the scheme is based on random oracle model, but the security of random oracle model is based on an ideal hash function and the ideal hash function does not actually prove existing. Therefore the security of the scheme remains to be studied.

Yu *et al.* [52] modified RSA to probabilistic encryption algorithm by bringing in random numbers and improved the efficiency. However, the new algorithm also loss the homomorphism, which results in that the new algorithm cannot be widely used in cryptography and information security. Makkaoui *et al.* [11] described an improved RSA encryption scheme, namely “Cloud-RSA”. The new scheme is able to resist many known brute force attacks and to maintain multiplicative homomorphism, but it does not guarantee the confidentiality of a key exchange and it is a deterministic encryption scheme, which cannot be applied to bit commitment and digital signature.

Liu *et al.* [34] constructed an improved RSA algorithm using two combinatorial identities based on RSA public key encryption algorithm. It can partly resist common modulus attack, but it is a deterministic encryption algorithm, which cannot resist chosen plaintext attack. Dhakar *et al.* [8] designed a modified RSA encryption algorithm with additive homomorphism, but its execution time is almost 6 times of RSA algorithm because it needs more modulus exponentiations.

In order to take full advantages of simple principle and low computational complexity of RSA, and to enable RSA algorithm to play an important role in oblivious transfer, zero-knowledge proof, bit commitment, secret sharing and secure multi-party computation, this paper proposes two secure and efficient RSA public key encryption variants. These schemes keep multiplicative homomorphism and have semantic security, and therefore can be widely applied to above new fields of cryptography. The improvement schemes not only have important theoretical significances in cryptography but also have important practical significances in constructing other cryptographic encryption protocols, and they have broad applications in privacy protection, secure multi-party computation, cloud storage and cloud computing.

Our contributions: The main contributions of this study are as follows.

1) We design two secure and efficient probabilistic RSA variants with homomorphism using theory of number and public key encryption algorithm, and prove that they are correct and semantically secure. Homomorphism and nondeterminacy will make the algorithms be able to be widely used in all kinds of new cryptography fields;

2) We present a probabilistic digital signature scheme and a digital commitment scheme based on one improved algorithm, and expand the scope of research fields and practical applications of public key cryptography.

Paper organization: The remainder of the paper is organized as follows: Section 2 describes some preliminaries. Section 3 proposes an improved probabilistic RSA algorithm and proves its correctness, multiplicative homomorphism and security. Section 4 presents an efficient probabilistic RSA algorithm with multiplicative homomorphism and proves that the algorithm is secure and correct. Section 5 presents concrete application examples of the improved algorithm. Section 6 analyzes the efficiency and illustrates the results of experiments. Section 7 concludes our work with possible further research directions.

2 Preliminaries

In this section, we introduce several basic knowlege about RSA.

2.1 Public Key System

Diffie and Hellman proposed public key cryptosystem in 1976 [9], which is also known as asymmetric cryptosystem. The most important characteristic of public key cryptosystem is as follows: the keys exist in pairs and it is intractable to compute one key from another key. One is called public key, and the other is called private key. Messages encrypted with public key can only be decrypted by using the corresponding private key in public key cryptosystem. A traditional public key cryptography algorithm usually consists of three algorithms [15]: KeyGen$_{\mathcal{E}}$, Encrypt$_{\mathcal{E}}$, and Decrypt$_{\mathcal{E}}$.

KeyGen$_{\mathcal{E}}$. Taking a security parameter λ as the input(The λ is the bits of large prime numbers), KeyGen$_{\mathcal{E}}$ outputs a private key sk, a public key pk and the corresponding plaintext space $\mathcal{P}$ and ciphertext space $\mathcal{C}$.

$$(skpk\mathcal{P}\mathcal{C}) \leftarrow \text{KeyGen}_{\mathcal{E}}(\lambda).$$

Encrypt$_{\mathcal{E}}$. Taking the public key pk and a plaintext $M \in \mathcal{P}$ as inputs, Encrypt$_{\mathcal{E}}$ outputs the corresponding ciphertext $C \in \mathcal{C}$.

$$(C) \leftarrow \text{Encrypt}_{\mathcal{E}}(pk, M)(M \in \mathcal{P}).$$

Decrypt$_{\mathcal{E}}$. Taking the private key sk and a ciphertext $C \in \mathcal{C}$ as inputs, Decrypt$_{\mathcal{E}}$ outputs the plaintext $M \in \mathcal{P}$.

$$(M) \leftarrow \text{Decrypt}_{\mathcal{E}}(sk, C)(C \in \mathcal{C}).$$

2.2 Homomorphic Encryption

Homomorphic encryption algorithm plays a very important role in secure multi-party computation. Homomorphism is the most important property of the ElGamal, the Paillier, the Okamoto-Uchiyama, NTRU and elliptic curve public key encryption algorithm, which makes these algorithms be powerful building blocks in constructing other cryptographic protocols. A homomorphic encryption algorithm $\mathcal{E}$ consists of algorithms KeyGen$_{\mathcal{E}}$, Encrypt$_{\mathcal{E}}$, Decrypt$_{\mathcal{E}}$ and Evaluate$_{\mathcal{E}}$, which inputs the public key pk, the operation S and ciphertext group $\mathbb{C} =< C_1, \cdots, C_l >$, and outputs the ciphertext of $S(M_1, \cdots, M_l)$.

$$\text{Encrypt}_{\mathcal{E}}(pk, S(M_1, \cdots, M_l)) \leftarrow \text{Evaluate}_{\mathcal{E}}(pk, S, \mathbb{C}).$$

2.3 RSA Public Key Cryptosystem

Rivest, Shamir and Adleman proposed the famous RSA public key cryptosystem in 1978. Its security is based on the large integer factorization problem. So far, it is the most mature public key encryption algorithm in cryptography.

KeyGen.

1) Choose two large prime numbers p and q;
2) Compute $n = p \times q$ and $\varphi(n) = (p-1)(q-1)$, where $\varphi(n)$ is the value of Euler toient function of n;
3) Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(\varphi(n), e) = 1$;
4) Compute d such that $d \times e \equiv 1 \bmod \varphi(n)$;
5) The public key is (e, n), and the private key is (d, n).

Encrypt. To encrypt message m, compute

$$c \equiv m^e \bmod n.$$

Decrypt. To decrypt ciphertext c, compute

$$m \equiv c^d \bmod n.$$

Specifically, RSA is multiplicatively homomorphic, that is,

$$\begin{aligned} E(M_1) \times E(M_2) \bmod n &= ({M_1}^e \bmod n) \times ({M_2}^e \bmod n) \\ &= (M_1 \times M_2)^e \bmod n \\ &= E(M_1 \times M_2) \bmod n. \end{aligned}$$

2.4 RSA Blinding

RSA blinding is usually used to sign a message. The details are as follows.

Blind RSA signature [17]: the author of the message computes the product of the message and blinding factor, i.e.:

$$m' = mr^e \bmod n$$

and sends m' to the signer. The signer then computes the blinded signature s' as:

$$s' = (m')^d \bmod n.$$

s' is sent back to the author of the message, who can then remove the blinding factor to reveal s, the valid RSA signature of m:

$$s = s'(r)^{-1} \bmod n.$$

This works because RSA keys satisfy the equation $r^{ed} \equiv r \bmod n$ and thus

$$s = s'(r)^{-1} = (m')^d(r)^{-1} = m^d r^{ed}(r)^{-1} = m^d \bmod n.$$

Hence s is indeed the signature of m.

This process clearly shows that who adds the blind factor can remove it. This property restricts its application in secure a communication where the sender can add a blind factor, but the receiver cannot remove it. RSA blinding attack may trick the signer into decrypting a message by blind signing another message [12]. Since the signing process is equivalent to decrypting with the signer's secret key, an attacker can provide a blinded version of a message m encrypted with the signer's public key, m' for them to sign. The encrypted message would usually be some secret information which the attacker observed being sent encrypted under the signer's public key which the attacker wants to learn more about. When the attacker removes the blindness of the signed version they will have the clear text:

$$\begin{aligned} m'' = m'r^e \bmod n &= (m^e(\mathrm{mod} n) \cdot r^e)(\mathrm{mod} n) \\ &= (mr)^e \bmod n. \end{aligned}$$

where m' is the encrypted version of the message. When the message is signed, the cleartext m is easily extracted:

$$\begin{aligned} s' = (m'')^d \bmod n &= ((mr)^e \bmod n)^d \bmod n \\ &= mr^{ed} \bmod n \\ &= m \cdot r \bmod n, \end{aligned}$$

since

$$ed = 1 \bmod \varphi(n).$$

Note that $\varphi(n)$ refers to Euler's totient function. The message is now easily obtained.

$$m = s' \cdot r^{-1} \bmod n = mr \cdot r^{-1} \bmod n = m \bmod n.$$

This attack works not only for signing the result of a cryptographic hash function applied to the message but also for signing the message itself.

2.5 Security

Semantic security [6,53] is an important index to measure the security of public key cryptosystem, and it means that an adversary cannot obtain any message about plaintexts. Generally, semantic security of an encryption scheme is characterized by an indistinguishable game, which is also called IND game. IND game is a kind of mental experiment, which has two participants. One is called challenger($\mathcal{B}$), and the other is called adversary($\mathcal{A}$). The IND game of public key cryptosystem is called the IND-CPA game under the chosen plaintext attack, which is defined as follows.

1) Initialization. A challenger generates the encryption system $\mathcal{E}$, and $\mathcal{A}$ gets the public key K_{Pub} of the system, which can be used to encrypt any plaintext;
2) Challenge. $\mathcal{A}$ chooses two same long plaintexts m_0 and m_1. The challenger randomly chooses $b \in \{0,1\}$ to encrypt $c^* = Enc_{K_{Pub}}(m_b)$, then send c^* to $\mathcal{A}$.
3) Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ of b. If $b = b'$, then output 1($\mathcal{A}$ wins the game); otherwise output 0.

Assume $Adv_{\mathcal{A},\mathcal{E}}^{ind-cpa}(\lambda)$ is the advantage of $\mathcal{A}$ winning $Game_{\mathcal{A},\mathcal{E}}^{ind-cpa}(\lambda)$. If there is a negligible function δ, such that

$$Adv_{\mathcal{A},\mathcal{E}}^{ind-cpa}(\lambda) = |Pr[Game_{\mathcal{A},\mathcal{E}}^{ind-cpa}(\lambda) = 1] - \frac{1}{2}| \leq \delta(\lambda),$$

then the scheme $\mathcal{E}$ is indistinguishably secure under the chosen plaintext attack, which means that the scheme is semantically secure.

3 Probabilistic RSA Encryption Algorithm

In this section, we modify the deterministic RSA encryption algorithm to a probabilistic RSA encryption algorithm by adding a random number into a ciphertext. The improved algorithm is still homomorphic, and it has higher security, which makes it more powerful in addressing many cryptography and information security problems.

3.1 Probabilistic RSA with Homomorphism

KeyGen.

1) Choose two large prime numbers p and q;
2) Compute $n = p \times q$ and $\varphi(n) = (p-1)(q-1)$, where $\varphi(n)$ is the value of Euler toient function of n;
3) Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(\varphi(n), e) = 1$;
4) Compute d such that $d \times e \equiv 1 \bmod \varphi(n)$;
5) The public key is (e, n), and the private key is (d, n).

Encrypt. To encrypt a message m, choose a random number $r(0 < r < n)$, and compute

$$c = (c_1, c_2) = (m^{r+e} \bmod n, m^{re} \bmod n).$$

Decrypt. To decrypt ciphertext c, compute

$$m = {c_1}^d \cdot {c_2}^{-d^2} \bmod n.$$

Generally, $r \in Z_n^*$, but r is selected by an encryption party, who does not know the factorization of n. Thus the encryption party can only select $r \in Z_n$. However, the analysis shows that the probability of $r \notin Z_n^*$ is negligible. Thus the encryption party just selects $r \in Z_n$. Moreover, d^2 can be processed before decryption, which will improve the efficiency of decryption.

3.2 Scheme Analyses

Correctness analysis. The encryption is to compute

$$c = (c_1, c_2) = (m^{r+e} \bmod n, m^{re} \bmod n).$$

Decryption is to compute

$$m = {c_1}^d \cdot {c_2}^{-d^2} \bmod n.$$

It is known by the decryption formula that obtaining the inverse of c_2 is the key of correctness proof. Probability of that the inverse of c_2 exists is not 100%, but the probability of that the inverse of c_2 does not exist is negligible. The reason is as follows [38].

Fact 1. Let $a \in \mathbb{Z}_n$. Then a is invertible if and only if $\gcd(a, n) = 1$.

Fact 2. (Euler's theorem)Let $n \geq 2$ be an integer. If $a \in \mathbb{Z}_n^*$, then $a^{\phi(n)} \equiv 1 (\mathrm{mod} n)$.

This means that if the inverse of $(a \bmod n)$ exists, then $(a^{\phi(n)-1} \bmod n)$ is the inverse, because $a \cdot a^{\phi(n)-1} \bmod n = a^{\phi(n)} \bmod n = 1$.

In our scheme, no matter how to choose r and e, $c_2 \equiv m^{re} \bmod n \in Z_n$, the elements in Z_n are all invertible except multiples of p or multiples of q, because $\gcd(up, n) = p$ and $\gcd(vq, n) = q$, where u and v are integers. Then we will analyse the probability that c_2 has no inverse in the following. As mentioned above, up and vq have no inverse in Z_n. The number of up is $q - 1$, and the number of vq is $p - 1$. For example, when $n = p \times q = 5 \times 7 = 35$, the number of up is 6(5, 10, 15, 20, 25, 30), and the number of vq is 4(7, 14, 21, 28). These numbers(5, 10, 15, 20, 25, 30, 7, 14, 21, 28) have not inverse. Suppose that the bits of p is approximately equal to the bits of q, then the number of up and vq is approximately equal to $2(p - 1)$. Thus the ratio of $2(p - 1)$ to $n - 1$ is the probability that c_2 has no inverse, that is,

$$\tfrac{2(p-1)}{n-1} \approx \tfrac{2}{q}.$$

Generally, because we need to ensure n is difficult to factorize, we must set the bits of p approximately equal to the bits of q and $(p \geq 2^{384}) \wedge (q \geq 2^{384})$. Thus the probability that c_2 has no inverse is less than $\frac{1}{2^{383}} = 2^{-383}$. Therefore the probability is negligible. Thus c_2 has inverse, and it can be obtained by computing $(c_2^{\phi(n)-1} \bmod n)$, or by following:

$$D(c) \equiv {c_1}^d \cdot {c_2}^{-d^2} \bmod n \equiv \tfrac{m^{(r+e)d} \bmod n}{m^{red^2} \bmod n} \equiv \tfrac{(m^{rd} \times m^{ed}) \bmod n}{(m^{rd})^{ed} \bmod n} \equiv \tfrac{m^{rd} \times m \bmod n}{(m^{rd}) \bmod n} = m \bmod n.$$

In conclusion, congruence expressions are not valid to division in the absence of inverses. However, c_2 has inverse in our scheme, so congruence expressions are valid to division, that is, both sides of the congruence expression are divisible. This kind of usage can also be seen in the Ref.[6], so it can be seen that congruence expressions are valid to division in the case of the inverse existing. This completes the proof of the correctness analysis.

Homomorphism analysis. The RSA probabilistic encryption algorithm keeps multiplicative homomorphism. The specific property is described as follows.

Evaluation. For given ciphertexts $E(M_1)$ and $E(M_2)$, compute

$$\begin{aligned} v &= E(M_1) \times E(M_2) \bmod n \\ &= (M_1^{r_1+e}, M_1^{r_1 e}) \times (M_2^{r_2+e}, M_2^{r_2 e}) \bmod n \\ &= (M_1^{r_1} M_2^{r_2} (M_1 M_2)^e \bmod n, (M_1^{r_1} M_2^{r_2})^e \bmod n) \\ &= (c_1, c_2). \end{aligned}$$

Decrypting v can obtain:

$$\begin{aligned} D(v) &= {c_1}^d \cdot {c_2}^{-d^2} \bmod n \\ &= \frac{[M_1^{r_1} M_2^{r_2} (M_1 M_2)^e]^d \bmod n}{[(M_1^{r_1} M_2^{r_2})^e]^{d^2} \bmod n} \\ &= \frac{M_1^{r_1 d} M_2^{r_2 d} (M_1 M_2)^{ed} \bmod n}{(M_1^{r_1 d} M_2^{r_2 d})^{ed} \bmod n} \\ &= \frac{M_1^{r_1 d} M_2^{r_2 d} M_1 M_2 \bmod n}{M_1^{r_1 d} M_2^{r_2 d} \bmod n} \\ &= M_1 M_2 \bmod n. \end{aligned}$$

Therefore the RSA variant is multiplicatively homomorphic, that is,

$$E(M_1) \times E(M_2) \bmod n \equiv E(M_1 \times M_2) \bmod n.$$

Security analysis. About the security of this scheme, we have the following theorem [39, 40].

Theorem 1. *If the RSA problem is difficult, then $\mathcal{E}$ has IND-CPA security, that is, the scheme is semantically secure. Assume $\mathcal{E}(Gen, Enc, Dec)$ is RSA variant. $\mathcal{A}$ is a polynomial time algorithm that attacks $\mathcal{E}$, and the advantage of $\mathcal{A}$ winning IND-CPA game is ξ. We can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to solve the RSA problem.*

Proof. The challenger($\mathcal{B}$) of the RSA problem works as follows.

1) Inputs λ. Runs GenRSA(λ) and obtains (n, e, d). The public key is (n, e), and the private key is (n, d);

2) Sends system parameter λ and the public key (n, e) to $\mathcal{A}$;

3) Obtains M_0 and M_1 of $\mathcal{A}$;

4) Randomly selects $b \in \{0, 1\}$;

5) Assumes $C^* = (T_1 M^{e-1} \bmod n, T_2^e \bmod n)$ and sends C^* to $\mathcal{A}$;

6) Supposes that $b' \in \{0, 1\}$ is the guess of $\mathcal{A}$;

7) Outputs s'(If $b = b'$, then $s' = 0$; if $b \neq b'$, then $s' = 1$).

□

The probability of $\mathcal{B}$ winning RSA security game can be solved by Bayes formula as follows:

$$\begin{aligned} &Pr[s = s'] \\ &= Pr[s = 0]Pr[s = s'|s = 0] + Pr[s = 1]Pr[s = s'|s = 1] \\ &= \frac{1}{2}Pr[s' = 0|s = 0] + \frac{1}{2}Pr[s' = 1|s = 1] \\ &= \frac{1}{2}Pr[b = b'|s = 0] + \frac{1}{2}Pr[b \neq b'|s = 1]. \end{aligned} \quad (1)$$

When $s' = 0$, $\mathcal{B}$ sets $T = (T_1, T_2) = (M^{r+1} \bmod n, M^r \bmod n)$. At this point, the view of $\mathcal{B}$ submitted to $\mathcal{A}$ is indistinguishable from the view of $\mathcal{A}$ attacking $\mathcal{E}$ in the IND-CPA game. Therefore when $s' = 0$, the probability of $b = b'$ is equal to the probability of $\mathcal{A}$ winning the IND-CPA game, that is,

$$Pr[b = b'|s = 0] = \frac{1}{2} + \xi. \qquad (2)$$

When $s' = 1$, $\mathcal{B}$ sets $T = \mathcal{R}_w = (\mathcal{R}_1, \mathcal{R}_2)$. Because R_w is uniformly distributed over Z_n, we can obtain that $(R_1 M^{e-1} \bmod n, R_2^e \bmod n)$ is uniformly distributed over (Z_n^*, Z_n^*), which is independent of n, M_0, M_1 and b. $(R_1 M^{e-1} \bmod n)$ and $(R_2^e \bmod n)$ are independent of M_0, M_1 and b. Therefore K_{Pub} and ciphertext C^* do not reveal any information about b, and guess b' outputed by $\mathcal{A}$ must be independent of b. Because the probability of $b = 0$ and b =1 are both 1/2, we can obtain

$$Pr[b \neq b'|s = 1] = \frac{1}{2}. \qquad (3)$$

By Equations (1), (2) and (3), we can obtain

$$\begin{aligned} Pr[s = s'] &= \frac{1}{2}(\frac{1}{2} + \xi) + \frac{1}{2} \times \frac{1}{2} \\ &= \frac{1}{2} + \frac{1}{2}\xi. \end{aligned}$$

Therefore the advantage of $\mathcal{B}$ winning the game is

$$|Pr[s = s'] - \frac{1}{2}| = (\frac{1}{2} + \frac{1}{2}\xi) - \frac{1}{2} = \frac{\xi}{2}.$$

We are aware of that $\mathcal{B}$ can only win the game with negligible advantage, so $\xi/2$ is negligible, which implies ξ is also negligible. Therefore $\mathcal{A}$ can only win the IND-CPA game with the negligible advantage ξ.

Thus, using this scheme to encrypt any two same long plaintexts M_0 and M_1, the correspongding ciphertexts C_0 and C_1 are indistinguishable, that is, $C_0 \overset{c}{\equiv} C_1$.

4 Efficient Probabilistic RSA with Homomorphism

In Section 3, we modify the deterministic RSA encryption algorithm to a probabilistic RSA encryption algorithm, and maintain the homomorphism. However, encryption efficiency is reduced, so we introduce another variant. The new variant not only keeps homomorphism and semantic security but also greatly improves the efficiency of encryption.

4.1 Efficient Probabilistic RSA with Homomorphism

KeyGen.

1) Choose two large prime numbers p and q;
2) Compute $n = p \times q$ and $\varphi(n) = (p-1)(q-1)$, where $\varphi(n)$ is the value of Euler toient function of n;
3) Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(\varphi(n), e) = 1$;
4) Compute d such that $d \times e \equiv 1 \bmod \varphi(n)$;
5) The public key is (e, n), and the private key is (d, n).

Encrypt. To encrypt a message m, choose a random number $r (0 < r < n)$, and compute

$$c = (c_1, c_2) = (r^e \bmod n, rm^e \bmod n).$$

Decrypt. To decrypt ciphertext c, compute

$$M = (c_2 {c_1}^{-d})^d \bmod n.$$

4.2 Scheme Analyses

Correctness analysis. The encryption is to compute

$$E(M) = (c_1, c_2) = (r^e \bmod n, rm^e \bmod n).$$

Decryption is to compute

$${c_1}^d \bmod n \equiv (r)^{ed} \bmod n \equiv r \bmod n$$

and

$$(\tfrac{c_2}{r})^d \bmod n \equiv m \bmod n.$$

Homomorphism analysis. The efficient RSA probabilistic encryption algorithm keeps multiplicative homomorphism. The specific property is described as follows.

Evaluation. For given ciphertexts $E(M_1)$ and $E(M_2)$, compute

$$\begin{aligned} & E(M_1) \times E(M_2) \bmod n \\ & \equiv ({r_1}^e, r_1 {M_1}^e) \times ({r_2}^e, r_2 {M_2}^e) \bmod n \\ & \equiv ((r_1 r_2)^e, r_1 r_2 (M_1 M_2)^e) \bmod n \\ & \equiv E(M_1 \times M_2) \bmod n. \end{aligned}$$

Therefore the RSA variant is multiplicatively homomorphic.

Security analysis. The proof method [41] is similar to the proof of Theorem 1, and we omit it here.

5 Applications

5.1 Probabilistic Digital Signature Scheme

The rapid development of Internet, Internet of things and car networking animate booming development of e-commerce of the world. Using handwritten signatures will greatly reduce transaction efficiency of e-commerce, so more and more people want to replace handwritten signatures with fast and convenient digital signature for signing the agreement in real life to improve the trade efficiency Moreover, digital signature not only ensure the security and accuracy of data transmission, but also confirm the identity of both parties. Therefore the applications of digital signature are increasingly wide [42, 43], and the research of digital signature is meaningful. In this paper, we improve the security of the signature by introducing a random number, which is sufficient to resist the homomorphic attack. The specific signature scheme is as follows.

Sign:

1) Suppose that the message is m;
2) Generate signature $S_1 \equiv m^{r+d} \bmod n$;
3) Generate signature $S_2 \equiv m^{rd} \bmod n$;
4) Output (m, S_1, S_2).

Verify:

1) Obtain (m, S_1, S_2);
2) Compute $h_1 = {S_1}^e \bmod n$;
3) Compute $h_2 = {S_2}^{e^2} \bmod n$;
4) Compute $h' = (h_1, h_2) = h_1 {h_2}^{-1}$;
5) Compare whether $m = h'$. If $m = h'$, then accept the signature; otherwise reject the signature.

Correctness analysis. The signature process is:

$$s = (s_1, s_2) = (m^{r+d} \bmod n, m^{rd} \bmod n).$$

Verifying s can obtain:

$$\frac{{s_1}^e \bmod n}{{s_2}^{e^2} \bmod n} \equiv \frac{m^{(r+d)e} \bmod n}{m^{rde^2} \bmod n} \equiv \frac{(m^{re} \times m^{de}) \bmod n}{(m^{re})^{de} \bmod n} \equiv \frac{m^{re} \times m \bmod n}{m^{re} \bmod n} = m \bmod n.$$

5.2 Homomorphism Attack

Homomorphic attack refers to that a malicious attacker uses homomorphism to forge a new signature in order to achieve attacks. The deterministic RSA signature algorithm cannot resist homomorphism attack. It mainly has the following two attacks:

1) If the attacker knows the messages M_1 and M_2, and the corresponding signatures S_1 and S_2, then the attacker can forge signature $S = (M_1 \times M_2)^d \bmod n$ of message $M = (M_1 \times M_2) \bmod n$, because $S = (S_1 \times S_2) \bmod n = (M_1^d \times M_2^d) \bmod n = (M_1 \times M_2)^d \bmod n$;
2) If the attacker knows the messages M_1 and M_2, and the corresponding signatures S_1 and S_2, then the attacker can forge signature $S = (M_1^a \times M_2^b)^d \bmod n$ of message $M = (M_1^a \times M_2^b) \bmod n$, where a and b are positive integers, because $S = S_1^a S_2^b \bmod n = (M_1^d)^a \times (M_2^d)^b \bmod n = (M_1^a M_2^b)^d \bmod n$.

If M is a valuable piece of information, then the signature of M will be very important, and it is very dangerous for a malicious attacker to hold such an important signature. In order to resist the above two attacks, this paper proposes a new probabilistic signature scheme based on the first RSA variant. Because it is the application part, this paper just does an intuitive analysis here. Suppose that the attacker knows the two messages M_1 and M_2 and the corresponding signatures S and S'. If an attacker can forge the signatures $S^* = (S_1^*, S_2^*) = ((M_1M_2)^{r+d}, (M_1M_2)^{rd})$ of M_1M_2 by multiplicatively transforming $S = (S_1, S_2) = (M_1^{r_1+d}, M_1^{r_1d})$ and $S' = ({S'}_1, {S'}_2) = (M_2^{r_2+d}, M_2^{r_2d})$, then the scheme cannot resist homomorphism attack. However, we can only obtain $S \times S' \bmod n = ((M_1M_2)^d M_1^{r_1} M_2^{r_2}, (M_1^{r_1} M_2^{r_2})^d) \bmod n$ by multiplicative transformation, that is, the forged signatures cannot be verified. Therefore our signature algorithm can resist homomorphism attack.

With the RSA-blinding, a message provider can add a blind factor to the message, ask the signer to blind sign the message, and then remove the blind factor. This approach is not applicable for secure communication, because the sender can add a blind factor but the receiver cannot remove the blind factor unless the sender send the blind factor by a different channel. Therefore, RSA-blinding is mainly used to obtain non-determinstic blind signatures [1]. It cannot be used to secure a communication, nor can it be used in general protocols such as secure multiparty computations. Our RSA variants can be used to sign a message, to secure a communication, or to construct secure multiparty computation protocols. Using RSA-blinding signature, a malicious attacker may lure a signer to sign a message that hurts his benefit. Our probabilistic RSA can prevent this attack because our scheme can resist homomorphism attack.

To sum up, although the approach is not new, our constructions are completely new and have significant advantages. our probabilistic schemes with homomorphism can be used either to encrypt a message (to secure a communication) or to sign a message, or to construct general cryptographic protocols such as secure multiparty computation protocols. RSA blinding can only be used to make a blind signature. These are the advantages of our scheme.

5.3 Digital Commitment Scheme

Digital commitment is an important module of cryptography. Besides it can be widely used in constructing zero knowledge proof protocols and coin-tossing protocols, it also has important applications in real life, for example, confidential bidding. In addition, digital commitment can be applied to electronic voting, electronic lottery and other aspects. Therefore studying more efficient digital commitment is of great significance. Generally, digital commitment scheme [44-46] is divided into the following two categories: the first is bit commitment, which means that the commitment information is limited to 0 and 1; the second is digital commitment, which means that the commitment information can be numbers or strings. In short, a digital commitment scheme is a two phase agreement with two parties taken part in. The two parties are the commitment maker and the receiver, and the two phases are commitment phase and revealing phase. The commitment maker achieves that the secret information is binded to a number through this protocol. The binding satisfies confidentiality and certainty.

However, the deterministic RSA encryption algorithm cannot be used to construct a digital commitment scheme, while the probabilistic RSA encryption algorithm can be used to construct the commitment scheme. Based on the second RSA variant, this paper proposes a non-malleable commitment scheme based on large prime factorization problem. The specific commitment is as follows:

KeyGen. Choose two private large primes p and q, and compute $n = p \times q$ and $\varphi(n) = (p-1)(q-1)$, where $\varphi(n)$ is the Euler toient function value of n. Two parties choose an integer e, where $1 < e < \varphi(n)$ and $\gcd(\varphi(n), e) = 1$.

Protocol 5.3.1 Non-malleable commitment scheme based on factorization.

Commitment phase. The commitment maker uniformly selects a random number of $r (0 < r < n)$, and compute

$$c(v) = (c_1, c_2) = (r^e \bmod n, rv^e \bmod n).$$

Revealing phase. The commitment maker sends r and v to the receiver, and the receiver verifies whether the following equation is true.

$$(r^e \bmod n, rv^e \bmod n) = (c_1, c_2).$$

If it is true, then accept the commitment; otherwise reject the commitment.

Confidentiality analysis. In this protocol, the commitment of any number is computationally indistinguishable from the commitment of other numbers. Assume commitments of v and $v + b$ respectively are

$$\begin{aligned} c(v) &= (c_1, c_2) \\ &= (r^e, rv^e \bmod n) \\ c(v+b) &= (c_1, c_2) \\ &= ((r+x)^e, (r+x)(v+b)^e \bmod n). \end{aligned}$$

The commitment maker uniformly selects a random number in commitment, so the commitment maker possibly selects r or $r + x$. However, the receiver cannot determine that which is the commitment of v, and which is a commitment of $v + b$. Because the above commitment values have been randomized, the receiver does not distinguish commitments between v and $v + b$.

Determinacy analysis. Because $c_1 = r^e \bmod n$, we can obtain that $r^e \bmod n$ is deterministic when the r is deterministic, that is, there is one-to-one correspondence between c_1 and r, which means that an attacker cannot forge r' such that $r'^e \bmod n = r^e \bmod n$. Analogously, because $\frac{rv^e}{r} \bmod n = v^e \bmod n$, we know that v^e is also deterministic. Thus the commitment scheme satisfies the requirement of certainty at the meaning of the computational feasibility.

Non-malleability analysis. The commitment information is $c(v)$. If the attacker wants to make a commitment of $v+b$ according to $c(v)$, he/she must know the value of the v. If the attacker wants to know the value of the v, then he/she need to factorize large number. However the problem of factoring large numbers is difficult, so this scheme is non-malleable.

6 Performance Analyses

6.1 Computational Complexity

In public key encryption system, the Paillier and the ElGamal are probabilistic encryption algorithms. Among them, Paillier's encryption algorithm and our two schemes are based on the same difficult problem, namely the factorization of large integers. ElGamal's encryption algorithm and our two schemes have same homomorphism, that is, multiplicative homomorphism. Because the schemes of our paper are mainly based on modulus exponentiations, we can measure the computation overhead of the algorithms by comparing modulus exponentiations. Suppose that the computation overhead of modulus n is x, the computation overhead of modulus n^2 is y, the computation overhead of modulus p is z, the computation overhead of modulus m is h, the computation overhead of modulus m^2 is k, and the computation overhead of modulus $m^2 - 1$ is t. The analysis of each scheme is shown in Table 1. To simplify the description, we define the RSA variant of Section 3 is PRSA 1, and the RSA variant of Section 4 is PRSA 2.

6.2 Experiments

In this section, we present two experimental results in terms of two RSA variants efficiency. The experimental settings are as follows, the operating system is Windows 10, CPU is Inter Core i5-6600 3.30GHz, and RAM is 8GB. We implement the schemes of this paper by using Java language and use the Experiment 1 and Experiment 2 to test the cost of PRSA 1 and PRSA 2. Execution

Table 1: Comparison of all solutions

	Scheme [33]	**PRSA 1**	**PRSA 2**	**Paillier's**	**ElGamal's**
Computation	$2x + h + k + t$	$4x$	$3x$	$3y + x$	$3z$
Number of keys	7	3	3	3	2

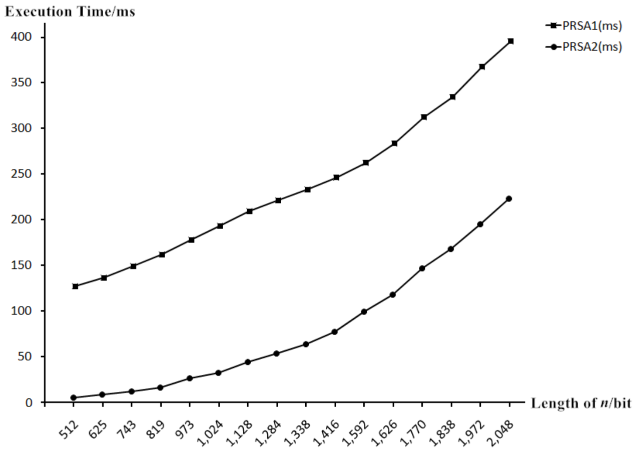

Figure 1: Comparison of the implementation of our two RSA variants($m = 20$)

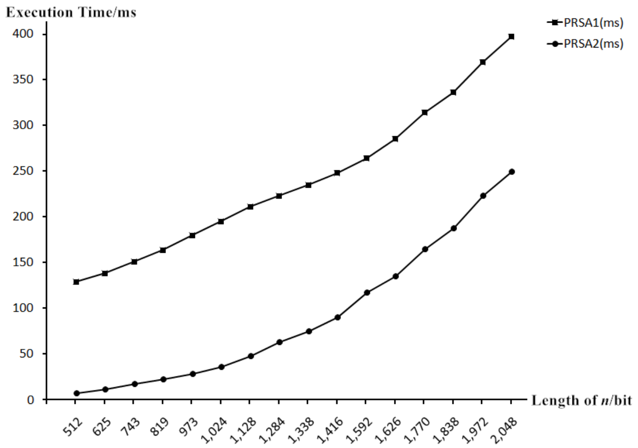

Figure 2: Comparison of the implementation of our two RSA variants($m = 2000$)

time of the four protocols verifies the computational complexity(Our implementation is not to compare the performance of our scheme with that of other schemes, but to show that our scheme is practical, so we did not use standard benchmark implementation [26, 45, 46, 51]).

Experiment 1. This experiment adopts control variable method. The length of n is independent variable. The range of n is [512, 2048]. The confidential datas are 20 and 2000. We implement PRSA1 and PRSA2, and each execution time in the experiment is the average time of 100 times encrypting and decrypting same plaintext. The results are shown in Figure 1 and Figure 2.

Figure 1 and Figure 2 show that the execution time grows as the length of the modulus increase, which has nothing to do with the length of confidential data. This is because that the execution time of the other operations is negligible compared with the execution time of the modulus exponentiations. In addition, the

modulus exponentiation operation is also uncertain. After modulus exponentiation operation, a small number may become a large number, and a large number may become a small number. Thus the length of confidential data does not affect the efficiency of the algorithm. The results show that the execution time of PRSA 1 is much larger than that of PRSA 2 under the condition of the limited length of confidential data, because modulus exponentiations of PRSA 1 is more than that of PRSA 2.

Experiment 2. This experiment adopts control variable method. The length of n is independent variable. The range of n is [512, 2048]. The confidential datas are 20 and 2000. We implement the Paillier, PRSA1, the ElGamal and PRSA2, and each execution time in the experiment is the average time of 100 times encrypting and decrypting same plaintext. The results are shown in Table 2.

The experimental datas show that the execution time of Paillier algorithm is much larger than that of PRSA 1 and PRSA 2, because the calculation of modulus n^2 of Paillier is more than the calculation of modulus n of PRSA 1 and PRSA 2. It can be seen that PRSA 2 is the most efficient.

7 Conclusion

RSA algorithm is of practical significances in information security, and it also has wide applications in public key cryptography. RSA algorithm can serve as the basic module of many cryptographic protocols, and it can be even widely used to guarantee secrecy communications, and confidentiality of the Internet, the Internet of things, and car networking business. In order to improve the security of RSA encryption algorithm and extend its applications, this paper modifies it to probabilistic encryption algorithms with semantic security. The RSA variants can resist homomorphic attack, which can be used for probabilistic encryptions, probabilistic signatures and digital

Table 2: Comparison of all solutions

Performance	**Paillier's**	**PRSA 1**	**ElGamal's**	**PRSA 2**
Alice(ms)	103	87	21	18
Number of keysBob(ms)	138	129	11	9
Total(ms)	241	216	32	27

commitments. Moreover, the variants can be widely used in designing various security protocols and it also provides a new and effective tool for designing cryptographic protocols. Theoretical analyses and experiments show that the algorithms are secure and efficient.

Acknowledgments

The authors would like to thank the anonymous reviewers for detailed and valuable comments. This work is supported by the National Natural Science Foundation of China (Grant no. 61272435).

References

[1] Bellare, Namprempre, Pointcheval, *et al.* "The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme," *Journal of Cryptology*, vol. 16, no. 3, pp. 185-215, 2003.

[2] M. Bellare and P. Rogaway "Optimal asymmetric encryption," *Lecture Notes in Computer Science*, vol. 950, no. 6, pp. 92-111, 1994.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology (CRYPTO'01)*, pp. 213-229, 2001.

[4] Z. J. Cao and M. L. Liu, "Improvement of signature scheme based on strong RSA," *Chinese Journal of Computers*, vol. 29, no. 9, pp. 1617-1621, 2006.

[5] C. C. Chang, M. S. Hwang, "Parallel computation of the generating keys for RSA cryptosystems", *Electronics Letters*, vo1. 32, no. 15, pp. 1365–1366, 1996.

[6] G. L. Chen, "Mathematical foundations in information security," *Beijing: Tsinghua University Press*, 2004.

[7] M. M. V. Deshmukh, P. A. Tijare and S. N. Sawalkar, "A survey on privacy preserving data mining techniques for clinical decision support system," *International Research Journal of Engineering and Technology*, vol. 3, no. 5, pp. 2064-2069, 2016.

[8] R. S. Dhakar, A. K. Gupta and P. Sharma, "Modified RSA encryption algorithm (MREA)," in *Second International Conference on Advanced Computing and Communication Technologies*, pp. 426-429, 2012.

[9] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

[10] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *Journal of IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469-472, 1984.

[11] K. El-Makkaoui, A. Ezzati and A. Beni-Hssane, "Cloud-RSA: An enhanced homomorphic encryption scheme," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*, pp. 471-480, 2017.

[12] M. Fischlin, D. Schroder "Security of blind signatures under aborts," in *International Conference on Practice and Theory in Public Key Cryptography*, pp. 297-316, 2009.

[13] M. Fischlin and R. Fischlin, "Efficient non-malleable commitment schemes," in *Annual International Cryptology Conference*, pp. 413-431, 2000.

[14] E. Fujisaki, T. Okamoto, D. Pointcheval, *et al.*, "RSA-OAEP is secure under the RSA assumption," *Journal of Cryptology*, vol. 17, no. 2, pp. 81-104, 2004.

[15] C. Gentry, "A fully homomorphic encryption scheme," *Stanford University*, 2009. ISBN: 978-1-109-44450-6

[16] S. Goldwasser "Lecture notes on cryptography," *Cite Seer X*, 1996. (`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.56.4314&rep=rep1&type=pdf`)

[17] O. Goldreich, "Secure multi-party computation," *Journal of Manuscript Preliminary Version*, 1998. (`https://www.researchgate.net/profile/Oded_Goldreich/publication/2934115_Secure_Multi-Party_Computation/links/00b7d52bb04f7027d4000000.pdf`)

[18] S. Goldwasser and S. Micali, "Probabilistic encryption and how to play mental poker keeping secret all partial information," in *Proceeding of the 14th ACM Symposium on the Theory of Computer*, pp. 365-377, 1982.

[19] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Science*, vol. 28, no. 1, pp. 270-299, 1994.

[20] S. Goldwasser, S. Micali and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 186-208, 1989.

[21] L. M. Gong, S. D. Li, D. S. Wang and J. W. Dou, "A randomized coding of plaintext encryption scheme," *Journal of Software*, vol. 2017, no. 2, pp. 372-383, 2017.

[22] L. M. Gong, S. D. Li and J. W. Dou, "A public-key cryptosystem secure against adaptive chosen ciphertext attack," *Journal of Cryptologic Research*, vol. 3, no. 1, pp. 42-55, 2016.

[23] L. M. Gong, S. D. Li, Q. Mao, *et al.*, "A homomorphic encryption scheme with adaptive chosen ciphertext security but without random oracle," *Theoretical Computer Science*, vol. 609, no. 1, pp. 253-261, 2016.

[24] J. Hoffstein, J. Pipher and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," *International Algorithmic Number Theory Symposium*, pp. 267-288, 1998.

[25] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, 1987.

[26] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems," in *International Cryptology Conference on Advances in Cryptology*, pp. 104-113, 1996.

[27] Y. F. Li, Q. Liu, T. Li, *et al.*, "An improved algorithm for batch RSA," *Journal of Chinese Mini-Micro Computer Systems*, vol. 33, no. 1, pp. 64-70, 2012.

[28] Q. Li and J. Y. Zhang, "An improved fast RSA algorithm," *Journal of Chinese Mini-Micro Computer Systems*, vol. 22, no. 1, pp. 70-72, 2001.

[29] S. D. Li and D. S. Wang, "Efficient secure multiparty computation based on homomorphic encryption," *Acta Electronica Sinica*, vol. 41, no. 4, pp. 798-803, 2013.

[30] S. D. Li, S. F. Zhou, Y. M. Guo, *et al.*, "Secure set computing in cloud environment," *Journal of Software*, vol. 27, no. 6, pp. 1549-1565, 2016.

[31] S. D. Li, J. W. Dou and D. S. Wang, "Survey on homomorphic encryption and its applications to cloud security," *Journal of Computer Research and Development*, vol. 52, no. 6, pp. 1378-1388, 2015.

[32] S. D. Li and D. S. Wang, "Modern cryptography: Theory, method and research forefront," *Science Press*, 2009.

[33] G. Liu and T. F. Jiang, "Research on homomorphic encryption technology and the applications of it in IOT," *Journal of Netinfo Security*, vol. 2011, no. 5, pp. 61-64, 2011.

[34] Y. H. Liu, Z. K. Dai and H. Li, "An improved public-key algorithm based on RSA," *Journal of Sichuan University (Natural Science Edition)*, vol. 42, no. 4, pp. 760-764, 2005.

[35] A. J. Menezes, P. C. V. Oorschot, S. A. Vanstone, "Handbook of applied cryptography," *CRC Press*, pp. 816, 1996. ISBN: 0-8493-8523-7

[36] V. S. Miller, "Use of elliptic curves in cryptography," in *Conference on the Theory and Application of Cryptographic Techniques*, pp. 417-426, 1985.

[37] M. Naor, "Bit commitment using pseudorandomness," *Journal of Cryptology*, vol. 4, no. 2, pp. 151-158, 1991.

[38] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 308-318, 1998.

[39] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223-238, 1999.

[40] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual International Cryptology Conference, Springer Berlin Heidelberg*, pp. 129-140, 1991.

[41] M. Qiu, S. S. Luo, W. Liu, *et al.*, "A solution of secure multi-party multi-data ranking problem based on RSA encryption scheme," *Acta Electronica Sinica*, vol. 37, no. 5, pp. 1119-1123, 2009.

[42] M. O. Rabin, "How to exchange secrets with oblivious transfer," *IACR Cryptology Eprint Archive*, 2005. (`https://www.semanticscholar.org/paper/How-To-Exchange-Secrets-with-Oblivious-Transfer-Rabin/1d2a3436fc7ff4b964fa61c0789df19e32ddf0ed`)

[43] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," *Journal of Massachusetts Institute of Technology*, 1979. (`https://dl.acm.org/citation.cfm?id=889813`)

[44] R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Journal of Communications of the ACM*, vol. 26, no. 1, pp. 96-99, 1978.

[45] M. A. Sadikin, R. W. Wardhani, "Implementation of RSA 2048-bit and AES 256-bit with digital signature for secure electronic health record application," in *International Seminar on Intelligent Technology and ITS Applications*, pp. 387-392, 2017.

[46] H. Siregar, E. Junaeti, T. Hayatno, "Implementation of digital signature using Aes and Rsa algorithms as a security in disposition system af letter," in *IOP Conference Series: Materials Science and Engineering*, vol. 180, no. 1, pp. 012-055, 2017.

[47] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[48] V. Shoup, "OAEP reconsidered (extended abstract)," *Proceeding of Cryptography*, vol. 15, no. 4, pp. 239-259, 2001.

[49] K. Suveetha and T. Manju, "Ensuring confidentiality of cloud data using homomorphic encryption," *Indian Journal of Science and Technology*, vol. 9, no. 8, pp. 1-7, 2016.

[50] L. C. Wang, J. Li and H. Ahmad, "Challenges of fully homomorphic encryptions for the internet of things," *Journal of Ieice Transactions on Information and Systems*, vol. 99, no. 8, pp. 1982-1990, 2016.

[51] H. Xia, Q. Pei, Y. Xi. "The analysis and research of freak attack based on OpenSSL," in *International Conference on Information Engineering for Mechanics and Materials*, 2016.

[52] M. S. Yu and H. Zou, "An improved RSA public key cryptosystem," *Journal of Dalian University of Technology*, vol. 43, no. 1, pp. 50-52, 2003.

[53] B. Yang, "Modern cryptography," *Beijing: Tsinghua University Press*, 2007.

[54] Y. B. Zhou, Z. F. Zhang, S. H. Qin, *et al.*, "A fair exchange protocol based on RSA signature scheme," *Journal of Software*, vol. 15, no. 7, pp. 1049-1055, 2004.

Biography

Yaling Geng was born in 1993. She is currently pursuing the M.S. degree with School of Computer Science in Shaanxi Normal University. Her research interests focus on modern cryptography and information security.

Shundong Li was born in 1963. He received the Ph.D. degree in Department of computer science and technology from Xian JiaoTong University in 2003. He is now a Professor with School of Computer Science in Shaanxi Normal University. His research interests focus on modern cryptography and secure multi-party computation.

Sufang Zhou was born in 1990. She is currently pursuing the PH.D. degree with School of Computer Science in Shaanxi Normal University. Her research interests focus on secure multi-party computation and information security.