IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 21, 2019, accepted August 30, 2019, date of publication September 3, 2019, date of current version September 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939227

Electronic Scoring Scheme Based on Real Paillier Encryption Algorithms

WANG RONG-BING[1], LI YA-NAN[1,2], XU HONG-YAN[1], FENG YONG[1], AND ZHANG YONG-GANG[3]

[1]College of Information, Liaoning University, Shenyang 110036, China
[2]Department of Senior Technician, Shandong Labor Vocational and Technical College, Jinan 250022, China
[3]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

Corresponding author: Feng Yong (fengyong@lnu.edu.cn)

This work was supported in part by the Project of Liaoning Provincial Engineering Laboratory of Big Data System of Public Opinion and Network Security under Grant 2016-294, in part by the National Nature Science Foundation of China under Grant 71771110, in part by the Social Science Planning Foundation of Liaoning Province of China under Grant L18AGL007, and in part by the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, under Grant 93K172018K01.

ABSTRACT Paillier algorithm has higher time efficiency than other homomorphic encryption algorithms due to only one power operation is required, but its encryption range is limited to integer, which cannot meet the need of electronic scoring system to encrypt real. To solve the problem, an electronic scoring scheme based on real Paillier algorithm is put forward. In the proposed scheme, compound transformation is introduced to convert real to large integer. The encryption range of Paillier algorithm is extended from integer to real by the large integer operation, which can make the scoring results more precision. In addition, a more secure and efficient two-way identity authentication strategy between clients and servers is designed by using Horner's rule. After hashing the username and password by Horner's rule, the obtained hash value is encrypted and transmitted, which can not only improve system security by avoiding disclosure of user's information during transmission, but also increase the computation efficiency due to the reduction of data size in encryption and decryption. In the experiment, compared with the existing encryption algorithms on different scoring data sets, the whole encryption and decryption time is slightly increased by 0.28% compared with the traditional Paillier, but computing real score is realized in the proposed method. The efficiency of addition and multiplication of the proposed method is increased averagely by 25.99% and 42.75%. The average shorten time of two-way identity authentication is 27.22 milliseconds due to the using of Horner's rule. When the number of clients is more than 105, the time efficiency is improved more obviously.

INDEX TERMS Real Paillier, homomorphic encryption, electronic scoring, privacy security, clouding service, E-voting.

I. INTRODUCTION

With the continuous progress of global democratization, all-people participation in vote has become a major symbol of a democratized country. With the rapid development of Internet, E-voting breaks through the time and space limitations to the traditional voting methods, realizes the possibility of remote voting in different places of the world [1], which can save a lot of voting costs [2], [3]. At present, E-voting schemes are mainly based on the following four technologies: blind signature, secret sharing, hybrid network and homomorphic encryption. Homomorphic encryption was proposed firstly by Rivest in 1978 [4]. It has a unique feature which is not possessed by the other three technologies [5]. The feature is that the required operations can directly perform on ciphertext, and the results are consistent with those got by executing the same operations on plaintext [6]–[9]. In other words, the technology allows people to make specific algebraic operation on ciphertext to obtain the results which are still encrypted and perform operations such as retrieval, comparison and other operations in the encrypted data to get the correct results without having to decrypt the data in the whole process [10]. This unique feature is the reason why the requirements of homomorphic

The associate editor coordinating the review of this manuscript and approving it for publication was Chunsheng Zhu.

encryption algorithms are increasing in the field of E-voting, cloud computing and E-commerce etc. LI Bei proposed an E-voting system based on homomorphic encryption strategy in 2015 [11], which used the homomorphism of ElGamal algorithm [12]–[14] to compute voting results, and protect voters' personal information from disclosure. HUANG Shi-jie proposed a homomorphism-based multi-candidate electronic voting scheme in 2017, which used the traditional Paillier algorithm to encrypt the contents of the votes and ensure the security of the election [15]. Compared with other homomorphic encryption algorithms, Paillier algorithm has the advantage of higher operational efficiency in the field of E-voting due to only one power operation is involved in.

With the popularity of smart phones and computers, many democratic evaluations are no longer limited to simply choose agreement or disagreement. For example, in many companies, all employees need to score the leaderships for year-end assessment via a scoring application. In this case, the new electronic scoring system can achieve not only a binary choice for vote, but also a numeric score to make the evaluation results more precision. In the existing research, the mainstream homomorphic encryption schemes mainly focus on integer-based encryption. Paillier algorithm is the most representative integer-based encryption. In order to encrypt the real scores with higher precision, an electronic scoring scheme based on real Paillier encryption algorithm is put forward. In the proposed scheme, the scoring range is extended from integer to real by expressing real as large integer based on the scheme in [15]. Guaranteeing the valid identities of clients and servers and preventing any unauthorized users to login the system are the premises for the application of electronic scoring system [16]. Therefore, challenge/response authentication based on dynamic password is improved to construct a more secure and less computational two-way authentication scheme by using Horner's rule. This scheme is compared with the electronic voting scheme in [15] and the multi-candidates E-voting scheme in [11]. The results show that the improved scheme achieves higher efficient in the homomorphic encryption of real and makes the evaluation results more precision. In addition, compared with the mobile terminal voting system in [16], the time efficiency of the two-way identity authentication scheme is analyzed. The results show that the proposed scheme not only guarantees the security of the electronic scoring system because of Paillier encryption and two-way identity authentication strategy, but also increases the authentication speed due to the introducing of Horner's rule.

The major contributions of this paper are as follows:

(1) Paillier algorithm is a widely used homomorphic encryption algorithm with high efficiency and safety. It is introduced in our proposed electronic scoring scheme to effectively avoid the possibility of exposing plaintext when obtaining counting results, which can make the electronic scoring system security and efficiency.

(2) Computing the data of scoring system will inevitably produce decimal, such as averaging and weighted summation, which means the range of data processed by encryption algorithms needs to be extended from integer to real. Therefore, compound transformation is used to convert real to large integer. The number of decimal places is recorded during the converting. With the aid of large integer operation, the encryption range of Paillier algorithm is extended to real, which can make Paillier used in more real fields, such as electronic scoring, medical data, quality assessment and financial data verification etc.

(3) Most data processing systems are based on distributed multi-server structure, so the one-way identity confirmation is not safe anymore. In the proposed scheme, the Horner's rule and Paillier algorithm are applied to implement a two-way identity authentication strategy between clients and servers. In the improved strategy, client needs to confirm the connected server is valid by the verification information from the server, which is a supplement to the one-way authentication and can make the system more secure and reliable.

(4) A safety, individual, and efficient electronic scoring scheme is designed in this paper. Its safety lies in the usage of Paillier algorithm and the two-way identity authentication strategy. Its individual comes from it can realize the encryption of real and make the results accuracy. Its efficiency can be verified by the experimental evidence. The more clients the scoring system connects to, the more significant improvement of the system time efficiency is.

In addition to Section I, the remainder of this paper proceeds as follows: Section II provides an overview of the related works in the literature, such as Paillier algorithm, identity authentication technologies and Horner's rule. Two core algorithms of the improved scheme, the security and protocol analysis are illustrated in detail in Section III. In Section IV, the experimental environment and data sets are introduced, and the simulation results and performance evaluation are also presented. Finally, the paper is concluded in Section V.

II. RELATED WORKS

E-voting system needs to meet the basic safety standards proposed by Fujioka in 1992 [17]. The standards are soundness, privacy, eligibility, fairness, completeness, un-reusability and verifiability. As an improved version of E-voting system, the design of electronic scoring scheme should also meet these standards. In order to make the system security and reliable, it is necessary to design appropriate security strategies. This section mainly introduces three security strategies involved in this scheme: Paillier algorithm, identity authentication strategy and Horner's rule.

A. OVERVIEW AND ANALYSIS OF PAILLIER ALGORITHM

Paillier algorithm was proposed by P. Paillier in 1999 [18]. The algorithm is a public key cryptosystem with semantic security and additive homomorphism [19]. Its high security performance mainly depends on the difficulty in factoring

the prime factor of large integer [20]. The core parts of the algorithm are described as follows:

(1) Key Generation
- Select two large prime numbers p and q randomly and independently of each other such that $gcd(pq, (p-1)(q-1)) = 1$ ($gcd()$ is used to find the greatest common divisor. This property is assured if both primes are of equal length).
- Calculate $n = pq$, $\lambda = lcm(p-1, q-1)$ ($lcm()$ is the function to find the least common multiple).
- Select a random integer g, where $g \in Z^*_{n^2}$.
- Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse:

$$\mu = \left(L\left(g^{\lambda} \bmod n^2\right)\right)^{-1} \bmod n,$$

$$\text{where } L(x) = \frac{x-1}{n}$$

(Note: notation $\frac{x}{y}$ denotes the quotient of x divided by y, i.e., the largest integer value $z \geq 0$ to meet the relation $x \geq zy$).
- The public encryption key pk is (n, g) and the private decryption key sk is (λ, μ).
- If using p and q with equivalent length, a simpler variant of the above key generation process would be:

$$g = n+1, \quad \lambda = \varphi(n), \ \mu = \varphi(n)^{-1} \bmod n$$

where $\varphi(n) = (p-1)(q-1)$.

(2) Encryption
- Let m be the plaintext to be encrypted, where $0 \leq m < n$.
- Selected random number r, where $0 < r < n$ and $r \in Z^*_{n^2}$ (i.e. ensure $gcd(r, n) = 1$)
- Compute ciphertext c by $c = g^m \cdot r^n \bmod n^2$.

(3) Decryption
- Let c be the ciphertext to be decrypted, where $c \in Z^*_{n^2}$.
- Compute the plaintext m by $m = L\left(c^{\lambda} \bmod n^2\right) \cdot \mu \bmod n$.

The additive homomorphic can hide the addition operation to implement the encrypted counting without decryption, which can effectively prevent the exposure of vote information and make the voting security and fair. The random number r makes two equivalent votes encrypted to same value with negligible likelihood, which can protect the privacy of voters [21]–[23].

The traditional Paillier algorithm can only achieve the encryption of integer [24]. Besides a simple binary vote, scoring systems which can provide specific scores are becoming more and more popular. Because the scores and operation results may be decimals, an efficiency and security encryption algorithm is desired to encrypt real, which can provide more evaluation methods and more accurate evaluation results in real field.

B. IDENTITY AUTHENTICATION TECHNOLOGIES

Because the user's information is transmitted in an insecure network, it is essential to ensure user's privacy by identity authentication technologies [25]. These technologies can also effectively prevent the intrusion of illegal third party.

Common authentication technologies are as follows:

(1) Traditional password checking method. This method compares the login username and password with the stored data on the server to authenticate clients. The transmitted ciphertext contains users' original information, which can be easily deciphered after interception. This technology is the simplest but least secure.

(2) Certificate authority (CA) [26]. CA is a safer technology because the trusted and authoritative third-party is responsible for issuing and managing digital certificates. Use public key theory and encryption technology to provide security guarantee. But it is not suitable for electronic scoring system because of its large amount of calculation and high complexity.

(3) One-time password (OTP): OTP is the most widely identity authentication method. The core of OTP is the transmitted ciphertext is composed of username, password and an unpredictable time-related random number. Therefore, the server receives different message from the same user due to the change of login time, which improves the security of user's privacy. Due to the disposable of one-time password, even if it is stolen by the third-party during the transmission, it will be invalid when re-validated, which can enhance the security of authentication.

Challenge/response is a dynamic password-based identity authentication method which is improved in this paper due to its high security and easy implementation. The authentication steps of challenge/response are as follows:

(1) After successful registration, server will automatically hash the username and password according to hash algorithm [27], the obtained hash value is stored on the server.

(2) When users login successfully, the client automatically hashes the username and password, encrypted the hash value and the random number to produce response number, and then passes it to the server.

(3) The server receives the response number and decrypts it. It takes out the hash value and compares it with the one stored on the server to confirm the user's identity.

(4) Different users own different secret keys, even if they get the same random number, different response numbers will be generated to ensure only the legal users can login.

The traditional challenge/response method has the following three aspects need to improve.

(1) It only considers single-server environments and achieves one-way authentication. It is unfit for the multi-server environments due to the clients also need

to confirm the connected server to make them access a valid server.

(2) When the number of clients is huge and the hash information is strings, choosing a hash function with high efficiency and less conflict is essential to challenge/response method.

(3) The transmitted information needs to be encrypted, considering timeliness and privacy, an efficient and secure encryption algorithm is desired.

C. HORNER'S RULE

In the electronic scoring system, the login information converts to integer by hash operation firstly, and then encrypts the integer to transmit. The reason to transmit the encrypted hash value is the original information cannot be disclosed even if the ciphertext is stolen, which is ensured by the irreversibility of hash operation.

Choosing a suitable hash function with high efficiency and low conflict is very important, especially in the case of a large number of users. If a character string convers to an integer, the integer will be a larger number. Horner's rule is a high efficient algorithm for polynomial evaluation, especially for large integer [28].The expression of Horner's rule is shown in equation (1) and the rewritten expression is shown in equation (2).

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 \tag{1}$$

$$\begin{aligned} f(x) &= a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 \\ &= (\ldots((a_n x + a_{n-1})x + a_{n-2})x + \ldots + a_1)x + a_0 \end{aligned} \tag{2}$$

In the expression of Horner's rule, a_i represents the ASCII value of a character in the string. For example, given a character string ''afe'', ASCII values of each character are 97, 102 and 101. Each ASCII value can be presented as an n-bit binary (i.e. $n = 7$), 97 is 1100001, 102 is 1100110 and 101 is 1100101. Concatenating the binary values in order, the most efficient way to compute the long binary is the use of expression $x = 97 \times 128^2 + 102 \times 128^1 + 101 \times 128^0$, where $128 = 2^n (n = 7)$. The number of arithmetic operations can be minimized by Horner's rule $x = (97 \times 128 + 102) \times 128 + 101$. The time complexity of Horner's rule is $O(L)$, where L is the length of the character string. The conflict can be reduced due to the range of computational result is enlarged by the increasing of n. To get a large integer x without overflow, modulo operator is applied to each step in the calculation by $x = x \bmod tablesize$, where $tablesize$ is a prime number as the length of hash table.

III. ELECTRONIC SCORING SCHEME BASED ON REAL PAILLIER ENCRYPTION ALGORITHM

Considering the security, efficiency and functional requirements of electronic scoring system, Paillier algorithm and the challenge/response method are both improved.

For security, two-way identity authentication strategy is introduced to replace the traditional challenge/response authentication, which can implement client and server confirmation each other in multi-server environment. In addition, the transmitted ciphertext is hash result with a random number, which guarantees the user's identity and the scores security with the aid of Paillier's asymmetric cryptosystem [18], [29].

For efficiency, Horner's rule is used to convert strings to integers due to its high efficiency in computation. Paillier algorithm is used to encrypt and decrypt the hash value because of its additive homomorphism. The information transmitted in the network and stored on servers are both hash value, which can not only reduce the data storage on the server, but also increase the time efficiency of transmission, encryption and decryption.

For functional requirements, the electronic scoring system is different from the simple binary E-voting system. The scores, intermediate results and final scores are probably decimals. Paillier algorithm is not suitable for the new requirements without improvement due to it can only encrypt integer. The improved real Paillier algorithm enlarge the range of encryption from integer to real owing to the introduction of compound transformation. Administrators can flexibly set the traditional voting mode or scoring mode according to the actual vote/scoring requirements. If the scoring mode is set, integer score or real score can be further set to meet the needs of different environments.

The highlights of the proposed scheme are the design of two-way identity authentication strategy and real Paillier encryption algorithm. The following two sections are the detailed introduction of the two key technologies.

A. TWO-WAY IDENTITY AUTHENTICATION

During the process of voting or scoring, not only the identity of client must be confirmed by server, but also the client must ensure that the connected server is correct. This is essentially important to the distributed multi-server environments. Two-way identity authentication strategy is an improvement of the challenge/response authentication, which is more secure and less computational. During the process of authentication, each entity implements authentication and guarantees the security of the key [30].

Users must register firstly before using the scoring system, input personal information by themselves, such as username, password and so on. When a user registers username, the system automatically monitors whether the username already exists to ensure that the username is unique. After successful registration, the server will hash the username (UID) and password (PWD) by $h = horner(UID, PWD)$, the hash value h is unique for all registered users and stored on the server. The username cannot be changed after registration, while the password can be updated, the server will compute the new hash value according to the changed password. From beginning to end, the stored hash value of each client is unique.

If users need to vote or scoring, UID, PSW and a verification code Y (Y is a four-digit time-related random number)

are required. When the user login successfully, the process of two-way identify authentication is shown in Fig. 1.

The acknowledge frames Ack_u and Ack_s are used to confirm the final connection. Ack_u is stored on client and Ack_s is on server. The structures of the two frames are shown in Table 1.

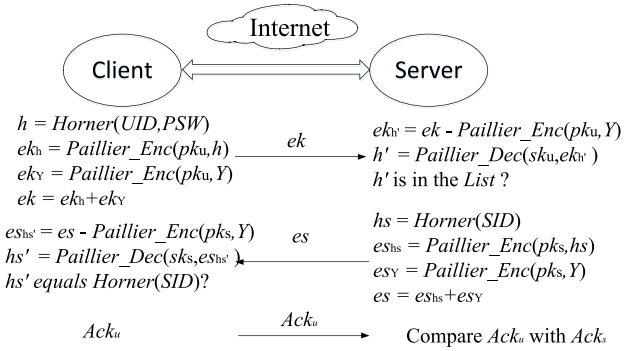

FIGURE 1. The process of two-way identity authentication.

TABLE 1. Structure of acknowledge frames.

Ack_s	$hash\_u$	$hash\_s$	f_1	f_2	f_3	T_1	T_2	T_3
Ack_u	$hash\_u$	$hash\_s$	f_1	f_2	f_3			

$hash\_u$ is the hash value of user's information (UID and PSW), $hash\_s$ is the hash value of server identity. f_1, f_2 and f_3 are the successful flags of each confirmation. T_1, T_2 and T_3 are the time of each confirmation. All hash values in frames are encrypted before transmission.

The detailed authentication steps are as follows:

Step 1: According to the UID and PSW, the client hashes the string into integer h by $h = horner(UID, PWD)$. Set $Ack_u.hash\_u = h$. Paillier encryption algorithm $Paillier\_Enc()$ is used to encrypt h and Y to obtain ciphertexts ek_h and ek_Y by $ek_h = Paillier\_Enc(pk_u, h)$ and $ek_Y = Paillier\_Enc(pk_u, Y)$, where pk_u is the public key to confirm the valid client. ek_h and ek_Y are added together to get the ciphertext ek by $ek = ek_h + ek_Y$. The ciphertext ek is sent to server. Set $Ack_u.f_1 = 1$.

Step 2: When the server receives the ciphertext ek from the client, homomorphic operation is used to subtract the ciphertext of the verification code Y from ek to get $ek_{h'}$ by $ek_{h'} = ek - Paillier\_Enc(pk_u, Y)$. Paillier decryption algorithm $Paillier\_Dec()$ is used to decrypt $ek_{h'}$ to get h' by $h' = Paillier\_Dec(sk_u, ek_{h'})$, where sk_u is the private key to confirm valid client. Determine whether h' exists in the server database, if exists, indicating the received UID is a legitimate user, set $Ack_s.hash\_u = h'$, $Ack_s.f_1 = 1$, $Ack_s.T_1 = now()$ and proceed to step 3. Otherwise the client authentication fails and the server terminates the conversation.

Step 3: Similarly, the server hashes the server identity (SID) to get the hash value hs by $hs = horner(SID)$. Set $Ack_s.hash\_s = hs$. The ciphertext es_{hs} is obtained by $es_{hs} = Paillier\_Enc(pk_s, hs)$, where pk_s is the public key to confirm the connected sever. es_Y is the ciphertext of Y encrypted by $es_Y = Paillier\_Enc(pk_s, Y)$. es_{hs} and es_Y are added to obtain ciphertext es. The server sends es back to the client. Set $Ack_s.f_2 = 1$ and $Ack_s.T_2 = now()$.

Step 4: The client receives the message es sent back from the server, using homomorphic operation to subtract the ciphertext of Y from es to get $es_{hs'}$ by $es_{hs'} = es - Paillier\_Enc(pk_s, Y)$. Decrypt $es_{hs'}$ to get hs' by $hs' = Paillier\_Dec(sk_s, es_{hs'})$, where sk_s is the private key to confirm the legal server. The client verifies that hs' is consistent with the local hash value of $Horner(SID)$. If consistent, SID is a legitimate server, set $Ack_u.hash\_s = hs'$ and $Ack_u.f_2 = 1$, proceed to step 5. If not, the server authentication fails and the client terminates the conversation with the server.

Step 5: Set $Ack_u.f_3 = 1$. The client sends the acknowledge frame Ack_u to the server.

Step 6: The server receives Ack_u. Set $Ack_s.f_3 = 1$ and $Ack_s.T_3 = now()$. The server compares Ack_u with Ack_s to ensure that whether client and server establish a successful connection.

The following is the algorithm description of the two-way identity authentication.

B. PAILLIER ENCRYPTION ALGORITHM IN REAL RANGE

The core improvement of Paillier algorithm lies in the introducing of compound transformation. Use k to record the number of decimal places in real number m before encryption. Move the decimal point to right by k-digit to realize the conversion from real to integer, and encrypt the obtained integer. Decrypt the ciphertext to get the plaintext of integer, move the decimal point to left by k-digit to obtain the original real. The scheme realizes real number encryption without changing the consistency of homomorphic operations.

The specific steps are as follows:

Step 1: Define variable k to record the number of decimal places in real number m.

Step 2: Decomposition m into $k+1$ parts according to decimal point: $m_0, m_1, m_2, \ldots$, and m_k, where m_0 is the integer part of m, $m_1, m_2, \ldots$, and m_k are the decimal part of m, m_i is a positive integer and $0 \leq m_i \leq 9$. The value of k is obtained by the length of decimal part.

Step 3: m is converted to integer m' by the operation $DoubleToInt(m)$ shown in equation (3).

$$m' = DoubleToInt\ (m) = m \times 10^k \tag{3}$$

Step 4: m' is input into the Paillier algorithm and the ciphertext c is obtained.

Step 5: The plaintext m' is obtained by decrypting c with Paillier decryption algorithm.

Step 5: Using k, m' is converted back to the original real m by the operation $IntToDouble(m')$ shown in equation (4).

$$m = IntToDouble\left(m'\right) = \frac{m'}{10^k} \tag{4}$$

Algorithm 1 Two-Way Identity Authentication Algorithms

Input: *UID*, *PSW* and verification code *Y*
Output: Acknowledgement frame Ack_u

function *ClientToServer* (*UID*,*PSW*,*Y*)
$h \leftarrow Horner(UID,PSW)$//Obtain hash values of *UID* and *PSW* by Horner's rule
$ek \leftarrow Paillier\_Enc(pk_u, h) + Paillier\_Enc(pk_u, Y)$// Encrypt *h* and *Y* by Paillier encryption algorithm and add them together.
return *ek*
end function *ClientToServer*
function *select_h* (*ek, Y*)
$h' \leftarrow Paillier\_Dec(sk,(ek\text{-}Paillier\_Enc(pk_u, Y)))$// Decrypt *ek* by Paillier decryption algorithm
if (h' is found in the user List of database)
ServerToClient(*Y,SID*)// Server verify client successfully
else
return ERROR
end function *select_h*
function *ServerToClient* (*Y,SID*)
$hs \leftarrow Horner(SID)$ //Obtain hash value of *SID* by Horner's rule
$es \leftarrow Paillier\_Enc(pk_s, hs) + Paillier\_Enc(pk_s, Y)$// Encrypt *hs* and *Y*, then add them together
return *es*
end function *ServerToClient*
function *verify_hs* (*es,Y*)
$hs\_S = Paillier\_Dec(sk_s,(es\text{-}Paillier\_Enc(pk_s, Y)))$
$hs\_C = Horner(SID)$ //Client calculates hash value of *SID* by Horner's rule
if($hs\_S == hs\_C$) // Client confirm server successfully
return Ack_u
else
return ERROR
end function *verify_hs*

Algorithm 2 Paillier Homomorphic Encryption Based on Real Number

Input: Real number *m* (plaintext)
Output: *c* (ciphertext)

function *getD_int*(*m*) //The integer part of *m* is stored in *d_integer*
d_integer = get integer part of *m*;
return *d_integer*
end function *getD_int*
function *getD_decimal*(*m*) //The deciaml part of *m* is stored in *d_decimal*
d_decimal = deciaml part of *m*;
return *d_decimal*
end function *getD_decimal*
function *getD_decimal_bits*(*m*)//*d_decimal_bits* stores the decimal places of *m*
d_decimal_bits= obtain the decimal places of *m*;
return *d_decimal_bits*
end function *getD_ decimal _bits*
function *DoubleToInt*(*m*) //Convert real to integer
$k \leftarrow getD\_\ decimal\ \_bits(m)$
return $m' = m \times 10^k$
end function *DoubleToInt*
function *DoublePaillier_Enc*(*pk*,m')
$c \leftarrow Paillier\_Enc(pk,m')$
return *c*
end function *DoublePaillier_Enc*
function *DoublePaillier_Dec*(*sk,c*)
$M' \leftarrow Paillier\_Dec(sk,c)$
return M'
end function *DoublePaillier_Dec*
function *IntToDouble*(M') //Convert integer to real
$k \leftarrow getD\_\ decimal\ \_bits(M')$
return $M = \frac{M'}{10^k}$
end function *IntToDouble*

The above two equations are only for a single real number. During encryption, decryption, addition and multiplication, the system needs to process a larger amount of decimals with different values of k. The different values need to be represented as a unified value in order to avoid misalignment in addition and multiplication. The method of the improved scheme is use the largest one as the value of k to convert the reals to integers. For example, the number of decimal places in 2.15 is $k_1 = 2$, and that of 1.3648 is $k_2 = 4$, compare k_1 and k_2, the final number of decimal places is $k = 4$. For addition operation, 2.15 is converted to 21500 by equation (3), while 1.3648 is converted to 13648, the sum of them is 35148. By equation (4), the integer 35148 can be converted to 3.5148 according to the value of k. For multiplication operation, 21500 times 13648 is 293432000, the decimal point moves to left by $2k$-digit (8-digit) to get the decimal number 2.93432000.

C. SECURITY ANALYSIS OF THE IMPROVED SCHEME

Security performance is another important index to the evaluating of electronic scoring system. Three security mechanisms are involved in the improved scheme.

(1) The login information is hashed by Horner's rule, and the transmitted message and the stored information on server are all hash values. Hash function is unidirectional. Let y is the hash value, x is the concatenating of username and password. It is difficult to find x to meet $y = horner(x)$. Hash function is anti-collision. For the given x, it is difficult to find x' to meet $horner(x) = horner(x')$, $x \neq x'$. In the improved system, the transmitted information is ciphertext of $h' = h + hy$, where $h = Horner(UID, PWD)$ and $hy = Horner(Y)$. Even if the ciphertext is decrypted by third-party, it is difficult to split h from h', which adds the difficulty to disclose the original user's information.

(2) The second is the two-way identity authentication. All clients and servers must confirm their identities each other before establishing a connection, which can ensure the client

and the server both valid. The server uses pk_u to decrypt ek and match h' with user table to ensure the user's identity. The hash value h' cannot be forged by valid client, which can avoid illegal attacks. Similarly, clients decrypt the hash value of server hs' by pk_s, and then compare hs' with the hash value y of the local stored identity of server $y = Horner(SID)$, if $y = hs'$, the server is valid. It is impossible for valid third party to counterfeit hs' to cheat the client. Finally, the server needs to compare Ack_u and Ack_s. If all flags are equal to 1, $Ack_u.hash\_u = Ack_s.hash\_u$, $Ack_u.hash\_s = Ack_s.hash\_s$ and $Ack_s.T_1 < Ack_s.T_2 < Ack_s.T_3$, which mean the connection can be established successfully.

(3) The last is Paillier encryption algorithm. The message transmitted, computed and stored are all ciphertexts encrypted by Paillier algorithm to avoid disclosing of plaintext. The same plaintext m encrypts twice, the obtained two ciphertexts are different due to the random number r, so Paillier algorithm is semantic security. In Paillier algorithm, a variable *BitLengthVal* is set as 128 to generate two 64-bit large prime numbers p and q. Use random numbers r and g, and $n = pq$ to obtain the ciphertext c of m by $c = g^m \cdot r^n \bmod n^2$. For an intercepted ciphertext c, it is impossible to reverse generation the corresponding plaintext m, because it is a problem of computing n-th residue classes. The private key $\lambda = lcm(p-1, q-1)$ is generated according to the large prime numbers p and q, which is hard to crack because of the difficulty of decomposition of larger prime factors.

The following detailed security analysis are used to illustrate the improved scheme are also meets the seven basic safety standards.

(1) **Soundness**: The dishonest voter cannot disrupt the voting.

All scores and final results stored on the server are ciphertexts. Illegal users login the server and try to modify the existing scores can cause data errors while computing ciphertexts, which can end the scores calculation. On the other hand, the system is equipped with log to record each operation, when database errors are detected, the system will automatically restore the data according to the log. These measures can effectively avoid cheating in the process of scoring.

(2) **Privacy**: All voters must be secret.

For the privacy of login message, the input string is converted to large integer by $h = horner(UID, PWD)$ firstly. The transmitted ciphertext is compose of h and verification code Y, when the ciphertext is intercepted during the transmission, it is impossible to decrypt the user's information because of the irreversibility of hash operation. For the privacy of scores, all scores are encrypted by the high security Paillier algorithm due to the two computationally difficulties, which cannot be solved in polynomial time on a classical computer.

(3) **Eligibility**: No one who isn't allowed to vote can vote.

In the two-way identity authentication, the encrypted hash value of user's information is transferred to the sever, the server decrypts the ciphertext by private key sk_u, and compares the obtained plaintext with the hash data on the server to confirm the identity of client, which can ensure that only the authorized user can vote/score.

(4) **Fairness**: Nothing must affect the voting.

To ensure the score fairness by avoiding the guidance of scoring result, the user cannot know the scores of other users, because the scores are all ciphertexts. In addition, the user cannot check the final result before scoring, which is controlled by checking choice. When the score casting is finished, the checking choice can be set true automatically to allow user to check the final results.

(5) **Un-reusability**: No voter can vote twice.

The user can only score once by scoring choice. When a user scores successfully, the scoring choice will be set false to disable the scoring function. After a successful scoring, users can only view their own scorings and check the final results, which can realize no voter can vote twice.

(6) **Completeness**: All valid votes are counted correctly.

All valid scores are encrypted by Paillier algorithm. The ciphertexts are all uploaded to the server and stored in the database. When the whole scoring process is over, all ciphertexts in the database will be calculated by real Paillier homomorphic encryption, and the final results are also stored in the form of ciphertexts.

(7) **Verifiability**: No one can falsify the result of the voting.

In the improved scheme, a history record management module is provided to retain the original ciphertexts of the scores. If verification is needed, the result calculation module can request, collect and compute the scores again to verify the published results.

D. PROTOCOL ANALYSIS

The protocol of the electronic scoring system includes the following four entities.

(1) Rater. Each certificated rater has a unique *Uno* and a hash value to present his valid identity. When scoring/vote is needed, valid rater can get a scoring ballot /vote with an authorization number *Vno* to cast. When the scoring/vote is finished, rater can check the final results.

(2) Authentication center (AC). Generate key pairs and keep them safe. Assign *Sno* and *Vno* to raters and scoring ballot/vote. Check whether the uploaded *Sno* and *Vno* are valid. Transfer the valid encrypted scores to counting center.

(3) Counting center (CC). Compute the encrypted scores without decryption, which is achieved by additive homomorphic of Paillier algorithm. Store the ciphertexts of results on the server.

(4) Publish center (PC). Release the scoring/vote announcement to the public. Publish the final results of the scoring/vote.

This protocol is different from FOO protocol. It is composed of five stages. Each stage is illustrated as followed.

(1) Registration. The designated raters/voters or users with participating wish need to register firstly. The username and password of $user_i$ are hashed as $hash_i$, encrypt and transfer to AC. AC ensures $hash_i$ is unique and allocate a system number Uno_i to $user_i$. Record the pair as (Uno_i, $hash_i$), which is used

to verify valid user. After successful registration, pk_u and sk_u are generated.

(2) Initialization. AC constructs the scoring encrypted environments, such as pk, pk_s, sk, and sk_s, according to the initialized parameters p and q. Create a unique identifier Sco_j to distinguish all scoring/vote entities. If a scoring/vote needs designated users, create a group g_k for these users, mark the participatory note Cer as $Cer_{j,k} = 1$, which means the users in group g_k has the qualification to score/vote Sco_j. Inform PA to publish the announcement.

(3) Scoring/vote. Only approved users can use the system, which is ensured by AC to check if $hash_i$ is valid via two-way identity authentication. After successful login, the system lists all scorings/votes can be performed by $user_i$, which is filtered by AC according to the Cer. The user can choose an item to score, AC will give users an authorized ballot with unique number Vno_m. For the completed scoring/vote, $user_i$ can only check the final results according to $sco\_choice = 0$ set by AC, which can ensure each voter can vote only once. $Score_n$ is encrypted as $Cmsg_n$, sent to AC and stored in database as a triple (Vno_m, Uno_i, $Cmsg_n$).

(4) Counting and publish. When Sco_i is finished, which is controlled by AC according to initial setting, such as deadline or reaching the vote limits. CC takes all $Cmsg_j$ of Sco_i from the database, compute the result $CFinalSco_i$ by additive homomorphism of Paillier algorithm and store it in database. While publishing, PC decrypts $CFinalSco_i$ and obtain the plaintext $PFinalSco_i$. PC needs to judge if $PFinalSco_i$ is real according to the number of decimal places k, if yes, transfer integer to real and present the real final score.

(5) Result verification. If the final results are challenged, creator of administrator takes all $Cmsg_j$ of Sco_i from CC to reexamine the results as $RCFinalSco_i$ by history record management module. Compare $RCFinalSco_i$ and $CFinalSco_i$ to verify the results of Sco_i.

IV. EFFICIENCY EVALUATION OF THE PROPOSED SCHEME

A. INTRODUCTION OF EXPERIMENTAL ENVIRONMENT

The simulation experimental environments are as follows:

(1) Server: Three servers are configured to implement user authentication and compute scores and so on.

(2) Client: The clients are 150 personal computers in the simulation experiment. Users can login the electronic scoring system from the browser, vote/score and check the final result.

(3) The programming platform is eclipse 7.5 Dreamweaver CS6, server is apache tomcat 8.0, programming language is java, toolkit is JDK 1.7 and database is Mysql 8.0.13.

(4)Original data sets: The experiment used the following data sets to verify the time efficiency of the proposed scheme.

- http://dataju.cn/Dataju/web/datasetInstanceDetail/208
- http://dataju.cn/Dataju/web/datasetInstanceDetail/451
- http://dataju.cn/Dataju/web/datasetInstanceDetail/226

B. EXPERIMENTAL RESULT ANALYSIS

(1) Analysis of improved Paillier algorithm

The time efficiency tests are under the condition of fixed length of key. Eight groups of real numbers were chosen in the experiment. The reals in each group have the same number of digits, i.e., the third group is composed of real numbers with 3 digits, such as 123, 15.3 and 1.57. There are 200 to 300 real numbers in each group. Each real was encrypted and decrypted by the real Paillier algorithm and record the running time, the average time is regarded as the final running time of this group.

The running time of real Paillier is composed of three parts: convert real to large integer (*RtoI*), encrypt and decrypt the large integer by traditional Paillier algorithm and convert large integer to real (*ItoR*). The times of *RtoI* and *ItoR* are shown in Fig. 2. For the traditional Paillier algorithm, the time performance is divided into two aspects, one is the time of addition and multiplication due to the two operations are the evaluation indicators of encryption algorithms, which is shown in Fig. 3. The other is the time of encryption and decryption as shown in Fig. 4.

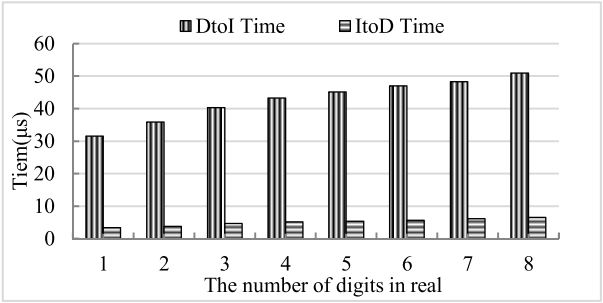

FIGURE 2. ***DtoI*** **and** ***ItoD*** **Operational Time.**

Compared with the traditional Paillier algorithm in [15], the improved Paillier realizes the encryption of real number, which can make the result more accurate. For example, the scores of candidate casted by 2 raters are 97.8 and 97.1. In traditional Paillier, the two scores are all truncated as 97, so the computing results must be with low precision. In the real Paillier, the scores are changed to 978 and 971 firstly. The integers are encrypted by Paillier algorithm. The two ciphertexts are transferred to the server and stored in database. While counting is needed, the system takes out the scores to calculate the average. The final result is stored as ciphertext of integer 9745. When the result is published, 9745 needs to convert to decimal 97.45.

As the results shown in Fig. 2 to Fig. 4, time units of *DtoI*, *ItoD*, addition and multiplication are microsecond, while that of encryption and decryption operations are millisecond. On average, the total time of encryption and decryption is about 19.08 milliseconds, and that of *DtoI* and *ItoD* is about 47.84 microseconds, which is about 0.28% of the total encryption and decryption time. The time efficiency of addition and multiplication is increased averagely by 25.99% and 42.75%. This scheme realizes real encryption by adding

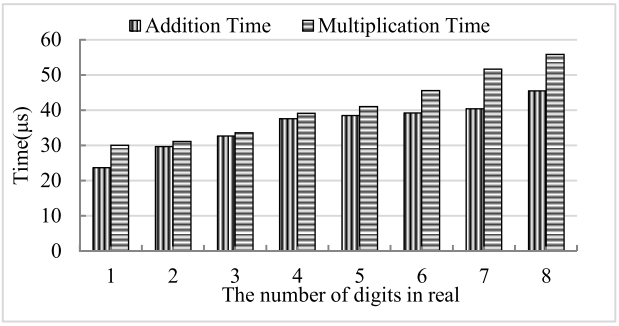

FIGURE 3. Addition and Multiplication Operational Time.

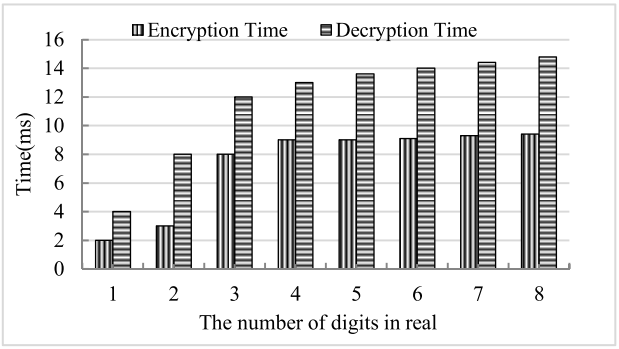

FIGURE 4. Encryption time and decryption time.

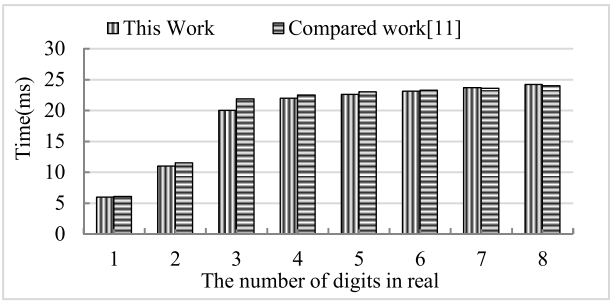

FIGURE 5. Encryption and decryption time of two real-based schemes.

two links, *DtoI* and *ItoD*, which is only a small percentage of the whole processing. Therefore, the time consumption of this method is slight increased over the original method which can even be neglected, but real number encryption and decryption is achieved by Paillier algorithm.

The improved Paillier is designed for the encryption of real number. The followed experiments are compared the time efficiency with the existing real encryption algorithms based on ElGamal cryptosystem [11]. The results are shown in Fig. 5 to Fig. 7.

As the results shown in Fig. 5, the performance of the new algorithm is slight more efficient than the method in [11] when the number of digits in real is between 2 and 4. In other cases, these two methods have similar performance. Because the numbers of digits of most data in scoring system are 2 to 4,

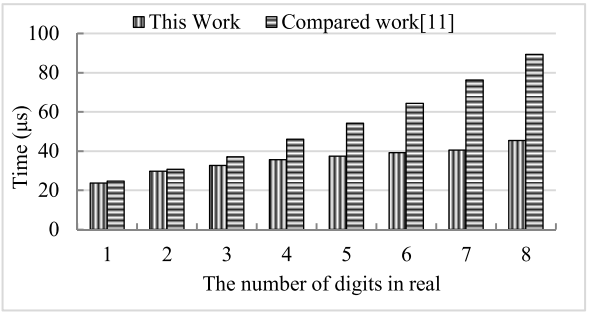

FIGURE 6. Addition time of two real-based schemes.

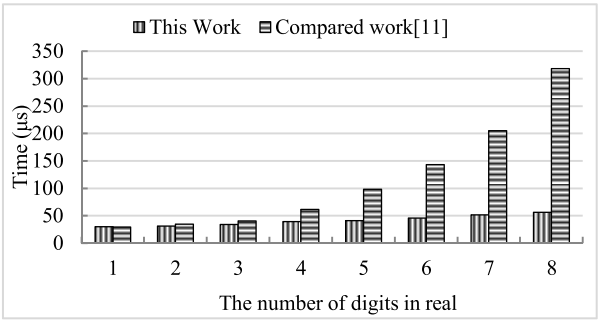

FIGURE 7. Multiplication time of two real-based schemes.

the real Paillier is more suitable for scoring system than other real-based encryption algorithms.

Analysis the results in Fig. 6 and Fig. 7, the time efficiency of the improved Paillier algorithm has a very significant improvement in addition and multiplication operations. Especially when the number of digits in real is larger than 3, time efficiency is more obvious than the compared method. For addition, the average time efficiency is increased by 25.99%, when the number of digits in real is larger than 3, the increase rate can reach 33.47%. For multiplication, the average increase rate is 42.75% and 56.06% when the number of digits in real is over 3.

(2) Analysis of two-way identity authentication

The two-way identity authentication is an improvement of traditional challenge/response. The key to this method is to find an efficient hash function to process the username and password. In the proposed scheme, Horner's rule is chosen to hash the string message. The time of two-way identity authentication can be divided into two stages, one is the time to confirm the valid client (shown in Fig. 8), and the other is the time to ensure the right connected server (shown in Fig. 9). The total time of two-way identity authentication is shown in Fig. 10. The compared scheme is also a hash-based two-way identity authentication [16].

Analyze the results in Fig. 8 to Fig. 10, the time to confirm the client and the time to ensure correct server are both shorten by introducing Horner's rule as the hash function. In Fig. 8, the average time of confirmation client is shorten by about 14.7ms, when the clients number is larger than 105, the average shorten time is about 23.78ms. In Fig. 9, the

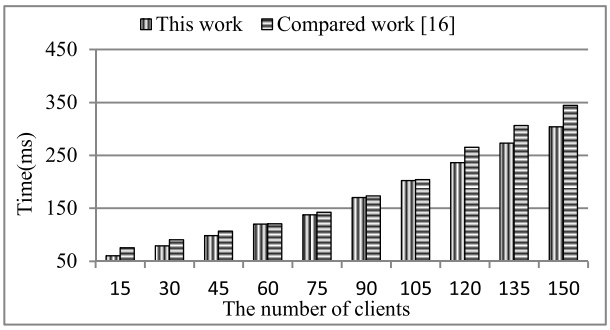

FIGURE 8. Time to confirm valid client of two hash-based schemes.

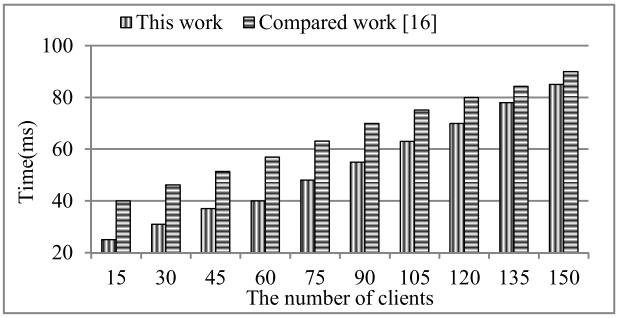

FIGURE 9. Time to confirm right sever of two hash-based schemes.

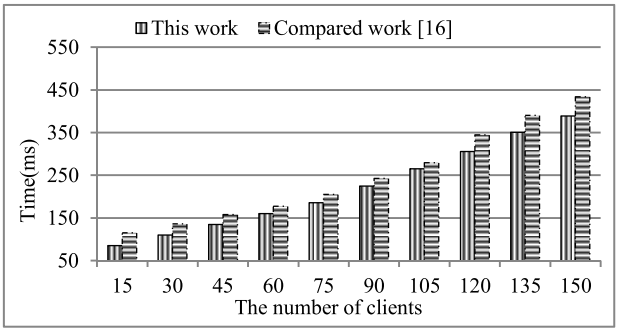

FIGURE 10. Total time of authentication of two hash-based schemes.

average time of ensuring server is about 12.52ms, when the clients number is larger than 105, the average shorten time is about 9.20ms. In Fig. 10, the shorten time of the whole authentication is 27.22ms, when the clients number is over 105, the average shorten time is about 34.43ms. According above analysis, the time of the whole authentication is mainly depends on the client confirmation, so the improvement of client confirmation decides the performance of the authentication. When the number of clients is over 105, the efficiency of this scheme is improved obviously. The more users log in, the more user information can be verified at the same time.

V. CONCLUSION

This paper presents an electronic scoring scheme based on real Paillier encryption algorithm to protect the privacy and security of the participants. The encryption and decryption range of traditional Paillier algorithm is extended from integer to real number, which can be applied in the field of real, such as medical data, quality assessment and financial data verification and so on. In addition, the two-way identity authentication is realized to confirm the security of users' privacy information. Horner's rule is effectively used to improve hash operation efficiency, further enhance the security of the scheme, and ensure the two-way identity authentication of client and server. On the basis of meeting the basic security standards of the E-voting system, the design of this scheme realizes the function of electronic scoring to better improve the precision of scores. This paper still has some problems worth further study, such as designing more efficient and practical anonymous channel to improve the security of network communication.

REFERENCES

[1] L. Cheng-Hui, "Design a novel sign mix-net scheme e-voting," M.S. thesis, Dept. Comput. Softw. Theory, Zhejiang Normal Univ., Hebei, China, 2014.
[2] R. Joaquim, P. Ferreira, and C. Ribeiro, "EVIV: An end-to-end verifiable Internet voting system," *Comput. Secur.*, vol. 32, no. 1, pp. 170–191, Feb. 2013.
[3] L. Fen-Fen *et al.*, "Receipt-freeness electronic voting scheme based on FOO voting protocol," *Comput. Sci.*, vol. 42, no. 8, pp. 180–184, Aug. 2015.
[4] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Found. Secure Comput.*, vol. 4, no. 11, pp. 169–180, 1978.
[5] G. Traverso, D. Demirel, and J. Buchmann, "Suitable homomorphic signature schemes for eVoting, smart grids, and eHealth," in *Homomorphic Signature Schemes*. Berlin, Germany: Springer, 2016, pp. 53–58.
[6] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–88, 1981.
[7] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "A homomorphic LWE based E-voting scheme," in *Proc. Post-Quantum Cryptogr. (PQCrypto)*, Fukuoka, Japan, 2016, pp. 245–265.
[8] K. Peng and F. Bao, "Efficient multiplicative homomorphic e-voting," in *Information Security*. Boca Raton, FL, USA: Springer, 2010, pp. 381–393.
[9] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.
[10] W. She, Z.-H. Gu, X.-K. Lyu, Q. Liu, Z. Tian, and W. Liu, "Homomorphic consortium blockchain for smart home system sensitive data privacy preserving," *IEEE Access*, vol. 7, pp. 62058–62070, May 2019.
[11] L. Bei, "Electronic voting systems based on homomorphic encryption scheme," *J. Comput. Appl.*, vol. 35, no. S1, pp. 66–68 and 88, Jun. 2015.
[12] A. S. Sodiya, S. A. Onashoga, and D. I. Adelani, "A secure e-voting architecture," presented at the 8th Inf. Technol., New Gener. (ITNG), Las Vegas, NV, USA, Apr. 2011.
[13] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," in *Advances in Cryptology—CRYPTO*. Berlin, Germany: Springer, 1998, pp. 13–25.
[14] X. Yang, X. Yi, S. Nepal, A. Kelarev, and F. Han, "A secure verifiable ranked choice online voting system based on homomorphic encryption," *IEEE Access*, vol. 6, pp. 20506–20519, Mar. 2018.
[15] H. Shi-Jie and H. Xuan, "An electronic voting scheme realizing multi candidates based on homomorphism," *Comput. Appl. Softw.*, vol. 34, no. 3, pp. 284–288, Mar. 2017.
[16] H. Mei-Da, "The design and implementation of a mobile terminal voting system," M.S. thesis, Dept. Comput. Appl. Technol., Agricult. Univ. Hebei, Hebei, China, 2014.
[17] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *Advances in Cryptology—AUSCRYPT'92*. Berlin, Germany: Springer-Verlag, 1992, pp. 244–251.
[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. EUROCRYPT*, Prague, Czech Republic, 1999, pp. 223–238.

[19] R. Harerimana, S.-Y. Tan, and W.-C. Yau, "A Java implementation of paillier homomorphic encryption scheme," presented at the 5th Int. Conf. Inf. Commun. Technol. (ICoICT), Washington, DC, USA, May 2017.
[20] S. M. Anggriane, S. M. Nasution, and F. Azmi, "Advanced e-voting system using Paillier homomorphic encryption algorithm," in *Proc. Int. Conf. Inform. Comput. (ICIC)*, Mataram, Indonesia, Oct. 2016, pp. 338–342.
[21] I. Damgård, M. Jurik, and J. B. Nielsen, "A generalization of Paillier's public-key system with applications to electronic voting," *Int. J. Inf. Secur.*, vol. 9, no. 6, pp. 371–385, Dec. 2010.
[22] H. Hussien and H. Aboelnaga, "Design of a secured e-voting system," presented at the Int. Conf. Comput. Appl. Technol. (ICCAT), Sousse, Tunisia, Jan. 2013.
[23] M. Nassar, A. Erradi, and Q. M. Malluhi, "Paillier's encryption: Implementation and cloud applications," in *Proc. ICAR*, Beirut, Lebanon, Oct. 2015, pp. 1–5.
[24] X. Liu, K.-K. R. Choo, R. H. Deng, R. Lu, and J. Weng, "Efficient and privacy-preserving outsourced calculation of rational numbers," *IEEE Trans. Depend. Sec. Comput.*, vol. 15, no. 1, pp. 27–39, Feb. 2018.
[25] T. Gao, X. Deng, Y. Wang, and X. Kong, "PAAS: PMIPv6 access authentication scheme based on identity-based signature in VANETs," *IEEE Access*, vol. 6, pp. 37480–37492, May 2018.
[26] B. Zhou, H. Li, and L. Xu, "An authentication scheme using identity-based encryption & blockchain," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Natal, Brazil, Jun. 2018, pp. 556–561.
[27] D. Giry. *Keylength—Cryptographic Key Length Recommendation BlueKrypt Version 31.0*. Accessed: Jun. 10, 2018. [Online]. Available: https://www.keylength.com/en/compare/
[28] M. Ceberio and L. Granvilliers, "Horner's rule for interval evaluation revisited," *Computing*, vol. 69, no. 1, pp. 51–81, Sep. 2002.
[29] L. Yao and X. Shuai, "Accelerate the paillier cryptosystem in CryptDB by Chinese remainder theorem," in *Proc. 20th Int. Conf. Adv. Commun. Technol. (ICACT)*, Chuncheon-si, South Korea, Feb. 2018, pp. 74–77.
[30] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for Internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

WANG RONG-BING received the Ph.D. degree in management science from Liaoning University, in 2015, where he is currently an Associate Professor. His research interests include data mining and cloud computing. He is a member of CCF.

LI YA-NAN is currently pursuing the master's degree with Liaoning University. She is also an Assistant with the Shandong Labor Vocational and Technical College. Her research interests include big data and information security.

XU HONG-YAN received the M.S. degree in computer software and theory from Liaoning University, in 1999, where she is currently an Associate Professor and an M.S. Supervisor. Her research interests include deep Web and personal recommendation.

FENG YONG received the Ph.D. degree in management science and engineering from Northeastern University, in 2007. He is currently a Professor and an M.S. Supervisor with Liaoning University. His research interests include data mining and personal recommendation. He is a member of CCF.

ZHANG YONG-GANG received the Ph.D. degree in computer software theory from Jilin University, in 2005, where he is currently a Professor and a Ph.D. Supervisor. His research interests include constraint solving, constraint optimization, and data mining. He is a member of CCF.