A Homomorphism Based Zero Knowledge Proof of Authentication for Chinese Remainder Theorem Based Secret Sharing

Parthajit Roy(✉)

Department of Computer Science, The University of Burdwan,
Burdwan 713104, West Bengal, India
roy.parthajit@gmail.com

Abstract. This paper proposes a secure computation model for zero knowledge proof of authentication for Chinese remainder theorem based secret sharing method. The model considers frauds in the system for more realism. The proposed model uses cryptographic hash function and discrete logarithm based ElGamal cryptosystems for its computations. The model computes the authentication in a homomorphic domain so that the information is not revealed, no matter whether all the persons are true shareholders or some of them are fraud. The proposed model definitely concludes that the system has a fraud.

Keywords: Homomorphic computation · Secure Hash Function
Discrete logarithm · Chinese remainder theorem
Secret sharing · Secure computation · Zero-Knowledge proof

1 Introduction

After the path breaking proposal of Public Key Cryptography by Diffie and Hellman [4] in the year 1976, four decades have gone and during this period, the subject has undergone many changes and has reached to its adolescence. The world has observed classical cryptosystems like RSA system [16], ElGamal system [5] to modern systems like elliptic curve based cryptography [10,12]. Several cryptographic protocols like digital signature, secure hash, secret sharing and many more have also been developed in the last four decades. There are a variety of such tools and protocols that made the field of cryptography rich enough.

All the techniques discussed above relies on the fact that the Eavesdropper is an outside entity. i.e. not the parties involved in the protocol. So, almost all of the techniques are computations on plain text to generate cipher texts. The Modern cryptographic protocols are even more realistic. They assume that the intruder may be present inside the system. So, trusting the insiders and revealing information to them may lead to information leakage. So, modern cryptographic

J. K. Mandal and D. Sinha (Eds.): CSI 2017, CCIS 836, pp. 562–572, 2018.
https://doi.org/10.1007/978-981-13-1343-1_47

research directing towards secure computation or computation over cipher texts. Such types of computations are called homomorphic computations [22].

Homomorphism is a branch of mathematics where mapping between two of finite groups and their properties are studies. Homomorphic computation is a fascinating model where the computation in the homomorphic domain retains the computations of the plain text. Some early such proposal was given by Goldwasser et al. [7] where a square root over the product of two large primes was computed in the homomorphic cipher domain. Homomorphic computation on nth residue classes has been proposed by Paillier [15]. Okamoto et al. [14] has proposed factoring based secure computing. A pairing bases secure homomorphic computation has been proposed by Boneh et al. [3]. In their method, they have used Elliptic curve based bilinear Weil pairing for homomorphic encryption. An elaborated foundation on homomorphic encryption is done by Yi et al. [22].

Homomorphic encryption has a number of good applications in the field of modern cryptography. Its main objective is to keep data or identity, or location of the user secret in the whole process of computation. In secure face detection, homomorphic computation has successfully been applied by Ma et al. [11] and Nassar et al. [13] whereas in the field of Internet-of-Things (IoT) homomorphic computation has been applied by Zouari et al. [23].

Homomorphism is an important tool that can be used for privacy. Privacy is an unconditional issue in the field of Bioinformatics where the gene codes become exposed for research and computations. A secret genome search based on homomorphism in a decentralized architecture has been developed by Yamamoto et al. [20]. A privacy preservation model for location based service using K nearest neighborhood has been proposed by Yi et al. [21].

In this paper, a secure secret sharing model has been proposed. Secret sharing, on the other hand, is a branch of cryptography where a secret is being distributed among n shareholders in such a way that any combination of k persons can fully retrieve the secret and any combination of less than k persons does not get slightest idea of the secret. The first such proposal was given by Shamir [19]. His proposal was based on interpolation of polynomials. Independently, secret sharing based on linear equations was proposed by Blakley [2].

Chinese Remainder Theorem (CRT) based Secret sharing is another stunning branch of secret sharing where the secret is shared among n persons using CRT. An early such model has been proposed by Asmuth et al. [1]. A generalized model of secret sharing on Chinese Remainder Theorem has been proposed Iftene [8]. He also showed how to use it in secure e-voting system.

Zero Knowledge proof of identity is another interesting branch of cryptography where the proof of ownership of an information is established without revealing the information itself and thus avoid leakage of information. The first such model was proposed by Goldwasser et al. [6]. Some recent application in the field of wireless sensor network has been proposed by Khernane et al. [9] whereas an application of the same in Mobile to Mobile communication has been proposed my Schukat et al. [18]. A state of the art discussions on Zero-Knowledge proof up to year 2006 has been done by Rosen [17].

In this paper the problem has been identified as follows. Let a secret has been shared using CRT among n users by a designated authority. For the purpose of further security, the authority did not disclose the identity of the shareholders. This means, no shareholder knows who the other secret shareholders are. Let in the absence of the authority, the organization needs to know the secret. So, the organization announces the news in their website and asks the secret shareholders to be present in a meeting. In the meeting, no body knows whether the other persons are true share holders or a fraud. As there may be frauds in the system revealing of information is a bad idea. The problem is, as the shareholders are unknown to each other, they don't want to revile their secrets without being sure about the identity of the other share holders. This paper proposes a solution that uses Secure Hash Functions and ElGamal encryption based models for authentication of the share holders in case of Chinese remainder theorem based secret sharing without revealing the information. Proposed model is a non-threshold secret sharing model, i.e. all the shareholders have to be present to recover the secret. The model gives a clear indication whether all the shareholders are true shareholders or not. The proposed model is also computationally efficient in the sense that it uses a single round for its computations.

The rest of the paper is organized as follows. Section 2 discusses the mathematical prerequisites that are needed to describe the model. Section 3 proposes the model of zero knowledge authentication. Strengths ans Weaknesses of the proposed model have been discussed in Sect. 4. Conclusion and future scopes are presented in Sect. 5 and references come thereafter.

2 Mathematical Preliminaries

This section deals with the mathematics related to group homomorphism, ElGamal cryptosystem and Chinese remainder theorem. The section also gives a brief idea of cryptographic hash functions and zero knowledge proof of identity.

A Homomorphism is a mapping $\psi : G \rightarrow H$, for two groups $(G, *)$ and $(H, \circ)$ s.t. $\forall a, b \in G, \psi(a), \psi(b) \in H$ and that $\psi(a * b) = \psi(a) \circ \psi(b)$. This also implies that if e_G, e_H are the identity elements of the group G and H respectively, then $\psi(e_G) = e_H$, which further implies that $\forall a \in G, \psi(a^{-1}) = \psi(a)^{-1}$ and therefore G forms a normal subgroup of H. Further, if the mapping $\psi(.)$ is a bijection and the set H is same as set G, then the homomorphism is called endomorphism.

The striking feature of homomorphism is that the computation can be performed in the homomorphic domain instead of the original group domain. Let us try to realize this by an example, Let us suppose that $a * b$ needs to be computed for some $a, b \in G$. What can be done is, instead of computing $a * b$ directly, they can be transfered to their corresponding homomorphic images as $\psi(a)$ and $\psi(b)$ and thereafter compute $h = \psi(a) \circ \psi(b) \in H$. Thereafter, compute the pre-image of h as $\psi^{-1}(h) = a * b$. Homomorphism ensures that $a * b$ will be recovered in the original domain.

The main advantage such type of indirect computation is secrecy of the original values a and b. Suppose Alice has a secret a and Bob has another secret b and they want to compute $a * b$ without revealing their actual information.

They will transform their plain text information a and b to $\psi(a)$ and $\psi(b)$ via homomorphic mapping and will reveal $\psi(a)$ and $\psi(b)$. The computation on $\psi(a)$ and $\psi(b)$ will be done and from $\psi(a) \circ \psi(b)$, which is actually $\psi(a * b)$ the pre-image will be computed. (For the time being, assume that from the product, they cannot recover the opponent's secret). The whole security of such model is based on the assumption that from $\psi(a), \forall a \in G$, is cryptographically hard. Obviously, computing a from $\psi(a) \forall a \in G$ must be easy if some extra information is known. This is called one way trap door functions. To summarize, an homomorphism is suitable for secure homomorphic computation, if and only if it has one way property.

ElGamal [5] system is the direct outcome of the discrete logarithm problem proposed by Diffie and Hellman [4]. The mathematical description of the system is as follows. Let p be a large prime and G is a multiplicative group realized over p. Let g be a be a generator of the group G. Let Alice owns these information. She then choses a secret number K_s and keeps it secret. She then computes P_k^A as public key and publishes $P_k^A = (G, g, K_p)$ where public key K_p is generated using Eq. 1.

$$K_p = g^{K_s} \; mod \, p \tag{1}$$

Let Bob wants to send a plain text to Alice. To send a plain text m Bob choses a random number $K_e, (K_e < p)$. This is known as *ephemeral key*. Bob then computes the following using Eqs. 2 and 3.

$$c_1 = g^{K_e} \; mod \, p \tag{2}$$

$$c_2 = m \times K_p^{K_e} \; mod \, p \tag{3}$$

He then sends the pair (c_1, c_2) over the insecure channel to Alice.

Alice in due course, performs her computations using Eq. 4 and subsequently she computes t^{-1}. She then multiplies t^{-1} with c_2 and retrieves the plain text m. This can be realized from Eq. 6.

$$t = c_1^{K_s} \tag{4}$$

$$t^{-1} = (c_1^{K_s})^{-1} \; mod \, p \tag{5}$$

$$\begin{aligned} t^{-1} \times c_2 \; mod \, p &= \left(c_1^{K_s}\right)^{-1} \times c_2 \; mod \, p \\ &= \left(\left(g^{K_e}\right)^{K_s}\right)^{-1} \times \left(m \times (K_p)^{K_e}\right) \; mod \, p \\ &= \left(g^{K_e \times K_s}\right)^{-1} \times \left(m \times \left(g^{K_s}\right)^{K_e}\right) \; mod \, p \\ &= \left(g^{K_e \times K_s}\right)^{-1} \times \left(m \times g^{K_s \times K_e}\right) \; mod \, p \\ &= m \times \left(g^{K_e \times K_s}\right)^{-1} \times \left(g^{K_s \times K_e}\right) \\ &= m \times 1 \\ &= m \end{aligned} \tag{6}$$

The striking fact is that the ElGamal structure has endomorphism. To understand this, let us first define the homomorphic mapping using Eq. 7.

$$\psi(a) = g^a \; mod \; p \tag{7}$$

Now, consider two integers m_1 and m_2 are in possession of two different persons B and C and they want to compute $m_1 * m_2$ secretly. Let Alice has an ElGamal system (G, g, K_p). B and C can compute $m_1 * m_2$ secretly using Alice's ElGamal system. To do this, let B and C choses ephemeral keys K_{eb} and K_{ec} respectively. Thereafter B and C encrypts their plain texts m_1 and m_2 using Alice's public key. Let B generates (c_{1b}, c_{1c}) and C generates (c_{1c}, c_{2c}) respectively. They multiplies them using the following equation and sends the result to Alice for decryption.

$$(c_{1b}, c_{2b}) \times (c_{1c}, c_{2c}) = (c_{1b} \times c_{1c}, c_{2b} \times c_{2c}) \tag{8}$$

Clearly, multiplication is done in the homomorphic domain so none of B, C will be able to learn about other's plain text. What Alice decrypts is as follows.

$$t = (c_{1b} \times c_{1c})^{K_s} \tag{9}$$

and computes t^{-1} the inverse using Eq. 5. She then multiplies t^{-1} with $(c_{2b} \times c_{2c})$. The result she gets is,

$$\begin{aligned}
& t^{-1}(c_{2b}.c_{2c}) \; mod \; p \\
&= \left((c_{1b} \times c_{1c})^{K_s}\right)^{-1} \times (c_{2b} \times c_{2c}) \; mod \; p \\
&= \left(\left(g^{K_{eb}} g^{K_{ec}}\right)^{K_s}\right)^{-1} \left(m_1 \left(K_p\right)^{K_{eb}} \times m_2 \left(K_p\right)^{K_{ec}}\right) \; mod \; p \\
&= \left(\left(g^{K_{eb}} g^{K_{ec}}\right)^{K_s}\right)^{-1} \left(m_1 \left(g^{K_s}\right)^{K_{eb}} \times m_2 \left(g^{K_s}\right)^{K_{ec}}\right) \; mod \; p \\
&= \left(\left(g^{K_{eb}} g^{K_{ec}}\right)^{K_s}\right)^{-1} \left(m_1 m_2 \left(g^{K_{eb}}\right)^{K_s} \left(g^{K_{ec}}\right)^{K_s}\right) \; mod \; p \\
&= m_1 m_2 \left(\left(g^{K_{eb}} g^{K_{ec}}\right)^{K_s}\right)^{-1} \left(g^{K_{eb}} g^{K_{ec}}\right)^{K_s} \; mod \; p \\
&= m_1 m_2
\end{aligned} \tag{10}$$

Thus, Alice gets the multiplication of m_1 and m_2. The important observation is that if $\psi(m_1) * \psi(m_2)$ is performed in the homomorphic domain, then the resultant is multiplication in the original domain. This is what this paper is going to exploit.

The third thing that a cryptographic hash function has been used. A cryptographic hash function is a mapping from input of arbitrary length to an output of fixed length in such a way that is impossible to revert back. Hash functions has a number interesting properties. Given the output of the secure hash, it is impossible to reverse engineered to get back the plain text. Secondly, hash functions are preimage resistant. i.e. finding a collision is computationally impossible.

Hash functions are typically used for secret agreements. In this paper, standard SHA-512 (Designed by NSA) has been used for authentication protocol design.

Chinese remainder theorem says that given n linear congruent equations $x = r_i \, mod \, m_i, \forall i = 1, 2, \cdots, n$ with $gcd(m_i, m_j) = 1, \forall i, j$ with $i \neq j$, x has a unique solution in modulo $m_1 \times m_2 \times \cdots \times m_n$ domain.

In CRT based secret sharing method, a secret x is considered, and the remainder r_i is computed when it is divided by some prime p_i (Relatively co-primes are also works well) for $i = 1, \cdots, n$. The secrets (r_i, p_i) are transfered to person P_is. In this way the shares are shared. Whenever the secret needs to be reconstructed, the shareholders solve the linear congruent equations as follows.

$$x = r_1 \, mod \, p_1 \tag{11}$$

$$x = r_2 \, mod \, p_2 \tag{12}$$

$$\vdots$$

$$x = r_n \, mod \, p_n \tag{13}$$

The Chinese remainder theorem ensures that it has a unique solution. Further, if one of the information is missing, x cannot be guessed. There are thresholding models also for Chinese remainder theorem based secret sharing, but the present proposed model considers non thresholding model only.

All the necessary mathematics have been discussed in this section. Now we are ready to present the proposed model. Next section proposes this.

3 Proposed Zero Knowledge Authentication Model

The proposed model is a zero knowledge proof based authentication model for Chinese remainder theorem (CRT) based secret sharing. In the proposed model, it is assumed that a designated authority (DA) will share a secret x among n persons using CRT mased secret sharing model. In the proposed model, instead of co-prime divisors, the DA choses primes for modulus of CRT based secret sharing model. Typically, large primes. The DA then computes the shares. Let the shares are $(r_1, p_1), (r_2, p_2), \cdots, (r_n, p_n)$ and have been distributed to shareholders $S_1, S_2, \cdots, S_n$ respectively. He then multiplies all the primes and produces the hash of the result and makes it public. The computation of DA is summarized in Table 1.

It is clear that making the *aggrement* public is absolutely safe. No one can guess a bit of idea about the values *mult* of the original values of $p_1, p_2, \cdots, p_n$.

For authentication of the shareholders, some protocols are followed. The protocol is simple and deterministic. The shareholders have to compute the value *mult*. This means, they have to know the values $p_1, p_2, \cdots, p_n$. This is the only way to get the value of *aggrement*.

This paragraph focuses on the secret generation process. As, the DA didn't disclose the identities of the shareholders, it is hard for any shareholder to authenticate other share holders. Suppose, in the absence of DA, the company

Table 1. The computations of the designated authority.

Designated authority's computation
1. DA identifies the information x that he wants to share secretly. 2. DA selects suitable large primes $p_1, \cdots, p_n$ 3. DA breaks the information x using CRT based secrets $(r_1, p_1), (r_2, p_2), \cdots, (r_n, p_n))$ and shares them to holders $S_1, S_2, \cdots, S_n$ respectively. 4. DA computes $mult = p_1 \times p_2 \times \cdots \times p_n$. 5. DA computes $aggrement = SHA(mult)$ 6. DA makes the number of holders n and *agreement* public in his web site.

needs to recover the secret. The company places the news in their website. In response, n persons turn out and claim that they are the actual shareholders. The next task of the company is to ask them to disclose their secrets. But, if one or some of them are frauds, then the information cannot be retrieved and the secrets of the valid shareholders will be disclosed to the frauds.

In the proposed model, no one, not even the company secretary, who is conducting the recovery process, will be able to learn about the secrets until all the n shareholders are authenticated. So, before revealing the actual secrets, the shareholders will be asked to prove their identity. This means, that they will have to prove that they posses the secrets. Shareholders then starts their protocol according to Table 2.

Table 2. The computations of the shareholder for proving their identity.

Shareholder's computation
1. Some person P (Secretary can be a good choice for this) makes her ElGaml system public. i.e. She makes (g, p, P_k) public. 2. Each S_i then does the following sequentially. S_1 encrypts his prime p_1 with the ElGamal system as $Secret_1 = (c_{11}, c_{21})$ and passes the result to S_2. S_2 then encrypts his prime p_2 with the ElGamal system as $Secret_2 = (c_{12}, c_{22})$. 3. S_2 then multiplies them as $secret_1 \times secret_2$ and passes to S_3. 4. S_3 then computes his $secret_3$ and multiplies $secret_1 \times secret_2 \times secret_3$. 5. The process continues up to nth shareholder computes $encrypt = secret_1 \times secret_2 \times \cdots \times secret_n$. 6. Person P then asked to recover the secret from *encrypt*. 7. Due to homomorphism of ElGamal as shown in equation 10 , P computes $result = p_1 \times p_2 \times \cdots \times p_n$ 8. all the hash of the *result* is then computed. 9. If $SHA(result) = aggrement$, computed by DA as shown in Table 1, then the persons are all true shareholders. Otherwise there is fraud in the system.

Let us explain the working principle of the model. From Eq. 10 it is clear that ElGaml system has homomorphism property for multiplication. Now, whenever $Secret_1 = (c_{11}, c_{21})$ has been created by shareholder S_1 if it is passed to S_2, S_2 will not be able to recover the value of p_1 because it is encrypted using Secretary's public key. So, other than secretary, nobody can recover p_1. Shareholder S_2 then encrypts his prime p_2 using Secretary's public key and multiplies it with $secret_1$ and passes the result to shareholder S_3. S_3 also cannot recover the message for same reason. In this way all the messages are multiplied. The final outcome is $encrypt = secret_1 \times secret_2 \times \cdots \times secret_n$. The value encrypt is then submitted to the secretary for decryption. If it is assumed that shareholder S_1 choses ephemeral key e_1, S_2 chooses $e_2 \cdots$ S_n choses e_n, then the secretary computes the following using his private key (G, g, K_s) (K_s is the corresponding secret exponent of his public exponent K_p).

$$t = (c_{11} \times c_{12} \times \cdots \times c_{1n})^{K_s} \tag{14}$$

and computes t^{-1} the inverse using Eq. 5. She then multiplies t^{-1} with $(c_{21} \times c_{22} \times \cdots \times c_{2n})$. The result she gets is,

$$\begin{aligned}
& t^{-1}.c_2 \; mod\, p \\
&= \left((c_{11} \times \cdots \times c_{1n})^{K_s}\right)^{-1} \times (c_{21} \times \cdots \times c_{2n}) \; mod\, p \\
&= \left(\left(g^{K_{e1}} \times \cdots \times g^{K_{en}}\right)^{K_s}\right)^{-1} \left(p_1 \left(K_p\right)^{K_{e1}} \times \cdots \times p_n \left(K_p\right)^{K_{en}}\right) \; mod\, p \\
&= \left(\left(g^{K_{e1}} \times \cdots \times g^{K_{en}}\right)^{K_s}\right)^{-1} \left(p_1 \left(g^{K_s}\right)^{K_{e1}} \times \cdots \times p_n \left(g^{K_s}\right)^{K_{en}}\right) \; mod\, p \\
&= \left(\left(g^{K_{e1}} \times \cdots g^{K_{en}}\right)^{K_s}\right)^{-1} \left(p_1 \times \cdots \times p_n \left(g^{K_{e1}}\right)^{K_s} \cdots \left(g^{K_{en}}\right)^{K_s}\right) \; mod\, p \\
&= p_1 \times \cdots \times p_n \left(\left(g^{K_{e1}} \times \cdots \times g^{K_{en}}\right)^{K_s}\right)^{-1} \left(g^{K_{e1}} \times \cdots \times g^{K_{ec}}\right)^{K_s} \; mod\, p \\
&= p_1 \times \cdots \times p_n
\end{aligned} \tag{15}$$

As a concrete example, let the secretary's ElGamal prime is $p = 83$ and the generator $g = 2$. Let the secretary's private key is $k_s = 11$. So, secretary's public key is $K_p = g^{K_s} \; mod\, p = 2^{11} \; mod\, 83 = 56$. Let B and C are the two shareholders. Let B has a prime for CRT and C has a prime for CRT. Let B's prime is $m_b = 5$ and C's prime is $m_c = 11$. Let B chooses the ephemeral key as $K_{eb} = 13$. B computes $c_{1b} = g^{K_{eb}} = 2^{13} \; mod\, 83 = 58$. He then encrypts the message and gets cipher text $c_{2b} = m_b \times K_p^{K_{eb}} \; mod\, p = 5 \times 56^{13} \; mod\, 83 = 64$. He then sends this pair $(c_{1b}, c_{2b}) = (58, 64)$ to C.

Let C's ephemeral key is $c_{ec} = 18$. So, he computes $c_{1c} = g^{K_{ec}} = 2^{18} \; mod\, 83 = 30$. He then encrypts the message and gets the cipher text $c_{2c} = m_c \times K_p^{K_{ec}} \; mod\, p = 11 \times 56^{18} \; mod\, 83 = 68$. So, his pair is $(c_{1c}, c_{2c}) = (30, 68)$. He then multiplies $((c_{1b}, c_{2b}) \times (c_{1c}, c_{2c})) \; mod\, p = ((58, 64) \times (30, 68)) \; mod\, 83$ and gets $(80, 36)$ and send this number to the secretary.

Secretary computes $80^{K_s}\ mod\ p = 80^{11}\ mod\ 83 = 58$. Then computes its inverse as $inv(58)\ mod\ 83 = 73$ and multiplies it with the second component i.e. she computes $73 \times 36\ mod\ 83$ and gets the result 55, which is the product of the two primes 5 and 11 of B and C respectively.

In this way, from Eq. 15, the secretary computes the product $result = p_1 \times \cdots \times p_n$. But $result$ is a product of large primes and the factorization of the product of large primes is a computationally impossible task. Also, from this product (which is now made public by secretary), shareholders cannot reveal any information. This is because, they can divide the $result$ by their number but the resultant is a product of $n - 1$ primes. So, if n is greater than or equal to 3, then this resultant is at least a product of two large primes, which is also computationally impossible to factorize. Thereafter the hash of the $result$ has been computed. If the hash matches with the $agreement$ value proposed by the DA, then the shareholders are all authentic and they go for CRT based retrieval of secret x, otherwise the system has frauds and the process stops.

4 Security and Weaknesses

Last section proposed and explained homomorphic model for authentication model. This section discusses the strengths and weaknesses of the model.

The proposed model finally reveals the product of some large primes. This is absolutely safe. In the field of public key cryptography, it is widely known that product of large prime is very hard to factor. So, from the product of the large primes no information about the individual prime numbers will be revealed. If one or more of the persons are fraud, even then also, the product will not give any idea about individual primes. Finally, as the hash of the product is made public, no outside eavesdropper can learn anything about the product. So, the system is secure from both internal cheaters as well as from external intruders.

Though the model is secure, the whole computation is computationally intensive. This is because the message in the present case is a product of a number of large primes. Therefore for successful computation on n shares, the ElGamal modulo prime p should be like $p > p_1 \times \cdots \times p_n$. But small primes cannot be chosen. In that case factorization will be easier. So, the present paper restricts the number of shares. For this reason, in a practical sense, the number of shareholders should not be too large.

5 Conclusion

This paper proposes zero knowledge proof of authentication for Chinese remainder theorem based secret sharing method. The proposed model uses homomorphic computation for authentication purpose. For this, the model uses ElGamal based encryption system. The model is a realistic model that assumes the presence of frauds both withing the system as well as in outside. Further, the proposed model doesn't reveal any information about the secret until all the true shareholders are true shareholders. The Model also considers secure hash functions for hiding the plain text information and makes the digest of the information public for proving the authentication of the shareholders.

Though the model is promising, there are some weaknesses also. First of all, the model is computationally intensive. This is because the model computes huge number of exponentiation on modulo large prime domain. Secondly, the model works only for is a non-threshold CRT based secret sharing schemes. i.e. all the shareholders have to come together.

For the first weakness, the model can be further improvised to achieve better computational efficiency. For this Elliptic curves cryptosystems may be used which works well with relatively smaller primes. For the second problem, hash of all possible combinations of ${}^{n}c_k$ multiplications of primes can be made public by the designated authority DA. Further some more sophisticated mathematical models can be used as a remedy of both the weaknesses.

References

1. Asmuth, C., Bloom, J.: A modular approach to key safeguarding. IEEE Trans. Inf. Theory **29**(2), 208–210 (1983)
2. Blakley, G.R.: Safeguarding cryptographic keys. In: International Workshop on Managing Requirements Knowledge, p. 313 (1979)
3. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_18
4. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976). https://doi.org/10.1109/TIT.1976.1055638
5. ELGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
6. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing. STOC 1985, pp. 291–304, ACM, New York (1985). https://doi.org/10.1145/22145.22178
7. Goldwasser, S., Micali, S.: Probabilistic encryption and how to play mental poker keeping secret all partial information. In: Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing. STOC 1982, pp. 365–377, ACM, New York (1982). https://doi.org/10.1145/800070.802212
8. Iftene, S.: General secret sharing based on the chinese remainder theorem with applications in e-voting. Electron. Notes Theor. Comput. Sci. **186**(Supplement C), 67–84 (2007). Proceedings of the First Workshop in Information and Computer Security (ICS 2006). http://www.sciencedirect.com/science/article/pii/S1571066107004604
9. Khernane, N., Potop-Butucaru, M., Chaudet, C.: BANZKP: a secure authentication scheme using zero knowledge proof for WBANs. In: 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 307–315, October 2016
10. Koblitz, N.: Elliptic curve cryptosystems. Math. Comput. **48**(177), 203–209 (1987)
11. Ma, Y., Wu, L., Gu, X., He, J., Yang, Z.: A secure face-verification scheme based on homomorphic encryption and deep neural networks. IEEE Access **5**, 16532–16538 (2017)

12. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-39799-X_31. http://dl.acm.org/citation.cfm?id=18262.25413
13. Nassar, M., Wehbe, N., Bouna, B.A.: K-NN classification under homomorphic encryption: application on a labeled eigen faces dataset. In: 2016 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) and 15th International Symposium on Distributed Computing and Applications for Business Engineering (DCABES), pp. 546–552, August 2016
14. Okamoto, T., Uchiyama, S.: A new public-key cryptosystem as secure as factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 308–318. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054135
15. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16. http://dl.acm.org/citation.cfm?id=1756123.1756146
16. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978). https://doi.org/10.1145/359340.359342
17. Rosen, A.: Concurrent Zero-Knowledge, 1st edn. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-32939-0
18. Schukat, M., Flood, P.: Zero-knowledge proofs in M2M communication. In: 25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), pp. 269–273, June 2014
19. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979). https://doi.org/10.1145/359168.359176
20. Yamamoto, Y., Oguchi, M.: A decentralized system of genome secret search implemented with fully homomorphic encryption. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–6, May 2017
21. Yi, X., Paulet, R., Bertino, E., Varadharajan, V.: Practical k nearest neighbor queries with location privacy. In: 2014 IEEE 30th International Conference on Data Engineering, pp. 640–651, March 2014
22. Yi, X., Paulet, R., Bertino, E.: Homomorphic Encryption and Applications. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12229-8
23. Zouari, J., Hamdi, M., Kim, T.H.: A privacy-preserving homomorphic encryption scheme for the internet of things. In: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1939–1944, June 2017