IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 November 2024, accepted 23 December 2024, date of publication 7 January 2025, date of current version 15 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3526632

RESEARCH ARTICLE

Secure User Authentication With Information Theoretic Security Using Secret Sharing-Based Secure Computation

KEIICHI IWAMURA[1], (Member, IEEE), AND AHMAD AKMAL AMINUDDIN MOHD KAMAL[2], (Member, IEEE)

[1]Department of Electrical Engineering, Tokyo University of Science, Tokyo 125-8585, Japan
[2]Department of Information and Computer Technology, Tokyo University of Science, Tokyo 125-8585, Japan

Corresponding author: Ahmad Akmal Aminuddin Mohd Kamal (ahmad.amin@rs.tus.ac.jp)

This work was supported in part by Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP23K16884.

ABSTRACT When using an insecure communication channel, the initial step involves authenticating the user (verifying the other party) to ensure the legitimacy of the communication partner, followed by an encrypted communication. Public key encryption-based digital signatures are widely used for user authentication; however, with the development of quantum computers they are highly likely to be deciphered. Studies are also ongoing on post-quantum cryptography, although they require significant computational resources and are challenging to implement in Internet of Things (IoT) devices. Therefore, this study suggests the implementation of user authentication and secure communication that guarantees information-theoretic security through the use of secure computation based on a computationally lightweight (k, n)-threshold secret sharing scheme. In this study, a user authentication system is proposed with information-theoretical security that utilizes constantly changing information. Subsequently, it is demonstrated that secure communication with information-theoretic security can be achieved without the need to distribute a substantial number of true random numbers by employing secure computation based on (k, n)-threshold secret sharing. The proposed methods are suitable for implementation in IoT environments because they require minimal processing overhead. The practical application of the proposed methods are also demonstrated through an implementation using C++. For example, the average execution time of the claimant was less than 0.1 [ms], proving that the proposed methods are very efficient.

INDEX TERMS User authentication, identification, secret sharing, secure computation, two-party computation, insecure channel, information theoretic, secure communication, information security.

I. INTRODUCTION

In communication over an insecure channel, the initial step involves performing entity authentication (user authentication) to verify the legitimacy of the parties involved. The purpose of user authentication is to confirm the identity of a person. Password-based identification is the most commonly used method of user authentication [1], [2], [3]. Passwords are commonly used for user authentication over communication channels and the Internet, where an encrypted password is transmitted and remains unchanged until it is updated. Consequently, if an attacker intercepts an encrypted password, they can potentially pose as a legitimate user. Therefore, it is advisable to regularly update the passwords.

For user authentication alternatives to passwords, the known methods involve shared secret information [4] (excluding those that utilize physical tokens such as identification cards or biometric traits). Several approaches have been discussed, including token-based authentication [5], [6], challenge-and-response (C-R) protocols using symmetric key encryption [7], [8], [9], and digital signatures [10], [11], [12].

The associate editor coordinating the review of this manuscript and approving it for publication was S. K. Hafizul Islam.

However, the token-based method requires a device that coordinates and produces random numbers, known as tokens, between the claimant and the verifier. This device must also be physically carried as required. The C-R method remains secure provided that the password is securely exchanged between the claimant and the verifier, thereby eliminating the need to transmit the password directly over the communication channel. However, this method requires a minimum of two communication phases, challenge and response. Furthermore, there are concerns that the existing cryptographic codes can be readily decrypted with the advent of quantum computing [13].

The Vernam cipher, recognized for its perfect information-theoretical security, was deemed secure [14]. However, its practical application is hindered by the necessity of exchanging a substantial volume of random numbers as keys between the claimant and the verifier. Because the conventional C-R method employs a combination of a password and a hash function that provides computational security, it provides only computational security. Moreover, although digital signatures are implemented using public key encryption, they are secured only under computational security assumptions and lack information-theoretic security. Computational security implies that the security can be compromised if an adversary has unlimited computing resources. However, information-theoretic security remains secure, even when an adversary has infinite computational power [15]. The threat that digital signatures based on traditional public key encryption may be vulnerable to quantum-computing attacks has recently become more pronounced. Consequently, there has been extensive research into a type of cryptography known as post-quantum cryptography, which remains secure against quantum computing attacks; however, it requires significant computational resources and effort.

In contrast, the (k, n)-threshold secret sharing scheme is recognized for its information-theoretic security. This method transforms a single secret into n distinct values, known as *shares* [16], [17]. This enables the reconstruction of the original secret by collecting any k of the n shares. Furthermore, possessing fewer than k shares does not yield knowledge of the secret. Examples include Shamir's (k, n) method [18], Blakley's method [19], and additive secret sharing method. These methods are commonly used in applications such as secure information storage and management across multiple server environments [20], multiparty computation [15], [21], [22], [23], [24], [25], and searchable encryption [26], [27]. They are yet to be applied to user authentication in communication channels, to the best of our knowledge.

Conversely, the Vernam cipher demands substantial storage space because of the pre-sharing of numerous true random numbers between the sender and the receiver, which renders it impractical for lightweight devices such as Internet of Things (IoT) devices. Looking ahead to Society 5.0, it is crucial to have user (or device) authentication and secure communication in IoT environments, which require lightweight and highly secure methods [28], [29], [30], [31], [32].

A. OUR CONTRIBUTIONS

In this study, we initially introduce a technique that enables the verifier to confirm the claimant as a legitimate individual who has previously registered information secretly in an environment without a secure communication channel. Subsequently, we aim to establish a secure communication channel with information-theoretical security suitable for IoT devices using minimal processing that does not require the exchange of substantial numbers of true random numbers, as in the Vernam cipher.

The contributions discussed above can be realized by deploying a secure two-party computation. Secure two-party computation, which is a branch of secure multiparty computation (MPC), has attracted considerable attention from the research community because of its wide range of potential applications. Two-party computation enables the two parties to jointly compute an arbitrary function of their inputs while keeping the actual values of those inputs hidden [21]. Secure computation methods, such as TUS methods (described in Section II-D), utilize the previously mentioned secret sharing to enable information-theoretically secure computation between the sender (data provider) and the receiver (data user). The key contributions of this study are summarized as follows:

- We demonstrate that secure computation with secret sharing enables user authentication with information-theoretic security by employing variables or information that vary for each instance.
- We demonstrate that employing secure computation with secret sharing facilitates the establishment of a secure communication channel with information-theoretic security without the need to distribute a substantial number of true random numbers, as in methods that employ the Vernam cipher.
- We implemented the aforementioned processes using a Dell PC and demonstrated that the proposed methods could be executed with minimal processing, making them appropriate for use in an IoT environment.

B. NOVELTY AND DIFFERENCE FROM PREVIOUS WORKS

The novelty and differences between our proposed method and previous studies are summarized as follows:

- We propose a new framework for applying a (k, n)-threshold secret sharing scheme based secure computation to realize a secure authentication and communication system with information-theoretic security.
- We achieved the application of a (k, n)-threshold secret sharing-based two-party secure computation for secure communication without the need for a trusted third party (TTP) or secure communication channel.

Since the proposal of the (k, n)-threshold secret sharing scheme in the early 1980s, many methods have been proposed for its implementation in fields such as secure computation and searchable secret sharing. However, based on the survey of the applications of secret sharing by the authors, there have been no examples of implementing (k, n)-threshold secret sharing-based secure computation to realize secure communication between two parties. One reason is that a (k, n)-threshold secret sharing-based secure computation typically requires at least $n > 2k - 1$ participants, where $k > 1$, and the existence of a secure channel between the two parties to achieve information-theoretic security [24]. Secure computation examples utilizing additive secret sharing also exist, where the parameters satisfy $n = k$ [33]. However, these approaches typically require additional elements, such as Beaver triples [34]. Alternative strategies involve employing a client-server model for MPC implementation [35]. However, the most widely recognized efficient method requires the number of server specification to be $n = 3$ and a threshold of $k = 2$. This implies that, beyond the two interacting parties, a separate server setup is necessary. In contrast, we demonstrated that secure computation between the two parties (sender and receiver) can be used to achieve identity authentication and secure communications with information-theoretic security.

Our proposed method is based on our previous work on secure computation, and can perform any arbitrary computation with a setting of $n \geq k$ [37], [38]. However, our previous work on secure computation focused on the conditionally secure approach. That is, conditions such as the need for a set of random numbers generated by a trusted third party (TTP) and the existence of a secure communication channel are required to achieve information-theoretic security. Here, we propose an improved method of secure two-party computation and demonstrate that it can be applied to situations where there is no secure communication channel or TTP based on our assumed application in secure communication. In other words, for applications in which information is transmitted from a sender to a receiver, we have shown that this can be realized by the sender playing the role of a TTP and sending the necessary information (including sets of random numbers) to the receiver. Table 1 presents a comparison between the proposed method and our previous work on the application of (k, n)-threshold secret sharing.

C. PAPER ORGANIZATION

The remainder of this paper is organized as follows. Section III discusses the related research. Section IV introduces the user authentication method with information-theoretic security. Section V describes the realization of a secure communication channel through secure computations. Sections VI and VII present a discussion and practical implementation of the proposed methods. Finally, Section VIII summarizes this study.

II. PRELIMINARIES

A. NOTATIONS AND DEFINITIONS

The following notations and definitions are implemented throughout the paper:

- $\mathbb{Z}_p$ denotes the finite field of prime p $(= \mathbb{Z}/p\mathbb{Z})$.
- n denotes the number of shares.
- k denotes the threshold number.
- p denotes a prime number, such that $p > n$.
- $H(X)$ denotes the Shannon's entropy of variable X.
- $H(X|Y)$ denotes the conditional entropy of X given Y.

In the security analysis of the proposed method, an information-theoretic approach is implemented using the information entropy or Shannon entropy function $H(X)$ of the variable X to demonstrate that the proposed scheme achieved perfect information-theoretic security. Entropy is often used as a mathematical measure of information or uncertainty and can be computed as a function of probability distribution [16]. Shannon initiated the research on information-theoretic security of a system in 1948. In Shannon's model for secure communication, the original message m was encrypted into ciphertext E using an encryption key K that was shared between the sender and receiver. Entropy was introduced to measure the amount of information associated with message m, represented as $H(m)$, and the amount of uncertainty associated with the possibilities of the encryption key, denoted as $H(K)$. In the context of perfect secrecy or information-theoretic security, the following equation applies, ensuring that the ciphertext E reveals no information regarding the message or key.

$$H(m) = H(m|E),\ H(K) = H(K|E)$$

Compared with conventional encryption methods with computational security, such as public key algorithms for key management in hybrid encryption methods, information-theoretic security algorithms are less vulnerable to man-in-the-middle attacks because of the randomness introduced to the systems. Moreover, information-theoretic security approaches achieve provable security that is robust against powerful eavesdroppers (adversaries) that possess unlimited computational resources [39]. However, to realize an information-theoretic approach, it is assumed that all the random numbers are selected according to a uniform distribution. We formally define Shannon's information entropy [40] as follows.

Definition 1 (Shannon Entropy): Let $P : U \to [0, 1]$ be a probabilistic function defined on a nonempty finite set U, then, its Shannon entropy is defined as:

$$H(P) = -\sum_{u \in U} P(u) \cdot \log P(u)$$

For variable X, the entropy $H(X) = H(P_X)$ of X can be expressed as:

$$H(X) = -\sum_{x \in X} p(x) \cdot \log(p(x))$$

TABLE 1. Brief comparison with previous works on secure computation.

Methods	Function	System model	Secret sharing	n and k	Secure channel	Conditions?
Our method	Secure comm.	Client-client	$(2,2)$ threshold	$n=k=2$	No	No
2P-Mult1 [22]	Two-party mult.	Client-server	(k,n) threshold	$n \geq 2k$	Yes	No
2P-Mult2 [23]	Two-party mult.	Client-server	(k,n) threshold	$n \geq 2k$	Yes	No
MP-Mult [25]	Multi-party mult.	Client-server	(k,n) threshold	$n \geq 2k$	Yes	Yes (limitation on input values)
SPDZ [33]	Multi-party comp.	Client-client	Additive	$n=k$	Yes	Yes (requires Beaver triples)
Sharemind [35]	Multi-party comp.	Client-server	Additive	$n=k=3$	Yes	Yes (secure against one corrupt server)
Araki [36]	Multi-party comp.	Client-server	Replicated	$n=3,k=2$	Yes	Yes (requires correlated randomness)
TUS 2 [37]	Multi-party comp.	Client-server	(k,n) threshold	$n \geq k$	Yes	Yes (limitation on input values, TTP)
TUS 4 [38]	Multi-party comp.	Client-server	(k,n) threshold	$n \geq k$	Yes	Yes (requires TTP)

The entropy $H(X)$ measures the uncertainty in the value of the random variable X. If $p(x|y) = Pr[X = x|Y = y]$, then the conditional entropy $H(X|Y)$ is defined as follows:

$$H(X|Y) = -\sum_{x\in X}\sum_{y\in Y} p(y)p(x|y)\cdot \log p(x|y))$$

B. (k, n) THRESHOLD SECRET SHARING

Secret sharing is known as (k, n)-threshold secret sharing when it satisfies the following conditions [16]:

- Any $k-1$ or fewer shares reveal no information about the original secret s.
- Any k or more shares enable the reconstruction of the original secret s.

Examples of threshold secret-sharing schemes include Shamir's (k, n) method [18], the XOR-based method [41], (k, L, n) ramp method [42], (n, n) additive method, and the replicated secret sharing method [36].

Shamir's (k, n) and XOR-based methods enable the secret to be reconstructed with only k of n shares, implying that recovering the secret is possible, even if $n-k$ shares (or servers) are missing. Although the XOR-based method offers faster distribution and reconstruction than Shamir's (k, n) method, it lacks the capability of secure computation over the computed shares. The (k, L, n) ramp method enhances the encoding efficiency of Shamir's (k, n) method by enabling simultaneous distribution of the L-tuple of the secret $\mathcal{S} = \{s_1, \ldots, s_L\}$. However, shares generated by this method are not appropriate for secure computation. In contrast, the (n, n) additive method facilitates efficient secure computation with $n = k$. However, it lacks server-loss resilience, which means that the secret cannot be recovered if any share is lost. To address this issue, replicated secret sharing increases the number of shares to be greater than the threshold k; however, it requires at least three shares (equivalent to servers) and is unsuitable for two-party computation. The comparisons are presented in Table 2.

This study was inspired by Shamir's (k, n) method to achieve a two-party computation. In this method, all computations were performed in a finite field $\mathbb{Z}_p$. Shamir's (k, n) method uses the following protocols for the distribution and reconstruction of secret s:

TABLE 2. Comparison of different threshold secret sharing schemes.

Threshold secret sharing	Parameters n and k	Addition/ Scalar Mult.	Multiplication/ Division
Shamir's (k, n) method	$n \geq k > 1$	Yes	Yes*
XOR-based method	$n \geq k > 1$	No	No
(k, L, n) ramp method	$n \geq k > 1$	No	No
(n, n) additive method	$n = k > 1$	Yes	Yes*
Replicated method	$n > k > 1$	Yes	Yes

*Requires certain conditions such as the use of Beaver triple [34], etc.

Protocol 2.1: Distribution

1) Dealer $\mathcal{D}$ selects any prime number p such that $s < p$ and $n < p$.
2) Dealer $\mathcal{D}$ selects $k-1$ random numbers a_l $(l = 1, \ldots, k-1)$ from $\mathbb{Z}_p$ and generates a random distribution polynomial $f(x)$, as follows:
$$f(x) = s + a_1 x + \cdots + a_{k-1}x^{k-1} \quad (1)$$
3) Dealer $\mathcal{D}$ inserts the ID x_i $(i = 1, \ldots, n)$ of each server into x in (1), calculates the shares $f(x_i) = W_i$ corresponding to each ID, and sends (x_i, W_i) to all servers.

Protocol 2.2: Reconstruction

1) Player $\mathcal{P}$ collects any k shares and their pairs of IDs. We assume that the shares are W_h $(h = 1, \ldots, k)$ and the corresponding IDs are x_h.
2) Player $\mathcal{P}$ substitutes x_h and W_h into (1) and solves k simultaneous equations to obtain secret s.

C. VERNAM CIPHER

The Vernam cipher, also referred to as a one-time pad encryption, is a robust cipher with perfect information-theoretic security. This ensures that even if an adversary knows ciphertext Y, they cannot derive any information regarding the original plaintext X. Consequently, the following holds:

$$H(X) = H(X|Y)$$

With the Vernam cipher, perfect security can be achieved under the following conditions:

- the encryption key must be truly random,
- the encryption key should be equal to or greater than the length of the plaintext, and
- the encryption key must be used only once.

Consequently, issues arise when encrypting substantial amounts of plaintext, as it requires the pre-sharing and generation of a large number of true random numbers (at least equal to the number of plaintexts to be encrypted) between the sender and receiver. The critical aspect is the secure handling of these random numbers.

D. TUS METHOD

The TUS method is a secure computation method based on (k, n)-threshold secret sharing proposed by the Iwamura Research Group at the **T**okyo **U**niversity of **S**cience (thus the name TUS) [37], [38]. This method enables secure arithmetic operations, including multiplication operations, to be executed while maintaining input secrecy using only $n = k$ servers. Typically, secure computations based on (k, n)-threshold secret sharing facilitate the addition operation with $n = k$ servers; however, they do not support multiplication with the same number of servers [43].

For example, in Shamir's (k, n) method, which distributes secrets using a polynomial $f(x)$ with $\deg(f(x)) = k - 1$, the multiplication operation will result in changes in the degree of the resulting polynomial from $k - 1$ to $2k - 2$. Consequently, the number of shares required to reconstruct the secret increases from k to $2k - 1$, thereby requiring at least $2k - 1$ servers.

An additional characteristic of the TUS method is that it does not directly share the secret as n shares using Shamir's (k, n) method. Instead, the secret is first masked using an initial random number and stored as an *encrypted scalar value*. To perform secure computations with a scalar value, the shares of a different random number (subsequently referred to as the conversion random number) are used to multiply the secret previously encrypted with the initial random number. This process transforms the encrypted scalar values into *shares* to facilitate the computation of these shares.

In this study, conversion random numbers are defined as a set that includes a single random number ε_j and a share $[\varepsilon]_j$ of the random number ε for $j = 0, \ldots, k - 1$, such that

$$\varepsilon = \prod_{j=0}^{k-1} \varepsilon_j$$

Therefore, each server S_j has the following conversion random number.

$$(\varepsilon_j, [\varepsilon]_j)$$

By configuring the parameters k and n to $n = k = 2$ in the TUS method, secure computations between two parties can be provided (that is, two-party computation). Therefore, secure two-party computation can be achieved between the sender and the receiver, which enables the implementation of our proposed method.

III. RELATED WORKS: USER AUTHENTICATION

This section describes the related studies on conventional user authentication methods. The following two methods provide only computational security.

A. CHALLENGE-AND-RESPONSE (C-R)

This section describes the challenge-and-response method [7], [8], [9]. A key aspect of this method is that the authentication server authenticates using a one-time random number known as a "challenge," without the need to send the password over the network. The detailed steps are as follows:

1) The user initiates a request for access to the server.
2) The server generates a unique challenge and sends it to the user.
3) The user creates a reply using the stored password and the challenge received and then transmits *Response A* to the server.
4) The server produces *Response B* using the challenge and the user password retained in the database.
5) The server compares *Response A* and *Response B*, and if they match, the user is authenticated.

B. DIGITAL SIGNATURE

The digital signature technology employs public key encryption to confirm the authenticity of an individual or another entity while validating messages to guarantee their integrity [12].

In public key encryption, the keys used for encryption and decryption are distinct, allowing one to be disclosed publicly while the other remains confidential. Moreover, in the authentication method using a public key, only one person can create the correct signature using a specified private signing key, that is, an authorized user possessing that key. Furthermore, a private key, also known as a secret key, cannot be deduced from a public key. Therefore, if a signature created using a private key is validated using a public key, it is confirmed to be produced by a legitimate user possessing an appropriate private key.

IV. STRONGLY SECURE USER AUTHENTICATION

The method of providing secure user authentication with information-theoretical security lies in the use of constantly changing information, such as keys in a Vernam cipher, and in the examination of the similarities and differences between these values.

For example, assume that a password is stored as a secret and encrypted using a true random number, as in the Vernam cipher. Here, a user authentication system with information-theoretic security can be easily realized if the claimant sends an encrypted password to the verifier. However, repeated use of the same encrypted password value allows an adversary to easily mimic an authorized user by performing a replay attack using an identical encrypted value [44], [45]. Therefore, the authentication data must vary with each use, and although the approach described

in Section III achieves this, it only ensures computational security.

In contrast, the TUS method has been proven to provide strong information-theoretic security. Therefore, if the claimant and verifier alter the registered information with each instance and perform a secure subtraction of this information using the secure computation facilitated by the TUS method, if the outcome is zero then it demonstrates that the claimant possesses the correct registered information.

Although the TUS method relies on the assumption of a secure communication channel to transmit necessary information, this study makes a stricter assumption; that is, we assume a worst-case scenario in which no such secure communication channel exists.

A. PROPOSED METHOD 1

1) OVERVIEW OF THE PROPOSED METHOD

This section introduces the proposed method for achieving secure user authentication using secure two-party computation based on the (k, n)-threshold secret sharing scheme. The proposed approach is divided into two primary processes: preprocessing and user authentication. An overview of the proposed technique is as follows.

- *Preprocessing*: In this step, every user initially registers their distinct verification details, such as a password and biometric data, with a verifier $\mathcal{V}$. Consider the verification details as a unique key, represented by *Key*; the user completes this step beforehand, presumably through an in-person information exchange.
- *User authentication*: In this step, the individual seeking verification is designated as a claimant $\mathcal{C}$ and provides verification details Key' to the verifier $\mathcal{V}$. Unlike conventional methods that compare the hashed values of the verification data or rely on public key encryption, this approach employs secure computation based on the information-theoretic principles of Shamir's $(2, 2)$ method. In particular, this approach applies secure computation to shares generated by the Shamir's $(2, 2)$ method. Secure computation is used to securely compute the following information A: if $A = 0$, the user is authenticated; otherwise, if $A \neq 0$, the authentication process is aborted.

$$A = Key - Key'$$

2) CONSTRUCTION OF THE PROPOSED METHOD 1

Consider a claimant $\mathcal{C}$ who seeks to demonstrate their legitimacy as the registrant of confidential authentication details, such as a password, and secretly transmits the necessary information to the verifier $\mathcal{V}$ confirming their knowledge of the authentication information.

The objective of this study is to implement a lightweight user authentication system in which a claimant $\mathcal{C}$ can request for authentication without disclosing sensitive information. For user authentication, the parameters n and k are assumed to be set to two in the (k, n)-threshold secret sharing scheme.

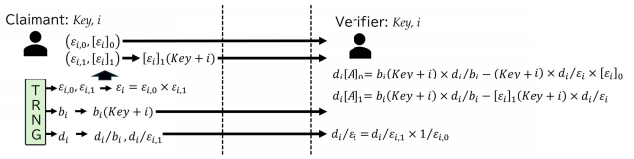

FIGURE 1. Overview of our proposed method 1.

Furthermore, it is assumed that secret information, such as a password, serves as *Key*, agreed upon and pre-shared beforehand by both the claimant $\mathcal{C}$ and the verifier $\mathcal{V}$, as described below. Figure 1 shows an overview of the process of proposed method 1.

In the TUS method, a secure communication channel is assumed between each participant, and all the random numbers required for the conversion process (as detailed in [38]) are generated based on the assumption of a TTP. In contrast, the method described below operates without the need for a secure communication channel or the assumption of TTP. Instead, the sender is responsible for generating the random numbers required for conversion (Steps 1 and 2 of Protocol 4.1). The authentication information is then sent to the receiver in encrypted form, with an additional layer of encryption applied using a separate random number (as shown in Steps 3 and 4 of Protocol 4.1). This approach enables secure computation without relying on secure communication channels or TTP. Note that all random numbers are chosen within $\mathbb{Z}_p \setminus \{0\}$.

Protocol 4.0: Preprocessing

1) Claimant $\mathcal{C}$ registers the legitimate *Key* (for example, a password) and the initial value of i to verifier $\mathcal{V}$. Therefore, $\mathcal{V}$ has the following information:

$$Key, i$$

Here, *Key* is considered as a predefined password; however, it can also be set as biometric data, such as fingerprints or facial images. Furthermore, i denotes the number (counter) of processes executed under the initial assumption that $i = 1$. The process in the ith iteration is as follows.

Protocol 4.1: User authentication process

1) Claimant $\mathcal{C}$ generates random numbers $\varepsilon_{i,0}$ and $\varepsilon_{i,1}$, computes $\varepsilon_i = \varepsilon_{i,0} \times \varepsilon_{i,1}$ and distributes ε_i by using Shamir's $(2, 2)$ method to produce the following two shares:

$$[\varepsilon_i]_0, [\varepsilon_i]_1$$

2) $\mathcal{C}$ stores $(\varepsilon_{i,1}, [\varepsilon_i]_1)$ and sends $(\varepsilon_{i,0}, [\varepsilon_i]_0)$ to $\mathcal{V}$.
3) $\mathcal{C}$ generates a random number b_i, computes, and sends the following to $\mathcal{V}$:

$$b_i(Key + i) = b_i \times (Key + i),$$
$$[\varepsilon_i]_1(Key + i) = [\varepsilon_i]_1 \times (Key + i)$$

4) $\mathcal{C}$ generates a random number d_i, computes, and sends the following to the verifier $\mathcal{V}$:

$$\frac{d_i}{b_i}, \frac{d_i}{\varepsilon_{i,1}}$$

5) If the received d_i/b_i and/or $d_i/\varepsilon_{i,1}$ are equal to 0, $\mathcal{V}$ assumes dishonest action and stops the process (because all random numbers are chosen within $\mathbb{Z}_p \setminus \{0\}$).
6) Else, $\mathcal{V}$ computes the following:

$$\frac{d_i}{\varepsilon_i} = \frac{1}{\varepsilon_{i,0}} \times \frac{d_i}{\varepsilon_{i,1}}$$

7) $\mathcal{V}$ uses the registration information Key and i to compute $Key + i$ in the second half of $d_i[A]_0$ and computes the following:

$$d_i[A]_0 = b_i(Key + i) \times \frac{d_i}{b_i} - (Key + i) \times \frac{d_i}{\varepsilon_i} \times [\varepsilon_i]_0$$
$$d_i[A]_1 = b_i(Key + i) \times \frac{d_i}{b_i} - [\varepsilon_i]_1(Key + i) \times \frac{d_i}{\varepsilon_i}$$

8) $\mathcal{V}$ reconstructs $d_i(A)$ using the two shares $d_i[A]_0$ and $d_i[A]_1$ and authenticates the claimant if the result is equal to zero; otherwise, it is rejected.
9) $\mathcal{V}$ sends the authentication result to $\mathcal{C}$, and if the authentication succeeds, updates $i = i + 1$.
10) $\mathcal{C}$ also updates $i = i + 1$ if authentication is successful.

This protocol can be described as a method in which the verifier $\mathcal{V}$ uses the registered legitimate value to subtract $b_i(Key + i)$, which is provided by the claimant $\mathcal{C}$, to verify whether the claimant $\mathcal{C}$ possesses an accurate value. $Key + i$ in the latter part of $d_i[A]_0$ is computed from the initially registered information; therefore, if this differs from $Key + i$ in the first part of the equation (provided by the claimant), the outcome of the reconstruction result will not equal zero.

Moreover, if the outputs from Steps 1, 2, to 3 are transmitted simultaneously, only a single communication is necessary. However, different values of Key' and i' may accidentally result in $Key + i = Key' + i'$. In such instances, Key and i can be merged to obtain $Key||i$. Here, both $Key + i$ and $Key||i$ represent single indivisible numerical values, and each individual value cannot be decomposed. For ease of explanation, we refer to both as $Key + i$.

3) SECURITY OF PROPOSED METHOD 1

In the security model, it is assumed that both the claimant and the verifier are honest and their information is handled securely. Under these conditions, the adversary $\mathcal{A}$ seeks to acquire Key and other registered details transmitted through the communication channel between the claimant and the verifier. If the adversary $\mathcal{A}$ succeeds in obtaining this information, then the attack is deemed successful.

In the protocol described above, the adversary $\mathcal{A}$ obtains the following during the ith communication:

$$\mathbf{A}_i = \left\{(\varepsilon_{i,0}, [\varepsilon_i]_0), b_i(Key + i), [\varepsilon_i]_1(Key + i), \frac{d_i}{b_i}, \frac{d_i}{\varepsilon_{i,1}}\right\}$$

Apart from Key and i, all the additional variables were generated from true random numbers and varied in each execution. Furthermore, it was not possible to decompose the combinations of these variables into separate components. In the proposed method 1, the same value $(Key + i)$ for $b_i(Key + i)$ and $[\varepsilon_i]_1(Key + i)$ is utilized twice. Calculating the ratio yields $b_i/[\varepsilon_i]_1$ and its inverse, $[\varepsilon_i]_1/b_i$. Furthermore, $\varepsilon_{i,1}/b_i$ can be derived using the information on d_i/b_i and $d_i/\varepsilon_{i,1}$. However, without knowledge of b_i, it is impossible to determine ε_i and $\varepsilon_{i,1}$.

Furthermore, information $(b_i - [e_i]_1)(Key + i)$ can be determined by subtraction; however, it remains non-decomposable. Moreover, by updating i to $i = i + 1$, the possibility of a replay attack was eliminated. For example, the information of $(b_i - b_{i+1})(Key + i) - b_{i+1}$ can be computed from two executions of the proposed method; however, distinct components remain undisclosed. Based on the above discussion, it is clear that adversary $\mathcal{A}$ was unable to extract any additional details from $\mathbf{A}_i$; therefore, the following holds true:

$$H(Key) = H(Key|\mathbf{A}_i) \tag{2}$$

From the above, it can be proved that proposed method 1 achieved information-theoretic security.

B. PROPOSED METHOD 2

1) OVERVIEW OF THE PROPOSED METHOD

In the proposed method 1, it was assumed that users directly provide their verification information, such as passwords, to the verifier $\mathcal{V}$. This approach raises potential issues regarding the leakage or misuse of registered information. To mitigate this risk, the second proposed method introduces a more secure technique for pre-sharing verification information with a verifier. The key improvements are as follows.

- *Preprocessing*: Consider a unique key Key that represents verification information. The user initially generates two shares of this verification data by using Shamir's (2, 2) method. Subsequently, only one share is provided to verifier $\mathcal{V}$. This approach guarantees that the verifier cannot reconstruct the original information without acquiring both the shares.

$$[Key]_0, [Key]_1$$

- *User authentication*: The user authentication process mirrors that of the proposed method 1, using secure computation with (k, n)-threshold secret sharing to securely evaluate the following, where, Key' represents the information provided by claimant $\mathcal{C}$. Authentication was considered successful if $A = 0$; otherwise, the process was terminated.

$$A = Key - Key'$$

2) CONSTRUCTION OF THE PROPOSED METHOD 2

Figure 2 shows an overview of the proposed method 2. Should the claimant $\mathcal{C}$ prefer to keep Key confidential from the verifier $\mathcal{V}$, $\mathcal{C}$ can opt to register $[Key]_1 d_1$ as a substitute for Key with $\mathcal{V}$, as described in Protocol 4.2 below. Moreover, in Step 3 of Protocol 4.3, we let $\mathcal{C}$ computes and sends $d_i/\varepsilon_i \times [\varepsilon_i]_1$ to $\mathcal{V}$. By this, as shown in Figure 2, the

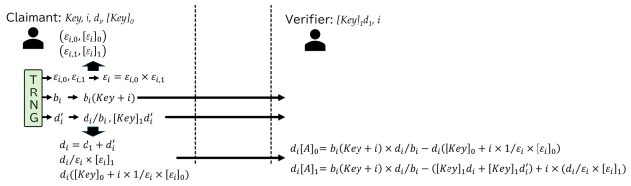

FIGURE 2. Overview of proposed method 2.

communication cost for transmitting $(\varepsilon_{i,0}, [\varepsilon_i]_0)$ to $\mathcal{V}$ can be omitted. Below, all all random numbers are chosen within $\mathbb{Z}_p \setminus \{0\}$.

Protocol 4.2: Preprocessing

1) Claimant $\mathcal{C}$ computes the shares $[Key]_0, [Key]_1$ of Key using Shamir's (2, 2) method, and generates a random number d_1. Then, it computes $[Key]_1 d_1$ and registers the following to the verifier $\mathcal{V}$:

$$[Key]_1 d_1, i$$

The modifications introduced in Protocol 4.2 necessitate corresponding adjustments in the original Protocol 4.1 outlined in proposed method 1. An improved user-authentication protocol is presented in Protocol 4.3.

Protocol 4.3: Authentication

1) Claimant $\mathcal{C}$ generates random numbers $\varepsilon_{i,0}$ and $\varepsilon_{i,1}$, computes $\varepsilon_i = \varepsilon_{i,0} \times \varepsilon_{i,1}$ and distributes ε_i by using Shamir's (2, 2) method to produce the following two shares.

$$[\varepsilon_i]_0, [\varepsilon_i]_1$$

2) $\mathcal{C}$ generates and stores a random number b_i, computes $b_i(Key + i)$, and sends it to the verifier $\mathcal{V}$.
3) $\mathcal{C}$ generates a random number d_i', computes the following as $d_i = d_1 + d_i'$, and sends it to $\mathcal{V}$.

$$\frac{d_i}{b_i}, [Key]_1 d_i', d_i([Key]_0 + i \times \frac{1}{\varepsilon_i} \times [\varepsilon_i]_0), \left(\frac{d_i}{\varepsilon_i} \times [\varepsilon_i]_1\right)$$

4) If the received d_i/b_i and/or $d_i[\varepsilon_i]_1/\varepsilon_i$ are equal to 0, $\mathcal{V}$ assumes dishonest action and stops the process.
5) $\mathcal{V}$ generates the second half of $d_i[A]_1$ using $[Key]_1 d_1$ and i from the registered information and computes the following:

$$d_i[A]_0 = b_i(Key + i) \times \frac{d_i}{b_i} - d_i([Key]_0 + i \times \frac{1}{\varepsilon_i} \times [\varepsilon_i]_0)$$

$$d_i[A]_1 = b_i(Key + i) \times \frac{d_i}{b_i} - ([Key]_1 d_1 + [Key]_1 d_i') + i \times \left(\frac{d_i}{\varepsilon_i}[\varepsilon_i]_1\right)$$

6) $\mathcal{V}$ reconstructs $d_i(A)$ using the two shares $d_i[A]_0$ and $d_i[A]_1$ and authenticates the claimant if the result is equal to zero; otherwise, it is rejected.
7) $\mathcal{V}$ sends the authentication result to $\mathcal{C}$, and if the authentication succeeds, updates $i = i + 1$.
8) $\mathcal{C}$ also updates $i = i + 1$ if authentication is successful.

3) SECURITY OF THE PROPOSED METHOD 2

In the proposed method 2, the data captured by the adversary $\mathcal{A}$ through the communication channel are summarized in $\mathbf{B}_i$.

$$\mathbf{B}_i = \left\{ b_i(Key + i), \frac{d_i}{b_i}, [Key]_1 d_i', d_i([Key]_0 + i \times \frac{1}{\varepsilon_i} \times [\varepsilon_i]_0), \left(\frac{d_i}{\varepsilon_i} \times [\varepsilon_i]_1\right) \right\}$$

In Protocol 4.3, the terms $d_i([Key]_0 + i \times 1/\varepsilon_i \times [\varepsilon_i]_0)$ and $(d_i/\varepsilon_i \times [\varepsilon_i]_1)$ are transmitted together as a single data unit; therefore, they cannot be separated into distinct components. Furthermore, $[Key]_1 d_1$ is transformed into $[Key]_1 d_i$ by adding $[Key]_1 d_i'$. Similarly, because the specific values remain undisclosed, Equation (2) remains true, ensuring information-theoretic security.

For enhanced security, the initial value of i, which is currently set to one, can be substituted with a random number and recorded alongside the key.

V. REALIZING STRONGLY SECURE COMMUNICATION

A. PROPOSED METHOD 3

1) OVERVIEW OF THE PROPOSED METHOD

As previously discussed, the Vernam cipher is recognized for its information-theoretic security; however, it requires the advance sharing of a substantial quantity of true random numbers between the sender and the receiver, which is inefficient. In contrast, our proposed method realizes the secure transmission of confidential data with information-theoretic security by exchanging only three random numbers. Figure 3 shows an overview of the proposed method 3.

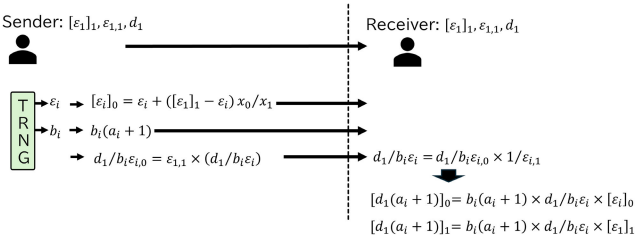

FIGURE 3. Overview of our proposed method 3.

The proposed method can be categorized into three primary stages: preprocessing, transmission, and reception of data by the sender $\mathcal{S}$ and the receiver $\mathcal{R}$. A brief outline of the procedure for each stage is provided below.

- *Preprocessing*: Before initiating secure communication between the sender $\mathcal{S}$ and the receiver $\mathcal{R}$, it is assumed that the following three random numbers have already been securely exchanged through face-to-face interactions.

$$([\varepsilon_1]_1, \varepsilon_{1,1}, d_1)$$

- *Sender* $\mathcal{S}$: To ensure that message a_i remains secure, $\mathcal{S}$ applies randomization by multiplying it by a true random number b_i, resulting in

$$b_i(a_i + 1)$$

To ensure that the randomized secret does not equal zero when $a_i = 0$, the value "1" is incorporated. Subsequently, $\mathcal{S}$ performs the required calculation and transmits the result to the receiver $\mathcal{R}$.

- *Receiver* $\mathcal{R}$: Upon receiving the necessary information, $\mathcal{R}$ performs secure computation based on Shamir's (2, 2) method and recovers the original message a_i.

2) CONSTRUCTION OF THE PROPOSED METHOD

In the following, we assume $n = k = 2$ and that the sender previously provided the receiver with the following three random numbers. Moreover, all random numbers are chosen within $\mathbb{Z}_p \setminus \{0\}$.

$$\left([\varepsilon_1]_1, \varepsilon_{1,1}, d_1\right)$$

Protocol 5.1: Secure communication

1) Sender $\mathcal{S}$ generates a random number b_i for secret a_i, computes and sends the following to receiver $\mathcal{R}$:

$$b_i(a_i + 1) = b_i \times (a_i + 1)$$

2) $\mathcal{S}$ generates a random number ε_i and computes c_i from the shared $[\varepsilon_1]_1$ that satisfies the following. Then, $\mathcal{S}$ computes $[\varepsilon_i]_0 = \varepsilon_i + c_i x_0$ using the computed c_i and sends it to $\mathcal{R}$.

$$[\varepsilon_1]_1 = \varepsilon_i + c_i x_1$$

3) $\mathcal{S}$ computes the following from the shared d_1, $\varepsilon_{1,1}$ and sends it to $\mathcal{R}$:

$$\frac{d_1}{b_i \varepsilon_{i,0}} = \frac{d_1 \varepsilon_{1,1}}{b_i \varepsilon_i}$$

4) $\mathcal{R}$ computes the following equation:

$$\frac{d_1}{b_i \varepsilon_i} = \frac{d_1}{b_i \varepsilon_{i,0}} \times \frac{1}{\varepsilon_{1,1}}$$

5) $\mathcal{R}$ computes the following equations:

$$[d_1(a_i + 1)]_0 = b_i(a_i + 1) \times \frac{d_1}{b_i \varepsilon_i} \times [\varepsilon_i]_0$$
$$[d_1(a_i + 1)]_1 = b_i(a_i + 1) \times \frac{d_1}{b_i \varepsilon_i} \times [\varepsilon_1]_1$$

6) $\mathcal{R}$ reconstructs $d_1(a_i + 1)$, and computes secret a_i as follows.

$$\frac{d_1(a_i + 1)}{d_1} - 1 = a_i$$

3) SECURITY OF THE PROPOSED METHOD

In the protocol described, it is assumed that both the sender and the receiver are honest and refrain from engaging in dishonest actions. Furthermore, it is assumed that the utilized device is not susceptible to internal scrutiny by the adversary $\mathcal{A}$.

In this case, the adversary $\mathcal{A}$ seeks to acquire confidential data transmitted through a communication channel. If the adversary $\mathcal{A}$ succeeds in obtaining secret information a_i, then the attack is deemed successful. During the ith communication, the adversary $\mathcal{A}$ obtains

$$\mathbf{C}_i = \left\{ b_i(a_i + 1), [\varepsilon_i]_0, \frac{d_1}{b_i \varepsilon_{i,0}} \right\}$$

From the information above, both b_i and a_i are arbitrarily determined. Furthermore, the set of conversion random numbers, $([\varepsilon_1]_1, 1/\varepsilon_{1,1})$ is fixed. Although ε_i is a true random number that differs each time, another set of conversion random numbers $([\varepsilon_i]_0, 1/\varepsilon_{i,0})$ is calculated. In this case, if we set $[\varepsilon_1]_1 = t$, then

$$t = \varepsilon_1 + c_1 x_1, \quad [\varepsilon_1]_0 = \varepsilon_1 + c_1 x_0 \tag{3}$$
$$t = \varepsilon_2 + c_2 x_1, \quad [\varepsilon_2]_0 = \varepsilon_2 + c_2 x_0 \tag{4}$$
$$t = \varepsilon_3 + c_3 x_1, \quad [\varepsilon_3]_0 = \varepsilon_3 + c_3 x_0 \tag{5}$$

From (3), because $\varepsilon_1 = t - c_1 x_1$, adversary $\mathcal{A}$ learns $[\varepsilon_1]_0 = t + c_1(x_0 - x_1)$. From (4), because $\varepsilon_2 = t - c_2 x_1$, adversary $\mathcal{A}$ learns $[\varepsilon_2]_0 = t + c_2(x_0 - x_1)$. From (5), because $\varepsilon_3 = t - c_3 x_1$, adversary $\mathcal{A}$ learns $[\varepsilon_3]_0 = t + c_3(x_0 - x_1)$.

Therefore, we could state that

$$t = [\varepsilon_1]_0 - c_1 (x_0 - x_1) = [\varepsilon_2]_0 - c_2 (x_0 - x_1)$$
$$= [\varepsilon_3]_0 - c_3 (x_0 - x_1) = \ldots$$

Because $[\varepsilon_1]_0$, $[\varepsilon_2]_0$, $[\varepsilon_3]_0$, $(x_0 - x_1)$ are known, if one c_i, that is, one ε_i, is leaked, all the secret information is leaked. If ε_i is assumed to be secret information a_i, then all the secret information is leaked if one ε_i is made public, as in a known plaintext attack. However, ε_i is a value generated by the transmitting device, combined with other random numbers, and deleted during reconstruction, and does not leak from $\mathbf{C}_i$.

Moreover, even when b_i and a_i are known, the subsequent values b_{i+1} and a_{i+1} are calculated without depending on the former, thereby ensuring resistance to known plaintext attacks. Furthermore, knowledge of b_i does not compromise ε_i, because it cannot be derived from $d_1/b_i\varepsilon_{i,0}$. Therefore, acquiring ε_i requires access to ε_i produced on the device of the sender or the share $[\varepsilon_1]_1$ configured for both the sender and receiver devices.

However, even when employing the Vernam cipher with perfect security, exposure to random numbers on the device will result in the leakage of sensitive information. Therefore, information-theoretic security in the proposed method is guaranteed, provided that the shared $([\varepsilon_1]_1, \varepsilon_{1,1}, d_1)$ remains secure.

In the proposed method, d_1 is fixed; however, its value remains uncertain as c_i, b_i, and $\varepsilon_{i,0}$ vary for each instance. Therefore, without the knowledge of $([\varepsilon_1]_1, \varepsilon_{1,1}, d_1)$, it can be asserted that secret information a_i is transmitted securely with information-theoretic security using the proposed method 3.

$$H\,(a_i) = H\,(a_i|\mathbf{C}i)$$

From another perspective, using the same random number to secure multiple pieces of secret inputs in the Vernam cipher ensures that the information remains secure unless the

random number itself is compromised. However, if any single piece of confidential data is disclosed in the Vernam cipher, the random number used is exposed, thereby compromising the security of the remaining secret.

Therefore, it can be said that the proposed method 3 uses secure computation to provide resistance to the disclosure of the secret input. In secure computation, the random numbers employed are kept confidential during the process, unlike known plaintext attacks. Therefore, provided that the random numbers are secure and assuming that b_i and a_i, which are related to the secret input, are determined independently of other random numbers, the integrity of these numbers remains intact, and their resistance is preserved even if the secret input is disclosed.

Moreover, in the Vernam cipher, if the adversary $\mathcal{A}$ has knowledge of the secret input a_i, it is possible for them to modify the secret input to the desired f_i by appending $a_i' = f_i - a_i$ to ciphertext $a_i + r_i$, even without knowing the actual random number r_i. In contrast, in the proposed method 3, modifying the secret input to the desired value is not feasible without knowing the random number b_i.

Finally, security can be further improved if x_0, or x_0 and x_1 are shared as random numbers at the beginning, as in the following information: $([\varepsilon_1]_1, \varepsilon_{1,1}, d_1)$.

VI. DISCUSSION

A. APPLICATIONS AND LIMITATIONS

It can be presumed that the proposed method 3 is appropriate for secure communication in IoT devices, which play a crucial role in transforming human life and delivering significant financial and social benefits. They can contribute significantly to essential applications in future smart cities, including intelligent waste management, environmental monitoring systems, smart transportation networks, and intelligent traffic control [46], [47]. One of the popular uses of the IoT is to enhance home automation systems, often referred to as smart homes. For example, IoT can be applied to gadgets such as smart lights, climate control systems, washing machines, door sensors, surveillance cameras, multimedia devices, and central air conditioning units. This facilitates remote home management and offers users a customized living environment. However, incorporating the IoT into smart homes requires that each device be interconnected with several other devices, allowing them to monitor operations and exchange information. Consequently, it is crucial to ensure the security and privacy of user data before it is shared, while minimizing computational and communication costs to accommodate the limited power and processing abilities of the IoT devices.

For example, when employing the Vernam cipher for secure communication, IoT devices must possess a substantial number of true random numbers beforehand, and the depletion of these numbers renders further encrypted communication infeasible. In contrast, in the proposed method 3, assuming that the transmitting device is an IoT device equipped with a true random number generator and the receiving device is a management device that collects data from the IoT device, initially, by establishing and distributing $(d_1, [\varepsilon_1]_1, \varepsilon_{1,1})$ among IoT devices, it is possible to attain information-theoretic security in communications without employing the Vernam cipher.

Furthermore, the implementation of secure computation using Shamir's (2, 2) method enables the proposed method to maintain a lower computational cost than methods such as digital signatures that use public key encryption (such as RSA). Therefore, the proposed method is efficient in terms of both processing and energy consumption. Furthermore, the use of secret sharing schemes for secure computation provides stronger information-theoretic security than computational security. Our authentication method also benefits from requiring only one communication round between the claimant and verifier, along with minimal additions and multiplications for secure computation, making it less susceptible to network delays. Table 3 provides a brief comparison of the security and energy efficiency of the proposed method with those of the other methods.

Moreover, if the transmitting device produces three new a_i and transmits them as secret inputs to the receiving device, a new set of random numbers $([\varepsilon_1]_1, \varepsilon_{1,1}, d_1)$ is exchanged. Subsequently, the previously used random numbers $([\varepsilon_1]_1, \varepsilon_{1,1}, d_1)$ are replaced. If the adversary remains unaware of this change, the confidentiality of the secret information remains intact, regardless of whether the adversary recorded the earlier communication data.

Furthermore, by combining the proposed methods 1 to 3, the following improved user authentication method is possible. Suppose that user A has registered Key and $(d_1, [\varepsilon_1]_1, \varepsilon_{1,1})$ on the authentication server.

i. User A initiates a request from the authentication server for the user authentication protocol.
ii. The authentication server creates a random number i and sends it to user A using the proposed method 3.
iii. User A reconstructs i and employs it in the proposed method 1 (or method 2) for user authentication.

All the described functionalities can be realized through the consistent application of secure computation using secret sharing. This enables the IoT devices to securely perform personal authentication (device authentication) using different values each time, while requiring minimal processing. Although the proposed authentication method offers robust information-theoretic security, it requires a prover and verifier to exchange keys and random numbers in advance. Public key encryption-based digital signatures enable the authentication of parties even on the first encounter, assuming the availability of a public key certificate issued by a trusted certification authority. Therefore, the remaining challenge for the proposed method is the pre-sharing of keys and random numbers. To address this challenge, it is essential to develop a framework that assumes a TTP similar to that of

TABLE 3. General comparison of security and energy consumption of our proposed method.

Method	Security	Energy consumption	Possible attacks
Preregistered encrypted password	Information-theoretic	Low (one communication)	Impersonation
Challenge-Response (C-R)	Computational	Low (one encryption and communication)	Brute-force
Digital signature (RSA)	Computational	High (modular exponentiation)	Quantum attack
Our method	Information-theoretic	Medium (one communication, and few additions and multiplications)	–

Kerberos [48] to distribute random numbers to parties during their initial encounter.

B. QUALITATIVE COMPARISON WITH THE CONVENTIONAL METHODS

Most conventional authentication and secure communication methods rely on public key or symmetric key encryption based on computational security. Rostampour et al. [49], Islam et al. [50], Uppuluri et al. [51], and Wazid et al. [52] introduced user authentication and communication methods that leveraged public key encryption to provide efficient and lightweight cryptographic solutions suitable for IoT environments. These methods predominantly utilize lightweight elliptical curve cryptography (ECC), thereby ensuring computational security. Alternately, Vishwakarma et al. [53] proposed a method for secure authentication and communication that offers computational security in a similar way by integrating blockchain with public and symmetric key encryptions. Sun et al. [54] introduced a user authentication system relying on passwords, utilizing a user cell phone and short message service (SMS) combined with AES-CBC symmetric key encryption to ensure secure authentication and communication. However, this approach is dependent on the computational security provided by the encryption method employed.

Currently, the security of most public key encryption schemes relies on integer factorization and discrete logarithm problems, which are difficult for computers to solve. However, Shor's quantum algorithm offers an exponential increase in the efficiency of solving these mathematical problems, enabling quantum computers to easily tackle them [55]. Specifically, Shor's algorithm has a complexity of $O((\log n)^2(\log\log n)(\log\log\log n))$ for factoring a positive integer n, which enables to efficiently resolve the discrete logarithm problem. Consequently, public key encryption systems based on factoring and discrete logarithm challenges are effectively compromised. A similar threat exists in symmetric key encryption, where Grover presented a quantum mechanical search algorithm that was polynomially faster than classical algorithms [56].

Examples of authentication methods that ensure security against quantum threats using digital signature approaches submitted to NIST for post-quantum cryptography include FALCON (lattice-based) [57], NTRUSign (lattice-based) [58], CRYSTALS-dilithium (lattice-based) [59], and SPHINCS+ (hash-based) [60] methods. Additional methods that maintain security in the presence of quantum computers include the contributions of Wang et al. [61] and Wang et al. [62], both employing lattice-based security to provide robust computational security for authentication and communication purposes. Moreover, in the digital signature proposed in FALCON, the implementation of a fast Fourier transform (FFT) means that the signature can be efficiently computed. However, many of the suggested methods incur extensive computational costs and involve large public keys and ciphertexts, rendering them impractical for certain applications such as those in IoT environments. Moreover, Chen et al. [63] introduced an identity-based multi-signature with robust computational security capable of consolidating multiple signatures on the identical message into a single compact form, thereby minimizing storage and transmission bandwidth consumption. However, this approach requires significant computational resources and assumes a trusted system known as a private key generator (PKG).

In contrast to the aforementioned methods, which implement encryption key methods and achieve computational security, our method employs a (k, n)-threshold secret sharing scheme for secure computation, offering enhanced information-theoretic security. Information-theoretic security implies that security assurance does not depend on the assumptions regarding computational difficulty. In contrast to public or symmetric key methods, including postquantum digital signatures, information-theoretic security strategies are less susceptible to this scenario. Therefore, information-theoretic security strategies offer a provable security that resists even the most capable eavesdroppers armed with limitless computational power, such as quantum computing. Moreover, these strategies are intrinsically more resistant to man-in-the-middle attacks [64], [65], [66] because of the generation of randomness.

Shannon demonstrated that only information-theoretically secure methods, such as one-time pad encryption, can achieve secure authentication and message communication. However, traditional one-time pad encryption requires symmetric key distribution between the communicating parties, and the volume of keys must match the number of messages to be exchanged. Our proposed method offers secure authentication and communication systems with enhanced security using the information-theoretic security provided by the (k, n) threshold secret sharing scheme established in [18]. Specifically, we introduce a novel framework for user authentication using secure computation based on Shamir's (2, 2) method. However, in contrast to one-time pad

encryption, our proposed method achieves the same level of robust security during communication by pre-sharing only three random numbers, regardless of the number of messages to be communicated. Moreover, secure computation utilizing (k, n)-threshold secret sharing is also recognized for its efficient computational performance, as demonstrated in Table 4 [67].

TABLE 4. Time taken to sort 1,000,000 20-bit data using secure computation (taken from [67]).

Method	Communication (bit)	Process time (s)
Secret sharing	(omitted)	1.1
Garbled circuit	2.43T	243
Homomorphic enc. (HE)	35.64T	3,564
Fully HE	0	48,877

Therefore, the proposed method is well suited for implementation in IoT environments, where a solution with a low computational cost is desirable. Table 5 summarizes the comparison between the proposed and conventional methods.

VII. EXPERIMENTAL EVALUATION

To assess the practical effectiveness of our protocol, the proposed methods 1, 2, and 3 described in Sections IV and V were implemented and the time required to complete the necessary computations were measured.

The implementation was performed using C++ language and was run on a Dell PC to simulate both the sender (claimant) and receiver (verifier) for simplicity. Furthermore, no optimization processes such as parallel computation were implemented in the simulation. The execution time was measured using the std::chrono library introduced in C++ 11. In particular, we used the high_resolution_clock with the highest accuracy to measure the execution time. Table 6 summarizes the implementation environment.

A. PROPOSED METHODS 1 AND 2: USER AUTHENTICATION

In this section, the results of proposed methods 1 and 2 are presented and compared. In our implementation, the variable i indicates the current count of user authentication protocol executions requested by the claimant $\mathcal{C}$, and user authentication are performed sequentially (for example, the tenth ($i = 10$) authentication request is handled only after the ninth ($i = 9$) is completed), and the average time required for each method is calculated.

For simplicity, it was also assumed that the claimant $\mathcal{C}$ provided the correct password for every requested user authentication process and that the password consisted solely of numerical digits. Passwords containing alphanumeric characters can also be used in the proposed methods, assuming that the characters are transformed into their respective ASCII or Unicode values. In addition, the proposed methods were implemented for five specific prime numbers to explore the effect of the size of the prime numbers used on the execution of our methods, particularly because our methods require the calculation of the modular multiplicative inverse.

- 23, 252, 729
- 2, 147, 483, 647
- 282, 671, 531, 609
- 29, 996, 224, 275, 833
- 26, 082, 833, 894, 132, 791, 297
- 2,297,271,634,742,810,443,154,153,338,805,764,573

Tables 7–12 clearly demonstrate that the proposed methods are very efficient in terms of processing time. In our implementation for the largest prime p, the claimant $\mathcal{C}$ and the verifier $\mathcal{V}$ required an average time of $35[\mu s]$ and $20[\mu s]$, respectively, to execute the computation shown in the proposed method 1. In contrast, for the same prime p, the claimant $\mathcal{C}$ and the verifier $\mathcal{V}$ need $45[\mu s]$ and $12[\mu s]$, respectively, to perform the operations described in our proposed method 2, making both the proposed methods of user authentication highly efficient.

Based on the data shown in Tables 7–12, we observed that the required time increased with the size of the prime number used; however, the difference was almost insignificant. Furthermore, when comparing our proposed methods 1 and 2, it was observed that claimant $\mathcal{C}$ required less time to execute the necessary process in proposed method 1 compared to proposed method 2, whereas verifier $\mathcal{V}$ requires less time in proposed method 2 than proposed method 1. This is because in proposed method 2, the claimant handles most of the complex calculations, leaving only simple arithmetic operations with precomputed values for the verifier, which resulted in reduced time compared to proposed method 1.

B. PROPOSED METHOD 3: SECURE COMMUNICATION

Table 13 presents the results of implementing the proposed method 3. In this method, it is assumed that the sender transmits m inputs to the receiver, and the total duration required for the inputs to be encrypted, sent, and decrypted by the receiver is measured. Furthermore, the results for the variation in m within the range $[10, 100000]$ are included. Here, we assume prime $p = 2, 297, 271, 634, 742, 810, 443, 154, 153, 338, 805, 764, 573$, and that all inputs are elements of $\mathbb{Z}_p$.

Additionally, we considered the time required to accomplish the same task using a traditional encryption key-based technique. Specifically, we utilize the one-time pad encryption method, in which each random key is used only once to encrypt a message. For a fair comparison with our proposed method 3, we assume that the plaintext consists of elements from $\mathbb{Z}_p$ and the key is randomly selected from $\mathbb{Z}_p \setminus \{0\}$. Similar to proposed method 3, we assume a scenario in which a sender aims to securely transmit his/her secret s_i

TABLE 5. Comparison with other secure authentication/communication works.

Methods	Encryption	Security	Quantum secure?	Description
Our method	Secret sharing	Information-theoretic	Yes (information-theoretic)	Secure two-party authentication and communication using (k, n)-threshold secret-sharing-based secure computation.
Rostampour et al. [49]	Public key	Computational	No (Shor's algorithm)	An efficient secure communication between the IoT edge devices and the cloud servers using a secure elliptic curve cryptography (ECC)-based authentication.
Islam et al. [50]	Public key	Computational	No (Shor's algorithm)	A lightweight cryptographic scheme for securing the communication between IoT home devices based on ECC.
Uppuluri et al. [51]	Public key	Computational	No (Shor's algorithm)	A secure user authentication and key agreement for IoT devices where data is secured through encryption using keys generated with ECC.
Wazid et al. [52]	Public key	Computational	No (Shor algorithm)	Three factor user authentication using one-way hash, XOR operations, and ECC.
Vishwakarma et al. [53]	Hybrid (public and symmetric)	Computational	No (Shor's and Grover algorithm)	A secure authentication of IoT devices and secure communication using the blockchain and hybrid cryptosystem (a combination of AES and ECDSA).
Sun et al. [54]	Symmetric key	Computational	No (Grover algorithm)	A user authentication that leverages the cellphone of a user and short message service in combination with AES-CBC to prevent password stealing and password reuse attack.
FALCON [57]	Public key	Computational	Yes (lattice based)	Digital signatures based on the difficult short integer solution (SIS) problem over the NTRU lattice that attains rapid signature generation by FFT.
NTRUSign [58]	Public key	Computational	Yes (lattice based)	A public-key encryption-based signature scheme based on solving the approximate closest vector problem (APPR-CVP) in the NTRU-type lattice.
Dilithium [59]	Public key	Computational	Yes (lattice based)	A lattice-based digital signature scheme that is strongly secure under chosen message attacks based on the hardness of finding short vectors in lattices.
SPHINCS+ [60]	Hash function	Computational	Yes (hash based)	A high-security post-quantum stateless hash-based signature scheme. It produces small signature size with fast signing and strong security.
Wang et al. [61]	Public key	Computational	Yes (lattice based)	A smartcard-based password authentication scheme whose security is based on the Ring-LWE problem.
Wang et al. [62]	Public key	Computational	Yes (lattice based)	Combination of post-quantum cryptography (learning with errors (LWE) and small integer solutions (SIS) problems) and quantum key distribution to realize secure authentication and message exchange.
Chen et al. [63]	Public key	Computational	Yes (lattice based)	An identity-based multi-signature based on the ring version of the SIS assumption (Ring-SIS).

TABLE 6. Implementation environment.

Device	Dell PowerEdge T150
CPU	Intel Xeon(R) E-2378G @ 2.80 GHz
Operating System	Ubuntu Desktop 20.04.6 LTS (64 bit)
Memory	16 GB (Usable memory: 15.3 GB)
Disk capacity	2.0 TB
Language	C++
Compiler	g++ (11.4.0)
OpenSSL version	3.0.2 15 Mar 2022

to the receiver for $i = 1, \ldots, m$. The detailed algorithm is as follows.

Protocol 7.1: Modified one-time pad encryption key-based secure communication

For $i = 1, \ldots, m$, do the following:

1) (Preprocessing) The sender creates a random key for encryption key_i.
2) (Encryption) The sender computes the following using the key and sends it to the receiver:

$$\mathrm{Enc}(s_i) = s_i + key_i \mod p$$

3) (Decryption) The receiver then determines the original secret s_i as follows.

$$\mathrm{Dec}(\mathrm{Enc}(s_i)) = \mathrm{Enc}(s_i) - key_i = s_i \mod p$$

Based on Table 13, approximately 3.8 [s] was required to transfer $m = 100,000$ inputs to the receiver. However, the traditional method employing a random symmetric key takes just 0.3 [s], making it about 10 times faster than our proposed method 3. This is because the encryption and decryption processes of Protocol 7.1 rely solely on the straightforward addition or subtraction within modulo p. In contrast, the proposed approach involves the sender performing various tasks, including multiple multiplication, addition, random number generation, and the expensive operation of modular inversion for each piece of data to be sent.Therefore, it can be stated that the proposed method 3 exhibits a slower processing time than the (modified) one-time pad encryption method presented in Protocol 7.1. However, it is crucial to highlight that the times listed in Table 13 represent the duration required for both the sender to encrypt and

TABLE 7. Time in microseconds to execute proposed methods 1 and 2 (for $p = 23, 252, 729$).

Counter, i	Proposed Method 1		Proposed Method 2	
	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$
1	85	20	72	13
2	26	13	28	9
3	22	13	27	9
4	20	13	27	9
5	24	12	29	12
6	20	13	27	9
7	22	13	29	9
8	21	12	28	9
9	21	12	31	9
10	20	13	31	9
11	27	13	30	9
12	22	12	29	9
13	21	13	32	9
14	22	12	29	9
15	23	12	31	9
16	23	12	30	9
17	25	12	30	9
18	23	12	30	9
19	23	12	28	9
20	24	12	19	9
Average	26	13	31	9

TABLE 8. Time in microseconds to execute proposed methods 1 and 2 (for $p = 2, 147, 483, 647$).

Counter, i	Proposed Method 1		Proposed Method 2	
	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$
1	87	19	71	13
2	22	13	28	10
3	20	12	28	10
4	21	12	27	9
5	20	12	27	9
6	20	12	27	9
7	20	13	27	9
8	21	12	28	9
9	20	12	27	9
10	20	13	27	9
11	21	12	28	10
12	20	12	28	9
13	20	12	27	10
14	20	13	28	9
15	20	13	28	9
16	20	12	28	9
17	20	12	27	10
18	20	12	27	9
19	20	12	29	10
20	20	12	28	10
Average	24	13	30	9

TABLE 9. Time in microseconds to execute proposed methods 1 and 2 (for $p = 282, 671, 531, 609$).

Counter, i	Proposed Method 1		Proposed Method 2	
	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$
1	85	21	75	13
2	25	14	33	10
3	22	13	31	9
4	22	13	32	10
5	23	13	32	9
6	24	13	31	9
7	23	13	30	9
8	22	13	29	9
9	22	13	30	9
10	24	14	30	9
11	22	13	29	10
12	21	13	31	9
13	21	13	32	10
14	25	13	32	9
15	22	13	32	9
16	22	13	31	10
17	21	13	29	9
18	22	13	30	9
19	21	13	29	9
20	23	13	29	9
Average	26	14	33	9

TABLE 10. Time in microseconds to execute proposed methods 1 and 2 (for $p = 29, 996, 224, 275, 833$).

Counter, i	Proposed Method 1		Proposed Method 2	
	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$
1	88	21	75	13
2	25	14	32	10
3	25	14	30	10
4	24	14	30	10
5	23	14	31	10
6	24	14	34	10
7	23	14	31	9
8	22	14	29	10
9	22	14	29	10
10	22	13	30	10
11	23	14	29	10
12	22	14	31	9
13	23	14	30	10
14	23	14	29	10
15	23	14	32	9
16	22	14	31	9
17	24	14	30	10
18	23	14	30	9
19	23	14	31	10
20	25	14	31	10
Average	26	14	33	10

the receiver to decrypt, excluding the time needed for the preliminary sharing of essential information, such as Step (1) in Protocol 7.1.

It can be noted that while our proposed method involves a computation time 10 times longer than the approach outlined in Protocol 7.1, it only requires prior sharing of three random numbers, regardless of the number of inputs m being transferred. However, to achieve strong security comparable to our proposed method 3, Protocol 7.1 requires encrypting each secret with a unique, truly random key, adhering to the criteria specified in Section II-C. As demonstrated in the Shannon model, achieving perfect secrecy, as in our proposed method 3, requires a number of unique keys to match or exceed the number of messages. Therefore, for $m = 100,000$, this requires that the sender also pre-arrange the secure distribution of 100,000 secret keys to the receiver, potentially introducing more computational and communication burdens than our proposed method.

Moreover, these findings were obtained using a single core. Therefore, our future research will explore parallelization

TABLE 11. Time in microseconds to execute proposed methods 1 and 2 (for $p = 26, 082, 833, 894, 132, 791, 297$).

Counter, i	Proposed Method 1		Proposed Method 2	
	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$
1	91	24	83	15
2	33	17	38	12
3	29	16	37	11
4	27	16	38	11
5	28	16	36	11
6	27	16	36	11
7	27	16	39	11
8	26	16	34	11
9	27	16	35	11
10	28	16	37	10
11	25	16	41	11
12	30	16	36	11
13	30	16	33	11
14	26	16	36	11
15	28	16	35	10
16	25	16	36	11
17	27	16	40	10
18	27	16	35	10
19	27	16	36	11
20	25	16	37	10
Average	31	16	39	11

TABLE 12. Time in microseconds to execute proposed methods 1 and 2 (for $p = 2, 297, 271, 634, 742, 810, 443, 154, 153, 338, 805, 764, 573$).

Counter, i	Proposed Method 1		Proposed Method 2	
	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$	Claimant, $\mathcal{C}$	Verifier, $\mathcal{V}$
1	99	26	88	15
2	33	20	45	11
3	34	19	45	12
4	31	29	42	12
5	31	19	41	12
6	31	19	43	11
7	30	19	42	12
8	31	19	44	11
9	32	18	41	11
10	32	18	41	11
11	31	19	41	11
12	31	19	42	11
13	31	19	42	12
14	31	19	42	11
15	30	18	41	12
16	31	18	42	11
17	30	19	41	11
18	30	18	42	11
19	31	18	42	12
20	31	22	43	11
Average	35	20	45	12

TABLE 13. Time in microseconds to execute Proposed Method 3 (compared with the encryption/encryption time of Protocol 7.1).

Number of inputs, m	Proposed method	Protocol 7.1
10	397	34
100	3740	311
1000	38048	3137
10000	378065	33314
100000	3748355	326102

techniques, such as spreading the workload across multiple cores, to enhance performance. Consequently, the time required to execute proposed method 3 can be further reduced.

VIII. CONCLUSION

In this study, two novel protocols aimed at achieving user authentication with information-theoretic security were introduced through a secure computation method using a (k, n)-threshold secret sharing scheme. By deploying and optimizing a conventional secure computation method to realize secure two-party computation between the two entities, user authentication strategies that offer robust protection against threats such as replay attacks were developed.

Furthermore, a new method for secure communication was introduced through two-party computation using (k, n)-threshold secret sharing. A comprehensive security evaluation for each proposed method was performed, demonstrating that our methods achieve robust information-theoretic security in contrast to the weaker computational security guaranteed by most conventional methods. Furthermore, these methods were implemented in C++ and their efficiency was demonstrated with minimal processing time.

Secure computation using secret sharing has enabled the realization of information-theoretic security for user authentication and simple, secure communication methods that were formerly dependent on computational security. Consequently, it will be possible to deploy technologies that were previously based solely on computational security to strengthen information-theoretic security using secure computation, which could broaden research opportunities.

In future studies, we will investigate the incorporation of a verification system to identify modifications in shares caused by malicious adversaries. Moreover, we intend to enhance the implementation strategy by implementing optimization techniques such as parallelization to improve the implementation results.

ACKNOWLEDGMENT

The authors express their heartfelt appreciation to anonymous peer reviewers who generously dedicated their time and expertise to review and provide constructive feedback on this research article.

REFERENCES

[1] C.-L. Lin and T. Hwang, "A password authentication scheme with secure password updating," *Comput. Secur.*, vol. 22, no. 1, pp. 68–72, Jan. 2003.

[2] I.-E. Liao, C.-C. Lee, and M.-S. Hwang, "A password authentication scheme over insecure networks," *J. Comput. Syst. Sci.*, vol. 72, no. 4, pp. 727–740, Jun. 2006.

[3] S. K. Sood, A. K. Sarje, and K. Singh, "Cryptanalysis of password authentication schemes: Current status and key issues," in *Proc. Int. Conf. Methods Models Comput. Sci. (ICM2CS)*, Dec. 2009, pp. 1–7.

[4] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proc. IEEE*, vol. 91, no. 12, pp. 2021–2040, Dec. 2003.

[5] M. Dammak, O. R. M. Boudia, M. A. Messous, S. M. Senouci, and C. Gransart, "Token-based lightweight authentication to secure IoT networks," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–4.

[6] H. Bojinov and D. Boneh, "Mobile token-based authentication on a budget," in *Proc. 12th Workshop Mobile Comput. Syst. Appl. (HotMobile)*, 2011, pp. 14–19.

[7] K. Rhee, J. Kwak, S. Kim, and D. Won, "Challenge-response based RFID authentication protocol for distributed database environment," in *Security in Pervasive Computing (SPC)* (Lecture Notes in Computer Science), vol. 3450, D. Hutter and M. Ullmann, Eds. Berlin, Germany: Springer, 2005, pp. 70–84, doi: 10.1007/978-3-540-32004-3_9.

[8] A. Hiltgen, T. Kramp, and T. Weigold, "Secure internet banking authentication," *IEEE Secur. Privacy*, vol. 4, no. 2, pp. 21–29, Mar./Apr. 2006.

[9] Y. Shoukry, P. Martin, Y. Yona, S. Diggavi, and M. Srivastava, "PyCRA: Physical challenge-response authentication for active sensors under spoofing attacks," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2015, pp. 1004–1015.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[11] M. Bellare and S. K. Miner, "A forward-secure digital signature scheme," in *Advances in Cryptology–(CRYPTO)* (Lecture Notes in Computer Science), vol. 1666, M. Wiener, Ed. Berlin, Germany: Springer, 1999, pp. 431–448, doi: 10.1007/3-540-48405-1_28.

[12] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.

[13] L. Chen, S. P. Jordan, Y.-K. Liu, D. Moody, R. C. Peralta, R. A. Perlner, and D. C. Smith-Tone, "Report on post-quantum cryptography," NIST Interagency/Internal Report (NISTIR), Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 8105, 2016, doi: 10.6028/NIST.IR.8105.

[14] G. S. Vernam, "Cipher printing telegraph systems: For secret wire and radio telegraphic communications," *J. A.I.E.E.*, vol. 45, no. 2, pp. 109–115, Feb. 1926.

[15] Y. Lindell, "Secure multiparty computation," *Commun. ACM*, vol. 64, no. 1, pp. 86–96, Dec. 2020.

[16] D. R. Stinson and M. B. Paterson, *Cryptography: Theory and Practice*, 4th ed. Boca Raton, FL, USA: Taylor & Francis, 2017.

[17] C. Yan, Z. Li, L. Liu, and D. Lu, "Cheating identifiable (k, n) threshold quantum secret sharing scheme," *Quantum Inf. Process.*, vol. 21, no. 1, pp. 1–24, 2022.

[18] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[19] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. Int. Workshop Manag. Requirements Knowl. (MARK)*, Jun. 1979, pp. 313–318.

[20] Cybernetica. *Secure Computing Platform*. Accessed: Apr. 25, 2024. [Online]. Available: https://sharemind.cyber.ee/secure-computing-platform

[21] A. C. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. Symp. Found. Comput. Sci. (SFCS)*, Chicago, IL, USA, Nov. 1982, pp. 160–164.

[22] A. A. A. M. Kamal and K. Iwamura, "(Server-Aided) two-party multiplication of encrypted shares using (k, n) threshold secret sharing with $n = k$ servers," *IEEE Access*, vol. 9, pp. 113117–113129, 2021.

[23] A. A. A. M. Kamal and M. Fujisawa, "Improved protocols for secure multiplication using secret sharing scheme," *ICT Exp.*, vol. 10, no. 5, pp. 1019–1025, 2024.

[24] R. Cramer, I. B. Damgrd, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[25] A. A. A. M. Kamal and K. Iwamura, "Privacy preserving multi-party multiplication of polynomials based on (k, n) threshold secret sharing," *ICT Exp.*, vol. 9, no. 5, pp. 875–881, 2023.

[26] A. A. Aminuddin Mohd Kamal, K. Iwamura, and H. Kang, "Searchable encryption of image based on secret sharing scheme," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Kuala Lumpur, Malaysia, Dec. 2017, pp. 1495–1503.

[27] A. A. A. Mohd Kamal and K. Iwamura, "Searchable encryption using secret sharing scheme that realizes direct search of encrypted documents and disjunctive search of multiple keywords," *J. Inf. Secur. Appl.*, vol. 59, Jun. 2021, Art. no. 102824.

[28] S. Nagaraj, A. B. Kathole, L. Arya, N. Tyagi, S. B. Goyal, A. S. Rajawat, M. S. Raboaca, T. C. Mihaltan, C. Verma, and G. Suciu, "Improved secure encryption with energy optimization using random permutation pseudo algorithm based on Internet of Thing in wireless sensor networks," *Energies*, vol. 16, no. 1, p. 8, Dec. 2022.

[29] A. B. Kathole, J. Katti, D. Dhabliya, V. Deshpande, A. S. Rajawat, S. B. Goyal, M. S. Raboaca, T. C. Mihaltan, C. Verma, and G. Suciu, "Energy-aware UAV based on blockchain model using IoE application in 6G network-driven cybertwin," *Energies*, vol. 15, no. 21, p. 8304, Nov. 2022.

[30] S. D. Patil, A. B. Kathole, S. Kumbhare, K. Vhatkar, and V. V. Kimbahune, "A blockchain-based approach to ensuring the security of electronic data," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 11s, pp. 649–655, 2024.

[31] A. Kathole and D. Chaudhari, "Secure hybrid approach for sharing data securely in VANET," in *Proc. Int. Conf. Comput. Sci. Appl.*, 2022, pp. 217–221.

[32] A. B. Kathole and D. N. Chaudhari, "Securing the adhoc network data using hybrid malicious node detection approach," in *Proc. Int. Conf. Intell. Vis. Comput. (ICIVC)*, 2022, pp. 447–457.

[33] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," *Advances in Cryptology–(CRYPTO)* (Lecture Notes in Computer Science), vol. 7417, R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer, 2012, pp. 643–662, doi: 10.1007/978-3-642-32009-5_38.

[34] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Advances in Cryptology–(CRYPTO)* (Lecture Notes in Computer Science), vol. 576, J. Feigenbaum, Eds. Berlin, Germany: Springer, 2001, pp. 420–432, doi: 10.1007/3-540-46766-1_34.

[35] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: A framework for fast privacy-preserving computations," in *Computer Security–(ESORICS)*, (Lecture Notes in Computer Science), vol. 5283, S. Jajodia and J. Lopez, Eds. Berlin, Germany: Springer, 2008, pp. 192–206, doi: 10.1007/978-3-540-88313-5_13.

[36] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, "High-throughput semi-honest secure three-party computation with an honest majority," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2016, p. 87.

[37] A. A. A. Mohd Kamal and K. Iwamura, "Conditionally secure multiparty computation using secret sharing scheme for n <2k-1 (Short Paper)," in *Proc. 15th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2017, pp. 225–2255.

[38] K. Iwamura and A. A. A. M. Kamal, "Communication-efficient secure computation of encrypted inputs using (k, n) threshold secret sharing," *IEEE Access*, vol. 11, pp. 51166–51184, 2023.

[39] Y. Liang, H. V. Poor, and S. Shamai (Shitz), "Information theoretic security," *Found. Trends Commun. Inf. Theory*, vol. 5, nos. 4–5, pp. 355–580, 2009.

[40] T. M. Cover and J. A. Thomas, "Entropy, relative entropy and mutual information," in *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2005, ch. 2, pp. 13–55.

[41] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A new (k, n)-threshold secret sharing scheme and its extension, in *Information Security (ISC)*, (Lecture Notes in Computer Science), vol. 5222, T. C. Wu, C. L. Lei, V. Rijmen, and D. T. Lee, Eds. Berlin, Germany: Springer, 2008, pp. 455–470, doi: 10.1007/978-3-540-85886-7_31.

[42] M. Iwamoto and H. Yamamoto, "Strongly secure ramp secret sharing schemes for general access structures," *Inf. Process. Lett.*, vol. 97, no. 2, pp. 52–57, 2006.

[43] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. 12th Annu. ACM Symp. Theory Comput. (STOC)*, 1988, pp. 1–10.

[44] V. Singh and S. K. Pandey, "Revisiting cloud security threats: Replay attack," in *Proc. 4th Int. Conf. Comput. Commun. Autom. (ICCCA)*, Dec. 2018, pp. 1–6.

[45] Y. Feng, W. Wang, Y. Weng, and H. Zhang, "A replay-attack resistant authentication scheme for the Internet of Things," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE) IEEE Int. Conf. Embedded Ubiquitous Comput. (EUC)*, vol. 1, Jul. 2017, pp. 541–547.

[46] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in IoT: The challenges, and a way forward," *J. Netw. Comput. Appl.*, vol. 125, pp. 251–279, Jan. 2019.

[47] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu, "Security vulnerabilities of Internet of Things: A case study of the smart plug system," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1899–1909, Dec. 2017.

[48] J. G. Steiner, B. C. Neuman, and J. L. Schiller, "Kerberos: An authentication service for open network systems," in *Proc. USENIX Winter Conf.*, 1988, pp. 191–202.

[49] S. Rostampour, M. Safkhani, Y. Bendavid, and N. Bagheri, "ECCbAP: A secure ECC-based authentication protocol for IoT edge devices," *Pervas. Mobile Comput.*, vol. 67, Sep. 2020, Art. no. 101194.

[50] T. Islam, R. A. Youki, B. R. Chowdhury, and A. S. M. T. Hasan, "An ECC based secure communication protocol for resource constraints IoT devices in smart home," in *Proc. Int. Conf. Big Data, IoT, Mach. Learn.* (Lecture Notes on Data Engineering and Communications), M. S. Arefin, M. S. Kaiser, A. Bandyopadhyay, M. A. R. Ahad, and K. Ray, Eds. Singapore: Springer, 2021, pp. 431–444, doi: 10.1007/978-981-16-6636-0_33.
[51] S. Uppuluri and G. Lakshmeeswari, "Secure user authentication and key agreement scheme for IoT device access control based smart home communications," *Wireless Netw.*, vol. 29, no. 3, pp. 1333–1354, Apr. 2023.
[52] M. Wazid, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Secure three-factor user authentication scheme for renewable-energy-based smart grid environment," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3144–3153, Dec. 2017.
[53] L. Vishwakarma and D. Das, "SCAB–IoTA: Secure communication and authentication for IoT applications using blockchain," *J. Parallel Distrib. Comput.*, vol. 154, pp. 94–105, Aug. 2021.
[54] H.-M. Sun, Y.-H. Chen, and Y.-H. Lin, "OPass: A user authentication protocol resistant to password stealing and password reuse attacks," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 651–663, Apr. 2012.
[55] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, Nov. 1994, pp. 124–134.
[56] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, pp. 325–328, 1997.
[57] P.-A. Fouque et al., "Falcon: Fast-Fourier lattice-based compact signatures over NTRU," *Submission NIST's Post-Quantum Cryptogr. Standardization Process*, vol. 36, no. 5, pp. 1–75, 2018.
[58] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman, and W. Whyte, "NTRUSIGN: Digital signatures using the NTRU lattice," in *Topics in Cryptology–(CT-RSA)* (Lecture Notes in Computer Science), vol. 2612, M. Joye, Ed. Berlin, Germany: Springer, 2003, pp. 122–140, doi: 10.1007/3-540-36563-X_9.
[59] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-dilithium: A lattice-based digital signature scheme," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2018, no. 1, pp. 238–268, 2018.
[60] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The SPHINCS$^+$ signature framework," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2019, pp. 2129–2146.
[61] L.-J. Wang, K.-Y. Zhang, J.-Y. Wang, J. Cheng, Y.-H. Yang, S.-B. Tang, D. Yan, Y.-L. Tang, Z. Liu, Y. Yu, Q. Zhang, and J.-W. Pan, "Experimental authentication of quantum key distribution with post-quantum cryptography," *npj Quantum Inf.*, vol. 7, p. 67, May 2021.
[62] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 193–208, Jan. 2023.
[63] X. Chen, Q. Huang, H. Li, Z. Liao, and W. Susilo, "A novel identity-based multi-signature scheme over NTRU lattices," *Theor. Comput. Sci.*, vol. 933, pp. 163–176, Oct. 2022.
[64] L. Lai, H. El Gamal, and H. V. Poor, "Authentication over noisy channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 906–916, Feb. 2009.
[65] J. L. Massey, "An introduction to contemporary cryptology," *Proc. IEEE*, vol. 76, no. 5, pp. 533–549, May 1988.
[66] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "A physical-layer technique to enhance authentication for mobile terminals," in *Proc. IEEE Int. Conf. Commun.*, May 2008, pp. 1520–1524.
[67] R. Kikuchi and D. Ikarashi, "Progress of secure computation: Basic constructions and dedicated algorithms," *Fundam. Rev.*, vol. 12, no. 1, pp. 12–20, 2018.

KEIICHI IWAMURA (Member, IEEE) received the B.S. and M.S. degrees in information engineering from Kyushu University, Japan, in 1980 and 1982, respectively, and the Ph.D. degree from The University of Tokyo.

From 1982 to 2006, he was with Canon Inc. He is currently a Professor with Tokyo University of Science. His research interests include coding theory, information security, and digital watermarking. He is a fellow of the Information Processing Society of Japan and the Chairperson of the Technical Committee of Information Hiding and its Criteria for Evaluation and the Technical Committee of Enriched Multimedia, Institute of Electronics, and Information and Communication Engineers, Japan.

AHMAD AKMAL AMINUDDIN MOHD KAMAL (Member, IEEE) was born in Penang, Malaysia, in 1994. He received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in engineering from Tokyo University of Science, Japan, in 2017, 2019, and 2022, respectively.

He is currently an Assistant Professor with Tokyo University of Science. His research interests include information security, multiparty computation using secret sharing, and its application into searchable encryption.