Hindawi
Security and Communication Networks
Volume 2019, Article ID 5134534, 16 pages
https://doi.org/10.1155/2019/5134534

Research Article

Linear (t, n) Secret Sharing Scheme with Reduced Number of Polynomials

Kenan Kingsley Phiri[1] **and Hyunsung Kim**[1,2]

[1]*Department of Mathematical Sciences, University of Malawi, P.O. Box 280, Zomba, Malawi*
[2]*Department of Cyber Security, Kyungil University, Kyungsan, Kyungbuk 38428, Republic of Korea*

Correspondence should be addressed to Hyunsung Kim; kim@kiu.ac.kr

Received 18 March 2019; Revised 13 June 2019; Accepted 16 July 2019; Published 4 August 2019

Guest Editor: Mehdi Hussain

Threshold secret sharing is concerned with the splitting of a secret into n shares and distributing them to some persons without revealing its information. Any $t \leq n$ persons possessing the shares have the ability to reconstruct the secret, but any persons less than t cannot do the reconstruction. Linear secret sharing scheme is an important branch of secret sharing. The purpose of this paper is to propose a new polynomial based linear (t, n) secret sharing scheme, which is based on Shamir's secret sharing scheme and ElGamal cryptosystem. Firstly, we withdraw some required properties of secret sharing scheme after reviewing the related schemes and ElGamal cryptosystem. The designed scheme provides the properties of security for the secret, recoverability of the secret, privacy of the secret, and cheating detection of the forged shares. It has half computation overhead than the previous linear scheme.

1. Introduction

Information security has been a major concern over the past years in communication technology. The main concern has been how to make information confidential, authenticating and protecting it from being altered before reaching the receiver. Cryptography is part of the answer to these concerns [1]. The idea in cryptography is to prevent unauthorized use or alteration of information using mathematical tools. In a symmetric cryptosystem, this is achieved by using a shared secret key and in asymmetric cryptosystem it is achieved by using a pair of keys, public and private. The use of secret key or public and private key pair raised another problem of securely storing it. To address this problem, secret sharing schemes allow reliable storage without any risk [2, 3].

Threshold secret sharing is a method of splitting a secret s into n shares and distributing them to users such that the shares do not reveal any information about the secret [4]. In this care, the secret is a secret key, which needs to be stored securely. The secret is reconstructed easily if the authorized number of users combines their shares together. A subset of unauthorized users cannot reconstruct the secret or gain any information about the secret. The shares are sent to users using private channels so that each user should not have information of shares of the other users before reconstruction is done. Schemes that achieve this are called secret sharing schemes. Some schemes reveal the shares of all users who take part in secret reconstruction. In such schemes, users know all shares after the secret is reconstructed. This is called open reconstruction. While other schemes do not reveal the shares even after reconstruction is done for the reason that the shares may be reused. This is called closed reconstruction. Such schemes use a trusted third party to take the role of secret reconstruction as discussed by Martin [5].

Shamir introduced a (t, n) threshold secret sharing scheme, which provides secure way of sharing a secret [6]. The scheme starts with secret information, which is divided into n pieces of information called shares. The shares are distributed by a dealer to n individuals, called users; each user gets at least a share. These shares do not reveal any information about the secret. The dealer is entrusted with sharing of the secret to n users using share generation and distribution algorithms. No user knows the share of the others because distribution is done through secure channel. The

secret is reconstructed by any authorized subset of users with cardinality t using Lagrange interpolation.

Tompa and Woll exposed the weakness of Shamir's scheme by introducing a cheating concept [7]. Malicious users present forged shares during reconstruction so that the honest users get an invalid secret. The cheaters will be able to reconstruct a valid secret since they know all correct shares. Therefore, Shamir's scheme cannot withstand this attack even if there is only one cheater. Furthermore, they proposed an improved scheme which uses redundant shares to detect cheating so that malicious users are prevented. Redundant shares are extra shares used in reconstructing the secret other than the required threshold. Many schemes also solve the same problem of cheating detection [8–14].

Apart from cheating detection, there are schemes that identify cheaters in [15–20]. These schemes provide the method of identifying any forged share once it has been detected that cheating takes place. Identification is also good since it helps to recognize who the cheater is and can be removed from the system during the next sharing. Other schemes reconstruct the secret even though there are forged shares called robust secret sharing scheme (RSS) [21–23]. RSS prevents cheating by allowing the reconstruction of the correct secret even if some participants submit forged shares. However, the probability of recovering the secret in RSS depends on the number of forged shares submitted during reconstruction phase. Furthermore, some schemes verify shares and are called verifiable secret sharing schemes (VSS) [24, 25]. VSS also prevents cheating by verifying the shares received from the dealer. In VSS, users assume that the adversary may corrupt the dealer, as a result it is no longer trusted. Once user receives the share from the dealer, he (or she) shares it to other users and creates a check vector, which helps to identify cheaters. The user rejects the share if it does not agree to the check vector, otherwise accepts it.

To achieve cheating prevention, users are given a share of the secret plus additional information, which is used for cheating detection. This makes share size $|v_i|$, $\forall i \leq t - 1$, to increase greatly as compared to the secret size. However, it is shown by Carpentier et al. that a lower bound for the problem should be given as $|v_i| \geq |s| / \varepsilon$, where ε is the cheating probability [26]. The lower bound is based on the assumption that $t - 1$ cheaters somehow know the secret before they cheat a user U_t. This is called Carpentieri, De Santis, and Vaccaro (CDV) assumption. However, Ogata and Kurosawa proposed a scheme that detects cheating based on the assumption that no cheating user knows the secret [27]. The scheme's share size reaches the lower bound of $|v_i| \geq (|s| - 1)/\varepsilon + 1$. This is called Ogata, Kurosawa, and Stinson (OKS) assumption.

Linear secret sharing schemes have been studied because of their application in multiparty computation and function sharing [28, 29]. The schemes are able to detect cheating behavior of malicious users during reconstruction of the secret; hence, an honest user cannot be fooled. Liu et al.'s scheme could be applied if system needs to share more than one secret [28]. Cramer et al.'s scheme depends its security on the universal hash function, which means that it is not unconditionally secure, where Lin and Harn's scheme has been proved to be easily broken by a simple attack as pointed out by Ghodosi [30, 31]. Liu et al.'s scheme uses two polynomials to detect cheating during secret reconstruction and reduce the share size given to a user. Use of two polynomials increases the number of computations the scheme undergoes. As a result, there is an increased computation overhead.

This paper proposes a new linear (t, n) threshold secret sharing scheme based on a polynomial called polynomial based linear scheme (PBLS), which optimizes the number of computation overhead while maintaining security and privacy concerns. The goals that achieve security and privacy concerns (SP) of PBLS are

(i) SP1: provide security of the secret.

(ii) SP2: provide recoverability of the secret once shared.

(iii) SP3: provide privacy of the secret and shares.

(iv) SP4: provide cheating detection feasibility not only for one cheater but also $t - 1$ cheaters so that any malicious behavior could be detected.

The goal that achieves computation overhead (CO) of PBLS is

(i) CO: reduce the number of polynomials used so that computational overhead is reduced as compared to Liu et al.'s scheme.

To achieve these goals, PBLS uses ElGamal cryptosystem and Shamir's scheme as core operations. ElGamal cryptosystem helps to design a basic scheme, which is the initialization of PBLS. The basic scheme aims at hiding the secret during share generation phase, which can be revealed during cheating detection. PBLS applies Shamir's secret sharing scheme to share the secret, which uses the polynomial $f(x)$ such that the element hiding the secret becomes the coefficient of x. Therefore, reconstruction of the secret in PBLS uses Lagrange interpolation, which comes up with the polynomial $f'(x)$. Revealing the secret helps to detect cheating. PBLS has an advantage over Liu et al.'s scheme in the computation overhead concern. Furthermore, PBLS provides cheating detection feasibility, which is not available in Shamir's scheme.

2. Preliminaries

This section provides some basic mathematical and cryptographic concepts, which are major tools in secret sharing schemes, and reviews related works. First of all, the definition of finite field is provided together with some properties [32–34]. Polynomials in a finite field and Lagrange interpolation are also discussed to give understanding on the concepts. ElGamal cryptosystem is briefly discussed because it is used in construction of basic scheme. After that, we review related works such as linear secret sharing schemes, access structure, and some previous schemes like Shamir's and Liu et al.'s schemes [6, 14].

2.1. Basic Mathematical and Cryptographic Concepts. This section gives an overview of some mathematical concepts, which are useful in secret sharing like finite field, polynomial

and Lagrange interpolation. For more details on finite field, refer to [32–34]. The section also gives an overview of ElGamal cryptosystem, which assists in construction of basic scheme.

2.1.1. Finite Field

Definition 1. A finite field $\mathbb{F}$ is a finite set on which addition, subtraction, multiplication, and division are defined and the following axioms are satisfied.

(1) Associative: for all a, b, and c in $\mathbb{F}$, $a + (b + c) = (a + b) + c$ and $a \cdot (b \cdot c) = (a \cdot b) \cdot c$.

(2) Commutative: for all a and $b \in \mathbb{F}$, $a + b = b + a$ and $a \cdot b = b \cdot a$.

(3) Existence of identity: there exists elements e and $e' \in \mathbb{F}$, such that $a + e = e + a = a$ and $a \cdot e' = e' \cdot a = a$.

(4) Existence of inverse: for every element $a \in \mathbb{F}$, there exists an element $-a$ such that $a + (-a) = e$. Similarly for every element $a \in \mathbb{F}$, there exists an element $a^{-1} \in \mathbb{F}$ such that $a \cdot a^{-1} = e'$.

(5) Distributive: for all a, b, and c in $\mathbb{F}$, $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

Note that an element $-a$ is called additive inverse and another element is called multiplicative inverse. An element e is an additive identity and an element e' is multiplicative identity. In this paper, we take $e = 0$ and $e' = 1$.

Definition 2 (finite field of order p). Let p be a prime. The set of integers $\mathbb{Z}_p = \{0, 1, 2, \ldots, p - 1\}$ with addition and multiplication performed modulo p is a finite field of order p and is denoted as $\mathbb{F}_p = \mathbb{Z}_p$.

Proposition 3 (multiplicative inverse). *Let p be a prime. Element $a \in \mathbb{Z}_p$, $a \neq 0$ has a multiplicative inverse $b = a^{-1}$ such that $a \cdot b \equiv 1 \bmod p$.*

Definition 4 (group of units). Let p be a prime. A group $\mathbb{F}_p^*$ is a set that contains nonzero elements and is called a group of units.

2.1.2. Polynomials over Finite Field

Definition 5 (polynomial over $\mathbb{F}_p$). Let $\mathbb{F}_p$ be a field. Any expression

$$f(x) = \sum_{i=0}^{t} a_i x^i \quad a_i \in \mathbb{F}_p, \tag{1}$$

where t is an arbitrary positive integer which is called a polynomial over $\mathbb{F}_p$.

Definition 6 (degree of polynomial $f(x)$). Given a nonzero polynomial $f(x) = \sum_{i=0}^{t} a_i x^i$, where $a_t \neq 0$, the number t is said to be the degree of $f(x)$ denoted as $\deg f(x)$.

Definition 7 (equal polynomials). Let $f(x) = \sum_{i=0}^{t} a_i x^i$ and $f'(x) = \sum_{i=0}^{m} b_i x^i$, where $a_t \neq 0$ and $b_m \neq 0$ are two polynomials of degrees t and m, respectively. The two polynomials are equal and write $f(x) = f'(x)$, if $t = m$ and $a_i = b_i$ for all $i = \{0, 1, 2, \ldots, t\}$.

Definition 8 (roots of a polynomial). An element $\alpha \in \mathbb{F}$ is called a root of $f(x)$ if $f(\alpha) = 0$.

Proposition 9. *A polynomial $f(x) = \sum_{i=0}^{t} a_i x^i$, $a_i \in \mathbb{F}$ of degree t cannot have more than t roots in the field $\mathbb{F}$.*

2.1.3. Lagrange Interpolation. This is a method of reconstructing a polynomial from given known points. The polynomial constructed by Lagrange interpolation is called Lagrange interpolation polynomial, which is unique. To reconstruct the polynomial of degree t, $t + 1$ values are required, i.e., (α_i, β_i) for all $i = 0, 1, 2, \ldots, t$ such that $f(\alpha_i) = \beta_i$.

Proposition 10. *Let α_i for all $i = 0, 1, 2, \ldots, t$ be distinct elements of $\mathbb{F}$ and β_i for all $i = 0, 1, 2, \ldots, t$ be arbitrary elements of $\mathbb{F}$. There exists no more than one polynomial $f(x)$ of degree at most t such that $f(\alpha_i) = \beta_i$ for all $i = 0, 1, 2, \ldots, t$.*

Theorem 11 (see [33]). *Let $\alpha_0, \alpha_1, \ldots, \alpha_t$ be distinct elements of $\mathbb{F}$ and $\beta_0, \beta_1, \ldots, \beta_t$ be arbitrary elements of $\mathbb{F}$. There exists a unique polynomial*

$$f(x) = \sum_{i=1}^{t} \beta_i \frac{(x - \alpha_0) \ldots (x - \alpha_{i-1})(x - \alpha_{i+1}) \ldots (x - \alpha_t)}{(\alpha_i - \alpha_0) \ldots (\alpha_i - \alpha_{i-1})(\alpha_i - \alpha_{i+1}) \ldots (\alpha_i - \alpha_t)} \tag{2}$$

of degree at most t such that $f(\alpha_i) = \beta_i$ for all $i = 0, 1, 2, \ldots, t$.

Proof. We adopt the proof by Slinko [33]. The polynomial in Equation (2) was constructed as follows. First construct polynomials $g_i(x)$ of degree t such that $g_i(\alpha_i) = 1$ and $g_i(\alpha_j) = 0$ for $i \neq j$. These polynomials are

$$g_i(x) = \frac{(x - \alpha_0) \ldots (x - \alpha_{i-1})(x - \alpha_{i+1}) \ldots (x - \alpha_t)}{(\alpha_i - \alpha_0) \ldots (\alpha_i - \alpha_{i-1})(\alpha_i - \alpha_{i+1}) \ldots (\alpha_i - \alpha_t)}. \tag{3}$$

Thus, the polynomials $g_0(x), g_1(x), \ldots, g_t(x)$ are constructed. Furthermore, we multiply by β_i for $i = 0, 1, 2, \ldots, t$ and obtain the polynomials $\beta_0 g_0(x), \beta_1 g_1(x), \ldots, \beta_t g_t(x)$. Summing the polynomials $\beta_i g_i(x)$ the desired polynomial $f(x)$ is constructed as Equation (4).

$$f(x) = \sum_{i=0}^{t} \beta_i g_i(x) \tag{4}$$

We set $f(\alpha_i) = \beta_i$ as required. This polynomial is unique by Proposition 10. □

2.1.4. ElGamal Cryptosystem. ElGamal cryptosystem is in a family of public key cryptography [35]. Public key cryptography uses public key and private key to encrypt and decrypt

messages, respectively, such that knowledge of private key makes decryption easy. Without knowing the private key, it is impossible to decrypt a message in acceptable time. Security of ElGamal is based on discrete logarithm problem (DLP). Therefore, an attacker has to solve DLP to decrypt an intercepted message on ElGamal cryptosystem.

Definition 12 (finite field DLP (FFDLP)). Given a finite field $\mathbb{F}_p$, a primitive element g of $\mathbb{F}_p$, and a nonzero element b of $\mathbb{F}_p$, the FFDLP of b to base g, written as $\log_g(b)$, is determining the least nonnegative integer i such that $d = g^i$.

Three algorithms are used in ElGamal cryptosystem, which are key generation, encryption, and decryption. We assume Alice and Bob want to communicate over an insecure channel. They have to generate a public and private key pair for encryption and decryption as follows.

Key Generation. Bob will do the following steps:

(i) Generate a large prime p and a generator g of a multiplicative group ${\mathbb{Z}_p}^*$

(ii) Select a random integer $b \in {\mathbb{Z}_p}^*$ such that $1 \leq b \leq p - 2$

(iii) Compute $Y \equiv g^b \bmod p$.

The public key for Bob is (p, g, Y) and b is the private key. Bob publishes the public key so that if anyone wants to send an encrypted message to him, they can use it. Y is an element in ${\mathbb{Z}_p}^*$, which has a multiplicative inverse in the group.

When Alice wants to send a message M to Bob, she needs to use Bob's public key to encrypt the message. The following are the steps she takes:

Encryption

(i) Encode the message M such that $1 \leq M \leq p - 1$.

(ii) Select a random exponent k.

(iii) Compute $C_1 = g^k$ and $C_2 = M \cdot Y^k$.

The encrypted message sent to Bob is a pair (C_1, C_2).

Once Bob receives the message, he uses his private key to decrypt it in the following way:

Decryption

(i) Compute ${C_1}^{-b} = g^{-bk}$

(ii) Compute $M = {C_1}^{-b} \cdot C_2 = g^{-bk} \cdot M \cdot g^{bk}$.

The element ${C_1}^{-b}$ is the multiplicative inverse of g^{bk}.

2.2. Related Works. This section reviews linear secret sharing schemes, access structure of secret sharing schemes, and some previous schemes like Shamir's and Liu et al.'s schemes [6, 14]. Furthermore, the section discusses the strong and weak properties of the reviewed schemes.

2.2.1. Linear Secret Sharing Scheme. Linear (t, n) secret sharing scheme is a special type of secret sharing scheme where all the n shares of the secret satisfy a linear relationship [6, 14]. The Definition 13 gives what linear secret sharing scheme is.

Definition 13 (linear secret sharing scheme). A (t, n) secret sharing scheme is a linear secret sharing scheme when the n shares, $v_1, v_2, \ldots, v_n$ can be presented as in Equation (5)

$$(v_1, v_2, \ldots, v_n) = (k_1, k_2, \ldots, k_t)\, H, \tag{5}$$

where H is a public $t \times n$ matrix whose any $t \times t$ submatrix is not singular. The vector $(k_1, k_2, \ldots, k_t)$ is randomly chosen by the dealer.

According to Definition 13, we can see that Shamir's (t, n) secret sharing scheme is a linear scheme. Let

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}. \tag{6}$$

The shares $v_i = f(i)$, $i = 1, 2, \ldots, n$ can be presented as in Equation (7)

$$(v_1, v_2, \ldots, v_n) = (a_0, a_1, a_2, \ldots, a_{t-1})\, H, \tag{7}$$

where $h_{i,j} = j^{i1}$ ($h_{i,j}$ denotes the entry at ith row and jth column of matrix H).

2.2.2. Access Structure. Assume that U is the set of users where $U = \{U_1, U_2, \ldots, U_n\}$ and D is the dealer who facilitates secret sharing. An access structure is defined as follows.

Definition 14 (access structure). Let 2^U be the power set of the set of all users U. The set $\Gamma \subseteq 2^U$ of all authorized coalitions is called the access structure of the secret sharing scheme [33].

If a subset is in the access structure, all sets that contain that subset should also form part of the access structure. Let X and Y be subsets of Γ such that $X \subseteq Y$. An access structure Γ may be any subset of 2^U such that

$$\begin{gathered} X \in \Gamma \\ \text{and } X \subseteq Y, \\ \text{then } Y \in \Gamma. \end{gathered} \tag{8}$$

The condition in Equation (8) attached to access structure is called monotone property, which shows that if a smaller subset can know the secret, then any other larger set containing the subset will know it too.

Definition 15. Let $\Gamma \subseteq 2^U$ be an access structure. A coalition $C \subseteq U$ is called minimal authorized coalition if it is authorized and any proper subset of C is not authorized [33].

For example, if $T \subset C$ and C is the minimal authorized coalition, then T is not authorized because $|T| < |C|$. The assumption is that every user is in at least one minimal coalition and otherwise is not useful in reconstructing the secret. In a linear (t, n) secret sharing scheme, the minimal coalition has subsets with t users.

2.2.3. Shamir's (t, n) Threshold Scheme. Shamir proposed a (t, n) threshold scheme that splits a secret $s \in S$ into n shares, which are distributed to n users [6]. Splitting is done by a dealer using an algorithm called share generation algorithm. The algorithm uses a polynomial $f(x)$ of degree t - 1 to generate and distribute shares. The secret is reconstructed based on interpolating a polynomial using Lagrange interpolation, which is reconstructed by t users. The users combine their shares to reconstruct a polynomial $f'(x)$ of degree t using reconstruction algorithm. The algorithm inputs the user's identity i and their share v_i, which forms a point or an ordered pair (i, v_i) for all $i = 1, 2, \ldots, t$ and outputs the secret $f'(0) = s$. Shamir's scheme has the following important properties.

(i) Share size is exactly equal to secret size.

(ii) If a new player joins or leaves, it is easy to add or delete shares without affecting the other shares.

(iii) It is easy to change the shares of the same secret just by changing the polynomial without breaching any security.

(iv) t - 1 users do not reveal any information about the secret.

However, Tompa and Woll discovered that the scheme cannot withstand cheating if there is an untrusted user during secret reconstruction [7]. As a result, Shamir's scheme faces the following challenges during secret reconstruction.

(i) Any malicious user can present a forged share without being noticed.

(ii) It is difficult to detect if the reconstructed secret is invalid.

(iii) A malicious user, once is successful in cheating other users, will be able to reconstruct the valid secret.

2.2.4. Cheating Prevention. Cheating prevention in secret sharing became a great concern after Tompa and Woll introduced cheating concept. As a result, many schemes with cheating prevention are proposed where some detect cheating, others identify cheaters, and so on. Some of the categories of cheating prevention are as follows.

(i) Cheating detection: schemes provide the method to detect any forged share submitted for secret reconstruction by malicious user [7, 12]. The assumption is that the dealer is trusted.

(ii) Cheater identification: schemes provide the method to detect and identify any forged share presented for secret reconstruction by a malicious user [15, 17]. The assumption is that the dealer is trusted.

(iii) Robust secret sharing: schemes assume the dealer is trusted. Schemes can reconstruct a correct secret even if there are a number of forged shares presented by untrusted user [22].

(iv) Verifiable secret sharing: schemes assume that the dealer is not trusted. Each user verifies the shares if valid using verification algorithm before reconstruction is done [9, 25].

2.2.5. Liu Et Al.'s Scheme. Liu et al. proposed a linear threshold secret sharing scheme, which is capable of cheating detection with share size $|v_i| \geq |s| / \varepsilon$, where $\varepsilon > 0$ is the probability of cheating [14]. Liu et al.'s scheme is a combination of two Shamir's schemes. Two polynomials are used to share a secret s. Cheating detection is done by finding a random element $r \in \mathbb{Z}_p$ during secret reconstruction.

Liu et al.'s scheme adopts Shamir's scheme in sharing the secret. This means that most properties of Liu et al.'s scheme are similar to Shamir's scheme. However, there are some properties, which Shamir's scheme does not have. The properties include the following.

(i) Share size given to each user is equal to or greater than the secret size, i.e., $|v_i| \geq |s| / \varepsilon$.

(ii) Detect cheating whenever a forged share is presented during reconstruction.

However, the scheme uses two polynomials to achieve the property of cheating detection, which makes number of computations to double as compared to Shamir's scheme.

3. New Linear (t,n) Secret Sharing Scheme

This section proposes a new linear (t, n) threshold secret sharing scheme called polynomial based linear scheme (PBLS), which is based on one polynomial to reduce computational overhead of Liu et al.'s scheme and improve security of Shamir's scheme in terms of cheating detection. PBLS is capable of cheating detection for any forged shares presented for secret reconstruction with the help of the coefficients of x in the polynomial $f'(x)$. The coefficients are determined by a basic scheme which is an initialization of PBLS that adopts its properties from ElGamal cryptosystem. The security of PBLS is based on Shamir's scheme and ElGamal cryptosystem. PBLS provides perfect secret sharing, which is a required feature in all secret sharing schemes. The designed properties of PBLS withdrawn from the previous schemes satisfy SP1, SP2, SP3, and SP4, and computation overhead concern of CO.

3.1. Fundamental Properties. Basic scheme and PBLS adopt their properties from already existing scheme of ElGamal and Shamir. This makes the security of basic scheme and PBLS similar to the security of Shamir's scheme and ElGamal cryptosystem.

3.1.1. Properties of Basic Scheme. Basic scheme adopts its properties from ElGamal cryptosystem, which uses finite field elements to hide information [35]. The aim of basic scheme is to hide a secret in share generation phase but can be revealed when cheating detection is taking place. Secret hiding is done by multiplying a random element r by the secret $s \in S$ to produce z for all r, s, and $z \in \mathbb{F}_p$. The secret is revealed when a multiplicative inverse of r is multiplied by z.

Security of ElGamal cryptosystem depends on the hardness of FFDLP. Therefore, basic scheme adopts the same security as ElGamal cryptosystem. Propositions 18 and 19 give the properties of basic scheme. However, to understand

these properties better, we first provide Definition 16 and Corollary 17 without proof. The proofs for definition and corollary can be obtained in [36].

Definition 16. $a \equiv b \bmod p$ if and only if a and b leave the same remainder when divided by p.

Corollary 17. *The integer c is the remainder when a is divided by p if and only if $a \equiv c \bmod p$, where $0 \leq c < p$.*

The following are the properties of basic scheme.

Proposition 18. *Let s and $r \in \mathbb{Z}_p$ be a secret and a random element, respectively, and $z \equiv s \cdot r \bmod p$ such that p is prime. It has FFDLP difficulty to withdraw s and r from the element $z \in \mathbb{Z}_p$.*

Proof. Since elements s and r are field elements, the operation $n = s \cdot r \equiv z \bmod p$ is a modulo multiplication. Thus z is a remainder when p divides the integer n. Assume that there exists only one integer n, which leaves a remainder z when $p \mid n$, then

$$n = t \cdot p + z \quad \forall t > 0. \tag{9}$$

By Corollary 17, $n \equiv z \bmod p$. Therefore, t is unique. Let $d = a \cdot b$ such that $a \neq s \neq r$ and $b \neq s \neq r$. By Definition 16, $n \equiv d \bmod p$ if and only if n and d leave the same remainder when they are divided by p. Thus

$$\begin{aligned} n &= t \cdot p + z \\ d &= t \cdot p + z \\ &\forall t > 0. \end{aligned} \tag{10}$$

This implies that

$$\begin{aligned} n &= d \\ s \cdot r &= a \cdot b. \end{aligned} \tag{11}$$

But $a \neq s \neq r$ and $b \neq s \neq r$. Therefore, t is not unique. This contradicts the fact that n is the only integer that leaves a remainder z when $p \mid n$. Therefore $n \equiv d \bmod p$, where $d \neq n$. Element z is a remainder whenever $t \cdot p$ divides d such that $d > t \cdot p$ and n such that $n > t \cdot p$. Integers d and n contain different factors since they are not equal hence difficult to determine s and r from z, which is FFDLP. □

Proposition 19. *Let $z \equiv s \cdot r \bmod p$ such that $s \in \mathbb{Z}_p$ is a secret, $r \in \mathbb{Z}_p$ is a random element, and p is prime. It is impossible to determine s from z, which is based on the difficulty of the fractional decomposition.*

Proof. By Proposition 18, it is difficult to determine s and r from z because z does not reveal s and r. If we assume that we know the value of r, then it is possible to determine s. Since r is known, the multiplicative inverse of r can be computed from finite field $\mathbb{F}_p$. Multiplying z by r^{-1} gives s as follows:

$$z \cdot r^{-1} \equiv s \cdot r \cdot r^{-1} \bmod p \equiv s \bmod p. \tag{12}$$

Knowledge of r helps to determine s. Therefore, by contrapositive, we cannot determine s from z, which is based on the fractional decomposition difficulty. □

It is noted that though FFDLP is applied in basic scheme, there is a difference with ElGamal cryptosystem. The difference is that basic scheme makes no use of exponentiation, which helps it to operate in polynomial time.

3.1.2. Properties of PBLS. Any secret sharing scheme should be secure from malicious users by denying them the opportunity to obtain the secret when the required number of users is not reached. At the same time, the secret should be able to be reconstructed after sharing. PBLS adopts its properties from Shamir's secret sharing scheme, which shares a secret to n users to be recovered by t users where $t \leq n$ using a polynomial of degree $t - 1$, where the coefficients of x^0 and x are s and z, respectively. Therefore, all the properties for Shamir's scheme also hold for PBLS. However, PBLS also consists of some properties of ElGamal cryptosystem because of the use of basic scheme. Two fundamental properties of designing PBLS, which are adopted from Shamir's scheme, are given in Proposition 20 such that every secret sharing scheme has to be achieved.

Proposition 20 (SP1 and SP3). *Let s be the secret and t the threshold. Any less than t users cannot know the secret.*

Proposition 21 (SP2). *Let t be the threshold of a secret sharing scheme. Any t or more than t users should be able to reconstruct the secret by combining their shares together.*

Since Shamir's scheme is linear that is n shares of secret satisfy a linear relationship, PBLS is also linear. However, PBLS has a property of SP4 as in Proposition 22.

Proposition 22 (SP4). *Let t be the threshold of a secret sharing scheme and there are any less than t forged shares used for secret reconstruction. The shares will be detected during secret reconstruction.*

Any secret sharing scheme, which prevents cheating, must give to each participant shares whose sizes are at least the size of the secret plus $\log 1/\varepsilon$, where ε is the probability of successful cheating [26]. The Proposition 23 gives a property of the share size of PBLS given to users.

Proposition 23. *Let $v_i = \{f(i), y\}$ be the share given to each user. The share size of PBLS attains the bounds of $|v_1| \geq |s|/\varepsilon$.*

Any secret sharing scheme has a set of users who are allowed to make reconstruction of the secret called the access structure based on Definitions 14 and 15. Proposition 24 provides an access structure of PBLS.

Proposition 24. *Let $U = \{U_1, U_2, \ldots, U_n\}$ be the set of users. The access structure of PBLS is a set $\Gamma \subseteq 2^{|U|}$ such that $X \subseteq \Gamma$, where $|X| \geq t$.*

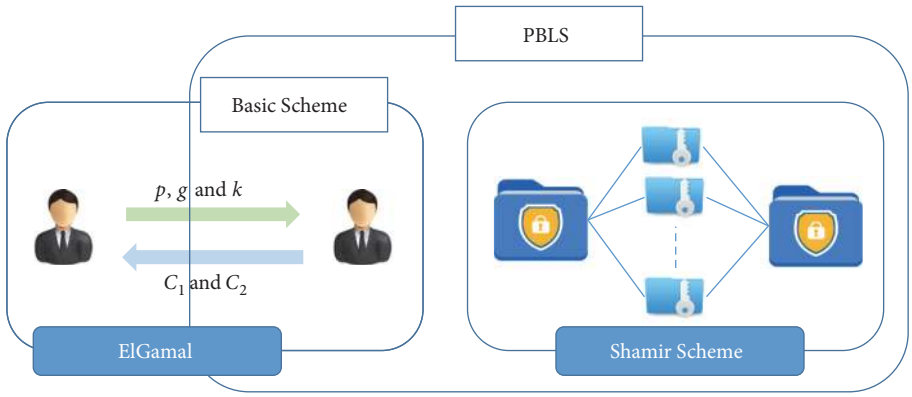

Figure 1: Adoption of ElGamal cryptosystem and Shamir's scheme in PBLS.

Proof. The scheme uses Lagrange interpolation to come up with a polynomial $f(x)$ of degree t - 1. By Theorem 11, t points are required to interpolate this polynomial. Thus any subset of t or more than t shares is authorized to reconstruct the secret. □

PBLS is composed of basic scheme and Shamir's scheme. However, exponentiation is not used in PBLS to reduce computation cost. Figure 1 shows the properties of ElGamal cryptosystem and Shamir's secret sharing scheme and how they are adopted in basic scheme and PBLS.

3.1.3. Adversary Model. In any cryptographic application, an attacker A has different goals to achieve for the attack mode to it. Secret sharing schemes face cheating attack, which was discovered by Tompa and Woll. Despite different applications of secret sharing schemes, a malicious user presents forged shares during secret reconstruction. Therefore, the following are some goals of cheaters in secret sharing schemes try to achieve:

(i) To recover the valid secret while the honest users are unable to detect cheating [29]. In this case, honest users believe the secret to be valid.

(ii) To recover the secret while the honest users are able to detect cheating [37]. The honest users will not have access to this secret; hence, they simply assist A to reconstruct it without their knowledge.

There are two assumptions in which cheaters behave. These are OKS and CDV as discussed [26, 27]. CDV assumes that cheaters already know the secret to be reconstructed. They only aim at blocking the correct reconstruction of the secret while they already have the secret. Honest users will get the invalid secret. On the other hand, OKS assumes that the cheater does not know the secret to be reconstructed. The aim is to block the correct reconstruction of the secret, but at the end they should be able to get a valid secret. PBLS considers OKS assumption because the aim of secret sharing is that the secret should not be known before reconstruction.

Cheating becomes successful when cheaters managed to reconstruct a valid secret while honest users failed to detect that cheating takes place. PBLS makes sure that A does not recover the secret whenever cheating detection is achieved. To prevent A from learning the secret, a closed reconstruction is done where no user can see the share of the other users. This also prevents malicious users who communicate their shares during reconstruction after learning the shares of honest user as described by [38]. Such users are called rushing cheaters.

3.2. Proposed Schemes. This subsection proposes basic scheme and PBLS. Basic scheme has two algorithms, which are secret hiding and secret revealing. The secret is hidden with field element in secret hiding algorithm. It is revealed using the multiplicative inverse of the element in secret revealing algorithm. PBLS has three algorithms, which are share generation, secret reconstruction, and cheating detection.

3.2.1. Basic Scheme. This subsection proposes basic scheme, which is the basis for constructing PBLS proposed in Section 3.2.2. Basic scheme provides a conceptual process of how PBLS detects cheating during secret reconstruction. A secret is hidden by a field element r and can be revealed by a multiplicative inverse b of the element r.

Secret Hiding. Consider a finite field $\mathbb{F}_p$ in which p is a prime such that p - 1 has at least one large prime factor. If p - 1 has only small prime factors, then computing FFDLP is easy as pointed out by [5]. By Proposition 3, all nonzero elements a in $\mathbb{F}_p$ have a multiplicative inverse b such that $a \cdot b \equiv 1 \bmod p$. A secret s is also the field element as $s \in \mathbb{F}_p$. Any random number r except 1 can be used to hide the secret s. The algorithm for hiding the secret avoids using 1 because it is a multiplicative identity therefore cannot hide s. After multiplying r by s, a different field element z is obtained. Hence Equation (13) follows

$$z \equiv r \cdot s \mod p. \tag{13}$$

Algorithm 25 describes how a secret s is hidden using field element in basic scheme.

Algorithm 25 (secret hiding).

Input. Secret s

Output. Element z

Process

(i) Choose a random element $r \in \mathbb{Z}_p$.

(ii) Compute z by multiplying r by s.

(iii) Output element $z \in \mathbb{Z}_p$.

Figure 2 illustrates how Algorithm 25 works.

The element z does not reveal any information of s and r in basic scheme.

Secret Revealing. Whenever one wants the secret s back, he (or she) simply computes the multiplicative inverse of r, given as b, and multiplies it by z to get the secret s. Therefore, multiplying z by multiplicative inverse of r is the same as multiplying r by r^{-1} by s. This means 1 is multiplied by s to get a result s as in Equation (14)

$$s \equiv b \cdot z \bmod p \equiv r^{-1} \cdot r \cdot s \bmod p \equiv s \bmod p. \quad (14)$$

Algorithm 26 shows how to reveal the secret s using the multiplicative inverse of r.

Algorithm 26 (secret revealing).

Input. Elements z and r

Output. Secret s

Process

(i) Compute multiplicative inverse b of element r, i.e., $b = r^{-1}$.

(ii) Compute s by multiplying b by z.

(iii) Output s.

Figure 3 illustrates Algorithm 26: how revealing the secret occurs.

Basic scheme requires r to be kept secret so that the secret remains private and secure. Otherwise, any adversary will be able to compute the inverse of r and reveal s as is done in Algorithm 26. We demonstrate this with a dummy example below.

Example 27. Let $p = 23$ and $s = 12$. Secret s can be hidden as follows. Choose a random element $r = 7 \in \mathbb{Z}_{23}$. Compute z by multiplying r by s to obtain z

$$z \equiv r \cdot s \bmod 23 \equiv 7 \cdot 12 \bmod 23 \equiv 15 \bmod 23. \quad (15)$$

The secret 12 is hidden as 15. It is difficult for an adversary A to know the secret 12 and the random 7 from 15 alone unless he (or she) solves FFDLP. The secret s is revealed by computing b, the multiplicative inverse of r

$$b = r^{-1} = 10, \quad (16)$$

and computing s by multiplying y by z

$$s \equiv b \cdot z \bmod 23 \equiv 10 \cdot 15 \bmod 23 \equiv 12 \bmod 23. \quad (17)$$

Note that in practice, p should be a large prime number for the scheme to be secure enough.

3.2.2. Polynomial Based Linear Scheme (PBLS). In this subsection, a linear (t, n) threshold secret sharing scheme, PBLS, is proposed, which provides cheating detection based on basic scheme and Shamir's secret sharing. We assume using two trusted third parties dealer D and combiner C. D generates and distributes shares to n users while C collects any t shares and reconstructs the secret. A trusted user can also be C depending on the application. C does not reveal shares after reconstruction and hence performs a closed reconstruction. In addition, C performs cheating detection in secret reconstruction phase. PBLS has three algorithms, which are share generation, secret reconstruction, and cheating detection.

Share Generation. As Shamir's scheme, share generation algorithm starts with D setting public parameters, prime p and threshold t. D chooses a random element r from a finite field $\mathbb{F}_p$, then hides the secret using Algorithm 25. The output is element $z \in \mathbb{F}_p$. D computes $b \in \mathbb{F}_p$, a multiplicative inverse of r. The element b is sent to n users on public channel while element z becomes a coefficient of x in polynomial $f(x)$. To share the secret s, D chooses a random polynomial of degree t - 1, which has constraints of two coefficients, $a_0 = s$ and $a_1 = z$. D computes $f(i)$ for all $i = 1, 2, \ldots, n$ and distributes $v_i = (i, f(i))$ to users where $i = 1, 2, \ldots, n$. Algorithm 28 shows how shares are generated and distributed to users.

Algorithm 28 (share generation).

Input. Secret s

Output. Secret shares v_i where $i = 1, 2, \ldots, n$

Process

(i) D uses Algorithm 25 to compute z.

(ii) D uses Algorithm 26 to compute b.

(iii) D chooses a random polynomial $f(x)$ of degree t - 1 over $\mathbb{F}_p$, i.e.,

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \quad (18)$$

such that $a_0 = s$ and $a_1 = z$.

(iv) D computes $f(i)$ and distributes $v_i = (i, f(i))$ and b to U_i for all $i = 1, 2, \ldots, n$ secretly.

Figure 4 illustrates the share generation algorithm in PBLS.

Users cannot obtain information of the secret s from z without the knowledge of r unless they have to solve FFDLP.

Secret Reconstruction. If the secret is required for use, any $t \leq n$ users combine their shares together to reconstruct it.

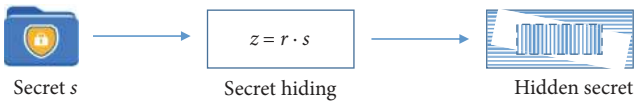

Figure 2: Secret hiding in basic scheme.

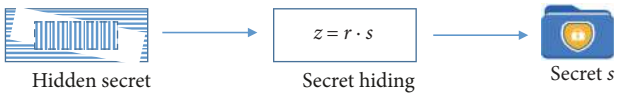

Figure 3: Secret revealing in basic scheme.

The combiner can be one of the user or a trusted third party who does the reconstruction without revealing the shares. Users send their shares to C together with b. C uses Lagrange interpolation to reconstruct the polynomial of degree t - 1 from at least t points, i.e., $(1, f(1))$, $(2, f(2))$, ... $(i, f(i))$ and the secret is $f(0)$ if all users are honest. Figure 5 illustrates how secret reconstruction is done in PBLS.

Algorithm 29 (secret reconstruction).

Input. Any list of t shares

Output. Secret $s' = s$ if there is no cheater or $s' \neq s$ if cheaters exist

Process

(i) C reconstructs $f'(x)$ from (i, v_i) using Lagrange interpolation

$$f'(x) = \sum_{i=1}^{t} v_i \prod_{j=1, j\neq i}^{t} \frac{x_i - x}{x_i - x_j} \tag{19}$$

(ii) C outputs polynomial $f'(x) = a_0' + a_1'x + a_2'x^2 + \ldots + a_{t-1}'x^{t-1}$.

By Theorem 11, the reconstructed polynomial $f'(x)$ is unique and if there is no cheating then polynomial $f'(x)$ is equal to polynomial $f(x)$. Therefore, the secret is $f'(0) = a_0' = s' = s$.

Cheating Detection. It is important to check if there are forged shares presented during secret reconstruction. In this phase, PBLS uses basic scheme to reveal the secret since it is hidden in the coefficient of x of the polynomial $f(x)$. Consequently, the polynomial $f'(x)$ should have the same coefficient. The secret is revealed by multiplying the coefficient of x by a multiplicative inverse of an element r. If the result gives term a_0', then there is no cheating. Therefore, the secret is valid. Otherwise, some forged shares are used during reconstruction of the secret. Algorithm 30 shows how cheating detection is done.

Algorithm 30 (cheating detection).

Input. Elements z and b

Output. No cheating or cheating

Process

(i) C computes $b \cdot z = a_0'$.

(ii) C outputs no cheating if a_0' holds or cheating otherwise.

C uses Algorithm 30 to detect if forged shares were presented to reconstruct the secret. C uses multiply a_1' by b to get a_0'. Once the output is not a_0', then cheating took place, secret reconstruction halted, and the reconstructed secret was not valid. C sends to users a signal that secret reconstruction has failed. In cases where all users are honest, Algorithm 30 outputs no cheating. Consequently, the output Algorithm 29 valid and C sends a_0' to each user, which shows that secret reconstruct is successful. A simple example below demonstrates how the new scheme works.

Example 31. Let $p = 23$. Given the secret $s = 12$, we can share it to $n = 6$ users such that any $t = 4$ of them can reconstruct the secret.

We use Algorithm 25 to hide the secret and select a random $r \in \mathbb{F}_{23}$ as 15

$$z \equiv s \cdot r \bmod p \equiv 12 \cdot 15 \bmod 23 \equiv 19. \tag{20}$$

We also compute a multiplicative inverse of r.

$$b \equiv r^{-1} \bmod p \equiv 15^{-1} \bmod 23 \equiv 20 \bmod 23. \tag{21}$$

Let the random polynomial be

$$f(x) = 12 + 19x + 20x^2 + 9x^3. \tag{22}$$

The shares given to users are

$$\begin{aligned}
U_1 &: f(1) = 12 + 19 \times 1 + 20 \times 1^2 + 9 \times 1^3 = 14 \\
U_2 &: f(2) = 12 + 19 \times 2 + 20 \times 2^2 + 9 \times 2^3 = 18 \\
U_3 &: f(3) = 12 + 19 \times 3 + 20 \times 3^2 + 9 \times 3^3 = 9 \\
U_4 &: f(4) = 12 + 19 \times 4 + 20 \times 4^2 + 9 \times 4^3 = 18 \\
U_5 &: f(5) = 12 + 19 \times 5 + 20 \times 5^2 + 9 \times 5^3 = 7 \\
U_6 &: f(6) = 12 + 19 \times 6 + 20 \times 6^2 + 9 \times 6^3 = 7.
\end{aligned} \tag{23}$$

Each user also receives 20 a multiplicative inverse of 15 $\in \mathbb{F}_{23}$.

When the secret is required, any 4 users send their shares to C to reconstruct the secret s. Let U_1, U_3, U_5, and U_6 send their shares to C. We use Lagrange interpolation to

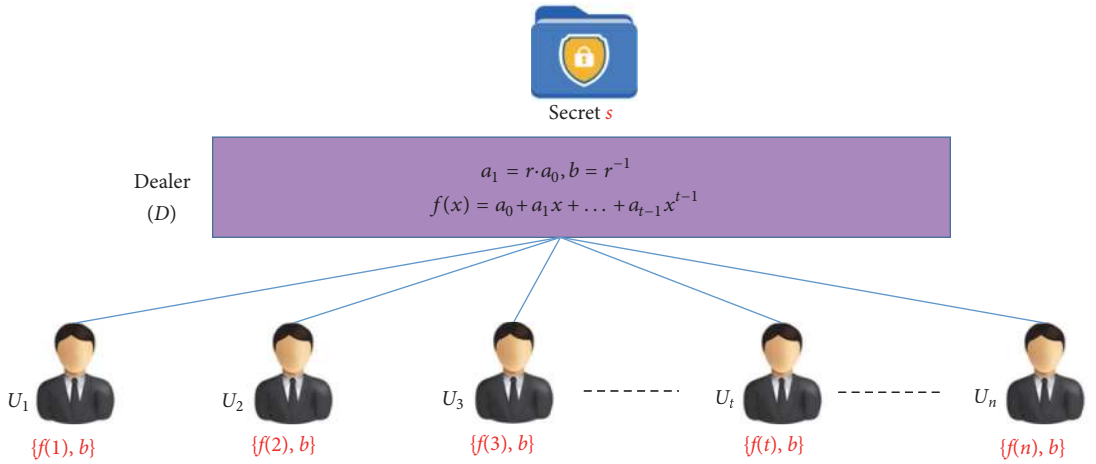

Figure 4: Share generation in PBLS.

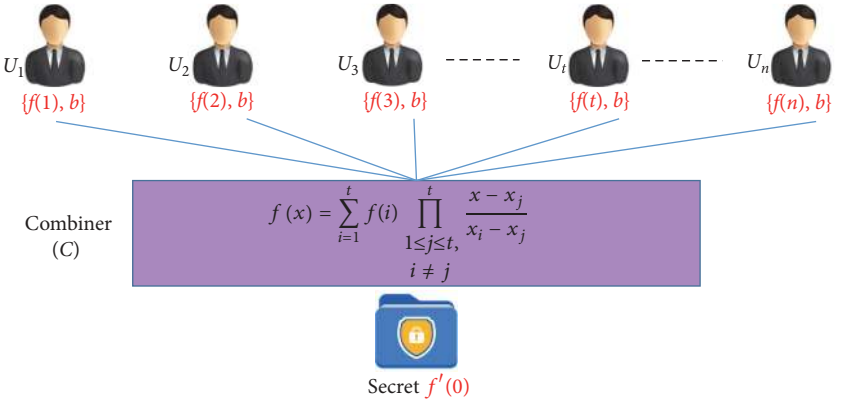

Figure 5: Secret reconstruction in PBLS.

reconstruct a polynomial $f'(x)$. We find the $g_i(x)$ $\forall i = \{0, 1, 2, 3\}$ as discussed in Section 2.1.3

$$
\begin{aligned}
g_0(x) &= \frac{(x-3)(x-5)(x-6)}{(1-3)(1-5)(1-6)} \\
&= 4\left(x^3 + 9x^2 + 17x + 2\right) \\
&= 4x^3 + 13x^2 + 22x + 8 \\
g_1(x) &= \frac{(x-1)(x-5)(x-6)}{(3-1)(3-5)(3-6)} \\
&= 2\left(x^3 + 11x^2 + 18x + 16\right) \\
&= 2x^3 + 22x^2 + 13x + 9 \\
g_2(x) &= \frac{(x-1)(x-3)(x-6)}{(5-1)(5-3)(5-6)} \\
&= 20\left(x^3 + 13x^2 + 4x + 5\right) \\
&= 20x^3 + 7x^2 + 11x + 8 \\
g_3(x) &= \frac{(x-1)(x-3)(x-5)}{(6-1)(6-3)(6-5)} = 20\left(x^3 + 14x^2 + 8\right) \\
&= 20x^3 + 4x^2 + 22.
\end{aligned} \tag{24}
$$

Therefore, the polynomial $f'(x)$ is

$$
\begin{aligned}
f'(x) &= f(1) g_0(x) + f(3) g_1(x) + f(5) g_2(x) \\
&\quad + f(6) g_3(x) \\
&= 14\left(4x^3 + 13x^2 + 22x + 8\right) \\
&\quad + 9\left(2x^3 + 22x^2 + 13x + 9\right) \\
&\quad + 7\left(20x^3 + 7x^2 + 11x + 8\right) \\
&\quad + 7\left(20x^3 + 4x^2 + 22\right) \\
&= 9x^3 + 20x^2 + 19x + 12
\end{aligned} \tag{25}
$$

The polynomial $f'(x) = 9x^3 + 20x^2 + 19x + 12$ is the same used to share the secret. Cheating detection is done by

multiplying 19 by 20, which gives 12. Assuming users U_1, U_3, and U_5 want to cheat U_6. Let the forged shares be U_1: (1, 8), U_3: (3, 10), U_5: (5, 17) and U_6: (6, 7). C reconstructs a polynomial

$$f'(x) = 3x^3 + 11x^2 + 10x + 7. \quad (26)$$

Multiplying 11 by 20 gives 12, which is not equal to 7. Cheating is detected by cheating detection algorithm in PBLS.

4. Analysis

This section provides the analysis of basic scheme and PBLS in terms of security and privacy with required features and computational overhead. We also compare the security, privacy, and computations in PBLS to Shamir's scheme and Liu et al.'s scheme [6, 14]. PBLS achieves the required features for the secret sharing schemes like SP1, SP2, SP3, SP4, and CO. Proofs for these requirements are provided. For this, we first provide the proof for the security of basic scheme, which is used by PBLS to detect cheating. The section also provides the proof of how secure PBLS is against cheating based on the assumption of OKS because the aim of secret sharing is to make the secret not known to users until reconstruction.

4.1. Security and Privacy Analysis. This subsection provides the analysis on the security and privacy of PBLS and proves that the required features for secret sharing schemes are achieved. We show that PBLS achieves the following properties.

(i) SP1: the secret is not known to all users and adversary A before reconstruction.

(ii) SP2: the secret can be reconstructed once it is shared to n users.

(iii) SP3: no less than required number of users can reconstruct the secret.

(iv) SP4: this is based on OKS assumption, which provides the guarantee that no cheating can be successful in PBLS.

First, we show that basic scheme is secure from A based on OKS adversary model in Section 3.1.3. At initialization of basic scheme, the secret is multiplied by a random element r with an aim of hiding it. Two security issues rise up in this case:

(i) security of s in z for the basic scheme and

(ii) security of s in the polynomial $f(x)$.

Proposition 32 proves the security of basic scheme from any adversary, i.e., dishonest user or anyone who is not taking part in the secret sharing cannot obtain the secret s without the knowledge of r and its multiplicative inverse.

Proposition 32. *Let p be prime and $z \equiv s \cdot r \bmod p$ such that $s \in \mathbb{Z}_p$ and $r \in \mathbb{Z}_p$ are a secret and a random number, respectively. An adversary* A *should solve FFDLP to obtain the secret from z in basic scheme without the knowledge of r and its multiplicative inverse.*

Proof. We showed in Proposition 19 that an element z cannot reveal any information about s without the knowledge of r. It was indicated that it is necessary to solve FFDLP to reveal s from z. Therefore, to know the secret from z in basic scheme, one has to solve FFDLP. Basic scheme is secure from A. □

However, we also need to show that the secret cannot be revealed by A in z, which is the polynomial used to distribute shares to users. Proposition 33 proves that basic scheme is secure in share generation algorithm.

Proposition 33. *Let z, s, and $r \in \mathbb{Z}_p$ as defined in Proposition 32 and p be prime. An adversary* A *should solve FFDLP to obtain the secret from z in the polynomial $f(x)$ even if multiplicative inverse of r is known.*

Proof. By Proposition 19, it is difficult to obtain s from z without the knowledge of r. However, if r is known, the secret s can be obtained. During share generation all users have access to the multiplicative inverse of r and their share; hence, it is possible to obtain the secret if z is known. However, the secret and the element z are coefficients of the polynomial, which are only known to the dealer. But the multiplicative inverse of r cannot give information of s and z. This is the same as solving the FFDLP. □

Security of secret sharing schemes depends on the private distribution of shares to user so that no user should know the shares of the other users. Therefore, each user has to receive the share from the dealer using a private channel. It is assumed that users do not communicate about their shares to each other unless they collaborate to cheat. Once the secret is divided, the shares do not show any information about the secret. As a result users do not have any information about the secret as assumed by OKS. In addition to this, shares are delivered privately to users and hence cannot know the share of the other users. Lemma 34 proves the fact of SP1 that users do not have access to the secret.

Lemma 34. *Any secret share given to a participant in PBLS does not reveal the secret s.*

Proof. In share generation, PBLS uses Shamir's method to share the secret, which uses the polynomial $f(x)$ of degree t - 1. Each share is evaluated from i to give $f(i)$ for all $i >$ 0 such that i is the identity of the user. The secret in the polynomial is $f(0)$. Shares in Shamir's scheme do not reveal any information of the secret. Since PBLS adopts Shamir's method, the shares generated have the same security as those generated by Shamir's scheme. □

Lemma 34 proves that no single user can have access to the secret using only his (or her) share. However, the secret can be reconstructed if t users pool in their shares together. Since the reconstruction is done by a trusted third party called combiner, no user is able to know the secret. However, the secret is obtained by the combiner. We

now prove Proposition 35, which gives a proof on SP2 in PBLS.

Proposition 35. *Let t be the threshold. Any t users can reconstruct the secret by combining their shares together in PBLS by using secret reconstruction algorithm.*

Proof. Secret reconstruction in PBLS is done by interpolating the polynomial $f'(x)$ by Lagrange interpolation. Given t points (i, v_i) for all i =1, 2, …, t, interpolated polynomial as in Equation (19). We rewrite the Equation below for clarity

$$f'(x) = \sum_{i=1}^{t} v_i \prod_{j=1, j\neq i}^{t} \frac{x - x_j}{x_i - x_j} = \sum_{i=0}^{t-1} a_i' x^i. \tag{27}$$

By Theorem 11 the polynomial $f'(x)$ is unique which has degree t - 1. The dealer used Equation (18) to generate shares and we write it

$$f(x) = \sum_{i=0}^{t-1} a_i x^i. \tag{28}$$

Therefore, the two polynomials in Equations (27) and (28) are equal. By Definition 7, $a_i = a_i'$. So, $a_0 = a_0'$. Since a_0 is the secret, a_0' is also a secret, which has been reconstructed from t shares. Therefore, the secret in PBLS can be reconstructed. □

If t shares can reveal the secret, then less than t shares should not be able to show any information about the secret. This is the recommendation of a secret sharing scheme to be achieved, which is called privacy. Proposition 36 proves SP3 of PBLS such that less than the required threshold can neither recover the secret not gain information about the secret.

Proposition 36. *Let t be the threshold. Less than t users cannot reconstruct or know any information of the secret in PBLS.*

Proof. We need to show that t - 1 shares do not reveal any information about the secret. Assume t - 1 participants collude to recover the secret, which means they will have t - 1 points to interpolate a polynomial $f'(x)$ of degree t - 1. However, these points will interpolate a polynomial of degree t - 2 as far as Theorem 11 is concerned, hence

$$f'(x) = \sum_{i=1}^{t-1} v_i \prod_{j=1, j\neq i}^{t-1} \frac{x - x_j}{x_i - x_j} = \sum_{i=0}^{t-2} a_i' x^i. \tag{29}$$

Equations (27) and (28) are not equal since they have different degrees. The fact that polynomial in Equation (29) is unique makes their coefficient differ as well. The other way is to try to solve a system of t - 1 equations with t unknowns as shown in the matrix Equation (30).

$$\begin{pmatrix} 1 & ID_1 & \dots & ID_1^{t-1} \\ 1 & ID_2 & \dots & ID_2^{t-1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & ID_{t-1} & \dots & ID_{t-1}^{t-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} f(ID_1) \\ f(ID_2) \\ \vdots \\ f(ID_{t-1}) \end{pmatrix} \tag{30}$$

But it is impossible to solve such equations unless the t-th term is guessed. Thus, we need at least t points to interpolate the polynomial, which might be not correct since the t-th share is secretly delivered to user. By Lemma 34, each share does not reveal the information about the secret. Therefore, t - 1 shares cannot reveal any information about the secret in PBLS. □

During secret reconstruction, t - 1 users may collude to cheat the t-th user. Since the aim of the cheaters is to prevent the correct recovery of the secret at the same time they should be able to reconstruct the valid secret. This is avoided in PBLS because all shares from participating users are not revealed during secret reconstruction. Therefore, if t - 1 users communicated their shares, it will be difficult to reconstruct the secret without the t-th share as proved in Proposition 35. Furthermore, PBLS will be able to detect cheating if t - 1 or less shares are forged. Proposition 37 proves that no cheating can be successful in PBLS even if t - 1 forged shares were presented during secret reconstruction.

Proposition 37. *Let t be the threshold and t - 1 be forged shares presented to the combiner. The forged shares will be detected during secret reconstruction in PBLS.*

Proof. Once fake shares are pooled to the combiner, the combiner uses Lagrange interpolation to compute a polynomial $f'(x)$ of degree t - 1, which is unique; hence $f'(x) \neq f(x)$. After coming up with the polynomial, the combiner will use the multiplicative inverse of the random element $r^{-1} \in \mathbb{F}_p$ to verify the secret as follows:

$$z \cdot r \equiv s \cdot r \cdot r^{-1} \bmod p \equiv s \bmod p, \tag{31}$$

where s is the coefficient of x^0. This works by Proposition 3. Since forged shares are used in secret reconstruction, the interpolated polynomial $f'(x)$ will not be equal to the polynomial used by the dealer in share generation algorithm; therefore, $a_i' \neq a_i$. Equation (31) will not work; thus $z \cdot r^{-1} \neq s$. Therefore, cheating is detected in PBLS. □

PBLS achieves the share size of $|v_i| = |s| / \varepsilon$ such that $\varepsilon > 0$ is the probability for successful cheating. Theorem 38 provides the security of PBLS and a proof on the share size of PBLS [30–39].

Theorem 38. *PBLS described in Section 3.2.2 realizing the access structure Γ is ε-secure against up to t - 1 cheaters who may somehow know the secret beforehand. Moreover, the share size is $|v_i| = p^2 = |s|/\varepsilon$, where $\varepsilon = 1/p$ and v_i denotes the share space of the i-th participant U_i.*

Proof. Since p is prime, $\mathbb{Z}_p$ is a finite field. We let the secret to be p bits long so

$$|s| = p. \tag{32}$$

For successful cheating, an adversary A has the sample space of p. Hence the probability is

$$\varepsilon = \frac{1}{p}. \tag{33}$$

Since every user receives $f(i)$ and y, the share is

$$v_i = \{f(i), y\} \quad \text{where } y = r^{-1}. \tag{34}$$

So $|f(i)| = p$ and $|y| = p$. Therefore,

$$|v_i| = p^2. \tag{35}$$

From Equations (32) and (33),

$$\frac{|s|}{p} = \varepsilon p. \tag{36}$$

This implies that

$$\frac{|s|}{\varepsilon} = p^2. \tag{37}$$

From Equation (35)

$$|v_i| = p^2 = \frac{|s|}{\varepsilon}. \tag{38}$$

Suppose an honest participant U_h having the share $v_h = (h, f(h))$ belongs to the k-th compartment. For a valid but incorrect secret $s' \in S$ to be accepted by U_h, after parsing another check polynomial $g'(x)$ with $g'(0) = s'$, the point $(x_h, g(x_h))$ should lie on the polynomial $g'(x)$. So, the successful cheating probability (PBLS, A) of cheaters A against PBLS is defined as

$$\varepsilon(\text{PBLS}, \mathsf{A}) = \Pr\left[g'(x) \text{ passes through a point } (x_h, g(x_h)) \text{ unknown to } \mathsf{A}\right]. \tag{39}$$

Since is a polynomial $g(x)$ of degree t - 1 over $\mathbb{F}_p$ from different constant term, $\varepsilon = 1/p$. This proves the theorem. □

Propositions 35 and 36 indicate that PBLS is perfect since the secret can be recovered by the required threshold but not less. This is the same for Shamir's scheme and Liu et al.'s scheme. Theorem 38 indicates that any cheating behavior can be detected, which is similar to Liu et al.'s scheme. Furthermore, PBLS and Liu et al.'s scheme have the same share size given to users. Therefore, PBLS and Liu et al.'s scheme have the same property of cheating detection.

We now compare the security and properties of PBLS with Shamir's scheme and Liu et al.'s scheme. Table 1 shows that comparison of SPs and CO of these schemes.

Table 1 shows that the schemes have similar properties in terms of SP1, SP2, and SP3. However, Shamir's scheme does not provide SP4. It also shows that PBLS achieves CO simultaneously. Even if PBLS provides good aspects in security and privacy, we need to mention that there is possibility that shareholders have some advantages over learning the secret since they have b as mentioned in Algorithm 28. So, our future research should focus on devising a new scheme to solve this probability.

4.2. Computation Analysis. In this subsection, we analyze the computation overhead of PBLS and give a comparison to Shamir's scheme and Liu et al.'s scheme. Three operations are used in this analysis, which are modulo addition (add), modulo multiplication (mul), and modulo inverse (inv). The analysis will consider all the two algorithms. However Algorithm 25 of basic scheme is not considered in this analysis since it is an initialization phase of PBLS.

4.2.1. Computations in PBLS. In PBLS, a share is computed from a polynomial $f(x)$ of degree t - 1. Therefore, the polynomial has t terms and t - 1 of them are multiplied by a variable x. Each share requires modulo addition and modulo multiplication to be computed. Therefore, Table 2 shows computation in share generation of PBLS.

The aim of secret reconstruction is to come up with a secret, which is done using Lagrange interpolation. A polynomial is interpolated using t points, which are the shares and the identity of the user, i.e., (i, v_i). The polynomial $f'(x)$ is obtained as in Equation (27), which is rewritten as

$$f'(x) = \sum_{i=1}^{t} v_i \cdot p_i(x), \quad \text{where } p_i(x) = \prod_{j=1, j\neq i}^{t} \frac{x - x_j}{x_i - x_j}. \tag{40}$$

To compute each $p_i(x)$, Table 3 shows the computations in secret reconstruction of PBLS.

4.2.2. Comparison. Now, we compare the computation overhead of PBLS with Shamir's scheme and Liu et al.'s scheme. We follow the method done in Section 4.2.1 to provide computation overhead comparison. Table 4 shows the computation

Table 1: Comparison of required properties for the schemes.

Scheme	Property				
	SP1	SP2	SP3	SP4	CO
Shamir's scheme	√	√	√	X	1
Liu et al.'s scheme	√	√	√	√	2
PBLS	√	√	√	√	1

√: the property exists; X: the property does not exist.

Table 2: Computations in share generation of PBLS.

Share generation				
Operation	mul	add	inv	Total
Number of operations	$n(t\text{ -}1)$	nt	-	$n(t\text{ -}1)$ mul + nt add

Table 3: Computations in secret reconstruction of PBLS.

Secret reconstruction				
Operation	mul	add	inv	Total
Number of operations	$t^3 + t + 1$	t	t	$(t^3 + t + 1)$ mul + t add + t inv

Table 4: Computation overhead of related schemes.

Share generation				
Scheme	Operation			
	mul	add	inv	Total
Liu et al.'s scheme	$2n(t - 1)$	$2nt$	-	$2n(t - 1)$ mul + $2nt$ add
Shamir's scheme	$n(t - 1)$	nt	-	$n(t - 1)$ mul + nt add
PBLS	$n(t - 1)$	nt	-	$n(t - 1)$ mul + nt add
Secret reconstruction				
Scheme	Operation			
	mul	add	inv	Total
Liu et al.'s scheme	$2(t^3 + t + 1)$	$2t$	$2t$	$2(t^3 + t + 1)$ mul + $2t$ add + $2t$ inv
Shamir's scheme	$t^3 + t + 1$	t	t	$(t^3 + t)$ mul + t add + t inv
PBLS	$t^3 + t + 1$	t	t	$(t^3 + t + 1)$ mul + t add + t inv

overhead comparisons in share generation phase and secret reconstruction phase.

Results in Table 4 show that PBLS and Shamir's scheme have the same computation overhead at share generation phase. The result is due to the use of a single polynomial when sharing and distributing the secret to users. However, comparing this result to Liu et al.'s scheme, the computation overhead is reduced by half in share generation phase. This indicates that PBLS has an efficient way to share the secret as compared to Liu et al.'s scheme but comparable with Shamir's scheme.

Considering secret reconstruction process, computation overhead on PBLS is higher by 1 mul as compared to Shamir's scheme. This is so because of cheating detection in PBLS in which the operation is 1 mul but Shamir's scheme does not have. However, results show that computation overhead of Liu et al.'s scheme at secret reconstruction still doubles as compared to PBLS. Therefore, PBLS is more efficient as compared to Liu et al.'s scheme in the concern of computation overhead.

5. Conclusion

In this paper, we proposed a new linear (t, n) threshold secret sharing scheme called PBLS, which is based not only on Shamir's scheme but also on ElGamal cryptosystem. PBLS satisfies the required properties like security, recoverability, privacy, cheating detection, and share size. PBLS is (t, n) threshold scheme, which requires at least t shares to reconstruct the secret while any less than t should not be able to do it.

Firstly, we draw the required features that secret sharing schemes satisfied by reviewing and analyzing some previous schemes like Shamir's and Liu et al.'s. The required features

drawn are security, recoverability of the secret, privacy of the secret, cheating detection of the forged shares presented for reconstruction of a secret and share size given to each user. We also reviewed some basic mathematical and cryptographic concepts, which assisted in designing methods for cheating detection such as finite fields and ElGamal cryptosystem.

Based on the withdrawn required features of secret sharing schemes, basic scheme and PBLS were designed. Basic scheme aims at hiding the secret, which is the initialization of PBLS. The secret is revealed during cheating detection. This is an idea of ElGamal who developed a cryptosystem that can hide a message using field elements. PBLS apples Shamir's secret sharing scheme to share the secret. Polynomial $f(x)$ is used in share generation phase such that the coefficient of x is the element hiding the secret. Secret reconstruction was done by interpolating a polynomial using Lagrange interpolation. Cheating detection was achieved by multiplying the coefficient of x of the polynomial $f(x)$ by multiplicative inverse of r to reveal the secret $f'(0)$.

After the design of PBLS, an analysis was made, which was presented in two ways. These were security analysis and privacy analysis with required features and computational overhead analysis. It was determined that the security with privacy of PBLS was similar to Liu et al.'s scheme. However, in terms of cheating, Shamir's scheme proved to be weak. Cheating detection was attained in both PBLS and Liu et al.'s schemes even though PBLS used only one polynomial. Furthermore, the required features like recoverability were analyzed to be similar to Liu et al.'s scheme. Computational analysis showed that number of operations in PBLS is almost equal to the computations in Shamir's scheme, which is half of Liu et al.'s scheme. This analysis made PBLS to be a better scheme in terms of efficiency than Liu et al.'s scheme and in terms of security than Shamir's scheme.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The results in this paper are part of Kenan Kingsley Phiri's Master degree thesis. Corresponding author is Hyunsung Kim. This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B04032598).

References

[1] R. Oppliger, *Contemporary Cryptography*, Artech House, Boston, MA, USA, 2005.

[2] S. Vaudenay, *A Classical Introduction to Cryptography: Applications for Communications Security*, Springer, New York, NY, USA, 2006.

[3] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Florida, Fla, USA, 1996.

[4] A. Beimel, "Secret-sharing schemes: a survey," *Lecture Notes in Computer Science*, vol. 6639, pp. 11–46, 2011.

[5] K. M. Martin, "Challenging the adversary model in secret sharing schemes: Coding and cryptography II," in *Proceedings of the Royal Flemish Academy of Belgium for Science and Art*, pp. 45–63, 2008.

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

[7] M. Tompa and H. Woll, "How to share a secret with cheaters," *Journal of Cryptology*, vol. 1, no. 3, pp. 133–138, 1989.

[8] B. Srikanth, G. Padmaja, S. Khasim, P. V. S. Likshmi, and A. Haritha, "Secure bank authentication using image processing and visual cryptography," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2432–2437, 2014.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98, New York, NY, USA, November 2006.

[10] T. Tassa, "Generalized oblivious transfer by secret sharing," *Designs, Codes and Cryptography*, vol. 58, no. 1, pp. 11–21, 2011.

[11] M. O. Rabin, "Randomized byzantine generals," in *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (SFCS '83)*, pp. 403–409, Tucson, AZ, USA, November 1983.

[12] P. Lin, Y. Chen, M. Hsu, and F. Juang, "Secret sharing mechanism with cheater detection," in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA '13)*, pp. 1–4, Kaohsiung, Taiwan, October 2013.

[13] Y. Liu, Y. Zhang, and Y. Hu, "Efficient (t, n) secret sharing scheme against cheating," *Journal of Computational Information Systems*, vol. 8, no. 9, pp. 3815–3821, 2012.

[14] Y. Liu, Z. Wang, and W. Yan, "Linear (k, n) secret sharing scheme with cheating detection," in *Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM '15)*, pp. 1942–1947, Liverpool, UK, October 2015.

[15] I.-C. Lin and C.-C. Chang, "A (t, n) threshold secret sharing system with efficient identification of cheaters," *Computing and Informatics*, vol. 24, no. 5, pp. 529–541, 2005.

[16] R. Kalombe and M. Kamble, "Cheater detection and identification based on shamir scheme," *International Research Journal of Computer Science Engineering and Application*, vol. 2, no. 2, pp. 255–259, 2013.

[17] S. Obana, "Almost optimum t-cheater identiable secret sharing scheme," *Lecture Notes in Computer Science*, vol. 6632, pp. 284–302, 2011.

[18] D. Pasaila, V. Alexa, and S. Iftene, "Cheating detection and cheater identification in CRT-based secret sharing schemes," *International Journal of Computing*, vol. 9, no. 2, pp. 107–117, 2010.

[19] C. Guo, R. Zhuang, L. Yuan, and B. Feng, "A group authentication scheme supporting cheating detection and identification," in *Proceedings of the 9th International Conference on Frontier of Computer Science and Technology (FCST '15)*, vol. 52, pp. 110–114, Dalian, China, August 2015.

[20] L. Harn, "Generalised cheater detection and identification," *IET Information Security*, vol. 8, no. 3, pp. 171–178, 2014.

[21] M. P. Jhanwar and R. Safavi-Naini, "Unconditionally-secure robust secret sharing with minimum share size," *Lecture Notes in Computer Science*, vol. 7859, pp. 96–110, 2013.

[22] A. Bishop and V. Pastro, "Robust secret sharing schemes against local adversaries," *Lecture Notes in Computer Science*, vol. 9615, pp. 327–356, 2016.

[23] M. P. Jhanwar and R. Safavi-Naini, "On the share efficiency of robust secret sharing and secret sharing with cheating detection," *Lecture Notes in Computer Science*, vol. 8250, pp. 179–196, 2013.

[24] J. S.-T. Juan, Y.-L. Chuang, and M.-J. Li, "An online verifiable and detectable (t, n) multi-secret sharing scheme based on a hyperelliptic function," *Journal of Information and Computational Science*, vol. 8, no. 4, pp. 688–696, 2011.

[25] M. Backes, A. Kate, and A. Patra, "Computational verifiable secret sharing revisited," *Lecture Notes in Computer Science*, vol. 7073, pp. 590–609, 2011.

[26] M. Carpentieri, A. De Santis, and U. Vaccaro, "Size of shares and probability of cheating in threshold schemes," *Lecture Notes in Computer Science*, vol. 765, pp. 118–125, 1994.

[27] W. Ogata and K. Kurosawa, "Optimum secret sharing scheme secure against cheating," *Lecture Notes in Computer Science*, vol. 1070, pp. 200–211, 1996.

[28] M. Liu, L. Xiao, and Z. Zhang, "Linear multi-secret sharing schemes based on multi-party computation," *Finite Fields and Their Applications*, vol. 12, no. 1, pp. 704–713, 2006.

[29] J. Pieprzyk and X. Zhang, "Cheating prevention in linear secret sharing," *Information Security and Privacy*, vol. 2384, no. 1, pp. 121–135, 2002.

[30] R. Cramer, I. B. Damgård, N. Döttling, S. Fehr, and G. Spini, "Linear Secret sharing schemes from error correcting codes and universal hash functions," *Lecture Notes in Computer Science*, vol. 9057, pp. 313–336, 2015.

[31] H. Ghodosi, "Comments on Harn–Lin's cheating detection scheme," *Designs, Codes and Cryptography*, vol. 60, no. 1, pp. 63–66, 2011.

[32] J. Hoffstein, J. Pipher, and J. H. Silverman, *An Introduction to Mathematical Cryptography*, Springer, New York, NY, USA, 2008.

[33] A. Slinko, *Algebra for Applications: Cryptography, Secret Sharing, Error Correcting, Fingerprinting, Compression*, Springer, Heidelberg, Germany, 2015.

[34] M. W. Baldoni, C. Ciliberto, and G. M. P. Cattaneo, *Elementary Number Theory, Cryptography and Codes*, Springer-Verlag, Berlin Heidelberg, Germany, 2009.

[35] T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *Lecture Notes in Computer Science*, vol. 196, pp. 10–18, 1985.

[36] T. Koshy, *Elementary Number Theory with Applications*, Academic press, New York, NY, USA, 2nd edition, 2007.

[37] T. C. Wu and T. S. Wu, "Cheating detection and cheater identification in secret sharing schemes," *IEE Transactions on Computers and Digital Techniques*, vol. 142, no. 1, pp. 367–369, 1995.

[38] A. Adhikari, K. Morozov, S. Obana, P. S. Roy, K. Sakurai, and R. Xu, "Efficient threshold secret sharing schemes secure against rushing cheaters," *Lecture Notes in Computer Science*, vol. 10015, pp. 3–23, 2016.

[39] J. Pramanik, P. S. Roy, S. Dutta, A. Adhikari, and K. Sakurai, "Secret sharing schemes on compartmental access structure in presence of cheaters," *Lecture Notes in Computer Science*, vol. 11281, pp. 171–188, 2018.