Journal of Discrete Mathematics and Its Applications 11 (3) (2026) 177–191

Journal of Discrete Mathematics and Its Applications

Available Online at: http://jdma.sru.ac.ir

Shahid Rajaee Teacher Training University

Research Paper

Efficient protocols for group key establishment based on secret sharing

Mahnaz Noroozi*, Maryam Ahmadi

Department of Computer Science, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran

Academic Editor: Alireza Abdollahi

Abstract. Secure group communication is a fundamental requirement in modern distributed systems, relying on efficient and secure group key establishment protocols. While secret sharing has been effectively utilized to construct such protocols, existing approaches often incur significant computational overhead. This paper addresses this efficiency challenge by presenting novel improvements to the group key agreement protocol of Harn and Lin and the group key distribution protocol of Harn and Hsu. Our key insight is to replace the use of Shamir's (n,n) threshold secret sharing scheme in the existing protocols with Shamir's $(2,2)$ threshold scheme, which consequently reduces the core polynomial operations from degree n to degree one. This optimization drastically lowers the computational cost associated with polynomial construction and interpolation, while maintaining the security guarantees of the original protocols. Security analysis demonstrates that the proposed protocols provably satisfy essential properties such as key confidentiality, authenticity, and forward secrecy under the hardness of the decisional Diffie-Hellman problem. Performance comparisons confirm that our constructions achieve superior efficiency in both computation and communication, making them highly practical for large-scale group applications.

Keywords. cryptography, group key establishment, key agreement, key distribution, secret sharing.
Mathematics Subject Classification (2020): 94*A*60, 94*A*62.

*Corresponding author (Email address: m.noroozi@alzahra.ac.ir)
Received 23 November 2025; Revised 08 December 2025; Accepted 22 December 2025
First Publish Date: 01 September 2026

1 Introduction

Secure communication among multiple parties has become a cornerstone of modern distributed systems, ranging from collaborative applications and sensor networks to large-scale cloud infrastructures. In such environments, the establishment of a common cryptographic key is essential to ensure confidentiality, integrity, and authenticity of group communication. Key establishment protocols are methods that generate a shared secret key among users, enabling them to employ it in subsequent applications within various cryptographic schemes.

Two principal paradigms have been studied in the literature of key establishment: 1)group key agreement, where all participants jointly contribute to the generation of the shared key, and 2) group key distribution, where a trusted entity derives and disseminates the key to the group. Agreement protocols emphasize fairness and contributory properties, ensuring that no single party dictates the key, while distribution protocols often achieve efficiency and centralized control at the cost of stronger trust assumptions.

The first two-party key agreement protocol was introduced in 1976 by Diffie and Hellman, based on the discrete logarithm problem [1]. In 2000, Joux proposed the first three-party key agreement protocol based on bilinear pairing [2] enabling participants to establish a shared key in a single round of communication. Both of these protocols, however, are vulnerable to man-in-the-middle attacks. To address this issue, authentication mechanisms were incorporated into key agreement protocols, ensuring that each participant can be confident of establishing a shared key only with legitimate parties [3–5].

To apply key agreement protocols to a group of users, multiple executions are required [11]. Clearly, as the number of group members increases, the number of communication rounds also grows. Consequently, employing these protocols for group key agreement leads to higher communication overhead, increased cost, and longer execution time. In 1982, Ingemarsson et al. proposed the first group key agreement protocol with reduced communication complexity [6]. According to the literature, a few approaches are used to extend the Diffie-Hellman key agreement protocol to group settings. One approach is the use of binary trees [7–9]. This method significantly improves the communication efficiency of the resulting protocols. Another approach places all group members in a logical ring, where each participant communicates only with the two immediate neighbors. This technique has been widely studied for extending Diffie-Hellman into group key agreement protocols, yielding highly efficient schemes that require only a limited number of communication rounds (two or three) [10, 12].

Secret sharing is another tool employed in the design of group key agreement (and distribution) protocols. In a secret sharing scheme, a secret is shared among a set of participants such that only certain subsets, called authorized subsets, can reconstruct the secret using their shares. Unauthorized subsets must not gain any information about the shared secret. The collection of all authorized subsets is referred to as the access structure of the scheme. Shamir [13] and Blakley [14] independently introduced the first secret sharing schemes. In these schemes, a secret is shared among n participants such that any $t < n$ of them can reconstruct the secret. The access structure of these schemes is known as a

threshold access structure. Following Shamir and Blakley, numerous secret sharing schemes have been proposed, either by adding new features or by supporting more general access structures [15–17].

In 1989, Laih introduced the first group key distribution protocol based on threshold secret sharing [18]. In this protocol, each participant first registers in a conference and shares a secret with the chairperson. The chairperson then selects a random key and uses secret sharing to distribute shares of the chosen key among the members. Other studies have adopted similar approaches to enable confidential broadcast of messages to multiple group members [19–21].

In 2014, Harn and Lin proposed an authenticated group key agreement protocol using secret sharing [22]. Later, in 2017, Harn and Hsu employed a similar idea to design a group key distribution protocol [23]. These protocols rely on Shamir's (n,n) threshold secret sharing scheme (where n is the number of group members) to extend the Diffie-Hellman protocol into group key agreement and distribution settings.

In [26], the authors used a secret sharing schemes over elliptic curve groups to propose more efficient group key agreement protocols. In [27], the authors used secret sharing alongside elliptic curves and Schnorr's signature to propose new authenticated key agreement protocols. Beyond classical Shamir threshold schemes, more recent protocols combine bivariate polynomial secret sharing with modern signature schemes. For instance, Cai et al. designed a group authenticated key agreement using bivariate polynomial-based secret sharing alongside BLS multi-signatures [24]. Also, in distributed key generation (DKG) settings, recent optimizations reduce the cost and trust assumptions: e.g., an SM9-based DKG scheme using verifiable secret sharing has been proposed to eliminate reliance on a central dealer [25].

The objective of this paper is to improve the efficiency of the group key agreement and distribution protocols proposed by Harn and Lin [22], and Harn and Hsu [23]. To this end, we employ secret sharing techniques and specifically utilize Shamir's (2,2) threshold scheme to design enhanced group key agreement and distribution protocols that outperform previous constructions. The presented protocols are secure and remain uncompromised even if certain components of a share corresponding to the secret are exposed. With the proposed protocols, the number of group members no longer poses a limitation. Whereas in the protocols of [22] and [23] participants were required to construct and recover a polynomial of degree n to derive the common key, our protocols reduce the degree of this polynomial to one, thereby achieving a significant reduction in computational effort.

The rest of this paper is organized as follows. In Section 2 the preliminaries of the paper are covered. Then, Section 3 reviews the group key agreement protocol proposed by Harn and Lin [22] and the group key distribution protocol proposed by Harn and Hsu [23]. In Section 4, we propose our improved group key establishment protocols. Sections 5 and 6 are dedicated to security and performance analysis of our proposed protocols, respectively. Finally, Section 7 concludes the paper.

2 Preliminaries

In this section, preliminaries needed in the paper including the Lagrange interpolation procedure and the decisional Diffie-Hellman problem which the security of the proposed protocols relies on its hardness assumption are described.

2.1 Lagrange Interpolation

Lagrange interpolation is a fundamental method for reconstructing a polynomial given a sufficient number of distinct points. For a set of k distinct points $\{(x_1,y_1),(x_2,y_2),\cdots,(x_k,y_k)\}$, where $y_i = f(x_i)$ for some polynomial $f(x)$ of degree at most $k-1$, the polynomial can be uniquely reconstructed as:

$$f(x) = \sum_{i=1}^{k} y_i \cdot L_i(x),$$

where $L_i(x)$ is the Lagrange basis polynomial, defined as:

$$L_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{k} \frac{x - x_j}{x_i - x_j}.$$

The constant term $f(0)$, which is often used to represent a shared secret in cryptographic schemes like Shamir's secret sharing, can be computed directly by:

$$f(0) = \sum_{i=1}^{k} y_i \cdot \prod_{\substack{j=1 \\ j\neq i}}^{k} \frac{-x_j}{x_i - x_j}.$$

2.2 The Decisional Diffie-Hellman (DDH) Problem

The security of one of our proposed protocols is based on the hardness of the Decisional Diffie-Hellman (DDH) problem. Let $\mathbb{G}$ be a cyclic group of prime order q with generator g.

- **DDH Assumption:** Consider the following two distributions:
  - $\mathcal{R} = (g, g^a, g^b, g^c)$, where $a,b,c \in_R \mathbb{Z}_q$ are chosen independently and uniformly at random.
  - $\mathcal{D} = (g, g^a, g^b, g^{ab})$, where $a,b \in_R \mathbb{Z}_q$ are chosen independently and uniformly at random.

  The DDH assumption states that for any probabilistic polynomial-time algorithm $\mathcal{A}$, the advantage $\mathbf{Adv}_{\mathbb{G}}^{DDH}(\mathcal{A})$ in distinguishing between the distributions $\mathcal{R}$ and $\mathcal{D}$ is negligible. That is,

  $$|\Pr[\mathcal{A}(\mathcal{D}) = 1] - \Pr[\mathcal{A}(\mathcal{R}) = 1]| \leq \mathsf{negl}(\lambda),$$

  where $\mathsf{negl}(\lambda)$ is a negligible function in the security parameter λ.

3 Review of two Related Works

In this section, we briefly review the group key agreement protocol proposed by Harn and Lin [22] and the group key distribution protocol proposed by Harn and Hsu [23].

3.1 Harn and Lin's Group Key Agreement Protocol

Assume that n users, $U_1,\ldots,U_n$, wish to establish a shared key among themselves. Let p and q be large prime numbers such that $q \mid (p-1)$, and let g be a generator of $\mathbb{Z}_q$. The details of the protocol proposed by Harn and Lin [22], which consists of three phases, are as follows[a]:

Phase 1: In this phase, each user U_i first selects a random value $x_i \in_R \mathbb{Z}_q$, computes $r_i = g^{x_i} \mod p$, and publishes this value as their public value.

Phase 2: In this phase, each user U_i, for every other user U_j, uses the public value r_j of U_j and applies the two-party Diffie-Hellman key agreement protocol to compute the shared secret $s_{i,j} = r_j^{x_i} \mod p$. Then, user U_i computes $k_i = \prod_{j=1}^{n} s_{i,j} \mod p$. Lagrange interpolation is applied to the points $\{(0,k_i),(r_1,s_{i,1}),\ldots,(r_{i-1},s_{i,i-1}),(r_{i+1},s_{i,i+1}),\ldots,(r_n,s_{i,n})\}$ to calculate the polynomial of degree n, $f_i(x)$. The values $f_i(k)$ for $k = 1,\ldots,n-1$ are then published publicly.

Phase 3: In this phase, each user U_j, for $i = 1,\ldots,n(\neq j)$, applies Lagrange interpolation on the n points $\{(1,f_i(1)),\ldots,(n-1,f_i(n-1)),(r_j,s_{i,j})\}$ to compute the polynomial $f_i(.)$, and sets the value of k_i to the constant value extracted from the polynomial. Finally, the group shared key K is computed as:

$$K = \prod_{i=1}^{n} k_i \mod p.$$

3.2 Harn and Hsu's Group Key Distribution Protocol

Suppose that n users $U_1,\cdots,U_n$ wish to establish a shared group key. Let p and q be large prime numbers such that $q \mid (p-1)$, and let g be a generator of $\mathbb{Z}_q$. The public key of user U_i is $y_i = g^{x_i} mod p$, where $x_i \in \mathbb{Z}_q$ is the private key of user U_i. The details of the protocol proposed by Harn and Hsu, consisting of an encryption phase and a decryption phase, are described below.

Encryption phase:

Step 1: The initiator or trusted party, who can be one of the group members, selects a one-time random secret $s \in \mathbb{Z}_q$, and computes $r = g^s \mod p$.

Step 2: Using the public keys of all group members $\{y_i \mid i = 1,\cdots,n\}$, the initiator with private key x_s computes the one-time shared key for each user as:

$$k_i = \left(y_i^{x_s+s} \mod p\right) \mod q, \quad (i = 1,\cdots,n).$$

Step 3: The initiator constructs a polynomial $f(x)$ of degree n over the set of $n+1$ points

$$\{(0,K),(ID_i,k_i) \mid i = 1,\cdots,n\},$$

[a] For simplicity, the protocol without authentication is considered here.

where ID_i is the public identifier of user U_i and K is the secret group key which would be transmitted by the initiator. Using Lagrange interpolation, the polynomial is given by:

$$f(x) = K\prod_{j=1}^{n}\frac{x-ID_j}{-ID_j} + \sum_{i=1}^{n} k_i \frac{x}{ID_i}\prod_{\substack{j=1\\ j\neq i}}^{n}\frac{x-ID_j}{ID_i-ID_j} \mod q.$$

Step 4: The initiator computes n public values $\{f(i) \mid i = 1, \cdots, n\}$ such that $i \neq ID_i$, and evaluates $h(TS\|K)$, where TS is a timestamp and $h(\cdot)$ is a one-way hash function. The initiator then broadcasts the following to all group members:

$$\{r, TS, h(TS\|K), (i, f(i)) \mid i = 1, \cdots, n\}.$$

Decryption phase:

Step 1: Each member U_i uses the private key x_i to compute $k_i = ((y_s \cdot r)^{x_i} \mod p) \mod q$ where y_s is the public key of the initiator.

Step 2: Given the private value k_i and the n public values $\{f(i) \mid i = 1, \cdots, n\}$, each user U_i reconstructs the group secret K' using:

$$k_i\prod_{j=1}^{n}\frac{-j}{ID_i-j} + \sum_{j=1}^{n} f(j)\frac{-ID_i}{j-ID_i}\prod_{\substack{v=1\\ v\neq j}}^{n}\frac{-v}{j-v} \mod q = K'.$$

Step 3: Each user U_i verifies the timestamp TS and checks whether

$$h(TS\|K') =? h(TS\|K).$$

If the equality holds, the group key K is authenticated and can be used for secure group communication.

4 The Proposed Improved Protocols

The protocols reviewed in the previous section employ secret sharing methods to provide group key agreement and group key distribution. However, these protocols do not optimally utilize secret sharing techniques. In this section, by employing secret sharing methods more efficiently, we propose new protocols for group key agreement and distribution that achieve substantially higher efficiency.

4.1 The Proposed Authenticated Group Key Agreement Protocol

Assume that there are n users $U_1, \cdots, U_n$ who wish to create a shared key among themselves. Let p and q be two large prime numbers such that $q|(p-1)$, and g be a generator of a group of

order q in the subset $\mathbb{Z}_p$. Assume that each user in the system has a pair of long-term private and public keys (x_i, y_i) where $y_i = g^{x_i} \pmod{p}$. Given these assumptions, the details of our improved authenticated group key agreement protocol, which consists of four phases, are as follows:

- **Phase 1:** In this phase, Each user U_i
  - Chooses a random value $d_i \in_R \mathbb{Z}_q$, computes $r_i = g^{d_i} \pmod{p}$, and publishes it as his/her temporary public value.
- **Phase 2:** In this phase, each user U_i
  - For each user U_j, uses the public value y_j and applies the Diffie-Hellman two-party key agreement protocol to compute the shared secret $s_{i,j} = (y_j r_j)^{d_i + x_i} \pmod{p}$.
  - Computes the sum $k_i = \sum_{j=1(\neq i)}^{n} s_{i,j} \pmod{p}$.
  - Chooses a random value $a_i \in_R \mathbb{Z}_q$ and constructs a degree-one polynomial $f_i(x) = k_i + a_i x \pmod{q}$.
  - Chooses a random value $b_i \in \mathbb{Z}_q$ and computes $f_i(b_i)$.
  - Computes the values $f_i(s_{i,1}), \cdots, f_i(s_{i,n})$.
  - Publishes the set of values $\{b_i, f_i(b_i), f_i(s_{i,1}), \cdots, f_i(s_{i,n})\}$.
- **Phase 3:** In this phase, each user U_j
  - For $i = 1, \cdots, n(\neq j)$, uses the shared secret $s_{j,i} (= s_{i,j})$, which is computed as $s_{j,i} = (y_i r_i)^{d_j + x_j} \pmod{p}$, and the values $f_i(s_{j,i})$, b_i, and $f_i(b_i)$ to reconstruct the degree-one polynomial $f_i(.)$ using the two points $(s_{j,i}, f_i(s_{j,i}))$ and $(b_i, f_i(b_i))$, and then computes k_i.
  - Computes the shared key $K = \sum_{i=1}^{n} k_i \pmod{p}$.
- **Phase 4:** In this phase, each user U_i
  - Computes and publishes their key confirmation $c_i = h(K'_i, U_i, r_i)$, where $K'_i = \sum_{i=1}^{n} k_j \pmod{p}$.
  - After receiving the key confirmations from all users, computes $h(K'_i, U_j, r_j) = c'_j$ for $j = 1, 2, \cdots, n \neq i$, and checks if $c'_j =^? c_j$ holds. If the relation holds for all j, the group key $K = K'$ is confirmed. Otherwise, the protocol must be run again.

4.2 The Proposed Group Key Distribution Protocol

Suppose that there are n users $U_1, \cdots, U_n$ who wish to establish a shared key among themselves. Let p and q be two large prime numbers such that $q|(p-1)$, and g be a generator of a group of order q in the subset $\mathbb{Z}_p$. Assume that each user in the system has a pair of long-term private and public keys (x_i, y_i) where $y_i = g^{x_i} \pmod{p}$. Given these assumptions, the details of the improved secret sharing-based group key distribution protocol, which consists of two phases (i.e., encryption and decryption), are as follows:

4.2.1 Encryption phase:

- **Step 1:** The initiator or trusted party (who may be one of the group members) selects a random one-time value $s \in \mathbb{Z}_q$, and computes $r = g^s \pmod{p}$.
- **Step 2:** After receiving the public values of all group members, i.e. $\{y_i, i = 1, \cdots, n\}$, the initiator computes the shared one-time keys $k_i = \{y_i^{x_s+s} \pmod{p}\} \pmod{q}$, $i = 1, 2, \cdots, n$, where x_s is the private key of initiator.
- **Step 3:** The initiator chooses a random value $a \in_R \mathbb{Z}_q$ and constructs a degree-one polynomial $f(x) = K + ax \pmod{q}$, where K is the shared key selected by the initiator. Then, the initiator selects a random value $b \in \mathbb{Z}_q$ and computes $f(b)$ and the values $f(k_1), \cdots, f(k_n)$.
- **Step 4:** The initiator computes $h(TS||K)$, where TS is the timestamp and $h(.)$ is a one-way function. The initiator then publishes the set of values $\{r, TS, h(TS||K), f(k_i) : i = 1, \cdots, n, (b, f(b))\}$.

4.2.2 Decryption phase:

- **Step 1:** Each user U_i, as an authorized group member, uses his/her private key x_i and computes $k_i = \{(y_s \cdot r)^{x_i} \pmod{p}\} \pmod{q}$, where y_s is the public key of the initiator.
- **Step 2:** Knowing the private value k_i and the points $(k_i, f(k_i))$ and $(b, f(b))$, each user U_i interpolates the linear polynomial $f(.)$ and computes the group secret value K' by taking the constant value of the polynomial, i.e., $K' = f(0)$.
- **Step 3:** Each user U_i checks the timestamp TS and verifies the validity of the relation $h(TS||K') =^? h(TS||K)$. If the relation holds, the group key K' is authenticated and used for group communications.

5 Security Analysis of the Proposed Protocols

In this section, we examine the security of our proposed protocols and prove that they provide the necessary security properties. Our proposed improved protocols are similar to the protocols presented in [22] and [23]; However, by intelligently using secret sharing schemes, in our protocols the need for interpolating a degree n polynomial is eliminated and instead, the

protocols work with interpolating a degree one polynomial and therefore, the efficiency is enhanced. Accordingly, the improved protocols, like the protocols in [22] and [23], satisfy all the security requirements. In the following, at first the security of the proposed group key agreement protocol is regarded and then, we will discuss the security of the proposed key distribution protocol.

5.1 Security Analysis of the Proposed Authenticated Group Key Agreement Protocol

Theorem 5.1. *The improved authenticated group key agreement protocol provides key confidentiality.*

Proof. In the third phase of the proposed protocol, each user U_i sets the value k_i as the constant of a degree one polynomial and publishes a point from the polynomial $(b_i, f_i(b_i))$ along with the values $f_i(s_{i,1}), \cdots, f_i(s_{i,n})$. It is clear that this can be treated as an example of a Shamir's threshold secret sharing scheme with a threshold of 2 and $n+1$ shares. Recovering the secret k_i requires access to 2 points of the degree one polynomial f_i, one of which is public, and only the group members can use their shared values with U_i to derive the second point on the linear polynomial. Once they retrieve the constant, they can compute k_i. Therefore, the security of k_i in this protocol reduces to the security of the shared secrets among users. Since the protocol uses the Diffie-Hellman two-party key agreement protocol to establish the shared secret between each pair of users, and the security of Diffie-Hellman protocol is based on the hardness of the decisional Diffie-Hellman problem, the security of the proposed protocol is also reduced to the decisional Diffie-Hellman problem. It is also worth mentioning that the values published in the fourth phase are obtained by applying a one-way function $h(.)$ to the value of key. Hence, according to the one-wayness of this function, it does not affect the confidentiality of the key. □

Theorem 5.2. *The proposed authenticated group key agreement protocol provides forward secrecy.*

Proof. In the proposed protocol, the group key is a function of a sequence of one-time keys selected by all the group members. Therefore, even if the long-term private keys of users are compromised, the security of the previously computed group keys is not jeopardized. For example, in a 2-member group, the group key is computed as follows, which requires simultaneous access to both the short-term and long-term private values of one of the users:
$K = (y_1 r_1)^{d_2+x_2} + (y_2 r_2)^{d_1+x_1} \ (mod\ p) = 2g^{(x_1+d_1)(x_2+d_2)} \ (mod\ p)$
In other words, simply obtaining a user's long-term private key is not enough to compute the group key. Therefore, the proposed protocol ensures forward secrecy. □

Theorem 5.3. *The proposed authenticated group key agreement protocol provides key independence.*

Proof. Since each user U_i participates in the final key computation by selecting a random integer d_i, if at least one user in the group selects the key randomly, the final key will also be random. For example, consider a 3-member group where $r_1 = g^{x_1} \ (mod\ p), r_2 = g^{x_2} \ (mod\ p), r_3 = g^{x_3} \ (mod\ p)$ are the public values corresponding to each user. In this case, the group key will be $K = 2(g^{x_1 x_2} + g^{x_1 x_3} + g^{x_2 x_3}) \ (mod\ p)$, and if at least one of the values x_i is chosen randomly,

the group key K will also be random. This proves the randomness of the group key and the independence of different keys generated by distinct sets of users. □

Theorem 5.4. *The proposed authenticated group key agreement protocol provides key authenticity.*

Proof. The purpose of authentication is that at the end of the protocol, each user should be convinced that all group members have achieved the same group key. In this protocol, users only accept the group key once all key confirmations have been validated. Note that the key confirmation for each user is obtained using their own version of the group key as one of the inputs of a one-way function. Therefore, all users must share the same group key so that all key confirmations could be verified successfully. Due to the lack of access to the group key (proven by the key confidentiality property provided by the protocol), adversaries cannot create a valid key confirmation. □

Theorem 5.5. *The proposed authenticated group key agreement protocol is secure against unknown key share attacks.*

Proof. The shared value $s_{i,j} = (y_j r_j)^{d_i + x_i} \ (mod\ p)$ between any two users U_i and U_j involves the long-term private and public keys of both users. Therefore, only authorized members can compute these values and subsequently compute the group key and also the valid key confirmation. As a result, adversaries are unable to generate valid key confirmations and thus cannot perform an unknown key share attack. □

5.2 Security Analysis of the Proposed Key Distribution Protocol

Theorem 5.6. *The proposed key distribution protocol provides key confidentiality.*

Proof. In step 1 of the decryption phase of the proposed protocol, each member U_i uses the long-term private key x_i to compute the shared key $k_i = (y_s.r)^{x_i} \ (mod\ p)(mod\ q)$. As a result, only group members are able to use the computed shared values along with the published public point $(k_i, f(k_i))$ on the linear polynomial to retrieve $f(x)$ and its constant value, which is $K = f(0)$. □

Theorem 5.7. *The proposed key distribution protocol provides key authenticity.*

Proof. The degree one polynomial $f(x)$ used for key distribution is a function that group members can compute using one of the shared values $k_i = (y_i)^{x_s + s} \ (mod\ p) \ (mod\ q) = (y_s.r)^{x_i} \ (mod\ p) \ (mod\ q)$ for $i = 1, \cdots, n$. Other values used for key computation are public values $(TS, h(TS||K), f(k_i) : \ i = 1, \cdots, n)$ which are published by the initiator. Given the assumption that the function $h(.)$ is one-way, it is impossible to find a value K' such that $h(TS||K) = h(TS||K')$. Therefore, if any group member verifies the above relationship for the recovered key value K', it can be ensured that $K' = K$. This proves that the proposed protocol provides key authenticity. □

6 Performance Analysis of the Proposed Protocols

In this section, we evaluate the performance of the proposed protocols and compare them with similar existing protocols.

6.1 Performance Analysis of the Proposed Authenticated Group Key Agreement Protocol

In this section, we compare the proposed protocol with that of Harn and Lin [22] based on communication rounds, computation costs and message length.

Communication Rounds: One of the important criteria for comparing key agreement protocols is the number of communication rounds required, which refers to how many times the users need to send messages to each other in order to complete the protocol. In the proposed authenticated group key agreement protocol, the parties need 3 communication rounds to agree on the key: 1) In the first phase of the protocol, the temporary public values of the users are published. 2) At the end of the second phase, the set of values required for other users to compute the portion of the key selected by each user is published. 3) In the fourth phase, the values required for key authentication are published by the users. Given this, and since all existing authenticated group key agreement protocols require at least 3 communication rounds, we can conclude that the improved protocol, like the one proposed by Harn and Lin, is optimal in terms of the number of communication rounds.

Computation Costs: To analyze the computation costs, we calculate the computational effort required by each user in the protocols and compare them. In this regard, we consider the basic time-consuming operations performed by each user. Suppose that the number of users in the group is n. In phase 1 of the proposed protocol, each user performs one modular exponentiation. In phase 2, in order to compute the shared secret value with other users, each user performs $n-1$ modular exponentiations and $n-1$ modular multiplications. Then to compute the values of a degree-one polynomial at n points, each user performs n modular multiplications. In phase 3 of the proposed protocol, each user must perform $n-1$ modular inversions, followed by $2n-2$ modular multiplications. In phase 4, each user only needs to compute the value of a one-way function on a single input, which is a negligible cost. Therefore, the total number of computations required to execute the proposed protocol is: n modular exponentiations, $4n-3$ modular multiplications and $n-1$ modular inversions.

Message Lengths: Another important communication parameter in a key agreement protocol is the size of the messages sent by each user. In the proposed protocol, in the first, second, and fourth phases, users send public messages. The message published by each user at the end of phase 1 consists of one value modulo p; the public message published at the end of phase 2 consists of n values modulo q; and the message published in phase 4 consists of one value modulo q. Thus, the total length of the messages published by each user in the proposed protocol is $|p| + (n+1)|q|$.

Table 1 compares the communication rounds, computation costs, and the length of messages sent in the proposed key agreement protocol with those of Harn and Lin [22]. As clearly shown, the proposed protocol is significantly more efficient in terms of computations and

message length, while the communication rounds are the same in both.

Table 1. Comparison of the Proposed Key Agreement Protocol with those of Harn and Lin.

Protocol	Communication Rounds	Computation Costs	Length of Messages
Ours	3	n modular exponentiations, $4n-3$ modular multiplications, $n-1$ modular inversions	$\|p\|+(n+1)\|q\|$
[22]	3	n modular exponentiations, $n^2\log_n^2+3n-3$ modular multiplications	$n\|p\|+\|q\|$

6.2 Performance Analysis of the Proposed Group Key Distribution Protocol

In this section, we compare the proposed key distribution protocol with that of Harn and Hsu [23] based on communication rounds, computation costs, and message length.

Communication Rounds: In the proposed protocol, key distribution requires only one communication round, which occurs in the fourth step of the encryption phase, where the values necessary to compute the key are published. Given this, the proposed protocol is optimal in terms of the number of communication rounds, just like the group key distribution protocol proposed by Harn and Hsu [23].

Computation Costs: To analyze the computation costs, we calculate the computational effort required by each user in the protocols and compare them. In this regard, we consider the basic time-consuming operations performed by each user. Suppose that the number of users in the group is n. In the proposed protocol, participants perform 3 modular multiplications, 1 modular exponentiation, and 1 modular inversion. The initiator, on the other hand, performs $n+1$ modular exponentiations and n modular multiplications.

Message Lengths: In the proposed protocol, only the initiator publishes the required values, which consist of $n+4$ values modulo q and 1 value modulo p.

Table 2 compares the communication rounds, computation costs, and the length of messages sent in the proposed group key distribution protocol with those of Harn and Hsu [23]. As shown, due to reduced computations, the proposed protocol is significantly more efficient.

7 Conclusions

This paper has addressed the critical challenge of efficiency in secret sharing based group key establishment protocols. We have presented improved protocols for both group key agreement and group key distribution by fundamentally rethinking the application of secret sharing. The core of our contribution lies in replacing the computationally expensive Shamir's (n,n) threshold scheme, used in the foundational works of Harn and Lin and Harn and

Table 2. Comparison of the Proposed Group Key Distribution Protocol with those of Harn and Hsu.

Protocol	Length of Messages		Computation Costs		Communication Rounds
	Participants	Initiator	Participants	Initiator	
ours	$-$	$\|p\| + (n + 4)\|q\|$	3 modular multiplications, 1 modular exponentiation, and 1 modular inversion	$n + 1$ modular exponentiations, and n modular multiplications	1
[23]	$-$	$n\|p\|$	1 modular exponentiation and $n \log^2 n$ modular multiplications	$n + 1$ modular exponentiations, $n^2 \log^2 n + n^2$ modular multiplications	1

Hsu, with a far more efficient Shamir's $(2,2)$ threshold scheme. This strategic substitution successfully reduces the core polynomial operations from degree n to degree one, resulting in a significant reduction in computational overhead without compromising security.

Through detailed security analysis, we have demonstrated that our proposed protocols provably satisfy essential security properties, including key confidentiality, authenticity, and forward secrecy, under the hardness of the decisional Diffie-Hellman problem. Furthermore, performance comparisons have conclusively shown that our constructions outperform the existing protocols in terms of computational cost and communication load, while maintaining an optimal number of communication rounds. This makes our protocols particularly suitable for dynamic groups and resource-constrained environments where efficiency is paramount.

Looking forward, an important research direction would be to investigate the plausibility of applying our efficiency-enhancing method to post-quantum two-party key exchange protocols. The goal would be to construct quantum-secure group key establishment protocols that inherit the computational and communication efficiency demonstrated in this work. Exploring this idea could yield robust and efficient solutions for secure group communication in the post-quantum era.

Funding

This research received no external funding.

Data Availability Statement

Data is contained within the article.

Conflicts of Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

[1] W. Diffie, M. E. Hellman, New Directions in Cryptography, IEEE Trans. Inf. Theory, 22(6) (1976) 644–654. https://doi.org/10.1109/TIT.1976.1055638

[2] A. Joux, A one round protocol for tripartite Diffie-Hellman, International algorithmic number theory symposium, (2000) 385–393. https://doi.org/10.1007/10722028_23

[3] H. Yang, Y. Zhang, Y. Zhou, X. Fu, H. Liu, A. V. Vasilakos, Provably secure three-party authenticated key agreement protocol using smart cards, Computer Networks, 58 (2014) 29–38. https://doi.org/10.1016/j.comnet.2013.08.020

[4] Y. Xie, L. Wu, J. Shen, L. Li, Efficient two-party certificateless authenticated key agreement protocol under GDH assumption, Int. J. Ad Hoc Ubiquitous Comput., 30(1) (2019) 11–25. https://doi.org/10.1504/IJAHUC.2019.097093

[5] A. Ostad-Sharif, H. Arshad, M. Nikooghadam, D. Abbasinezhad-Mood, Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme, Future Gener. Comput. Syst., 100 (2019) 882–892. https://doi.org/10.1016/j.future.2019.04.019

[6] I. Ingemarsson, D. Tang, C. Wong, A conference key distribution system, IEEE Trans. Inf. Theory, 28(5) (1982) 714–720. https://doi.org/10.1109/TIT.1982.1056542

[7] Y. Kim, A. Perrig, G. Tsudik, Group key agreement efficient in communication, IEEE Trans. Comput., 53(7) (2004) 905–921. https://doi.org/10.1109/TC.2004.31

[8] A. Shoufan, S. A. Huss, High-performance rekeying processor architecture for group key management, IEEE Trans. Comput., 58(10) (2009) 1421–1434. https://doi.org/10.1109/TC.2009.88

[9] Y. Sun, S. Yin, J. Liu, L. Teng, A Certificateless Group Authenticated Key Agreement Protocol Based on Dynamic Binary Tree, Int. J. Netw. Secur. 21(5) (2019) 843–849. https://doi.org/10.6633/IJNS.201909_21(5).17

[10] M. Burmester, Y. Desmedt, A secure and efficient conference key distribution system, Advances in Cryptology - EUROCRYPT'94 (1995) 275–286. https://doi.org/10.1007/BFb0053443

[11] Z. Eslami, M. Noroozi, S. Kabiri Rad, Provably Secure Group Key Exchange Protocol in the Presence of Dishonest Insiders, Int. J. Netw. Secur. 18(1) (2016) 33–42. https://doi.org/10.6633/IJNS.201601.18(1).03

[12] R. Dutta, R. Barua, Provably secure constant round contributory group key agreement in dynamic setting, IEEE Trans. Inf. Theory, 54(5) (2008) 2007–2025. https://doi.org/10.1109/TIT.2008.920224

[13] A. Shamir, How to share a secret, Commun. ACM, 22(11) (1979) 612–613. https://doi.org/10.1145/359168.359176

[14] G. Blakley, Safeguarding cryptographic keys, AFIPS Conference Proceedings (1979). https://doi.org/10.1109/MARK.1979.8817296

[15] P. Feldman, A practical scheme for non-interactive verifiable secret sharing, 28th Annual Symposium on Foundations of Computer Science (sfcs 1987) (1987). https://doi.org/10.1109/SFCS.1987.4

[16] T. Tassa, Hierarchical threshold secret sharing, Journal of cryptology, 20(2) (2007) 237–264. https://doi.org/10.1007/s00145-006-0334-8

[17] Z. Eslami, N. Pakniat, M. Noroozi, Hierarchical threshold multi-secret sharing scheme based on Birkhoff interpolation and cellular automata, 2015 18th CSI International Symposium on Computer Architecture and Digital Systems (CADS) (2015). https://doi.org/10.1109/CADS.2015.7377795

[18] C. S. Laih, J. Y. Lee, L. Harn, A new threshold scheme and its application in designing the conference key distribution cryptosystem, Inf. Process. Lett., 32(3) (1989) 95–99. https://doi.org/10.1016/0020-0190(89)90008-2
[19] S. Berkovits, How to broadcast a secret, Advances in Cryptology - EUROCRYPT'91 (1991) 535–541. https://doi.org/10.1007/3-540-46416-6_50
[20] C. Li, J. Pieprzyk, Conference key agreement from secret sharing, Australasian Conference on Information Security and Privacy, (1999) 64–76. https://doi.org/10.1007/3-540-48970-3_6
[21] G. Saez, Generation of key predistribution schemes using secret sharing schemes, Discrete Appl. Math., 128(1) (2003) 239–249. https://doi.org/10.1016/S0166-218X(02)00448-1
[22] L. Harn, C. Lin, Efficient group Diffie–Hellman key agreement protocols, Comput. Electr. Eng., 40(6) (2014) 1972–1980. https://doi.org/10.1016/j.compeleceng.2013.12.018
[23] L. Harn, C. Hsu, A practical hybrid group key establishment for secure group communications, The Computer Journal, 60(11) (2017) 1582–1589. https://doi.org/10.1093/comjnl/bxx003
[24] J. Cai, Z. Zhang, M. Li, N. Li, A group authenticated key agreement protocol for secure communication between distributed power terminal devices, Comput. Electr. Eng., 118 (2024) 109214. https://doi.org/10.1016/j.compeleceng.2024.109214
[25] J. Chen, X. Zhou, W. Fu, Y. Mao, Enhancing Efficiency in Trustless Cryptography: An Optimized SM9-Based Distributed Key Generation Scheme, Sensors, 24(24) (2024) 7874. https://doi.org/10.3390/s24247874
[26] R. Subrahmanyam, N. R. Rekha, Y. V. S. Rao, Authenticated Distributed Group Key Agreement Protocol Using Elliptic Curve Secret Sharing Scheme, IEEE Access, 11 (2023) 45243–45254. https://doi.org/10.1109/ACCESS.2023.3274468
[27] P. Dzurenda, S. Ricci, R. C. Marques, J. Hajny, P. Cika, Secret sharing-based authenticated key agreement protocol, Proceedings of the 16th international conference on availability, reliability and security, (2021) 1–10. https://doi.org/10.1145/3465481.3470057

Citation: M. Noroozi, M. Ahmadi, Efficient protocols for group key establishment based on secret sharing, J. Disc. Math. Appl. 11(3) (2026) 177–191.
https://doi.org/10.22061/jdma.2025.12734.1178