Scientific Reports

https://doi.org/10.1038/s41598-026-41463-w

Article in Press

A user centric group authentication scheme for secure communication

Received: 4 April 2025

Accepted: 20 February 2026

Published online: 03 April 2026

Cite this article as: Gerenli O., Karabulut-Kurt G. & Ozdemir E. A user centric group authentication scheme for secure communication. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-41463-w

Oylum Gerenli, Gunes Karabulut-Kurt & Enver Ozdemir

A User Centric Group Authentication Scheme for Secure Communication

Oylum Gerenli[1,*], Gunes Karabulut-Kurt[2], and Enver Ozdemir[1]

[1]Istanbul Technical University, Informatics Institute, Istanbul, Turkey
[2]Polytechnique Montréal, Department of Electrical Engineering, Montréal, Canada
[*]gerenli23@itu.edu.tr

ABSTRACT

Group Authentication Schemes (GAS) are methodologies developed to verify the membership of multiple users simultaneously. Numerous GAS methods have been explored in the literature and can be classified into three generations based on their underlying mathematical principles. First-generation GASs rely on polynomial interpolation and the multiplicative subgroup of a finite field. Second-generation GASs also employ polynomial interpolation but distinguish themselves by using elliptic curves over finite fields. While third-generation GASs offer a promising solution for scalable environments, they have limitations in certain applications. Such applications typically require identifying users participating in the authentication process. In the third-generation GAS, users can verify their credentials while remaining anonymous. However, user identification is necessary in various applications. In this study, we propose an improved version of third-generation GAS that uses inner product spaces and polynomial interpolation to resolve this limitation. We address the issue of preventing malicious actions by legitimate group members. The current third-generation scheme allows members to share group credentials, which can jeopardize group confidentiality. Our proposed scheme mitigates this risk by preventing individual users from distributing valid credentials. However, a potential limitation of our scheme is its reliance on a central authority for authentication in certain scenarios.

Introduction

The Internet of Things (IoT) is a network that utilizes the internet to interconnect physical and virtual devices, enabling intelligent decision-making. IoT has numerous potential for real-time applications, such as environmental monitoring, healthcare service, inventory and production management, food supply chain (FSC), transportation, workplace, and home support[1]. However, IoT security has become a pressing concern as IoT systems expand rapidly, requiring protection of both networks and hardware components[2]. Among the various security aspects, authentication is particularly vital to the efficient operation of digital communication networks, as it ensures that devices on the network are legitimate. Standard authentication and key establishment mechanisms may not be suitable for devices communicating over wireless mediums, particularly IoT devices. Due to resource constraints of IoT devices, implementing public-key algorithm-based methods[3] for authentication is not ideal, as such methods involve operations on large numbers. Additionally, current and next-generation wireless communication systems (5G and beyond)[4,5] will connect a vast array of devices with diverse architectures. Furthermore, the transition of infrastructure from ground to space for internet service providers (ISPs) requires integrating non-traditional security measures. Assuming that all elements of the wireless systems are mobile, frequent authentication and key establishment are required for a large number of devices[6]. It is evident that the current one-by-one methods may not provide a desirable solution for ensuring secure communication among devices in such systems.

Next-generation communication systems demand authentication methods that are scalable, lightweight, secure, and user-friendly. With numerous devices exchanging data simultaneously, hundreds may require authentication at once. Group authentication schemes, which enable the simultaneous authentication of multiple users while establishing a shared secret key with minimal communication and computational overhead, emerge as a practical solution to this challenge. Although several group authentication schemes have been proposed, a reliable implementation has yet to emerge. This is due to several reasons, some of which are outlined below:

- Group authentication schemes based on polynomial interpolation are vulnerable to denial-of-service attacks, as they cannot reliably detect unauthorized join attempts. A single illegitimate request can abort the protocol, thereby disrupting the authentication process.
- The most recent promising group authentication scheme (GAS)[7], which employs inner product spaces, cannot identify the participants in the process.
- Each member of the group must be able to identify malicious actors.

The proposed scheme defines processes for key generation, group key generation, group authentication, the inclusion of new members by any existing member, and the identification of malicious actors. For group key generation, users compute a shared key by performing projection operations on their private information and public data, which is broadcast by a manager or any participant before the authentication process. Interestingly, all participants arrive at the same result despite utilizing their unique secrets. This approach guarantees private and autonomous key establishment, eliminating the necessity for direct data exchange.

The group manager (GM) generates keys by defining a subspace W within a universal space E and constructing a random polynomial $f(x)$. Each group member is then assigned a unique private key derived from the polynomial and the selected basis of the subspace W. User authentication is accomplished through the member's secret, which serves as a basis for specific details about the function. Additionally, the new member feature allows an existing member to securely add another user without the GM. While the scheme effectively prevents non-members from accessing the communication channel, it also includes a mechanism to detect malicious members. This is achieved using a recursive subgroup division algorithm to isolate and identify adversarial members within the group.

The proposed method recommends using publicly shared information for group authentication. This approach removes the necessity for members to exchange private information, thereby minimizing security risks and reducing communication overhead. Furthermore, it facilitates the identification of malicious actors and prevents unauthorized participation in the authentication or key establishment processes. Most importantly, this method enables group members to verify one another, ensuring that only authorized individuals can engage in the group's communication. Considering the recent comprehensive survey on post-quantum blockchain security for IoT[8], transitioning from registering individual identities on the blockchain to a group-based authentication paradigm provides a clear strategic advantage. By validating group membership rather than exposing individual identities, privacy-preserving guarantees are substantially enhanced. Moreover, the proposed method directly addresses the overhead challenges highlighted in the survey, particularly the strain imposed by large public keys and signatures on blockchain scalability.

In summary, this proposed method offers the following contributions:

- A method for group membership confirmation and group key establishment, where the process cost is independent of the group size.
- A method for authenticating members participating in the process.
- A method for detecting and identifying malicious members during the authentication process.
- A practical solution suitable for scalable environments.

The structure of this paper is organized as follows: Section II reviews related publications and evaluates them. Section III covers the preliminaries. Section IV introduces the proposed scheme. Section V presents the security analysis and discussion. Section VI compares the performance of the proposed scheme with other group authentication methods. Finally, Section VII concludes by highlighting the contributions to the field and outlining future research directions.

Related Work

The first- and second-generation group authentication schemes are based on polynomial interpolation over finite fields. This concept is inspired by secret-sharing schemes; therefore, we begin by presenting some fundamental concepts. The first practical methods for sharing a secret among some users are presented by Shamir[9] and Blakley[10]. Their approach ensures that when t or more participants collaborate, they can reconstruct the secret using their respective shares. However, if fewer than t participants are involved, no information about the secret can be obtained. In Shamir's proposal, the secret is divided into several pairs and distributed among the shareholders. Only those who hold a number of shares that meet or exceed the threshold can retrieve the secret. Blakley's proposal addresses the secret-sharing problem using hyperplane geometry. Each of the n participants is assigned a hyperplane equation within a t-dimensional space over a finite field. Each hyperplane occasionally intersects a specific point, and the intersection of these hyperplanes collectively represents the secret. To reconstruct the secret, participants must solve the corresponding system of equations. Unlike Shamir's secret-sharing scheme, Blakley's approach may not be practical for certain use cases due to efficiency constraints[11].

Chaum and Van Heyst[12] explore four distinct techniques for group signatures. Their findings are highly significant in the field of group authentication, which is why we present them in this work.

- In the first group signature scheme, a trusted authority selects a public key system and assigns a unique private key to each group member. The authority then makes a publicly accessible list of corresponding public keys. When a group member signs a message using their private key, the recipient verifies the signature by checking it against the public key list.

- The second group signature scheme, which is based on an RSA factorization problem, allows group members to sign anonymously and enables the verification of these signatures. The scheme preserves the signer's anonymity while also allowing them to reveal their identity when necessary. In this approach, a trusted authority selects two large prime numbers, p and q, and computes $N = p \times q$. The modulus N is then made public. A function f is chosen such that its output is coprime with N. Each group member receives a secret key s_i, which is a large random prime. The authority computes $v = \prod s_i$ and publishes N, v, and f. To generate a signature, a member transforms the message m using f and computes $S = (f(m))^{s_i} \mod N$. A zero-knowledge proof is then provided to verify the signature's validity without revealing the signer's identity .

- The third group signature scheme is similar to the second but introduces a "Trusted Public Directory" that contains each member's RSA modulus, $N_i = p_i \times q_i$. Each member's secret key consists of their factors of their modulus, p_i and q_i. During the setup phase, a trusted authority generates an independent RSA modulus $N = p \times q$. To sign a message, a member randomly selects a group of participants (including themselves) and constructs the signature using their private key's prime factor p_i, $S = (f(m))^{p_i} \mod N$. The member then provides a zero-knowledge proof that p_i divides the product of the selected members' moduli, without revealing p_i. Verification is also carried out using zero-knowledge proofs to ensure anonymity.

- The fourth group signature scheme is based on a large prime p and modular arithmetic, enabling group members to sign messages anonymously. Each member has a secret key s_i along with a corresponding public key $k_i = g^{s_i} \mod p$, where g and h are public generators. To sign a message m, a member selects a random subset of members. The signature is then computed using the member's secret key as $S = m^{s_i} \mod p$. The signer subsequently provides a zero-knowledge proof to demonstrate that the signature corresponds to a valid public key while maintaining their anonymity.

Harn's work[13] illustrates an efficient application of Shamir's secret-sharing scheme in group authentication. Unlike conventional authentication schemes, which require one-to-one authentication between each pair of nodes and incur high communication costs, this approach significantly reduces communication overhead. The study that focuses on groups is specifically intended for group authentication. Based on Shamir's (t,n) secret sharing scheme (SSS), it proposes a basic t-secure m-user n-group authentication scheme (t,m,n), where t is the proposed scheme's threshold, m is the number of users who participated in the group authentication, and n is the total number of group members. Harn introduced three types of group authentication schemes, and various studies in the literature have analyzed the security of each scheme. For instance, two of the schemes are vulnerable to an attack similar to a replay attack. Specifically, if the scheme allows multiple authentication attempts, an attacker could repeatedly attempt to access system secrets and users' private tokens. Conversely, if each secret is restricted to a single authentication attempt, regardless of its success or failure, the system becomes susceptible to DoS attacks. In this scenario, a malicious entity could inject a false value, disrupting both the authentication process and the overall group authentication mechanism.

Harn's work has served as a foundation for subsequent studies, including[14] and[15]. These studies build on Harn's approach by introducing alternative methods. For instance, in[14], Li *et al.* applied Harn's scheme in conjunction with ECC-pairing to implement group authentication and key agreement within the LTE network. Later, Mahalle *et al.* proposed a solution in[15] leveraging Paillier threshold cryptography as a core tool. Both studies also include a comparative analysis of their performance relative to Harn's work.

A physically unclonable function (PUF) is defined as a representation of a unique and unclonable characteristic inherent to a physical object. An ideal PUF functions as a one-way mechanism, where its output consistently depends on the physical system. It is easy to evaluate and construct, operates like a random function with unpredictable outputs, and is inherently unclonable[16]. PUFs are widely used in group authentication studies, such as[17]. In[17], the protocol comprises two key steps: the registration phase and the mutual authentication and key agreement phase. To achieve mutual authentication and key agreement, the protocol employs a group authentication and data transmission technique for NB-IoT, using the PUF's output as a shared root key. Additionally, PUFs have been applied in conjunction with the Chinese Remainder Theorem (CRT). Studies, including[18], have explored integrating CRT with PUF. For instance, the study in[18] introduces a lightweight key distribution and group authentication scheme that combines CRT, factorial trees, and PUF. An enhanced version of this work is further presented in[18]. Another related study[19] employs PUFs in combination with Shamir's Secret Sharing (SSS). Leveraging the homomorphic property of SSS, the scheme supports multiple group authentications using the same shared values. In addition, the authors provide a thorough comparative evaluation covering storage requirements, computational complexity, and communication overhead.

The work[20] focuses on aggregating users' signatures and improving the verification process through a batch verification mechanism. It claims that source authentication is achieved since signature verification is performed during system execution. From this perspective, the scheme can be regarded as an alternative approach to group authentication.

Blockchain technology has seen widespread adoption across various fields. In[21], if a new block receives a valid group aggregate signature from the group to which the block author belongs, it will be considered legitimate. Furthermore, the study[22] provides a detailed explanation of the authentication and key exchange processes when mobile devices join or leave blockchain-based mobile edge computing (BMEC), and a new block will be considered valid only if it obtains a valid aggregate group signature from the group to which the block creator belongs.

GAS has become a critical research topic due to the massive and concurrent access requirements of IoT devices in next-generation networks. With the advent of quantum computing, even quantum-resistant authentication methods are now being actively explored. For example, this work[23] proposes a lattice-based group authentication protocol that enables simultaneous authentication of multiple IoT devices while resisting quantum attacks.

The works of[13],[24], and[7] have made significant contributions to this field, each considered to define distinct generations of group authentication. Consequently, their studies are detailed below.

First Generation Group Authentication Scheme

One of Harn's group authentication methods is a synchronous (t,m,n) scheme, in which all participants must reveal their secret tokens simultaneously. If they fail to do so, an unauthorized participant might forge valid tokens by leveraging the tokens already revealed by others. The other two methods include the asynchronous (t,m,n) group authentication scheme and the asynchronous (t,m,n) scheme with multiple authentication attempts. All of these schemes consist of two main phases: token generation and group authentication.

Basic (Synchronous) (t, m, n) group authentication:

This fundamental scheme is intended for synchronous communication environments, where all members must engage and respond within a specified time window.

Asynchronous (t; m; n) group authentication:

In this method, a (t,m,n) group authentication scheme is proposed that enables m users (where $t \leq m \leq n$) to asynchronously release their values during a group authentication process.

Asynchronous (t; m; n) group authentications scheme for multiple authentication scheme:

The (t,m,n) asynchronous scheme with multiple authentications allows tokens to be reused.

Second Generation Group Authentication Scheme

In the study[24], the elliptic curve discrete logarithm problem (ECDLP) underpins the proposed group authentication algorithm. The algorithm operates in two distinct stages: Initialization and Confirmation, each described in detail below.

The Initialization Phase

- The GM selects a cyclic group G along with a generator P for G. Additionally, the GM determines the encryption algorithm $E = Encryption(\cdot)$, the decryption algorithm $D = Decryption(\cdot)$, and a hashing function $H(\cdot)$. A polynomial of degree $t-1$ is also chosen by the GM, with the constant term defined as the group key s.
- The GM selects a public key x_i for each user U_i and generates the corresponding private key $f(x_i)$ for $i = 1,\ldots,n$.
- The GM calculates $Q = sP$.
- The GM publishes P, Q, E, D, $H(s)$, $H(\cdot)$, and x_i, while ensuring that $f(x_i)$ is shared exclusively with the respective user U_i for $i = 1,\ldots,n$.

The Confirmation Phase

- Each user computes $f(x_i)P$ and sends $f(x_i)P \parallel ID_i$ to the GM and other users, where ID_i is the identification number of the user, and $\parallel$ represents the concatenation of two values.
- If the GM is not involved in the verification process, any user in the group computes:

$$C_i = \left(\prod_{\substack{r=1 \\ r \neq i}}^{m} \frac{-x_r}{x_i - x_r} \right) \left(f(x_i)P \right)$$

for each user, where m represents the number of users in the group, and m must be greater than or equal to t.

- The user verifies whether:

$$\sum_{i=1}^{m} C_i \stackrel{?}{=} Q$$

holds. If this condition is met , the authentication process is complete. Otherwise, the process needs to be restarted from the initialization phase.

Third Generation Group Authentication Scheme

In this method, a novel mathematical tool, the inner product space, has been employed to confirm group membership. Each member has a unique basis for the subspace W. This basis enables them to verify their group membership. The nature of subspace W allows one to select infinitely many bases for it, but knowing any basis for W is sufficient to obtain the group's secret key. However, the algorithm's design ensures that breaking the group's authentication scheme requires knowing the chosen basis, which the group manager GM keeps secret.

The group manager GM employs a randomly selected function $f(x)$ during the distribution of secrets. This function $f(x)$ can be a polynomial of degree d, which the security analysis suggests should be larger than the expected number of users in group G.

Any user U_i in group G is assigned a public key x_i (preferably an integer) and a secret key

$$B_i = \{f(x_i)v_1, f(x_i)r_1v_2, f(x_i)r_2v_3, \ldots, f(x_i)r_{n-1}v_n\}.$$

Here, $r_1, r_2, \ldots, r_{n-1}$ are random numbers selected by the group manager (GM), and these values remain the same for all users.

Each group member's secret B_i is linearly independent over W and serves as their secret key. This ensures that each user's private information is independent of others'.

The group manager or any member publishes two random vectors v and g. Participants who successfully compute the inner product of g with the projection of v, $\text{Proj}_W v$, are confirmed as group members. Below, we present the steps of the algorithm.

- A random element $v \leftarrow E$ is selected by the group manager or a group member.
- A nonce vector $g \leftarrow E$ is also selected and published alongside v.
- Each user U computes the inner product $\langle g, \text{Proj}_W v\rangle$ to derive a shared secret.
- The shared secret s is calculated as:

$$s \leftarrow \langle g, \text{Proj}_W v\rangle$$

- Finally, the user U_i releases the requested bits of s for further verification.

In this work, we enhance group authentication by introducing additional flexibility through inner-product spaces. To this end, we present the mathematical framework underlying the scheme.

Preliminaries

Lagrange Interpolation Polynomials

Lagrange interpolation describes a method for constructing the unique polynomial $p_n(x)$ of degree less than n that satisfies the $p_n(x_i) = y_i$ for each $i = 1, ..., n$ given a set of points (x_i, y_i).

The equation is:

$$p_n(x) = \sum_{j=0}^{n} y_i \mathscr{L}_{n,j}(x)$$

The cardinal functions $\mathscr{L}_{n,j}(x)$ satisfy:

$$\mathscr{L}_{n,j}(x) = \prod_{k=0, k\neq j}^{n} \frac{x - x_k}{x_j - x_k}$$

and

$$\mathscr{L}_{n,j}(x_i) = \begin{cases} 1 & if \quad i = j \\ 0 & if \quad i \neq j \end{cases}$$

In order to find $p_n(0)$, we use:

$$p_n(0) = \sum_{j=0}^{n} y_j \prod_{k=0, k\neq j}^{n} \frac{-x_k}{x_j - x_k}$$

Inner Product Space

A vector space E over a field $\mathbb{F}$ with an inner product is called an inner product space. An inner product is denoted by

$$\langle\cdot,\cdot\rangle : E \times E \to \mathbb{F}$$

An inner product $\langle\cdot,\cdot\rangle$ on a vector space E is an assignment that for any two vectors $u, v \in E$, there is a real number $\langle u, v\rangle$ satisfying the following properties:

1. **Linearity**: $\langle au + bv, w\rangle = a\langle u, v\rangle + b\langle v, w\rangle$
2. **Symmetric Property**: $\langle u, v\rangle = \langle v, u\rangle$
3. **Positive Definite Property**: For any $u \in E$, $\langle u, u\rangle \geq 0$ and $\langle u, u\rangle = 0$ if and only if $u = 0$.

The following observation presents the concept of using inner product spaces in group authentication schemes.

Theorem 1. *Let V be a vector space of dimension n over the real numbers. The probability that the randomly selected $d \leq n$ vectors are linearly dependent is negligible.*

The following theorem presents a method to find the unique projection of a vector onto any space whose basis is known.

Theorem 2. *Let $S = \{b_1, ..., b_n\}$ be an orthogonal basis for a vector space E . Then every vector $w \in V$ can be written uniquely as a linear combination of vectors in the basis S. In fact, if*

$$w = c_1 b_1 + c_2 b_2 + ... + c_n b_n$$

then

$$c_j = \langle w, b_j\rangle = \frac{w \cdot b_j}{b_j \cdot b_j}$$

If S is not an orthogonal basis, the Gram-Schmidt method is used to find an orthogonal basis from S.

Gram-Schmidt method

The Gram-Schmidt process is a method for converting a set of vectors into an orthogonal basis[25].

Theorem 3. *Given a basis $\{x_1, \ldots, x_p\}$ for a nonzero subspace W of $\mathbb{R}^n$, define*

$$\begin{aligned}
v_1 &= x_1, \\
v_2 &= x_2 - \frac{x_2 \cdot v_1}{v_1 \cdot v_1} v_1, \\
v_3 &= x_3 - \frac{x_3 \cdot v_1}{v_1 \cdot v_1} v_1 - \frac{x_3 \cdot v_2}{v_2 \cdot v_2} v_2, \\
&\vdots \\
v_p &= x_p - \frac{x_p \cdot v_1}{v_1 \cdot v_1} v_1 - \frac{x_p \cdot v_2}{v_2 \cdot v_2} v_2 - \cdots - \frac{x_p \cdot v_{p-1}}{v_{p-1} \cdot v_{p-1}} v_{p-1}.
\end{aligned}$$

Then $\{v_1, \ldots, v_p\}$ is an orthogonal basis for W. In addition,

$$Span\{v_1, \ldots, v_k\} = Span\{x_1, \ldots, x_k\}.$$

The theorem ensures that any nonzero subspace possesses an orthogonal basis.

Authentication Scheme

To ensure clarity, we provide two definitions that aim to eliminate any potential ambiguity in the study of group authentication schemes:

Definition 1. *A group authentication scheme (GAS) is a method that enables the simultaneous verification of multiple users belonging to a specific group.*

Definition 2. *A fully functional group authentication scheme (FGAS) simultaneously verifies the identities of multiple users and confirms their membership in a specific group.*

The proposed method focuses on FGAS, exploring its fundamental principles and potential advantages for secure authentication systems. The scheme is designed based on the following system and network model.

System Model

The players in the scheme:

- G: The set of users.
- U_i: Any member in the group in G.
- GM: The group manager likely has superior computational capabilities compared to any other member.

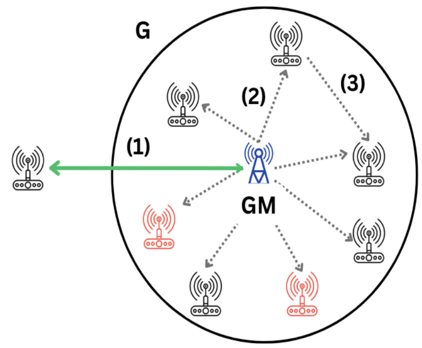

Figure 1. Communication Model: Channel (1) is designated as a dedicated line, while channels (2) and (3) are publicly accessible.

The registration of a user U_i to the group G is handled by the group manager GM. Therefore, we assume there is a secure channel between U_i and GM. On the other hand, there might be more than one group manager, or a member U_j might act as a manager for certain situations. Apart from the registration phase, all other communication channels are assumed to be open to the public as depicted in Figure 1.

Network Model

The basic structure of the network consists of 3 channels:

1. The first communication channel is between a user and the group manager responsible for handling registration. In this scenario, we are assuming it is the user's first time requesting to join the network.
2. The second communication channel enables communication between the GM and a user in the group, allowing the GM to communicate directly with group members.
3. The third communication channel is designated for users to interact and communicate with one another.

Group members can communicate with each other over the publicly accessible channel. As previously stated, the only secure channel is the one used for a member's initial registration . We assume that the group manager responsible for the first registration possesses superior computational and communication capabilities. The red color users are non-legitimate users who need to be detected. Additionally, we assume the communication channel is reliable, meaning no bit errors occur during data exchange between any two entities.

Key Generation

Typically, the responsibility for distributing user keys within a group is given to the group manager. For each group, the scheme uses subspaces of a predetermined universal inner product space E. One possible choice for E is an infinite-dimensional vector space; for instance, E could consist of all polynomials over a finite field $\mathbb{F}$.
$KeySpace$= Any subspace W of E.
$KeySpace_n$= Any subspace W of E with a dimension n.

The dimension n should be small in situations with limited memory. However, the value of n is also tied to the scheme's security parameter , so it must be chosen carefully , considering the trade-off between security and cost. We will investigate the selection of n in the sections that follow.

The first task of the manager is to select a suitable subspace W for the group G. For the sake of simplicity, we select the universal space E as $\mathbb{R}^d$ for a natural number d. In this case, the group manager's choice for dimension, n, of W should be less than d. In other words,

$$KeySpace_n = \text{ Any subspace } W \text{ of } \mathbb{R}^d \text{ with dimension } < d.$$

The GM determines the subspace W by randomly selecting n vectors in E. In other words, GM selects and keeps the set $B = \{v_1, ..., v_n\}$ for the subspace W as a secret. Theorem 1 implies that the set B is linearly independent, and as $n < d$, we can definitely conclude that the set B is a basis for W. The group manager GM also employs a randomly selected degree-1 function $f(x)$ to distribute members' secrets. Any user U_j in the group G is given a public key x_j which, in general, is selected to be an integer. The user's U_j private key is obtained via its public information x_j, the group manager's basis, and the functions that were selected by the group manager:

$$B_j = \left\{ f(x_j)v_1, f(x_j)v_2, f(x_j)v_3, \ldots, f(x_j)v_n \right\}$$

Each user's private information is unrelated to that of others; i.e., one user cannot construct any of the other members' private sets. Algorithm 1 provides a step-by-step outline of how the GM generates a unique key for every user.

Algorithm 1 KeyGen: Key Generation

Require:
$B : \{v_1, v_2, v_3, ..., v_n\} \leftarrow$ random n vectors in E.
$f(x) \leftarrow$ random element in $\mathbb{F}[x]$ of degree 1.
Key Generation:
Public key: $U_j \leftarrow x_j : x \in \mathbb{F}$.
Private key: $U_j \leftarrow B_j : \{f(x_j)v_1, f(x_j)v_2, ..., f(x_j)v_n\}$.

Group Key Generation

We present an algorithm designed to generate a secret key for the group members. For a given subspace W, each member has a distinct basis. A member chooses two vectors, v and h, randomly from the vector space E, with a preference for v and h not belonging to W. If v is already part of W, the projection operation would return the vector itself, which would not be meaningful from a computational perspective. The vectors v and h are publicly disclosed, and the s is derived from them by calculating

$$s = \langle Proj_W v, h \rangle$$

Computation of s requires knowledge of a basis for the subspace W, and the projection of v onto the subspace of W remains the same regardless of a basis for W. The projection of v onto W is given by $\text{Proj}_W v$, which can be computed as the sum of the projections of v onto each basis vector of W.

Algorithm 2 GKeyGen: Group Key Generation

Require: Two random vectors $v, h \in E$.
Group Key Generation:
Each member U_j computes $s \leftarrow \langle \text{Proj}_{B_j} v, h \rangle$.

Each user can compute s by executing a single projection operation and utilize it as a key. This key can then be employed for secure communication within the group.

Group Authentication

The purpose of group authentication is to verify the membership of many users in a certain group. The GM establishes the set $B = \{v_1, \ldots, v_n\}$ and the functions $f(x)$. The public key x_j of a user U_j is known by everyone. Each user knows their own set B_j and the value x_j:

Public key	Basis for each user
x_1	$B_1 = \{f(x_1)v_1, f(x_1)v_2, \ldots, f(x_1)v_n\}$
x_2	$B_2 = \{f(x_2)v_1, f(x_2)v_2, \ldots, f(x_2)v_n\}$
...	...
x_j	$B_j = \{f(x_j)v_n, f(x_j)v_2, \ldots, f(x_j)v_n\}$

Lagrange Interpolation gives the following equation:

$$f(0) = \sum_{j=0}^{r} f_i(x_j) \prod_{k=0,k\neq j}^{r} \frac{-x_k}{x_j - x_k} \tag{1}$$

The operations each user joining the process must complete are listed below.
Part 1: The user U_j computes A_j:

$$A_j = \prod_{k=0,k\neq j}^{r} \frac{-x_k}{x_j - x_k} \tag{2}$$

where x_k represents the public identities of the members joining the process.
Part 2: Each user multiplies the result of its computation, A_j, by a basis vector whose position in the set is either predetermined or agreed upon during the process. If no prior agreement is made, they all use the first vector from their basis set. In other words, they all compute:

$$U_1 : f(x_1)v_iA_1$$

$$U_2 : f(x_2)v_iA_2$$

$$\vdots$$

$$U_j : f(x_j)v_iA_j$$

where we assume that they all know to use the i^{th} position vector of their basis set.
Part 3: Each user computes the inner product of their result with the random vector g published by the group manager.

$$U_1 : \langle f(x_1)v_iA_1, g\rangle$$

$$U_2 : \langle f(x_2)v_iA_2, g\rangle$$

$$\vdots$$

$$U_j : \langle f(x_j)v_iA_j, g\rangle$$

Note that each computation gives

$$\langle f(x_j)v_iA_j, g\rangle = f(x_j)A_j\langle v_i, g\rangle$$

Each user encrypts their result with the group key s and sends it to GM:

$$Enc_s[f(x_j)A_j\langle v_i, g\rangle]$$

Part 4: The group manager (GM) performs the following operation using the inputs received from the participants:

$$\sum_{i=1}^{r} f(x_j)A_i\langle v_i, g\rangle = \langle v_i, g\rangle \sum_{j=1}^{r} f(x_j)A_j$$

GM confirms authentication if its computation gives the following:

$$\langle v_i, g\rangle \sum_{j=1}^{r} f(x_j)A_j \stackrel{?}{=} \langle v_i, g\rangle f(0)$$

The above equality holds since by equations (1) and (2), we have

$$\sum_{j=1}^{n} f(x_i)A_j = f(0)$$

The following algorithm summarizes the steps each participant must perform for authentication.

Algorithm 3 GroupAuth: Group Authentication

Require: x_j, B_j, g
Public key: $U_j \leftarrow x_j : x \in \mathbb{F}$
Private key: $U_j \leftarrow \{f(x_j)v_1, f(x_j)v_2, \ldots, f(x_j)v_n\}$
$g \leftarrow$ random element in E
Group Authentication:
Each user computes
Part 1: A_j: $\prod_{k=0,k\neq j}^{n} \frac{-x_k}{x_j - x_k}$
Part 2: $U_j : f(x_j)v_i A_j$
Part 3: $U_j : \langle f(x_j)v_i A_j, g\rangle$
GM receives results from each user and confirms authentication if the equation below holds.
Part 4: $\langle v_i, g\rangle \sum_{j=1}^{n} f(x_j)A_j \stackrel{?}{=} \langle v_i, g\rangle f(0)$

New Member Added to the Group by a Member

When the GM is unavailable to add new members to the group, an existing member should be able to add them. This ensures that the new member can securely communicate with the rest of the group. Also, the GM should be able to easily identify which member added it to the group.

Note that U_j has,

$$B_j = \left\{ f(x_j)v_1, f(x_j)v_2, f(x_j)v_3, \ldots, f(x_j)v_n \right\}$$

U_j selects a random number t and creates a new basis for W,

$$B_{new} = \left\{ tf(x_j)v_1, tf(x_j)v_2, tf(x_j)v_3, \ldots, tf(x_j)v_n \right\}$$

The host U_j does not need to know the function $f(x)$ to include the new member in the group conversation. It is important to note that the new member can easily participate in group communication by using its own basis.

Algorithm 4 GroupMemAdd: New Member Added to the Group by a Member U_j

Require: x_j, $B_j = \{f(x_j)v_1, f(x_j)v_2, f(x_j)v_3, \ldots, f(x_j)v_n\}$
$t \leftarrow$ Random number selected by U_j
Member Additon:
U_j creates a new basis for W,
$B_{new} = \{tf(x_j)v_1, tf(x_j)v_2, tf(x_j)v_3, \ldots, tf(x_j)v_n\}$

Malicious Actor Detection in Groups

This algorithm is designed to identify a malicious actor within a group. It starts by dividing the group of n people into two subgroups, G_1 and G_2. If either of them contains malicious users, it is further divided into smaller subgroups, and the authentication is repeated. This process continues recursively until the subgroup size is reduced to two. If both subgroups are found to be free of them, they are combined into a single group. The algorithm ensures that all malicious actors are isolated and identified through systematic division and verification of subgroups. We should note here that, unlike the first- and second-generation group authentication schemes, a non-member cannot join the authentication process, since it must have a basis for W to create the group secret key s.

Security Analysis

In this section, we examine cryptoanalysis and prominent threat models and assess the proposed algorithm's robustness against these attacks.

Cryptanalysis

The scheme has the following setup: the *KeyGen* algorithm determines a subspace W within the universal space V. Deciding a space W means determining a suitable basis for it, denoted as $B = \{w_1, \ldots, w_n\}$. Following this, the scheme dictates selecting

Algorithm 5 MalActDetect: Malicious Actor Detection in Groups

Require: Number of people in the group, n, Number of enemies in the group (unknown)
Initial:
n is the number of people in the group
Number of enemies in the group (unknown)
Goal: Find out who the enemies are
Algorithm:
while $n > 1$ **do**
  Divide n into two subgroups: G_1 and G_2
  Check whether each subgroup contains malicious actors by checking both subgroups
  if G_1 contains malicious actors **then**
    Divide G_1 into two subgroups and repeat the process
  end if
  if G_2 contains malicious actors **then**
    Divide G_2 into two subgroups and repeat the process
  end if
  if both G_1 and G_2 are determined to be valid users **then**
    Combine these subgroups into a single group
  end if
end while

a function $f(x)$. While the function can theoretically be of any degree, for mutual authentication to be feasible, the scheme requires that $f(x)$ be linear. In other words, the key consists of two components: the basis of the chosen subspace W and a function $f(x) = ax + b$ where a and b are randomly selected from the base field. This discussion leads us to the following description.

Definition 3. *The key space of the scheme consists of all subspaces of the universal space V.*

Theorem 4. *Let W be the subspace selected by the central authority. Define R as a random variable representing the dimension of the space chosen by any central authority. For a randomly chosen a, we have*

$$Pr(R = a) = \varepsilon$$

where ε is a negligible quantity.

Proof. The universal space V has been selected to have infinite dimension. Therefore, every natural number has an equal probability of being the dimension of the selected subspace. □

In scenarios where the universal space selected for the scheme is infinite-dimensional, an external eavesdropper lacks the capacity to accurately infer the dimension of the chosen subspace W. Theorem 4 is introduced to formalize the constraints and considerations involved in selecting an appropriate universal space for the scheme.

As previously noted, a polynomial space over an arbitrary field $\mathbb{F}$ constitutes a viable candidate for practical implementation. However, for the sake of simplicity and computational tractability, in our experiments we adopt the finite-dimensional Euclidean space $V = \mathbb{R}^n$ as the universal space over the real field $\mathbb{R}$. In this setting, the dimensionality of V is publicly known and thus accessible to an eavesdropper, specifically $\dim(V) = n$. Moreover, the adversary may possess a non-negligible probability of correctly guessing the dimension of the selected subspace $W \subset V$.

Despite this, the subsequent analysis (Theorems 5 and 6) demonstrates that such dimensional information does not compromise the secrecy of W. Crucially, knowledge of the dimension alone does not yield any substantive insight into the structure or identity of the secret subspace. Even within the finite-dimensional case $V = \mathbb{R}^n$, for any $d < n$, there exist uncountably many distinct subspaces of dimension d. This inherent abundance ensures that the dimension of W, while potentially guessable, does not enable an adversary to identify or reconstruct it .

It is important to emphasize that, in practical applications, the scheme typically operates over low-dimensional subspaces. Consequently, the preceding theorem may be of limited relevance in real-world scenarios, where the dimensionality of the selected subspace is often below 100. Nonetheless, the following theorems establish that an eavesdropper remains unable to verify whether a randomly selected dimension coincides with that of the actual secret subspace.

Theorem 5. *Let V be the publicly known key space over a base field $\mathbb{F}$. Then, an eavesdropper has no means of verifying whether a randomly guessed dimension coincides with the true dimension of the secret subspace $W \subset V$.*

Proof. The only information available to the eavesdropper consists of a finite collection of vectors in V, which are indistinguishable from uniformly random elements in the absence of structural knowledge about W. Since no additional data is revealed regarding the construction or dimension of W, the adversary lacks any statistical or algebraic basis to confirm the correctness of a guessed dimension. Thus, any such guess remains unverifiable. □

Corollary 1. *From an outsider's perspective , the scheme provides an information-theoretically secure approach to authentication and key establishment.*

Theorem 6. *Assume that the eavesdropper is aware of the dimension of the subspace W. This knowledge does not leak any secrecy of the group communication to the eavesdropper.*

Proof. Let d be the dimension of the subspace W. The universal space V has infinitely many subspaces of dimension d. Therefore, knowing only d does not reveal any information about the specific subspace W that has been selected. □

The previous discussion holds from an outsider's perspective. Next, we analyze the scheme from an insider's perspective. The following information is available to group members.

1. A basis B for the subspace W.
2. Two publicly known vectors, v and h, are broadcast during the authentication and key generation phases.

Theorem 7. *A legitimate user U possesses its own basis B_u, and it is infeasible to derive the group secret function $f(x)$ or the group secret basis B.*

Proof. Each vector in the basis of the user is a multiple of $f(x_u)$ where x_u is the public identity of the user U. As u does not have the knowledge of the function $f(x)$, retrieving the vectors in the secret group basis is impossible. □

One might argue about what happens when two or more users get together to obtain the function in order to access the group manager's secret. We discuss this and other situations below.

Sybil Attack

Douceur[26] first introduced the concept of a Sybil attack in peer-to-peer networks. This attack involves an adversary generating multiple fake identities, known as Sybil entities, to impersonate numerous users, either concurrently or at different times. In the proposed work, the Sybil attack will be non-functional, as shown in the proposition below.

Proposition 1. *Even if an adversary $\mathscr{A}$ acquires the bases of multiple users, it still cannot determine the group manager's secret.*

Proof. In practice, the inner product spaces are generally considered over large finite or real fields. Assume that the adversary A has the following bases:

$$\{f(x_i)v_1,\ldots,f(x_i)v_n\},\{f(x_j)v_1,\ldots,f(x_j)v_n\}$$

Based on this knowledge, isolating $f(x_i)$ from the basis elements is not possible. To determine the group manager's selected basis and the function $f(x)$, it is necessary to distinguish $f(x_i)$ and v_k from their product, $f(x_i)v_k$. However, since both entities belong to an infinite field, they can take on any value. In fact, for each a in the base field, there is nothing preventing the assignment $f(x_i) = a$, as there is no additional information to verify whether this assignment is correct.

□

Denial of Service (DoS) Attack

An earlier version of the group authentication method based on polynomial interpolation lacks the ability to identify malicious actors. If one or more actors send incorrect results during authentication or key establishment, the process must be halted. More critically, pinpointing the culprit during such an intervention is impossible. In other words, in case of the presence of an attacker, neither authentication nor key establishment can be performed via the first and second generation group authentication schemes. Since detecting a malicious actor is not feasible, the entire system may eventually need to be shut down.

The third-generation group authentication scheme does not rely on malicious actors for authentication or key establishment. As mentioned earlier, although the third-generation scheme prevents non-members from joining the process from the beginning, it cannot determine which users to involve. In other words, it only verifies an entity's membership. Our proposed work enhances the scheme by adding an additional feature to enable user authentication. That is, the proposed scheme not only confirms the membership of entities joining the process but also verifies their identities simultaneously. The structure of the scheme inherently prevents non-members from participating in the authentication process. Furthermore, it enables mutual authentication, allowing any member to easily detect a malicious actor when mutual authentication is applied. In the proposed scheme, only users possessing a valid basis set for a specific subdomain are allowed to participate in the key negotiation phase, and only valid users can perform authentication. This design effectively prevents attacker interference and safeguards the scheme against DoS attacks.

Existential Forgery Attack

In this class of strategies, the attacker attempts to construct a valid basis that enables unauthorized traversal of the authentication protocol. However, even when the secret space is constrained to a one-dimensional subspace, the attacker faces a fundamental obstacle: the universal space E admits uncountably many such subspaces, rendering the identification of the correct basis computationally and theoretically infeasible.

Moreover, possession of the group secret does not confer any meaningful advantage to the attacker. This is because the group secret alone does not reveal structural information about the underlying space, as any subspace of E, regardless of its construction, retains the potential to yield the final secret through legitimate protocol execution[7]. Thus, the entropy and ambiguity inherent in the space selection process act as a robust defense against basis reconstruction.

Even in the hypothetical scenario where an attacker manages to obtain a legitimate basis, that basis remains unusable for authentication. The reason lies in the individualized integration of the key pair $(x_i, f(x_i))$ into each participant's basis by the GM. During the authentication phase, the GM performs a verification step by checking whether the public key x_i corresponds to its embedded private counterpart $f(x_i)$. Since the function f is known exclusively to the GM and cannot be derived from public or private information, the attacker cannot replicate or validate the required key pair. Consequently, the authentication process is effectively safeguarded against impersonation or unauthorized access.

Formal Security Model

Definition 4 (Game-Based Security Model)**.** *The traditional game-based approach to cryptographic security defines an experiment between an adversary and a challenger to evaluate a protocol's security guarantees. In this framework, the adversary $\mathscr{A}$ interacts with the system by issuing queries intended to reveal sensitive information, such as user secrets or session keys. A logical predicate P specifies the* bad event, *the condition under which the adversary is considered successful. The probability of this event occurring quantifies the adversary's advantage.*

The formal game model introduced in[27] defines a security game G that operates over a structured state:

$$State = (LSID, SST, EST, LST, MST)$$

where:

- *LSID: Local session identifiers,*
- *SST: Session-specific state,*
- *EST: Global protocol-related information,*
- *LST: Local session state,*
- *MST: Global game-related metadata.*

The game is further characterized by the tuple:

$$(setupE, setupG, Q, Valid, \chi, P)$$

where:

- *setupE and setupG initialize the environment and global state respectively,*
- *Q is the set of allowed adversarial queries (e.g.,* `Send`*,* `Reveal`*,* `Corrupt`*,* `Test`*),*
- *χ is the behavior algorithm that processes queries,*
- *Valid is a predicate that determines whether a query is admissible,*
- *P is the bad event predicate.*

Each query from $\mathscr{A}$ is processed by χ, and a response is returned only if the query satisfies the Valid predicate. The security experiment is denoted by:

$$Exp^{G}_{\mathscr{A},\pi}(1^{\lambda})$$

where π is the protocol under evaluation and λ is the security parameter. The output of the experiment is a bit $b \in \{0,1\}$, indicating whether the adversary has succeeded in triggering the bad event P.

We extend the traditional game-based cryptographic model to capture the notion of *basis vector leakage*, which is critical in protocols where vector-based secrets underpin security. If an adversary $\mathscr{A}$ obtains any part of a user's basis set, the protocol's confidentiality may be compromised. To formally analyze this threat, we define the *basis-secrecy game* G^{BSec}, which models the adversary's ability to distinguish genuine basis vectors from random noise.

State. The game operates over the structured state:

$$\text{State} = (\text{LSID}, \text{SST}, \text{EST}, \text{LST}, \text{MST})$$

Game Definition. The basis-secrecy game is defined by the tuple:

$$G^{\mathsf{BSec}} = (setupE, setupG, Q, Valid, \chi, P_{\mathsf{BSec}})$$

Setup Phase. The challenger runs $(setupE, setupG)$ to generate the game environment and common parameters. The internal state is initialized as $(\text{LSID}, \text{SST}, \text{EST}, \text{LST}, \text{MST})$.

Query Phase. The adversary $\mathscr{A}$ interacts with the challenger by issuing queries from the set Q. Each query is processed by the algorithm χ, which returns a response only if the query satisfies the *Valid* predicate.

Challenge Phase. At some point, the adversary issues a challenge. To evaluate its success, the challenger samples a random bit $b \leftarrow \{0,1\}$ and responds as follows:

- If $b = 0$, the challenger reveals a genuine basis set $B_j = \{w_1, \dots, w_n\}$, split into a known subset $\{w_1, \dots, w_r\}$ and the remaining vectors $\{w_{r+1}, \dots, w_n\}$.
- If $b = 1$, the challenger reveals the same known subset $\{w_1, \dots, w_r\}$ along with $n - r$ uniformly random vectors from the underlying vector space.

The adversary outputs a guess b', and is said to win the game if $b' = b$. The adversary's advantage in this game quantifies its ability to distinguish genuine basis vectors from random noise, thereby measuring the leakage resilience of the protocol.

Leakage Bound. Let W_i be a vector space with basis $B_j = \{w_1, \dots, w_n\}$. For any adversary $\mathscr{A}$, the probability that $\mathscr{A}$ can correctly infer the remaining basis vectors $\{w_{r+1}, \dots, w_n\}$ given partial knowledge of $\{w_1, \dots, w_r\}$ is negligible. Formally,

$$P\left(\bigwedge_{j=r+1}^{n} \left(w'_j \in W_i \wedge w'_j \not\parallel \{w_1, \dots, w_r\}\right)\right) = \mathsf{negl}(\eta)$$

where each w'_j is linearly independent of the known subset, and $\mathsf{negl}(\eta)$ denotes a negligible function in the security parameter η.

Proof. Let $W \subset E$ be a subspace of a real vector space E, where $\dim W < \dim E$. Since E is defined over the field of real numbers $\mathbb{R}$, it admits uncountably many subspaces of any fixed dimension less than $\dim E$, including those of dimension $\dim W$. Suppose an adversary is given $\dim W - 1$ linearly independent vectors $\{w_1, \dots, w_{\dim W - 1}\} \subset W$. Despite this partial knowledge, there still exist uncountably many distinct subspaces of dimension $\dim W$ that contain these vectors. Consequently, the probability of correctly identifying the exact subspace W from among all such candidates is zero in the measure-theoretic sense. This reflects the inherent ambiguity in reconstructing a full basis from partial information in high-dimensional real vector spaces. □

Formal Analysis Using the Scyther Tool

The Scyther tool provides formal security analysis and is used to test the proposed group authentication method. Using the SPDL language in this program, the protocol roles can be embedded in the test environment, and attack resistance can be observed with this tool. The code of the proposed method is provided (see the Appendix), and as seen in the code, an environment with three users and one group manager has been designed. This tool shows that the proposed method passes the aliveness, weak agreement, non-injective agreement, non-injective synchronisation, and secrecy of information tests. According to the Figure 2, the tool indicates that the proposed method does not conflict with any unintended entity during the process; it is weak, but there is still consensus between the parties. In addition, the protocol does not contain illegitimate values and preserves information secrecy .

The Scyther code is designed with three users and one GM. Although this setup represents a small-scale prototype, it provides valuable insight into how the system would behave in a large-scale environment. Our proposed method consists of four main parts: the first three correspond to the user side, while the fourth belongs to the GM. On the user side, each user independently computes values such as A_j, $f(x_j)v_iA_j$, and $\langle f(x_j)v_iA_j, g\rangle$ without interacting with other users; their communication occurs solely with the GM. In the final stage, the GM verifies whether the users are legitimate group members. During authentication, all users follow the same procedure, so U_1, U_2, and U_3 are isomorphic copies of the same user role. As the number of users increases, they continue to behave as they do in this Scyther scenario. Because users do not communicate directly and the outputs of this phase are encrypted with group keys, increasing the group size does not introduce any additional structural vulnerabilities to the attack surface. In summary, although an attacker may collect information from a large number of users, the Scyther analysis remains unaffected.

Claim				Status		Comments
GroupAuth3	GM	GroupAuth3,GM1	Alive	Ok	Verified	No attacks.
		GroupAuth3,GM2	Weakagree	Ok	Verified	No attacks.
		GroupAuth3,GM3	Niagree	Ok	Verified	No attacks.
		GroupAuth3,GM4	Nisynch	Ok	Verified	No attacks.
		GroupAuth3,GM5	Secret Agg(Agg(M1,M2),M3)	Ok	Verified	No attacks.
	U1	GroupAuth3,U11	Secret B1	Ok	Verified	No attacks.
		GroupAuth3,U12	Alive	Ok	Verified	No attacks.
		GroupAuth3,U13	Weakagree	Ok	Verified	No attacks.
		GroupAuth3,U14	Niagree	Ok	Verified	No attacks.
		GroupAuth3,U15	Nisynch	Ok	Verified	No attacks.
	U2	GroupAuth3,U21	Secret B2	Ok	Verified	No attacks.
		GroupAuth3,U22	Alive	Ok	Verified	No attacks.
		GroupAuth3,U23	Weakagree	Ok	Verified	No attacks.
		GroupAuth3,U24	Niagree	Ok	Verified	No attacks.
		GroupAuth3,U25	Nisynch	Ok	Verified	No attacks.
	U3	GroupAuth3,U31	Secret B3	Ok	Verified	No attacks.
		GroupAuth3,U32	Alive	Ok	Verified	No attacks.
		GroupAuth3,U33	Weakagree	Ok	Verified	No attacks.
		GroupAuth3,U34	Niagree	Ok	Verified	No attacks.
		GroupAuth3,U35	Nisynch	Ok	Verified	No attacks.

Figure 2. The Scyther tool results show that our proposed method is resistant to all attack scenarios considered by the tool.

Performance Analysis

Real-time tests were conducted to evaluate the practicality of the proposed group authentication scheme using SageMath. The test series was run on a macOS computer with an Apple M2 processor and 16 GB of RAM. The comparison results in[24] indicate that their approach requires less time and energy compared to others. While implementing their study, which uses elliptic curves, in our test environment, we follow their suggested implementation method. For[28], we select the BLS12-381 elliptic curve because it is a well-known pairings-friendly elliptic curve. Therefore, the test environment uses the same elliptic curve for all other EC-based works[24],[29]. In addition, the *py_ecc* Python library is used to perform pairing operations.

Table 1 shows the computational overhead of the benchmark methods, where n denotes the number of users participating in the group authentication process.

Reference	Entity	Mult	Div	EC_mult	EC_add	B_pairing	Mod_exp	In_prod
Semal *et al.*[28]	User	-	-	$2(n-1)$	$n-1$	-	1	-
	GM	-	-	-	-	$n-1$	$2(n-1)$	-
	Total	-	-	$2(n-1)$	$n-1$	$n-1$	$2n-1$	-
Zhang *et al.*[29]	User	$2(n-1)$	-	-	-	-	-	-
	GM	$3n$	-	$n+2$	1	-	-	-
	Total	$5n-2$	-	$n+2$	1	-	-	-
Aydin *et al.*[24]	User	n	$n-1$	1	-	-	-	-
	GM	-	-	1	-	-	-	-
	Total	n	$n-1$	2	-	-	-	-
Our work	User	n	$n-1$	-	-	-	-	1
	GM	1	-	-	-	-	-	1
	Total	$n+1$	$n-1$	-	-	-	-	2

Table 1. Comparison of computational costs for user and *GM* in different protocols. In this table, **Mult** denotes multiplication, **Div** denotes division, **EC_mult** is elliptic curve point multiplication, **EC_add** is elliptic curve point addition, **B_pairing** denotes bilinear pairing, **Mod_exp** is modular exponentiation, and **In_prod** represents inner product computation.

Operation	Average Time (ms)
Bilinear pairing	352.361
Modular exponentiation	18.711
EC multiplication	6.415
Inner product	0.02
EC addition	0.017
Division	0.001
Multiplication	0.00055

Table 2. Average computation times of cryptographic operations.

These operations can be sorted by complexity by computing each operation. This table 2 is arranged from the most costly to the least costly. We did not analyze field multiplication and division, and real-number multiplication and division separately, because their differences are negligible. In addition, the execution time of the inner product operation is calculated based on a 10-element vector.

For a comprehensive understanding, the total authentication process for each group method was analyzed in Figure 3, and cumulative time measurements were recorded. Due to the bilinear pairing operation, the work of Semal *et al.*[28] is not suitable for large-scale group settings. To better illustrate performance differences, especially at small group sizes, we additionally provide zoom-in views for 2 to 10 users. Comparesd with the other three methods, our method is more efficient because it does not involve any costly operations and relies only on multiplication and division. Aydin *et al.* and our method are much faster than the other two methods; therefore, at small group sizes, their execution times differ only slightly, appearing almost the same. The results presented in Figure 3 clearly demonstrate that the proposed method surpasses the other methods in terms of total processing time.

The analysis Figure 4 involves identifying the operations performed by GM, CA, Key Generation Center (KGC), and Verifier, and implementing them to measure the real-time cost of the scheme for individual users. For the same reason as in Figure 3 (Total Execution Times vs. Number of Users), the left side illustrates the results for 100 to 500 users, while the right side provides a zoomed-in view for 2 to 10 users. As shown, Semal *et al.*[28] is significantly slower, whereas Aydin *et al.* and our method remain lightweight, with our method consistently achieving the lowest execution time.

It is clear that we have two studies Aydin *et al.*[24] and our proposed lightweight method. Figure 5 shows their scalability as the number of users increases from 1,000 to 10,000. As the number of users increases, execution time naturally rises because both methods exhibit an exponential trend; however, even with 10,000 users, our method remains highly efficient.

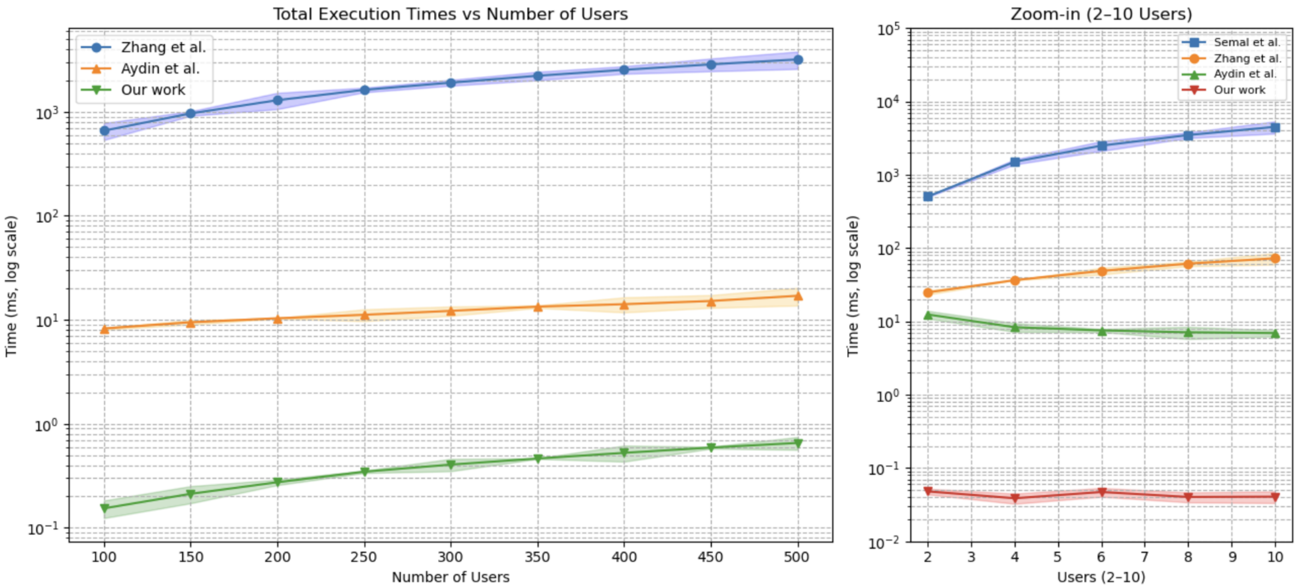

Figure 3. Total performance comparison with respect to the number of users. Since the work of Semal *et al.* takes a long time to run with a large number of users, the four studies were evaluated in the zoom-in graph for 2–10 users. For fast algorithms, it is normal for the line to appear irregular for a small number of users because the differences are very small. The other three studies were evaluated for 100–500 users.

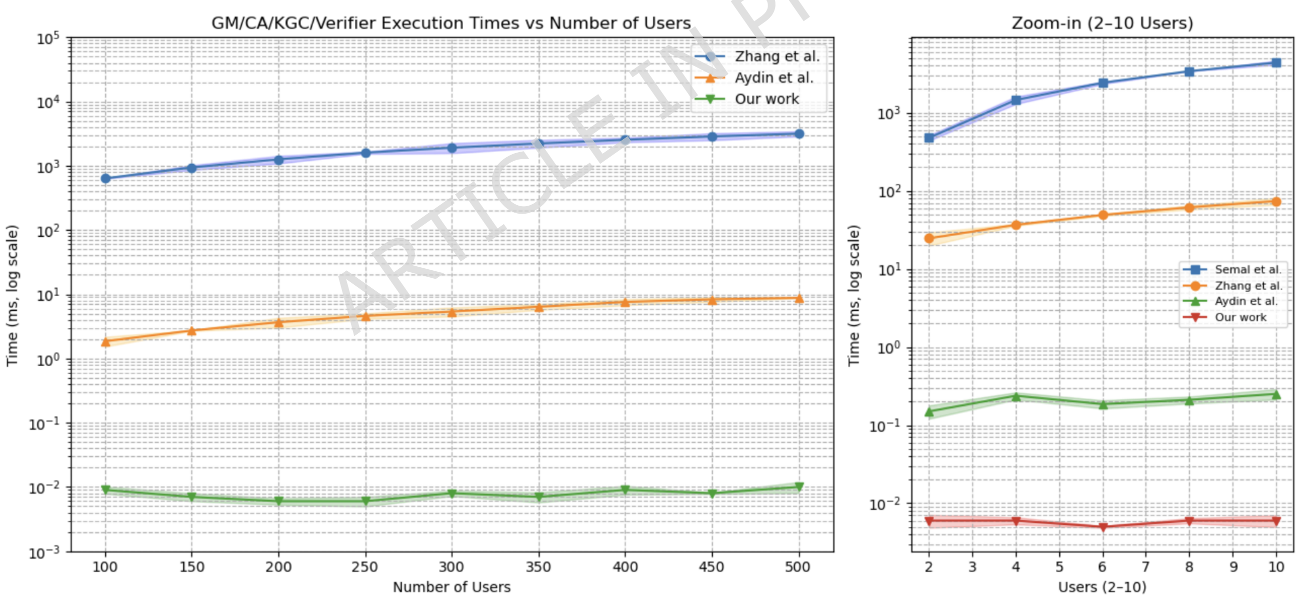

Figure 4. GM/CA/KGC/Verifier performance comparison with respect to the number of users. Since the work of Semal *et al.* takes a long time to run with a large number of users, the four studies were evaluated in the zoom-in graph for 2–10 users. For fast algorithms, it is normal *et al.*for the line *et al.*to appear irregular for a small number of users because the differences are very small. The other three studies were evaluated for 100–500 users.

To evaluate the practicality of our scheme under constrained computational environments, we conducted experiments on a Raspberry Pi 4 Model B. This device features 8 GB of RAM and a 1.5 GHz Quad-Core 64-bit ARM Cortex-A72 CPU. Notably, our implementation is strictly single-threaded, utilizing only one core for all operations. This design choice ensures that the performance metrics reflect realistic conditions encountered in IoT-class or embedded devices, which often operate under limited power and processing budgets.

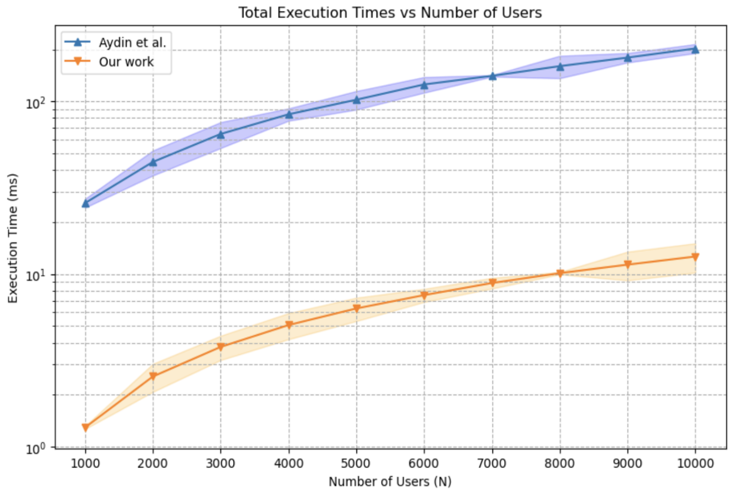

Figure 5. Total performance comparison between Aydin *et al.* and our proposed scheme. Since they demonstrated good performance, the evaluation was extended to 1,000–10,000 users. For these user numbers, the performance of our proposed scheme still appears satisfactory.

Despite its modest specifications, the Raspberry Pi 4B provides sufficient computational capability to execute the scheme reliably. Its low power consumption, approximately 3.0 W in the idle state and 6.7 W under full CPU load, further reinforces its suitability for energy-sensitive applications.

Figure 6 illustrates the execution times observed when one of the group members operates on such a constrained device. The left-hand graph shows performance metrics for single-core execution, demonstrating that the scheme remains efficient and practical even in low-power environments.

This experiment was conducted to observe individual user performance and *GM* performance under constrained computational resources, separately, since the workload increases proportionally with the number of users. Even on a resource-constrained device like the Raspberry Pi, the user execution time remains around 20 ms for up to 1000 users, demonstrating the efficiency and scalability of the proposed scheme. In addition, each user consumed only 0.0032 MB of memory. Also, the execution time of *GM* increases exponentially with the number of users; however, it remains below 10 ms even for 1,000 users, demonstrating the scheme's high efficiency. Memory consumption, on the other hand, was not explicitly measured, as the *GM* requires only a negligible amount of memory throughout the process. The right-hand graph illustrates the total memory consumption throughout the process, encompassing both the Lagrange interpolation and summation steps. The results show that overall memory usage increases almost linearly with the number of users, with only about a 5 MB difference between 100 and 1,000 users. This indicates that the scheme imposes a lightweight memory load, even as the number of users scales up.

Even when executed on a low-power device, the proposed scheme takes approximately 20 ms for up to 1,000 users, suggesting strong potential to meet the stringent low-latency demands of next-generation communication technologies, including those beyond 5G and 6G.

The findings reinforce the notion that the proposed scheme not only optimizes each user's workload but also accelerates overall group authentication, which is essential for scalability and reliability in real-world applications.

Conclusion

Contribution

In the first version of the linear group authentication scheme[7], a malicious member can provide a random basis to any party, allowing that party to join the group communication effortlessly. The confidentiality of group conversations is ensured by a scheme that constructs a shared group secret. This secret will serve as the key for encrypting and decrypting the group members' messages. The group secret can easily be obtained by anyone having a basis for the subspace assigned to the group. Therefore, any member easily constructs a random basis by selecting random coefficients for its basis elements and then getting a linear combination of these vectors. Although the first version of the linear group authentication scheme successfully eliminated a major weakness of its predecessors, which were susceptible to DoS attacks, it still has limitations. The scheme can only verify group membership but cannot identify individual users participating in the communication. Additionally, the algorithm grants each user the ability to add non-members to the group.

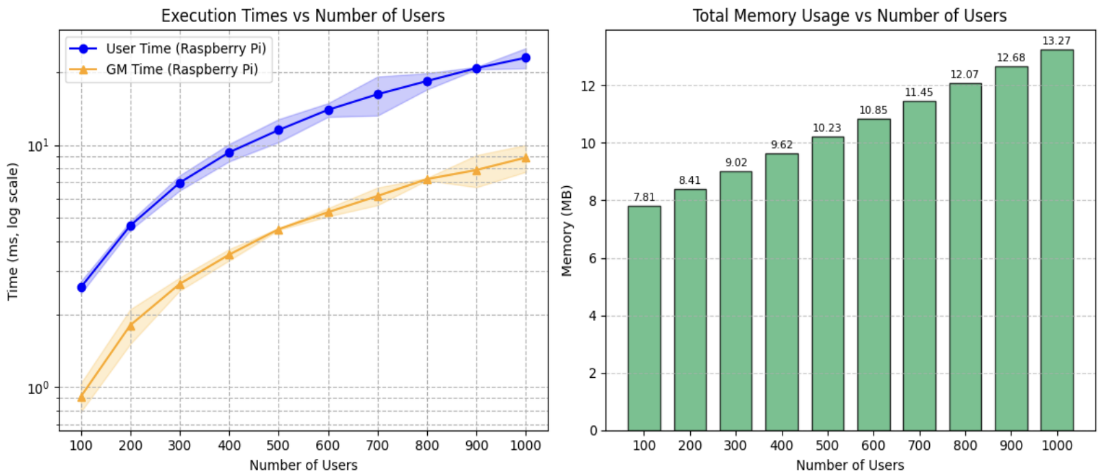

Figure 6. To evaluate the performance of the proposed scheme on limited-power devices, the implementation was tested on a Raspberry Pi. The left-hand graph shows the execution time of a single user during the authentication phase, while the right-hand graph shows the total memory consumption of the overall process.

While addressing the vulnerability to DoS attacks has introduced significant flexibility for practical use, it also highlights the need for further improvements in user identification. However, it cannot identify users participating in the communication. This creates a potential risk that non-members, added to the group without others' consent could gain access to all data exchanged among group members. In this study, we enhance the inner-product-based group authentication scheme by incorporating features that support both user identification and membership verification. Since authentication and membership confirmation occur simultaneously, this approach is particularly suitable for large-scale environments. The capability to authenticate users joining the process effectively prevents unauthorized access from both within and outside the group.

Future Work

The proposed work enhances the third-generation group authentication scheme to enable individual authentication. Future wireless communication systems are expected to incorporate space-based entities, implying that authentication for thousands of users may need to be conducted simultaneously, with authentication results relayed to others. One of our goals is to improve the current handover scheme in practice. The other one is to remove a central entity to make the method a suitable candidate for autonomous and intelligent systems.

As mentioned in the previous sections, the operation required by the *GM* is quite lightweight; therefore, a powerful centralized system is unnecessary. In this case, decentralized options can also be considered to avoid single-point-of-failure risks. Thus, multiple managers can be employed through blockchain-based group authentication, and instead of relying on a single *GM* for verification, the process is carried out through consensus on the blockchain.

Data Availability

All data generated or analyzed during this study are included in this published article.

References

1. Da Xu, L., He, W. & Li, S. Internet of Things in Industries: A Survey. *IEEE Transactions on Ind. Informatics* **10**, 2233–2243 (2014).
2. Frustaci, M., Pace, P., Aloi, G. & Fortino, G. Evaluating critical security issues of the IoT world: Present and future challenges. *IEEE Internet Things J.* **5**, 2483–2495 (2017).
3. Chaudhary, R., Aujla, G. S., Kumar, N. & Zeadally, S. Lattice-based public key cryptosystem for Internet of things environment: Challenges and solutions. *IEEE Internet Things J.* **6**, 4897–4909 (2018).
4. Chettri, L. & Bera, R. A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. *IEEE Internet Things J.* **7**, 16–32 (2019).
5. Nguyen, D. C. *et al.* 6G Internet of Things: A comprehensive survey. *IEEE Internet Things J.* **9**, 359–383 (2021).

6. Aman, M. N., Javaid, U. & Sikdar, B. A privacy-preserving and scalable authentication protocol for the Internet of vehicles. *IEEE Internet Things J.* **8**, 1123–1139 (2020).
7. Guzey, S., Karabulut Kurt, G. & Ozdemir, E. Group authentication and key establishment scheme. *IEEE Internet Things J.* **11**, 35086–35099 (2024).
8. Gharavi, H., Granjal, J. & Monteiro, E. Post-quantum blockchain security for the Internet of Things: Survey and research directions. *IEEE Commun. Surv. & Tutorials* **26**, 1748–1774 (2024).
9. Shamir, A. How to share a secret. *Commun. ACM* **22**, 612–613 (1979).
10. Blakley, G. R. Safeguarding cryptographic keys. In *International Workshop on Managing Requirements Knowledge (MARK)*, 313–318 (1979).
11. Alfred, M., Scott, V. *et al. Handbook of Applied Cryptography* (CRC press, 1997).
12. Chaum, D. & Van Heyst, E. Group signatures. In *Advances in Cryptology—EUROCRYPT'91: Workshop on the Theory and Application of Cryptographic Techniques*, 257–265 (1991).
13. Harn, L. Group authentication. *IEEE Transactions on Comput.* **62**, 1893–1898 (2012).
14. Li, J., Wen, M. & Zhang, T. Group-based authentication and key agreement with dynamic policy updating for MTC in LTE-A networks. *IEEE Internet Things J.* **3**, 408–417 (2015).
15. Mahalle, P. N., Prasad, N. R. & Prasad, R. Threshold cryptography-based group authentication (TCGA) scheme for the Internet of things (IoT). In *International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, 1–5 (2014).
16. Gope, P., Das, A. K., Kumar, N. & Cheng, Y. Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks. *IEEE Transactions on Ind. Informatics* **15**, 4957–4968 (2019).
17. Ren, X., Cao, J., Ma, M., Li, H. & Zhang, Y. A novel PUF-based group authentication and data transmission scheme for NB-IoT in 3GPP 5G networks. *IEEE Internet Things J.* **9**, 3642–3656 (2021).
18. Singh, K. R. & Sethia, D. ALPAK: Anonymized Lightweight PUF-based group Authentication with Key distribution for IoT Devices. In *International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–7 (2023).
19. Nakkar, M., AlTawy, R. & Youssef, A. Lightweight group authentication scheme leveraging Shamir's secret sharing and PUFs. *IEEE Transactions on Netw. Sci. Eng.* **11**, 3412–3429 (2024).
20. Zhao, S. *et al.* PPMM-DA: Privacy-preserving multidimensional and multisubset data aggregation with differential privacy for fog-based smart grids. *IEEE Internet Things J.* **11**, 6096–6110 (2023).
21. Zhang, C., Lu, R., Lin, X., Ho, P.-H. & Shen, X. An efficient identity-based batch verification scheme for vehicular sensor networks. In *IEEE INFOCOM*, 246–250 (2008).
22. Zhang, S. & Lee, J.-H. A group signature and authentication scheme for blockchain-based mobile-edge computing. *IEEE Internet Things J.* **7**, 4557–4565 (2019).
23. Xu, P. *et al.* Anti-quantum certificateless group authentication for massive accessing IoT devices. *IEEE Internet Things J.* **11**, 16561–16577 (2024).
24. Aydin, Y., Karabulut Kurt, G., Ozdemir, E. & Yanikomeroglu, H. A flexible and lightweight group authentication scheme. *IEEE Internet Things J.* **7**, 10277–10287 (2020).
25. Hoffmann, W. Iterative algorithms for Gram-Schmidt orthogonalization. *Computing* **41**, 335–348 (1989).
26. Douceur, J. R. The sybil attack. In *International Workshop on Peer-to-Peer Systems*, 251–260 (2002).
27. Brzuska, C., Fischlin, M., Warinschi, B. & Williams, S. C. Composability of bellare-rogaway key exchange protocols. In *Proceedings of the 18th ACM conference on Computer and communications security*, 51–62 (2011).
28. Semal, B., Markantonakis, K. & Akram, R. N. A certificateless group authenticated key agreement protocol for secure communication in untrusted uav networks. In *2018 IEEE/AIAA 37th digital avionics systems conference (DASC)*, 1–8 (IEEE, 2018).
29. Zhang, J., Cui, J., Zhong, H., Chen, Z. & Liu, L. Pa-crt: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks. *IEEE Transactions on Dependable Secur. Comput.* **18**, 722–735 (2019).

Funding

This work was supported by Istanbul Technical University under the Graduate Thesis Project – Doctoral Thesis Project (Project No: MDK-2025-47377).

Author contributions statement

The theoretical layout was developed by E.O. and O.G. The implementation and analysis were conducted by O.G. and G.K-K, while the security analysis was handled by E.O. and O.G.