IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 March 2023, accepted 28 April 2023, date of publication 8 May 2023, date of current version 12 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3274468

RESEARCH ARTICLE

Authenticated Distributed Group Key Agreement Protocol Using Elliptic Curve Secret Sharing Scheme

ROLLA SUBRAHMANYAM, N. RUKMA REKHA, AND Y. V. SUBBA RAO
School of Computer and Information Sciences (SCIS), University of Hyderabad, Hyderabad 500046, India
Corresponding author: Rolla Subrahmanyam (subbucse596@gmail.com)

ABSTRACT One of the fundamental construction blocks in safeguarding group communications is group key establishment protocols. Group key agreement protocols are more suitable for distributed environments where the participant from various places can agree upon the group key. Group key related information is distributed to various group participants, mostly using techniques such as using polynomials, bilinear pairing, and secret sharing scheme (SSS). Out of these, secret sharing schemes are more efficient compared to other techniques. Recent growth in Internet of things (IoT) related applications stresses the need for such key agreement protocols in resource-constrained environments. Elliptic curves are quite popular in resource-constrained environments to produce enough security with smaller key sizes. Elliptic curve secret sharing scheme (ECSSS) is proposed in this paper for resource-constrained environments. Using the same scheme, Authenticated distributed group key agreement protocol using Elliptic curve secret sharing scheme (ADGKAP) is proposed, which can be used as a group key agreement protocol, and the group key related information is shared using ECSSS. To the best of our knowledge, we know no group key agreement protocol in the literature that uses ECSSS in a distributed environment. Our proposed ADGKAP provides equal security with less key size, storage space, faster computation, and less computational cost without compromising on the number of rounds compared to existing schemes. This scheme security relies on the Elliptic curve discrete logarithm problem (ECDLP), and the security analysis of the scheme is discussed.

INDEX TERMS Secret sharing scheme (SSS), key agreement protocol (KAP), group key agreement protocol (GKAP), elliptic curve (EC), elliptic curve discrete logarithm (ECDLP).

I. INTRODUCTION

The advent of new networking technologies and evolving of good speed in the internet and its connectivity resulted in various group applications where participants work collaboratively even from remote places. However, when group participants work together and share common interests in such applications, there is also a dire need for secure group communication. The basic necessity for such secure group communications is a secure group key establishment protocol. Most of the group key establishment protocols rely on a trusted third party, such as a Key generation centre (KGC), to distribute the session key among the group participants. Malicious KGC can compromise the security of the established group key, and it also introduces a single point of failure in various contexts. Establishing a secure group key in a distributed environment with no KGC will be extremely difficult. In addition, establishing a secure group key in a resource-constrained distributed environment will be much more challenging as the number of computations that can be done on lightweight devices is limited. Computational speed and key size are quite essential in secure group communication for resource-constrained environments like the Internet of things (IoT).

Authentication, Confidentiality, and Integrity are three major goals for secure group communication in a distributed environment. In a distributed environment, authentication is required to verify the genuineness of the key, as the

The associate editor coordinating the review of this manuscript and approving it for publication was Shu Xiao.

key generation process involves multiple participants of the group. Confidentiality ensures that the transmitted messages are from valid participants and that only authorized participants can decrypt the received messages. In addition to these, integrity ensures that the messages are never modified during their entire transmission.

Although group key establishment protocols use polynomials and bilinear pairings, the usage of a secret sharing scheme for distributing group key related information is a popular strategy used in literature to attain efficiency. Cao et al. [32] proposed a constant round group key establishment protocol using a secret sharing scheme. Harn and Lin [33] proposed an authenticated group key agreement protocol using a secret sharing scheme. However, these schemes have limitations to use in resource constrained environments as their key size is too big for resource constrained devices like RFID (radio frequency identification).

The term ''resource-constrained environment'' refers to a system or device with limited resources, including memory, processing power, battery life, storage space, and bandwidth [62]. In low power systems such as embedded systems, Internet of Things devices, mobile devices, and other devices, these environments are common. In a resource-constrained environment, there is a dire need for a secure group key to communicate or exchange ideas. Zhang et al. [48], Cheng et al. [49], Li et al. [61], Zhang et al. [60], and Zhang et. al. [59] introduced group key agreement protocols (GKAP), which are useful for resource-constrained environments, however, the computational cost involved is too high. All these schemes use very costly operations such as bilinear pairing and modular exponentiation and a secure channel for distributing group key related information.

Although Harn and Lin [1] introduced group key agreement protocol using a secret sharing scheme, the computational cost involved is too high. Many of the secret sharing schemes in literature used Elliptic curves to achieve a good amount of security with smaller key sizes, making them a great fit for resource constrained environments.

Extending Harn and Lin [1] scheme using Elliptic curves for secret sharing is not feasible as we will not be able to reconstruct the polynomial from the pairs of points generated in the secret distribution phase. Hence a novel, Authenticated distributed group key agreement protocol using Elliptic curve secret sharing scheme(ADGKAP) is proposed in this paper to achieve equal security with a smaller key size and less storage, and faster computation. This scheme is more suitable for resource-constrained environments compared to Harn and Lin [1].

However, to the best of our knowledge, no secret sharing scheme using an elliptic curve in a distributed environment is available in the literature. Hence we propose a novel Elliptic curve secret sharing scheme (ECSSS) in a distributed environment first and use that in our proposed ADGKAP.

A. OUR CONTRIBUTIONS

The following are our's contributions:

- Elliptic curve secret sharing scheme (ECSSS) is proposed.
- Authenticated distributed group key agreement protocol using Elliptic curve secret sharing scheme (ADGKAP) is proposed by using ECSSS to achieve equal security with relatively smaller key size, storage, fast computation, and less computational cost requirements.
- To the best of our knowledge, no Authenticated distributed group key agreement protocol has used Elliptic curve secret sharing scheme for share distribution till date.

The salient features of the ADGKAP are as below:

- A novel Elliptic curve secret sharing scheme is designed to generate points that are used as shares by the users.
- All the shares of the scheme are shared through a public channel in a distributed environment.
- After the group key is reconstructed by each individual user, the user can verify the authentication of the group key by comparing the hash of his reconstructed key with the hash of the other user's reconstructed key.
- ADGKAP is proven secure enough provided the elliptic curve discrete logarithm problem is intractable. This ADGKAP provides equal security with a smaller key size, less storage, faster computation, and less computational cost without compromising on the number of rounds. In this ADGKAP, each group user can reconstruct the key individually, but an attacker cannot reconstruct the key.

The paper's organization is as follows: Related work is explained in section II. In section III, Preliminaries, definitions of the Elliptic curve, Elliptic curve discrete logarithm problems, Vandermode matrix, Background, Authenticated group Diffie–Hellman key agreement protocol are explained. Proposed schemes, Elliptic curve secret sharing scheme (ECSSS), Authenticated distributed group key agreement protocol using Elliptic curve secret sharing scheme (ADGKAP), and the Numerical example is explained in section IV. Security analysis and Comparisons are described in sections V and VI, respectively. Finally, the Conclusion and future work is given in section VII.

II. RELATED WORK

In secure multiparty computing, a group DH (Diffie Hellman) key is computed for a group of n members, each with a private key k_i, and calculates a function $f(k_1, k_2, \ldots, k_n)$ [2]. Tzeng and Tzeng [3], [4] introduced a round-efficient conference key with $f(k1, k2, .., k_n) = g^{k1+k2+,\ldots,+k_n}$. This is an extension to Burmester and Desmedt [5] two round protocol with $f(k1, k2, .., k_n) = g^{k_1k_2+,\ldots,+k_nk_1}$ which is round efficient but has a malicious participant attack. The basic DH key agreement protocol was substantially generalized for many group DH key approaches. This technique was utilized by Ingemarsson et al. [6], Steer et al. [7], Burmester and Desmedt [5], and Steiner et al. [8] to exchange group DH public keys by arranging group members in a logic ring.

Burmester achieved computational efficiency by reducing the number of rounds from $(n-1)$ to 3. In contrast, Lee et al. [9] and Kim et al. [10], [11] used a binary tree to arrange group members and exchange DH public keys. In 1996, Steiner et al. [8] introduced the GDH (Group DH) key exchange (KE) as an extension to DH protocol.

Steiner et al. [12] protocol was improved with authentication services in 2001 and is proven secure. In 2006, Bohli [13] developed a scheme for secure group key agreement protocols, which allows unverified point-to-point networks to be secure against internal and external attackers.

Later, in 2007, Bresson et al. [14] developed a secure generic authenticated Group DH key exchange protocol. In 2007 itself, Katz and Yung [15] developed the first provably secure constant-round and completely scalable GDH protocol in the standard model. Brecher et al. [16] added robustness to the GDH protocol's tree-DH method by making it resistant to system failures, network outages, and member misconduct. Jarecki et al. [17] developed a group key agreement system that can withstand up to t node failures out of n nodes. Secure digital signatures are used to provide authentication for DH public keys. The computational cost of each group member is a crucial concern when implementing these protocols, especially when the group size is large.

Joux [18] first proposed to use pairings in a one-round tripartite key exchange. Later [19], [20], [21], [22], [23] developed several versions of authenticated group key exchange protocols. However, most of the pairing-based group KE methods are inefficient, as they result in a rise in the number of rounds as the group size increases. Choi et al. [21] developed a pairing-based group KE protocol that required a fixed number of rounds so that each member will be calculating two pairings and $4n$ modular exponentiation where n is a message.

A tree-based pairing-based group KE was developed by Barua et al. [19]. Du et al. [22] proposed an authenticated ID-based group KE mechanism with a constant number of rounds. In 2008, Desmedt and Lange [24] created a constant round pairing-based authenticated group KE that had lower processing complexity per member than previous protocols. Wu et al. [25] and Zhao et al. [48] developed an asymmetric group key agreement model that establishes secure networks for group communications. In a hierarchical access control system, Gu et al. [26] represented a group key agreement technique to speed up rekeying time. Konstantinou [27] developed an ID-based group key agreement protocol with efficient constant rounds for adhoc networks. Sun et al. [34] developed a new key agreement mechanism with provable security that does not require the use of a certificate.

In certain group key transfer protocols, secret sharing is used for group key communication to achieve efficiency. Laih et al. [28] introduced the first group key transmission technique using a (t, n) secret sharing method in 1989. Each scheme participant must enroll with a conference chairperson and reveal a secret to the chairperson. The conference chairperson will select a random conference key as the secret and uses the secret-sharing mechanism to distribute shares of the secret among members. However, the number of users that the key can be shared is limited to $(t-1)$ users only out of n users. [29], [30], [31] took a similar approach to distribute group communications to a large number of people securely. Cao et al. [32] developed a constant-round group KE protocol based on secret sharing with universally composable security. Harn and Lin [33] developed an authenticated group key transfer protocol based on the secret sharing technique. However, members do not have equal priority in this arrangement. To overcome this problem, Harn and Lin [1] designed an efficient group DH key agreement protocol using a secret sharing scheme.

Yang and Chang [52] introduced a key agreement-based elliptic curve scheme. It has no perfect forward secrecy, impression attack, and provable security. Yoon and Yoo [53] designed a key agreement protocol that has provable security but suffers from perfect forward secrecy. Debio et al. [38] developed a key agreement protocol with perfect forward secret secrecy that is provable and secure. In all these three schemes, communication happens between KGC and users. Chien [41] introduced a key agreement protocol based on an elliptic curve where communication happens between server and tag in multiple rounds. After that, Liu [40] introduced a key agreement protocol based on ECC (Elliptic curve cryptography). This protocol has one round of communication between the server and the tag. After that, Shen et al. [50] introduced that ECC could be used to create secure session keys between authorized users and devices. Based on ECC, Islam and Biswas [39] developed an unpaired authentication group key protocol. It minimizes computational costs and gets rid of public key certificates. In literature, [43], [44], [45] introduced group key agreement protocols based on an elliptic curve where communication happens between KGC and users.

Dzurenda et al. [51] introduced an authenticated key agreement protocol based on secret sharing schemes. However, communication happens between KGC and users. Cheng et al. [49] introduced a group key agreement protocol based on bilinear pairing, which is suitable for mobile environments. Zhang et al. [48] introduced a group key agreement protocol based on bilinear pairing. It has been shown that this protocol can withstand harmful attacks, such as active and passive attacks. Although Cheng et al. [49] and Zhang et al. [48] developed key agreement protocols without KGC, the cost involved is more. Hence we proposed an Authenticated Distributed key agreement protocol using ECSSS to improve efficiency.

Raghunandan et al. [58] introduced an encryption scheme. The keys for encrypting the data are generated by the chaotic maps using pseudo-random numbers. Unauthorized users have a lot of difficulty accessing or changing the data because of the encryption procedure. One advantage of chaotic-map-based encryption is that it can be used on edge devices with low computing power. This scheme can be applied for encryption in a distributed environment.

Liu et al. [35], Binu et al. [36], and Wang et al. [37] introduced centralized secret sharing schemes based on elliptic curve and pairing. However, these schemes are not suitable for distributed environments. Liu et al. [40], Sheikhi-Garjan [42] send shares through a public channel in a centralized environment which is a general tendency. However, to the best of our knowledge, other than Harn and Liu [1], no secret sharing scheme sends shares through the public channel in a distributed environment.

III. PRELIMINARIES

In this section, the Elliptic curve (EC), Elliptic curve point addition and point doubling, Elliptic curve discrete logarithm problem (ECDLP), Vandermode matrix definitions, and Background, Authenticated group Diffie–Hellman key agreement protocol are explained.

Elliptic Curve: Let $\mathbb{F}_q$ be a field, where $q > 3$ is a prime. A curve of the form $E : y^2 = x^3 + ax + b$ over $\mathbb{F}_q$ is an elliptic curve if the discriminator $\Delta = 4a^3 + 27b^2 \neq 0$, where a, b are constants. The set of all points on E over $\mathbb{F}_q$ is denoted as $E(\mathbb{F}_q)$, along with point at infinity $\mathcal{O}$ [46], [47]. The Elliptic curve has point addition and point doubling operations.

Elliptic Curve Point Addition and Point Doubling: Let E be an elliptic curve, and P and Q be two points on it. Let R be sum of P and Q. The sum can be obtained as follows. Draw a line L passing through P and Q, which intersect the curve at another point, say R'. The reflection of R' about $x-$ axis is R, and we write $R = P + Q$. If P and Q are the same, the line L is tangent at P and intersects the curve at another point, R'. The reflection of R' about $x-$axis is R, and we write $R = 2P$. Examples of point addition and point doubling on elliptic curves are depicted in 1 and 2, respectively.

Suppose that $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ then the formula for R is given below. Let $R = P + Q = (x_3, y_3)$

If $P \neq Q$, then

$$x_3 = m^2 - x_1 - x_2$$
$$y_3 = m(x_1 - x_3) - y_1$$

where $m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$ is slope of line L passing through P and Q.

If $P = Q$, then $R = P + P$.

$$x_3 = m^2 - 2x_1$$
$$y_3 = m(x_1 - x_3) - y_1$$

where $m = \frac{(3x_1^2 + a)}{(2y_1)}$ is slope of tangent line L at P. This is called Point doubling [46], [47].

Elliptic curve discrete logarithm problem: Let E be an elliptic curve, and P is a point on it. For a given point $Q = xP$, $1 \leq x \leq q$, the problem of finding private key x from public points Q and P is called an Elliptic curve discrete logarithm problem [56].

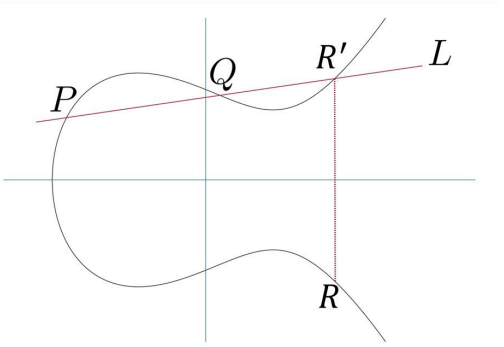


FIGURE 1. Point addition on Elliptic curve.

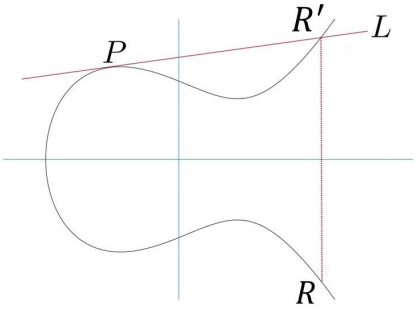


FIGURE 2. Point doubling on Elliptic curve.

Vandermode matrix: A vandermode matrix [55] is a matrix of the form

$$A = \begin{bmatrix} 1 & z_1 & z_1^2 & \cdots & z_1^{n-1} \\ 1 & z_2 & z_2^2 & \cdots & z_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_n & z_n^2 & \cdots & z_n^{n-1} \end{bmatrix}_{n \times n}$$

with $z_i \neq z_j$ for $i \neq j$. The matrix A is always invertible. Every vandermode matrix is invertible as the determinant of the matrix A is $\prod_{1 \leq i < j \leq n} (z_j - z_i)$, which is non-zero.

A. BACKGROUND

In this section, we briefly discussed Harn and Liu [1], Authenticated group Diffie–Hellman key agreement protocol. In this protocol [1], a group of users wants to reconstruct group key S in a distributed environment through a public channel. It has three rounds. In round 1, each user U_i, $1, 2, \cdots, n$, computes $n-1$ private shares and $n-1$ public shares of his key K_i and the private shares are sent via a public channel in an encrypted manner. In round 2. each user U_i, $i = 1, 2, \cdots, n$, reconstructs the key K_j of U_j, $j = 1, 2, \cdots, n$, $j \neq i$ by using his share and the $n-1$ public shares. Then, U_i reconstructs the group key $S = \prod_{j=1,}^{n} K_j$. In round 3, each individual user reconstructs the group key, and the user authenticates the group key. The scheme is explained in section III-A1 below.

1) AUTHENTICATED GROUP DIFFIE-HELLMAN KEY AGREEMENT PROTOCOL

A group with n users $(U_1, U_2, \cdots, U_n)$, want to construct group key S collaboratively in a public channel. It consists of the following set up and rounds.

SET UP

- Any user choose two primes p and q such that $p = 2q+1$, and choose generator $\alpha \in \mathbb{F}_q$.
- Every user $U_i, i = 1, 2, \cdots, n$, chooses private keys k_i and x_i, computes $r_i = \alpha^{k_i}, y_i = \alpha^{x_i}$ and makes r_i and y_i public.

ROUND:1

- Every user U_i computes shares $y_{ij} = (r_j y_j)^{x_i+k_i}, j = 1, 2, \cdots, n, j \neq i$.
- Every user U_i constructs key $K_i = \prod_{j=1, j\neq i}^{n} (r_j y_j)^{k_i+x_i}$.
- U_i constructs polynomial $f_i(x)$ of degree $n-1$ from n points: $(0, K_i), (r_j, y_{i1}), \cdots, (r_j, y_{ij}), \cdots, (r_j, y_{in})$, $j = 1, 2, \cdots, n, j \neq i$.
- U_i computes $n-1$ shares from $f_i(x), x = 1, 2, \cdots n-1$ and makes them public.

ROUND:2

- U_i computes his corresponding share from U_j, $y_{ji} = (r_j y_j)^{x_i+k_i}, j = 1, 2, \cdots, n, j \neq i$.
- U_i reconstructs key $K_j, j = 1, 2, \cdots, n, j \neq i$ of U_j from $n-1$ public shares and his respective share by using Lagrange's interpolation formula.
- U_i reconstructs group key $S = \prod_{j=1,}^{n} K_j$.

ROUND:3

- Each user U_i computes $C_i = h(S, U_i, r_i)$ and makes it public.
- Each user U_i computes $C_j' = h(S, U_j, r_j)$, for $j = 1, 2, \cdots, n, j \neq i$. If $C_j' = C_j$ the group key S is valid.

Harn and Liu [1] scheme cannot be extended using Elliptic curve for secret sharing as we cannot reconstruct the polynomial from the pairs of points in the secret distribution phase (Round 1). Hence a novel Authenticated distributed group key agreement protocol using Elliptic curve secret sharing scheme is proposed in this paper.

IV. PROPOSED SCHEMES

In this section, we propose two schemes, namely, the Elliptic curve secret sharing scheme (ECSSS) and Authenticated distributed group key agreement protocol using Elliptic curve secret sharing scheme (ADGKAP).

ECSSS has two rounds. In round 1, user U_1 computes n shares of key k_1P generated from elliptic curve E. Out of n shares, $n-1$ shares are made public, and 1 is kept as a private share. The private share is sent to $n-1$ users through a secure channel. In round 2, U_2, U_3, U_4 reconstructs key k_1P.

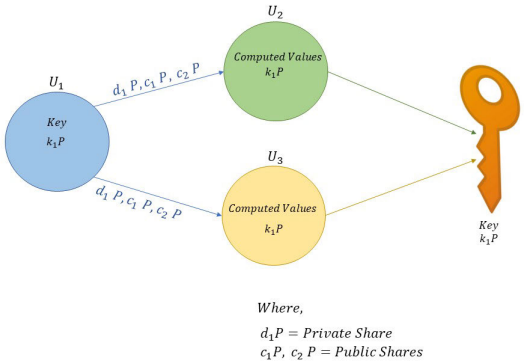


FIGURE 3. Framework of ECSSS.

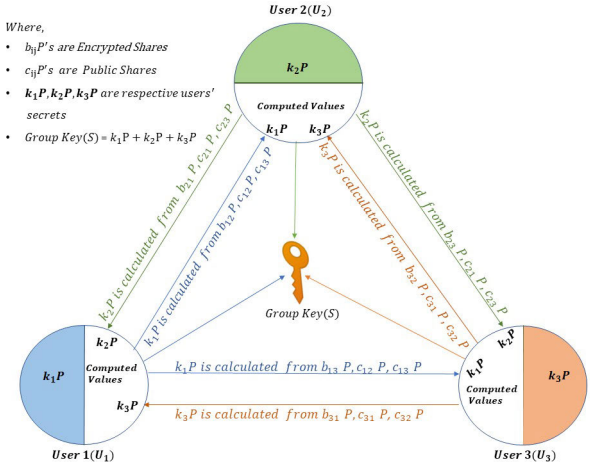


FIGURE 4. Framework of ADGKAP.

The ECSSS and the Correctness of the key reconstruction are explained in sections IV-A and IV-A1, respectively.

Next, we propose ADGKAP by using ECSSS. ADGKAP has three rounds. In round 1, each user $U_i, i = 1, 2, \cdots, n$ computes $n-1$ private shares and $n-1$ public shares of point k_iP. The private shares are sent in an encrypted manner via a public channel. Then, each user reconstructs his respective share using his private key and encrypted share. In round 2, every user $U_i, i = 1, 2, \cdots, n$ reconstructs the key k_jP of $U_j, j = 1, 2, \cdots, n, j \neq i$ by using his respective share and $n-1$ public shares. Then, each user U_i reconstructs the group key $S = \sum_{j=1}^{n} k_jP$. In round 3, the group key is reconstructed by each individual user. Also, the user authenticates of group key by comparing the hash of each user's reconstructed key with the hash of the other user's reconstructed key. This ADGKAP and the Correctness of the group key reconstruction are explained in sections IV-B and IV-B1, and the numerical example of the ADGKAP is explained in section IV-C, respectively.

A. PROPOSED ELLIPTIC CURVE SECRET SHARING SCHEME (ECSSS)

The proposed Elliptic curve secret sharing consists of two rounds, namely secret distribution and key reconstruction.

In secret distribution, a key k_1P generated from the elliptic curve E is distributed as shares to all the participants of the group. In key reconstruction, each user will be able to reconstruct the key k_1P.

Assume that user U_1 wants to secretly send the shares of key, k_1P to $n-1$ users $(U_2, U_3, U_4, \cdots, U_n)$.

ROUND 1: SECRET DISTRIBUTION

- Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_q$, q is prime power.
- Point $P \in E(\mathbb{F}_q)$ and $G =< P >$ be a group of order ℓ, where ℓ is prime.
- User U_1 makes E, q, P, and ℓ public.
- User U_1 chooses a $n \times n$ vandermonde matrix A_1 and makes it public

$$A_1 = \begin{bmatrix} 1 & z_1 & z_1^2 & \cdots & z_1^{n-1} \\ 1 & z_2 & z_2^2 & \cdots & z_2^{n-1} \\ 1 & z_3 & z_3^2 & \cdots & z_3^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_n & z_n^2 & \cdots & z_n^{n-1} \end{bmatrix}_{n\times n}$$

- U_1 selects values, $r_1, r_2, \cdots, r_{(n-1)}, k_1 \in [1, \ell - 1]$ randomly.
- U_1 calculates $d_1, c_1, \cdots, c_{(n-1)}$ using $(d_1, c_1, c_2, \cdots, c_{(n-1)})^T = A_1(k_1, r_1, r_2 \cdots, r_{(n-1)})^T$
- U_1 calculates $(d_1P, c_1P, c_2P, \cdots, c_{(n-1)}P)^T$. $j = 1, 2, \cdots n, j \neq i$.
- U_1 makes $n-1$ shares $c_1P, c_2P, \cdots, c_{(n-1)}P$ as public and sends share d_1P secretly to $n-1$ users through secure channel.

ROUND 2: KEY RECONSTRUCTION

- Each user $U_i, i = 2, 3, \cdots, n$, can reconstruct the key k_1P by computing ${A_1}^{-1}(d_1P, c_1P, \ c_2P, \cdots, c_{(n-1)}P)^T$, this result is the same as $(k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T$.

Figure 3 describes ECSSS as follows: Suppose there are three users in the scheme. User U_1 chooses a key k_1P, and computes shares: one private share d_1P and public shares $c_{12}P$ and $c_{13}P$. User U_1 sends private share to U_2, U_3 through a secure channel. Finally, U_2, U_3, reconstruct the key k_1P from d_1P and $c_{12}P$ and $c_{13}P$.

1) CORRECTNESS OF THE KEY RECONSTRUCTION

Each user $U_i, \ i = 2, 3, \cdots, n$ has n shares: one private share d_1P,and $n-1$ public shares $c_1P, \ c_2P, \cdots, c_{(n-1)}P$. The user U_i computes ${A_1}^{-1}$ and multiply with shares as $(d_1P, c_1P, c_2P, \cdots, c_{(n-1)}P)$. i.e

$$\begin{aligned} &{A_1}^{-1}(d_1P, c_1P, \ c_2P, \cdots, c_{(n-1)}P)^T. \\ &= {A_1}^{-1}(A_1(k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T) \\ &= I(k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T \\ &= (k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T \end{aligned}$$

where $I = {A_1}^{-1}(A_1)$ is identity matrix of order n. Therefore each user $U_i, i = 2, 3, \cdots, n$ can reconstruct the key.

Note 1: Each user U_i has shares $d_1P, c_1P, \ c_2P, \cdots, c_{(n-1)}P$. That is, $A_1(k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T = (d_1P, c_1P, c_2P, \cdots, c_{(n-1)}P)^T$.

Next, we propose an Authenticated distributed group key agreement protocol using Elliptic curve secret sharing scheme using (ECSSS), which uses public channels for share distribution and gives authentication to the group key.

B. PROPOSED AUTHENTICATED DISTRIBUTED GROUP KEY AGREEMENT PROTOCOL USING ELLIPTIC CURVE SECRET SHARING SCHEME (ADGKAP)

The goal of the scheme is to create a group key S among n users, $U_1, U_2, \cdots, U_n$, collaboratively using public channel. This scheme consists of three rounds: Secret distribution, Key reconstruction, and Authentication.

ROUND 1: SECRET DISTRIBUTION

- Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_q$, q is prime power.
- Point $P \in E(\mathbb{F}_q)$ and $G = \langle P \rangle$ be a group of order ℓ, where ℓ is prime.
- Any user $U_i, \ i = 1, 2, \cdots, n$, can make E, q, P and ℓ public.
- Each user U_i chooses a $(2n-2) \times n$ matrix

$$A_i = \begin{bmatrix} 1 & z_{i1} & z_{i1}^2 & \cdots & z_{i1}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_{i(n-1)} & z_{i(n-1)}^2 & \cdots & z_{i(n-1)}^{n-1} \\ 1 & z_{in} & z_{in}^2 & \cdots & z_{in}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_{i(2n-2)} & z_{i(2n-2)}^2 & \cdots & z_{i(2n-2)}^{n-1} \end{bmatrix}$$

  where $z_{im} \neq z_{ik}$ for $m \neq k$, $z_{im}, z_{ik} \in [1, \ell - 1]$, for $1 \leq i \leq n$ and $1 \leq m, k \leq (2n-2)$, and U_i makes A_i public.
- Each U_i chooses $r_{ij}, k_i \in [1, \ell - 1]$ randomly, $j = 1, 2, \cdots, n-1$.
- U_i computes $d_{i1}, d_{i2}, \cdots, d_{i(i-1)}, d_{i(i+1)}, \cdots, d_{in}$, and $c_{i1}, c_{i2}, \cdots, c_{i(i-1)}, c_{i(i+1)}, \cdots, c_{in}$ as $(d_{i1}, d_{i2}, \cdots, d_{i(i-1)}, d_{i(i+1)}, \cdots, d_{in}, c_{i1}, c_{i2}, \cdots, c_{i(i-1)}, c_{i(i+1)}, \cdots, c_{in})^T = A_i(k_i, r_{i1}, \cdots, r_{i(n-1)})^T$.
- U_i computes key k_iP.
- U_i computes $d_{i1}P, d_{i2}P, \cdots, d_{i(i-1)}P, d_{i(i+1)}P, \cdots, d_{in}P$, and $c_{i1}P, c_{i2}P, \cdots, c_{i(i-1)}P, c_{i(i+1)}P, \cdots, c_{in}P$.
- U_i makes $c_{i1}P, c_{i2}P, \cdots, c_{i(i-1)}P, c_{i(i+1)}P, \cdots, c_{in}P$ as public.
- Each user U_i chooses a private key $v_i \in [1, \ell - 1]$ and makes v_iP public.
- User U_i computes encrypted share $b_{ij}P$ as $b_{ij}P = d_{ij}P + v_iv_jP$ and sends $b_{ij}P$ publicly to the user $U_j, j = 1, 2, \cdots n, j \neq i$.

- User U_j will get the private share $d_{ij}P$ of U_i by computing $d_{ij}P = b_{ij}P - v_jv_iP$.

Note 2: $b_{ij}P$ are encrypted shares and $c_{ij}P$ are public shares.

ROUND 2: KEY RECONSTRUCTION

- Each user U_i reconstructs the key of $U_j, j = 1, 2, \cdots, n$ and $j \neq i$, as $M_j{}^{-1}(d_{ji}P, c_{j1}P, \ c_{j2}P, \cdots, c_{jn}P)$, this result is same as, $(k_jP, r_{j1}P, r_{j2}P, \cdots, r_{j(n-1)}P)$ where

$$M_j = \begin{bmatrix} 1 & z_{ji} & {z_{ji}}^2 & \cdots & {z_{ji}}^{n-1} \\ 1 & z_{jn} & z_{jn}^2 & \cdots & z_{jn}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_{j(2n-2)} & z_{j(2n-2)}^2 & \cdots & z_{j(2n-2)}^{n-1} \end{bmatrix}_{n\times n}$$

  is a submatrix of A_j corresponding to the shares $d_{ji}P, c_{j1}P, \ c_{j2}P, \cdots, c_{jn}P$.
  Note that M_j is a Vandermonde matrix.
- Finally, each user U_i reconstructs the group key $S = \sum_{j=1}^{n} k_jP$.

Round 3: Authentication

- Each user U_i computes $C_i = h(S)$ and makes it public, where $S = \sum_{j=1}^{n} k_jP$.
- Each user U_i computes $C_j{}' = h(S)$, for $j = 1, 2, \cdots, n$, $j \neq i$. If $C_j{}' = C_j$, the group key S is valid.

Figure 4 describes ADGKAP as follows: Assume three users are in the scheme. In round 1, user U_1 chooses a key k_1P and computes encrypted shares, $b_{12}P$ for U_2, $b_{13}P$ for U_3 and also computes public shares $c_{12}P, c_{13}P$ and sends them to both users, U_2, and U_3. Similarly, users, U_2 and U_3 chooses keys k_2P and k_3P respectively. User U_2 computes encrypted shares $b_{21}P$ for U_1 and $b_{23}P$ for U_3 and U_3 computes encrypted shares, $b_{31}P$ for U_1 and $b_{32}P$ for U_2. User U_2 computes public shares $c_{21}P, c_{23}P$ sends them to both users U_1 and U_3. User U_3 computes public shares $c_{31}P, c_{32}P$ sends them to both users U_1 and U_2. In round 2: User U_1 computes k_2P using $b_{21}P$ and public shares $c_{21}P, c_{23}P$. And also computes k_3P using $b_{31}P$ and public shares $c_{31}P, c_{32}P$. Similarly U_2 computes k_1P and k_3P, and U_3 computes k_1P and k_2P. Finally, users, U_1, U_2, and,U_3 can compute the group key as $S = k_1P + k_2P + k_3P$.

1) CORRECTNESS OF THE GROUP KEY RECONSTRUCTION

Each user U_i has $n-1$ private shares $d_{i1}P, d_{i2}P, \cdots, d_{i(i-1)}P, d_{i(i+1)}P, \cdots, d_{in}P$, and public shares $c_{j1}P, c_{j2}P, \cdots, c_{j(i-1)}P, c_{j(i+1)}P, \cdots, c_{jn}P$, for $j = 1, 2, \cdots, n, j \neq i$.

User U_i first computes $M_j{}^{-1}, j = 1, 2, \cdots, n$ and $j \neq i$. Then the user computes

$$\begin{aligned} &M_j{}^{-1}(d_{ji}P, c_{j1}P, \cdots, c_{j(i-1)}P, c_{j(i+1)}P, \cdots, c_{jn}P)^T. \\ &= M_j{}^{-1}(M_j(k_jP, r_{j1}P, \cdots, r_{j(n-1)}P)^T) \\ &= I(k_jP, r_{j1}P, \cdots, r_{j(n-1)}P)^T \\ &= (k_jP, r_{j1}P, \cdots, r_{j(n-1)}P)^T \end{aligned}$$

where $I = M_j{}^{-1}(M_j)$ is the identity matrix of order n.

Finally, user U_i computes group key $S = \sum_{j=1}^{n} k_jP$.

Note 3: $M_j(k_j, r_{j1}, \cdots, r_{j(n-1)})T$ represents corresponding rows in $A_i(k_i, r_{i1}, \cdots, r_{i(n-1)})^T$.

C. NUMERICAL EXAMPLE

- Let $E : y^2 = x^3 + 11x + 12$ be an elliptic curve over $\mathbb{F}_{467}$.
- Point $P = (360, 185) \in E(\mathbb{F}_{467})$ and $G = \langle P \rangle$ be a group of order $\ell = 79$.
- Any user $U_i, i = 1, 2, 3$, can make E, q, P and ℓ public.
- We discuss in detail the secret distribution (round 1), key reconstruction (round 2), and authentication (round 3) in Tables 1, 2, and 3, respectively.

Table 1 is described as follows: In step 1, each user U_i, $i = 1, 2, 3$ chooses a matrix of order 4×3 and makes it public. In step 2, each user U_i, $i = 1, 2, 3$, chooses three random integers $k_i, r_{i1}, r_{i2} \in [1, 78]$ randomly. In step 3, user U_1 computes d_{12}, d_{13}, c_{12} and c_{13}, user U_2 computes d_{21}, d_{23}, c_{21} and c_{23}, user U_3 computes d_{31}, d_{32}, c_{31} and c_{32}. In step 4, user U_i computes key k_iP for $i = 1, 2, 3$. In step 5, user U_1 computes $d_{12}P, d_{13}P, c_{12}P, c_{13}P$ and makes $c_{12}P$ and $c_{13}P$ public, user U_2 computes $d_{21}P, d_{23}P, c_{21}P, \ c_{23}P$ and makes $c_{21}P$ and $c_{23}P$ public, user U_3 computes $d_{31}P, d_{32}P, c_{31}P, \ c_{32}P$ and makes $c_{31}P$ and $c_{32}P$ public. In step 6, user $U_i, i = 1, 2, 3$ chooses private key v_i and computes public key v_iP. In step 7, user U_1 computes $b_{12}P$ and $b_{13}P$, user U_2 computes $b_{21}P$ and $b_{23}P$, user U_3 computes $b_{31}P$ and $b_{32}P$. Then the users make $b_{12}P, b_{13}Pb_{21}P, b_{23}P, b_{31}P, b_{32}P$ public. In step 8, user U_1 computes his shares $d_{21}P$ and $d_{31}P$ from v_2P and v_3P respectively, user U_2 computes his from shares $d_{12}P$ and $d_{32}P$ from v_1P and v_3P respectively, user U_3 computes his shares $d_{13}P$ and $d_{23}P$ from v_1P and v_2P respectively.

Table 2 is described as follows: In step 1, user U_1 computes $k_2P, r_{21}P, r_{22}P$ from M_2 and $d_{21}P, c_{21}P, c_{23}P$, user U_2 computes $k_1P, r_{11}P, r_{12}P$ from M_1 and $d_{12}P, c_{12}P, c_{13}P$, user U_3 computes $k_1P, r_{11}P, r_{12}P$ from M_1 and $d_{13}P, c_{12}P, c_{13}P$. In step 2, user U_1 computes $k_3P, r_{31}P, r_{32}P$ from M_3 and $d_{31}P, c_{31}P, c_{32}P$, user U_2 computes $k_3P, r_{31}P, r_{32}P$ from M_3 and $d_{32}P, c_{31}P, c_{32}P$. user U_3 computes $k_2P, r_{21}P, r_{22}P$ from M_2 and $d_{23}P, c_{21}P, c_{22}P$. In step 3,user $U_i, i = 1, 2, 3$ computes group key $S = k_1P + k_2P + k_3P$.

Table 3 is described as follows: In step 1, user U_i computes $C_i, i = 1, 2, 3$ and makes it public. In step 2, user U_1 computes $C_2{}^{-1}, C_3{}^{-1}$, then compares C_2 with $C_2{}'$ and C_3 with $C_3{}'$, if they are equal, group key is valid. Similarly, U_2 and U_3 verify if their group key is valid or not.

V. SECURITY ANALYSIS

In this section, we discussed in detail that the user could reconstruct the group key, but an attacker can not reconstruct the group key in detail.

TABLE 1. Secret distribution: Round 1.

Steps	User U_1	User U_2	User U_3
1	U_1 chooses 4×3 matrix $A_1 = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 5 & 25 \\ 1 & 3 & 9 \\ 1 & 6 & 36 \end{bmatrix}$ A_1 is public	U_2 chooses 4×3 matrix $A_2 = \begin{bmatrix} 1 & 3 & 9 \\ 1 & 7 & 49 \\ 1 & 4 & 16 \\ 1 & 2 & 4 \end{bmatrix}$ A_2 is public	U_3 chooses 4×3 matrix $A_3 = \begin{bmatrix} 1 & 8 & 64 \\ 1 & 4 & 16 \\ 1 & 9 & 81 \\ 1 & 10 & 100 \end{bmatrix}$ A_3 is public
2	U_1 chooses randomly $k_1 = 14, r_{11} = 32, r_{12} = 27$	U_2 chooses randomly $k_2 = 19, r_{21} = 21, r_{22} = 35$	U_3 chooses randomly $k_3 = 48, r_{31} = 61, r_{32} = 12$
3	U_1 computes $\begin{bmatrix} d_{12} \\ d_{13} \\ c_{12} \\ c_{13} \end{bmatrix} = A_1 \begin{bmatrix} 14 \\ 32 \\ 27 \end{bmatrix} = \begin{bmatrix} 186 \\ 849 \\ 353 \\ 1178 \end{bmatrix}$	U_2 computes $\begin{bmatrix} d_{21} \\ d_{23} \\ c_{21} \\ c_{23} \end{bmatrix} = A_2 \begin{bmatrix} 19 \\ 21 \\ 35 \end{bmatrix} = \begin{bmatrix} 397 \\ 1881 \\ 663 \\ 201 \end{bmatrix}$	U_3 computes $\begin{bmatrix} d_{31} \\ d_{32} \\ c_{31} \\ c_{32} \end{bmatrix} = A_3 \begin{bmatrix} 48 \\ 61 \\ 12 \end{bmatrix} = \begin{bmatrix} 1304 \\ 484 \\ 1569 \\ 1858 \end{bmatrix}$
4	U_1 computes Key $k_1P = (99, 290)$	U_2 computes Key $k_2P = (116, 97)$	U_3 computes Key $k_3P = (410, 375)$
5	$\begin{bmatrix} d_{12}P \\ d_{13}P \\ c_{12}P \\ c_{13}P \end{bmatrix} = \begin{bmatrix} (424, 193) \\ (21, 346) \\ (108, 135) \\ (288, 416) \end{bmatrix}$ $c_{12}P, c_{13}P$ make public	$\begin{bmatrix} d_{21}P \\ d_{23}P \\ c_{21}P \\ c_{23}P \end{bmatrix} = \begin{bmatrix} (435, 7) \\ (11, 159) \\ (410, 92) \\ (275, 427) \end{bmatrix}$ $c_{21}P, c_{23}P$ make public	$\begin{bmatrix} d_{31}P \\ d_{32}P \\ c_{31}P \\ c_{32}P \end{bmatrix} = \begin{bmatrix} (88, 50) \\ (316, 252) \\ (221, 195) \\ (387, 157) \end{bmatrix}$ $c_{31}P, c_{32}P$ make public
6	U_1 chooses Private key $v_1 = 24$ Public key $v_1P = (261, 186)$	U_2 chooses Private key $v_2 = 57$ Public key $v_2P = (133, 124)$	U_3 chooses Private key $v_3 = 63$ Public key $v_3P = (121, 391)$
7	U_1 computes encrypted shares $b_{12}P = d_{12}P + v_1v_2P$ $\implies b_{12}P = (394, 229)$ $b_{13}P = d_{13}P + v_1v_3P$ $\implies b_{13}P = (24, 435)$ $b_{12}P, b_{13}P$ make public	U_2 computes encrypted shares $b_{21}P = d_{21}P + v_2v_1P$ $\implies b_{21}P = (220, 253)$ $b_{23}P = d_{23}P + v_2v_3P$ $\implies b_{23}P = (99, 177)$ $b_{21}P, b_{23}P$ make public	U_3 computes encrypted shares $b_{31}P = d_{31}P + v_3v_1P$ $\implies b_{31}P = (424, 274)$ $b_{32}P = d_{32}P + v_3v_2P$ $\implies b_{32}P = (250, 169)$ $b_{31}P, b_{32}P$ make public
8	U_1 computes shares $d_{21}P = b_{21}P - v_1v_2P$ $\implies d_{21}P = (435, 7)$ $d_{31}P = b_{31}P - v_1v_3P$ $\implies d_{31}P = (88, 50)$	U_2 computes shares $d_{12}P = b_{12}P - v_2v_1P$ $\implies d_{12}P = (424, 193)$ $d_{32}P = b_{32}P - v_2v_3P$ $\implies d_{32}P = (316, 252)$	U_3 computes shares $d_{13}P = b_{13}P - v_3v_1P$ $\implies d_{13}P = (21, 346)$ $d_{23}P = b_{23}P - v_3v_2P$ $\implies d_{23}P = (11, 159)$

TABLE 2. Key reconstruction: Round 2.

Steps	U_1	U_2	U_3
1	$M_2 = \begin{bmatrix} 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 2 & 4 \end{bmatrix}$ $M_2{}^{-1}(d_{21}P, c_{21}P, c_{23}P)^T$ $= \begin{bmatrix} (116, 97) \\ (203, 437) \\ (285, 426) \end{bmatrix} = \begin{bmatrix} k_2P \\ r_{21}P \\ r_{22}P \end{bmatrix}$	$M_1 = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 6 & 36 \end{bmatrix}$ $M_1{}^{-1}(d_{12}P, c_{12}P, c_{13}P)^T$ $= \begin{bmatrix} (99, 290) \\ (140, 153) \\ (220, 253) \end{bmatrix} = \begin{bmatrix} k_1P \\ r_{11}P \\ r_{12}P \end{bmatrix}$	$M_1 = \begin{bmatrix} 1 & 5 & 25 \\ 1 & 3 & 9 \\ 1 & 6 & 36 \end{bmatrix}$ $M_1{}^{-1}(d_{13}P, c_{12}P, c_{13}P)^T$ $= \begin{bmatrix} (99, 290) \\ (140, 153) \\ (220, 253) \end{bmatrix} = \begin{bmatrix} k_1P \\ r_{11}P \\ r_{12}P \end{bmatrix}$
2	$M_3 = \begin{bmatrix} 1 & 8 & 64 \\ 1 & 9 & 81 \\ 1 & 10 & 100 \end{bmatrix}$ $M_3{}^{-1}(d_{31}P, c_{31}P, c_{32}P)^T$ $= \begin{bmatrix} (410, 375) \\ (121, 391) \\ (163, 367) \end{bmatrix} = \begin{bmatrix} k_3P \\ r_{31}P \\ r_{32}P \end{bmatrix}$	$M_3 = \begin{bmatrix} 1 & 4 & 16 \\ 1 & 9 & 81 \\ 1 & 10 & 100 \end{bmatrix}$ $M_3{}^{-1}(d_{32}P, c_{31}P, c_{32}P)^T$ $= \begin{bmatrix} (410, 375) \\ (121, 391) \\ (163, 367) \end{bmatrix} = \begin{bmatrix} k_3P \\ r_{31}P \\ r_{32}P \end{bmatrix}$	$M_2 = \begin{bmatrix} 1 & 7 & 49 \\ 1 & 4 & 16 \\ 1 & 2 & 4 \end{bmatrix}$ $M_2{}^{-1}(d_{23}P, c_{21}P, c_{23}P)^T$ $= \begin{bmatrix} (116, 97) \\ (203, 437) \\ (285, 426) \end{bmatrix} = \begin{bmatrix} k_2P \\ r_{21}P \\ r_{22}P \end{bmatrix}$
3	Group key $S = \sum_{j=1}^{3} k_jP = (435, 7)$	Group key $S = \sum_{j=1}^{3} k_jP = (435, 7)$	Group key $S = \sum_{j=1}^{3} k_jP = (435, 7)$

A. USER U_i CAN RECONSTRUCT GROUP KEY S BY USING $n-1$ PUBLIC SHARES

Suppose the user U_i wants to compute $k_jP, j = 1, 2, \cdots, n$, $j \neq i$, the secret point of U_j with the help of the private share $d_{ji}P$ and $n-1$ public shares. The user U_i can form a system of equations as follows

$$\begin{aligned} k_jP + r_{j1}z_{ji}P + \cdots + r_{j(n-1)}z_{ji}^{n-1}P &= d_{ji}P \\ k_jP + r_{j1}z_{jn}P + \cdots + r_{j(n-1)}z_{jn}^{n-1}P &= c_{j1}P \\ k_jP + r_{j1}z_{j(n+1)}P + \cdots + r_{j(n-1)}z_{j(n+1)}^{n-1}P &= c_{j2}P \\ &\vdots \\ k_jP + r_{j1}z_{j(2n-2)}P + \cdots + r_{j(n-1)}z_{j(2n-2)}^{n-1}P &= c_{jn}P \end{aligned}$$

The same matrix form can be represented as

$$\begin{bmatrix} 1 & z_{ji} & z_{ji}^2 & \cdots & z_{ji}^{n-1} \\ 1 & z_{jn} & z_{jn}^2 & \cdots & z_{jn}^{n-1} \\ 1 & z_{j(n+1)} & z_{j(n+1)}^2 & \cdots & z_{j(n+1)}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_{j(2n-2)} & z_{j(2n-2)}^2 & \cdots & z_{j(2n-2)}^{n-1} \end{bmatrix} \begin{bmatrix} k_jP \\ r_{j1}P \\ r_{j2}P \\ \vdots \\ r_{jn}P \end{bmatrix} = \begin{bmatrix} d_{ji}P \\ c_{j1}P \\ c_{j2}P \\ \vdots \\ c_{jn}P \end{bmatrix}$$

The above coefficient matrix of the system has n unknowns and n equations, as the rank of the coefficient matrix is n. Thus the system of equations has a unique solution. Hence a user can get key k_jP, $j = 1, 2 \cdots n$, by inverting the coefficient matrix and multiplying it with the share matrix $(d_{ji}P, c_{j1}P, c_{j2}P, \cdots, c_{jn}P)^T$. Finally, the user U_i computes the group key $S = \sum_{j=1}^{n} k_jP$, using his secret key k_iP.

B. AN ATTACKER CANNOT RECONSTRUCT GROUP KEY S by USING $n-1$ PUBLIC SHARES

Suppose an attacker wants to get k_iP with $n-1$ public shares. The attacker can form a system of equations as follows.

$$\begin{aligned} k_P + r_{i1}z_{in}P + \cdots + r_{i(n-1)}{z_{in}}^{n-1}P &= c_{i1}P \\ k_iP + r_{i1}z_{i(n+1)}P + \cdots + r_{i(n-1)}z_{i(n+1)}^{n-1}P &= c_{i2}P \\ &\vdots \\ k_iP + r_{i1}z_{i(n+i-2)}P + \cdots + r_{i(n-1)}z_{i(n+i-2)}^{n-1}P &= c_{i(i-1)}P \end{aligned}$$

TABLE 3. Key authentication: Round 3.

Steps	U_1	U_2	U_3
1	$h(S) = 34$ $C_1 = 34$ is public	$h(S) = 34$ $C_2 = 34$ is public	$h(S) = 34$ $C_3 = 34$ is public
2	$C_2' = 34$ $C_3' = 34$ $C_2' = C_2$ $C_3' = C_3$ Group key S is Valid	$C_1' = 34$ $C_3' = 34$ $C_1' = C_1$ $C_3' = C_3$ Group key S is Valid	$C_1' = 34$ $C_2' = 34$ $C_1' = C_1$ $C_2' = C_2$ Group key S is Valid

$$k_iP + r_{i1}z_{i(n+i)}P + \cdots + r_{i(n-1)}z_{i(n+i)}^{n-1}P = c_{i(i+1)}P$$
$$\vdots$$
$$k_iP + r_{i1}z_{i(2n-2)}P + \cdots + r_{i(n-1)}z_{i(2n-2)}^{n-1}P = c_{in}P$$

The same matrix form can be represented as

$$\begin{bmatrix} 1 & z_{in} & \cdots & z_{in}^{n-1} \\ 1 & z_{i(n+1)} & \cdots & z_{i(n+1)}^{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & z_{i(n+i-2)} & \cdots & z_{i(n+i-2)}^{n-1} \\ 1 & z_{i(n+i)} & \cdots & z_{i(n+i)}^{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & z_{i(2n-2)} & \cdots & z_{i(2n-2)}^{n-1} \end{bmatrix} \begin{bmatrix} k_iP \\ r_{i1}P \\ \vdots \\ r_{i(n-1)}P \end{bmatrix}$$
$$= \begin{bmatrix} c_{i1}P \\ c_{i2}P \\ \vdots \\ c_{i(i-1)}P \\ c_{i(i+1)}P \\ \vdots \\ c_{in}P \end{bmatrix}$$

The above coefficient matrix of the system has n unknowns and only $n-1$ equations. Thus the system of equations has finitely many solutions, and each solution is of equal probability of $\frac{1}{\ell^n}$. For large ℓ, $\frac{1}{\ell^n}$ is negligible. Thus an attacker cannot get key k_iP and hence $S = \sum_{i=1}^{n} k_iP$.

VI. COMPARISONS

This section compares our proposed scheme, Authenticated distributed group key agreement protocol using Elliptic Curve Secret Sharing Scheme (ADGKAP) with Harn and Liu [1] on various parameters such as Key size, a technique used, no.of rounds and so on. In literature, Shanks' and Index calculus algorithms [54] are popular for solving discrete logarithm algorithms in sub exponentiation time. However, till date, there are no such algorithms for ECDLP [57].

Our ADGKAP scheme's security relies on ECDLP(Elliptic Curve Discrete Logarithm Problem), but Harn et al. rely only on DLP (Discrete Logarithm Problem). It is widely known that ECDLP is more secure compared to DLP, hence our scheme is more secure comparatively. Harn et al. worked on the finite field $\mathbb{F}_q$, but our ADGKAP scheme worked on the elliptic curve over a finite field $E(\mathbb{F}_q)$. This ensures that our ADGKAP scheme, key size, and storage space are significantly less and have fast computation compared to Harn and Liu [1]. A comparison of Harn et al. with our proposed ADGKAP scheme with respect to various computational efficiency parameters is listed in Table 4.

TABLE 4. Comparison of computation parameters between Harn Scheme and ADGKAP scheme.

Parameters	Harn et al. [1]	Our ADGKAP
Key Size(in bits)	1024,2048,3072	160,224,256
Hard problem	DLP	ECDLP
Crypto technique	Polynomial SSS	Elliptic curve SSS
Rounds	3	3
Attack	DLP attacks(Index calculus,Shanks)	No ECDLP attack

In Cheng et al. [49] and Zhang et al. [48] schemes, the key is distributed to a group of participants using an encryption mechanism. Decryption relies on a single key making it less secure because of a single point of failure. However, our proposed ADGKAP scheme is more secure because shares of the key are distributed to a group of participants instead of a key. Also, all of these use a secure channel for communication among users. But our ADGKAP uses public channel for communication among participants. Li et al. [61] uses $8n+6$ exponentiation, Zhang et al. [60] uses $(2n+10)$ exponentiation, Cao et al. [32] uses 2 bilinear parings and Zhang et al. [59] uses 16 exponentiation. All these four schemes use a secure channel for communication between KGC and users. Cheng et al. [49] uses $(2n+2)$ bilinear operations, and Zhang et al. [48] uses 5 bilinear operations. Both Cheng et al. [49], and Zhang et al. [48] protocols were designed by using no KGC. Harn et al. used $2n$ exponentiation and a secret sharing scheme without KGC. However, our scheme uses n^2 scalar multiplications only. The cost of bilinear parings and exponentiation are costly operations compared to scalar multiplications [48]. Hence ADGKAP has less computational cost compared to existing schemes. Cost comparisons among various schemes are listed in table 5.

In our proposed scheme, the matrix A_i and inverse of the matrix M_j of user $U_j, j = 1, 2, \cdots, n$ are pre computed and made public by user U_i. Harn et al. require, (n^2+n-2) additions, multiplications $(6n^2-2n-4)$, $(2n)$ divisions, $(2n)$

TABLE 5. Cost comparisons among various schemes.

Author	KGC	SSS	Channel	Total cost
Li J et al. [61]	Yes	No	Secure	$(8n + 6)$ exponentiation
Zhang L et. al [59]	Yes	No	Secure	16 exponentiation
Zhang Q et al. [60]	Yes	No	Secure	$(2n + 10)$ exponentiation
Cheng et al. [49]	No	No	Secure	$(2n + 2)$ bilinear pairings
Qikum et al. [48]	No	No	Secure	5 bilinear pairings
Cao et al. [32]	Yes	Yes	Secure	2 bilinear pairings
Harn et al. [1]	No	Yes	Public	$2n$ exponentiation
ADGKAP scheme	No	Yes	Public	n^2 scalar multiplications

TABLE 6. Cost comparison between Harn Scheme and ADGKAP scheme.

Operations	Harn Scheme [1]	ADGKAP scheme
Additions	$O(n^2)$	Nil
Multiplications	$O(n^2)$	$O(n^2)$
Divisions	$O(n)$	Nil
Exponentiation	$O(n)$	Nil
Point additions	Nil	$O(n^2)$
Scalar multiplications	Nil	$O(n^2)$
Hashing	$O(n)$	$O(n)$

exponentiation and $(3n)$ hashing . However, our proposed scheme (ADGKAP) requires $(2n^2 - 2n)$ multiplications, $(3n^2 - 3n)$ point additions, $(3n^2 + n - 1)$ scalar multiplications and (n) hashing. A comparison of the computational cost between the Harn scheme and our proposed ADGKAP scheme is given in table 6. The observation from the table is that the costlier operation exponentiation is avoided in our scheme and replaced by point additions and scalar multiplications that deals with smaller key size with an equal level of security. This reduces the computational cost of our scheme very much compared to Harn's scheme. This makes our scheme efficient in terms of computational cost and key size, which makes it a better choice for resource constrained environments. Hence our proposed ADGKAP scheme gives similar security with a smaller key size.

VII. CONCLUSION AND FUTURE WORK

We propose a novel Elliptic curve secret sharing scheme (ECSSS) for share distribution that is secure enough with a relatively smaller key size and storage. Then, an Authenticated distributed group key agreement Protocol using Elliptic curve secret sharing scheme (ECSSS) is proposed. The proposed scheme can be used efficiently in a distributed environment for group key agreement. In comparison to existing schemes, our ADGKAP offers equal security with much smaller key sizes, less storage space, and less computational cost without compromising on the number of rounds. Every user in this scheme can reconstruct the group key, but the attacker cannot do so. ECDLP is a major aspect of this scheme's security. Compared to the existing scheme, Authenticated group Diffie–Hellman key agreement protocol our proposed scheme ADGKAP is more appropriate for resource constrained devices in a distributed environment.

In the future, we will extend our work to distributed multi-group key agreement protocol based on an elliptic curve secret sharing scheme.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] L. Harn and C. Lin, "Efficient group Diffie–Hellman key agreement protocols," *Comput. Electr. Eng.*, vol. 40, no. 6, pp. 1972–1980, Aug. 2014.

[2] M. Ben-Or and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 1–10.

[3] W.-G. Tzeng and Z.-J. Tzeng, "Round-efficient conference key agreement protocols with provable security," in *Proc. ASIACRYPT*, in Lecture Notes in Computer Science, vol. 1976, 2000, pp. 614–627.

[4] W.-G. Tzeng, "A secure fault-tolerant conference-key agreement protocol," *IEEE Trans. Comput.*, vol. 51, no. 4, pp. 373–379, Apr. 2002.

[5] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 950, 1995, pp. 275–286.

[6] I. Ingemarsson, D. Tang, and C. Wong, "A conference key distribution system," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 5, pp. 714–720, Sep. 1982.

[7] D. G. Steer, L. Strawczynski, W. Diffie, and M. Wiener, "A secure audio teleconference system," in *Proc. CRYPTO*, in Lecture Notes in Computer Science, vol. 403, 1988, pp. 520–528.

[8] M. Steiner, G. Tsudik, and M. Waidner, "Diffie–Hellman key distribution extended to group communication," in *Proc. 3rd ACM Conf. Comput. Commun. Secur. (CCS)*, 1996, pp. 31–37.

[9] P. P. C. Lee, J. C. S. Lui, and D. K. Y. Yau, "Distributed collaborative key agreement and authentication protocols for dynamic peer groups," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 263–276, Apr. 2006.

[10] Y. Kim, A. Perrig, and G. Tsudik, "Group key agreement efficient in communication," *IEEE Trans. Comput.*, vol. 53, no. 7, pp. 905–921, Jul. 2004.

[11] A. Shoufan and S. A. Huss, "High-performance rekeying processor architecture for group key management," *IEEE Trans. Comput.*, vol. 58, no. 10, pp. 1421–1434, Oct. 2009.

[12] E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater, "Provably authenticated group Diffie–Hellman key exchange," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, 2001, pp. 255–264.

[13] J.-M. Bohli, "A framework for robust group key agreement," in *Proc. Int. Conf. Comput. Sci. Appl. (ICCSA)*, in Lecture Notes in Computer Science, vol. 3982, 2006 pp. 355–364.

[14] E. Bresson, O. Chevassut, and D. Pointcheval, "Provably-secure authenticated group Diffie–Hellman key exchange," *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 3, pp. 255–264, 2007.

[15] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," *J. Cryptol.*, vol. 20, no. 1, pp. 85–113, Jan. 2007.

[16] T. Brecher, E. Bresson, and M. Manulis, "Fully robust tree-Diffie–Hellman group key exchange," in *Proc. 8th Int. Conf. Cryptol. Netw. Secur. (CANS)*, in Lecture Notes in Computer Science, vol. 5888, 2009, pp. 478–497.

[17] S. Jarecki, J. Kim, and G. Tsudik, "Flexible robust group key agreement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 879–886, May 2011.

[18] A. Joux, "A one round protocol for tripartite Diffie–Hellman," in *Proc. ANTS*, in Lecture Notes in Computer Science, vol. 1838, 2000, pp. 385–394.

[19] R. Barua, R. Dutta, and P. Sarkar, "Extending Joux's protocol to multi party key agreement," in *Proc. INDOCRYPT*, in Lecture Notes in Computer Science, vol. 2904, 2003, pp. 205–217.

[20] R. Barua, R. Dutta, and P. Sarkar, "Provably secure authenticated tree based group key agreement protocol using pairing," in *Proc. ICICS*, in Lecture Notes in Computer Science, vol. 3269, 2004, pp. 92–104.

[21] K. Y. Choi, J. Y. Hwang, and D. H. Lee, "Efficient ID-based group key agreement with bilinear maps," in *Proc. PKC*, in Lecture Notes in Computer Science, vol. 2947, 2004, pp. 130–144.

[22] X. Du, Y. Wang, J. Ge, and Y. Wang, "An improved ID-based authenticated group key agreement scheme," ePrint Arch., Tech. Rep. 2003/260, 2003.

[23] K.-A. Shim, "A round-optimal three-party ID-based authenticated key agreement protocol," *Inf. Sci.*, vol. 186, no. 1, pp. 239–248, Mar. 2012.

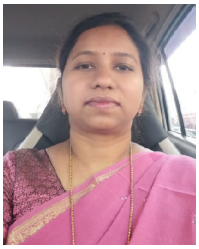
[24] Y. Desmedt and T. Lange, "Revisiting pairing based group key exchange," in *Proc. FC*, in Lecture Notes in Computer Science, vol. 5143, 2008, pp. 53–68.

[25] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer, "Asymmetric group key agreement," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 5479, 2009, pp. 153–170.

[26] X. Gu, Y. Zhao, and J. Yang, "Reducing rekeying time using an integrated group key agreement scheme," *J. Commun. Netw.*, vol. 14, no. 4, pp. 418–428, Aug. 2012.

[27] E. Konstantinou, "An efficient constant round ID-based group key agreement protocol for ad hoc networks," in *Proc. NSS*, in Lecture Notes in Computer Science, vol. 7873, 2013, pp. 563–574.
[28] C. Laih, J. Lee, and L. Harn, "A new threshold scheme and its application in designing the conference key distribution cryptosystem," *Inf. Process. Lett.*, vol. 32, pp. 95–99, Aug. 1989.
[29] S. Berkovits, "How to broadcast a secret," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 547, 1991, pp. 536–541.
[30] C.-H. Li and J. Pieprzyk, "Conference key agreement from secret sharing," in *Proc. 4th Australas. Conf. Inf. Secur. Privacy (ACISP)*, in Lecture Notes in Computer Science, vol. 1587, 1999, pp. 64–76.
[31] G. Sáez, "Generation of key predistribution schemes using secret sharing schemes," *Discrete Appl. Math.*, vol. 128, no. 1, pp. 239–249, May 2003.
[32] C. Cao, C. Yang, J. Ma, and S. Moon, "Constructing UC secure and constant-round group key exchange protocols via secret sharing," *EURASIP J. Wireless Commun. Netw.*, vol. 2008, no. 1, pp. 1–9, Dec. 2008.
[33] L. Harn and C. Lin, "Authenticated group key transfer protocol based on secret sharing," *IEEE Trans. Comput.*, vol. 59, no. 6, pp. 842–846, Jun. 2010.
[34] H. Sun, Q. Wen, H. Zhang, and Z. Jin, "A novel pairing-free certificateless authenticated key agreement protocol with provable security," *Frontiers Comput. Sci.*, vol. 7, no. 4, pp. 544–557, Aug. 2013.
[35] D. Liu, D. Huang, P. Luo, and Y. Dai, "New schemes for sharing points on an elliptic curve," *Comput. Math. Appl.*, vol. 56, no. 6, pp. 1556–1561, Sep. 2008.
[36] V. P. Binu and A. Sreekumar, "Threshold multi secret sharing using elliptic curve and pairing," Mar. 2016, *arXiv:1603.09524*.
[37] S.-J. Wang, Y.-R. Tsai, and J.-J. Shen, "Dynamic threshold multi-secret sharing scheme using elliptic curve and bilinear maps," in *Proc. 2nd Int. Conf. Future Gener. Commun. Netw.*, Dec. 2008, pp. 405–410.
[38] H. Debiao, C. Jianhua, and H. Jin, "An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security," *Inf. Fusion*, vol. 13, no. 3, pp. 223–230, Jul. 2012.
[39] S. H. Islam and G. P. Biswas, "An improved pairing-free identity-based authenticated key agreement protocol based on ECC," *Proc. Eng.*, vol. 30, pp. 499–507, Jan. 2012.
[40] Y. Liu, Q. Sun, Y. Wang, L. Zhu, and W. Ji, "Efficient group authentication in RFID using secret sharing scheme," *Cluster Comput.*, vol. 22, no. S4, pp. 8605–8611, Jul. 2019.
[41] H.-Y. Chien, "Elliptic curve cryptography-based RFID authentication resisting active tracking," *Wireless Pers. Commun.*, vol. 94, no. 4, pp. 2925–2936, Jun. 2017.
[42] M. Sheikhi-Garjan, M. Bahramian, and C. Doche, "Threshold verifiable multi-secret sharing based on elliptic curves and Chinese remainder theorem," *IET Inf. Secur.*, vol. 13, no. 3, pp. 278–284, May 2019.
[43] B. A. Alzahrani, S. A. Chaudhry, A. Barnawi, A. Al-Barakati, and M. H. Alsharif, "A privacy preserving authentication scheme for roaming in IoT-based wireless mobile networks," *Symmetry*, vol. 12, no. 2, p. 287, Feb. 2020.
[44] W. Cui, R. Cheng, K. Wu, Y. Su, and Y. Lei, "A certificateless authenticated key agreement scheme for the power IoT," *Energies*, vol. 14, no. 19, p. 6317, Oct. 2021.
[45] D.-Z. Sun, "Security and privacy analysis of Vinoth et al.'s authenticated key agreement scheme for industrial IoT," *Symmetry*, vol. 13, no. 10, p. 1952, Oct. 2021.
[46] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.
[47] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, Aug. 1985, pp. 417–426.
[48] Q. Zhang, L. Zhu, R. Wang, J. Li, J. Yuan, T. Liang, and J. Zheng, "Group key agreement protocol among terminals of the intelligent information system for mobile edge computing," *Int. J. Intell. Syst.*, vol. 37, no. 12, pp. 10442–10461, Dec. 2022.
[49] Q.-F. Cheng, C.-G. Ma, and F.-S. Wei, "Analysis and improvement of a new authenticated group key agreement in a mobile environment," *Ann. Telecommun.*, vol. 66, nos. 5–6, pp. 331–337, Jun. 2011.
[50] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future Gener. Comput. Syst.*, vol. 78, pp. 956–963, Jan. 2018.
[51] P. Dzurenda, S. Ricci, R. C. Marqués, J. Hajny, and P. Cika, "Secret sharing-based authenticated key agreement protocol," in *Proc. 16th Int. Conf. Availability, Rel. Secur.*, Aug. 2021, pp. 1–10.
[52] J.-H. Yang and C.-C. Chang, "An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Comput. Secur.*, vol. 28, nos. 3–4, pp. 138–143, May 2009.
[53] E.-J. Yoon and K.-Y. Yoo, "Robust ID-based remote mutual authentication with key agreement scheme for mobile devices on ECC," in *Proc. Int. Conf. Comput. Sci. Eng.*, vol. 2, Aug. 2009, pp. 633–640.
[54] D. Stinson, *Cryptography: Theory and Practice*. Boca Raton, FL, USA: CRC Press, Nov. 2005.
[55] A. Klinger, "The Vandermonde matrix," *Amer. Math. Monthly*, vol. 74, no. 5, pp. 571–574, May 1967.
[56] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Germany: Springer, Nov. 2009.
[57] R. Haakegaard and J. Lang. (Dec. 2015). *The Elliptic Curve Diffie–Hellman (ECDH)*. [Online]. Available: https://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Haakegaard+ Lang.pdf
[58] K. R. Raghunandan, R. Dodmane, K. Bhavya, N. S. K. Rao, and A. K. Sahu, "Chaotic-map based encryption for 3D point and 3D mesh fog data in edge computing," *IEEE Access*, vol. 11, pp. 3545–3554, 2023.
[59] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and Z. Dong, "Round-efficient and sender-unrestricted dynamic group key agreement protocol for secure group communications," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2352–2364, Nov. 2015.
[60] Q. Zhang, Y. Gan, L. Liu, X. Wang, X. Luo, and Y. Li, "An authenticated asymmetric group key agreement based on attribute encryption," *J. Netw. Comput. Appl.*, vol. 123, pp. 1–10, Dec. 2018.
[61] J. Li, Z. Qiao, and J. Peng, "Asymmetric group key agreement protocol based on blockchain and attribute for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 8326–8335, Nov. 2022.
[62] G. H. Forman and J. Zahorjan, "The challenges of mobile computing," *Computer*, vol. 27, no. 4, pp. 38–47, 1994.

ROLLA SUBRAHMANYAM received the M.Tech. degree in artificial intelligence from the University of Hyderabad, India, where he is currently pursuing the Ph.D. degree in computer science. His research interests include secret sharing schemes, cryptography, and AI-crypto.



N. RUKMA REKHA received the M.Tech. and Ph.D. degrees in computer science from Andra University. She is currently an Associate Professor with the University of Hyderabad. Her research interests include blockchain technology, cryptography, and security.

Y. V. SUBBA RAO received the M.Tech. degree from the Indian Statistical Institute (ISI), Kolkata, and the Ph.D. degree in computer science from the University of Hyderabad. He is currently an Associate Professor with the University of Hyderabad. His research interests include blockchain technology and cryptography.