Contemporary Mathematics
http://ojs.wiserpub.com/index.php/CM/

Research Article

Cryptographic Signature Scheme for Mobile Edge Computing Using DLFP Over Semiring

Sethupathi S, Manimaran A*

Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore, India
E-mail: manimaran.a@vit.ac.in

Received: December 7 2023; **Revised:** February 19 2024; **Accepted:** March 21 2024

Abstract: The Undeniable Signature Scheme (USS) was first introduced by chaum and van in 1989. The advantage of the (USS) scheme is the signer's co-operation during the verification process. In this paper, the Discrete Logarithm Problem with Factor Problem ($DLFP$) is introduced. It's security and complexity is analyzed. Then, based on ($DLFP$), a key exchange protocol is given. An undeniable signature scheme based on $DLFP$ over semiring is proposed along with illustration we have also given application of the proposed USS in Mobile Edge Computing (MEC). Then the proposed scheme's security and complexity is examined.

***Keywords*:** semiring, centralizer, discrete logarithm problem, factor problem, undeniable signature, mobile edge computing

MSC: 11T71, 06E20, 20A10

Abbreviation

CDL	Conjugacy Search Problem and Discrete Logarithm Problem
CSP	Conjugacy Search Problem
DLCSP	Discrete Logarithm Conjugacy Search Problem
DLFP	Discrete Logarithmic Factor Problem
DLP	Discrete Logarithm Problem
DP	Disavowal Protocol
DSA	Digital Signature Algorithm
EC	Edge Computing
ECDLP	Elliptic Curve Discrete Logarithm Problem
ES	Edge Server
FDLP	Factorization Discrete Logarithm Problem
FP	Factor Problem
IoT	Internet of Things
MEC	Mobile Edge Computing

DOI: https://doi.org/10.37256/cm.5220244061

PCC	Public Cloud Center
RSA	Rivest Shamir Algorithm
TD	Terminal Device
USS	Undeniable Signature Scheme
VP	Verification Protocol

1. Introduction

A digital signature technique is an essential component of any system for communication. Signatories are bound to the material they sign when using such Digital Signature techniques. The implementation of a secure signature technique ensures the integrity and validity of a message and also eliminates forgeries. Diffie-Hellman presented the notion of Digital Signature in 1976 [1], and in the meanwhile, Rivest Shamir Adleman (*RSA*)-based digital signature scheme [2] was suggested. In [3], introduced two consequences of decryption key one is consequences need not to transmit key which is publicly revealed by the intended recipient. The second consequences is can be signed using a privately held decryption key this article author used power d key system. Many signature systems, such as the Digital Signature Algorithm (*DSA*) and Elgamal's signature scheme, were presented as a result. There are group signature systems that use verifiable random numbers [4], as well as proxy-based blind signature schemes that use the Elliptic Curve Discrete Logarithm Problem (*ECDLP*) [5]. Although they offer non-repudiation and are publicly verifiable, these systems are not usually preferred for this feature.

In 1989, chaum and van introduced (*USS*) [6] and also mended the digital signature flaw for undeniable signature scheme [7]. Undeniable signature have gained much attention after being used in real world situations, when is more efficient and suitable application than digital signature.

There are three parts in the signature scheme. The first part is a signing algorithm , the second part is a Verification Protocol (*VP*) and the third part is a Disavowal Protocol (*DP*). In contrast to Digital Signature Scheme, without the consent of the signatory an *USS*'s validity cannot be verified. This restricts a third party's ability to validate the authenticity of the sign. A *DP* is a component of an *USS* that allows the signatory to contest a sign by validating that it is fake. A signatory is a legal signatory to the object, even if the signatory chooses not to participate in the rejection procedure. The disavowal protocol can be used by the verifier to determine the cause of an incorrect sign, such as whether the signatory lied during the process of verification or the sign was forgeried. The comparison between USS and digital signature schemes is given in Table 1.

Table 1. Comparison table between USS and digital signature schemes

Feature	USS	DSS
Non-repudiation	Provides non-repudiation.	Also ensues non-repudiation.
Key-dependency	Typically relies on a secret key.	Utilizes public-private key pair.
Key-management	Requires secure key management.	Involves secure key pair management.
Algorithm type	Often based on cryptographic hash functions.	Utilizes asymmetric cryptography.
Verification process	Verification may involve challenging the signer to reveal the secret key.	Verification relies on the public key.
Flexibility	May offer more flexibility in terms of design and implementation.	Standardized interoperability across different systems.
Key distribution	Key distribution may be more for this.	Public keys can be distributed through a public key infrastructure.

In [8], elliptic curve undeniable signature scheme based on the Elliptic Curve Cryptography (*ECC*) is introduced which is the improvised model of undeniable group signature scheme. It is more simpler, secure and efficient than the

undeniable group signature scheme. In [9] and [10] a polynomial time attack against the protocols proposed and proves that by using the attack, it can be reduced to factoring. In [11], a new concept Discrete Logarithm Conjugacy Search Problem (*DLCSP*) is introduced which is the combination of two problems: Discrete Logarithm Problem (*DLP*) and Conjugacy Search Problem (*CSP*) and an *USS* based on *DLCSP* in a non-abelian group over groupring is proposed with its security and time complexity analysis. These studies offer numerous distinct *USS* with varying degrees of security, provability, and other features.

Based on Conjugacy Search Problem with Discrete Logarithm Problem (*CDL*) a group signature scheme is given, in [12]. In [13], factorization algorithm developed and they presented the advantages and disadvantages of the proposed algorithm for better understanding The platform group of this scheme is general linear group with all the entries in group ring. In [14], a new digital signature is designed based on double conjugacy over non-commutative rings. The security of the *DLCSP* based protocols and schemes over $GL_n(f_q[S_r])$ is examined, in [15]. It shows that solving this problem is equivalent to solving *DLP* in polynomial time. A new hill cipher that uses the non-singular transformation is propose, in [16]. It is resistant towards the known-plaintext attack. The first tight secure group authenticated key exchange protocol is introduce, in [17]. This protocol is constructed using strong computational Diffie Hellman assumption.

In [18], a secure practical payment protocol is proposed for internet purchases with applications without any requirement of the private information of customers, the verifier certifies the re-encrypted data. The author claims that the scheme provides improved security and higher anonymous certificate. [19] demonstrates that a cryptanalyzing algorithm is made practical by using the decomposition of matrices over group ring. The first *USS* based on lattices is presented [20]. The scheme depends on the hardness of the In homogeneous Small Integer Solution problem and resistant to quantum attacks. [21] surveys the public key cryptography based on Semigroup Action Problem and security. In [22], some *USS* are proposed along with security.

The total number of devices connected to the internet is expanding quickly as a result of technology, and these devices require effective and quick data transfer and processing. The cloud computing paradigm has recently made it possible to connect gadgets to the internet safely and affordably. Due to the rapid expansion of mobile and different smart devices, remotely located cloud servers are unable to respond rapidly to delay-sensitive applications. Additionally, cloud computing is not appropriate for programmers that need a minimal latency network, effective processing, and a high degree of mobility, precision, scalability, and dependability.

By placing the processing close to the source of information, Edge Computing (*EC*) technology overcomes the issues mentioned above. *EC* permits data processing and analysis close to data collecting sources, rather than simply transferring to a centralized cloud, which lowers latency and bandwidth usage. *EC* gives assurance with no lag time, quick data streaming and prompt reaction for smart devices since it is a distributed platform. Three levels are used to represent the edge ecosystem's structure. First layer is the cloud which stores and processes data. Second layer is the edge nodes which has centers and acts as a connection between the cloud and the edge devices. Equipment known as "edge devices" are utilized to transfer data between a local network and the cloud. It is the third layer.

Mobile Edge Computing (*MEC*) is a widely used version of *EC*. The telecommunication and mobile firms use it to provide media information close to their mobile devices. To solve the aforementioned problems, scientists carefully examined the *MEC* strategy, which unifies connectivity, storage, and computing resources with the base station. Replicated transmissions, back-haul traffic, and communication efficiency are all enhanced by *MEC*. It provides users with distribution of data that is more dependable. Additionally, it significantly lowers latency and the amount of traffic on the fifth-generation mobile Internet. The *MEC* infrastructure is shown in Figure 1. In one sense, unlike cloud computing, *EC* guarantees greater data safety since processing takes place closer to Internet of Things (*IoT*) devices. *EC* presents greater security and privacy challenges as a result of network layout. Since numerous service providers are opening edge servers at remote network edges, the susceptibility to multiple assaults has increased.

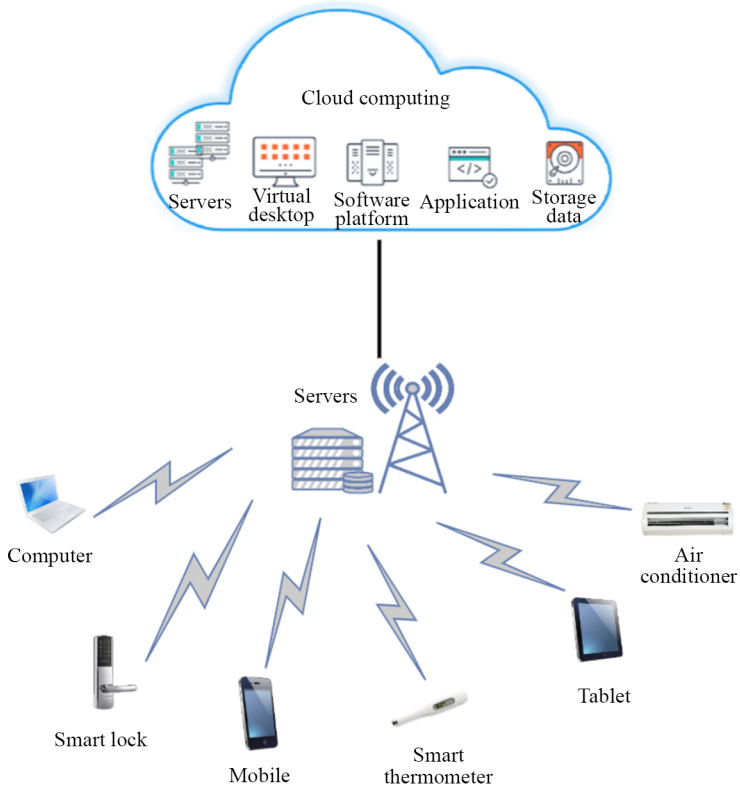

Figure 1. MEC architecture

A secure data processing model based on homomorphic encryption for *IoT* healthcare systems is proposed in [23].

In this article propose a provably secure authenticated key agreement protocol based on Jia et al.'s scheme. As a formal security proof, we simulate our scheme with the widely used Automated Validation of Internet Security Protocols and Applications (AVISPA) tool.

It claims that the scheme is better in storage complexity, communication overhead and computational overhead. [24] proposes a hybrid scheme that includes encryption deduplication with increased two level security and ownership proof achieved through homomorphism of algebraic signatures and cuckoo filters. A new technique using homomorphic encryption to overcome the problem of task assignment in *EC* based large-scale crowdsensing is introduced in [25]. In mobile crowdsensing, a homomorphic public key cryptosystem is presented in [26] to preserve privacy. [27] suggests a privacy-preserving authentication mechanism in *EC* for the vehicular networks using fifth generation. Using Elliptic Curve Cryptography in blockchain, an authentication technique which is privacy preserving is given in [28]. A trusted verification scheme for sixth generation using blockchain technology is presented in [29]. A multi-authority attribute based encryption scheme over fog computing is introduced in [30]. In Mishra et al. [31] suggests a no middle man authentication and key agreement scheme for *MEC*. A *MEC* based certificate less group signature scheme is given in [32]. In [33] proposes a data aggregation scheme for preserving privacy in *MEC* aided *IoT* devices. Rakeei et al. [34] examines Jia et al.'s method, which is one of the most recent methods of authentication used in the *MEC* system. They demonstrate that this technique is susceptible to attacks such as temporary secret leaks. The research carried out so far in *MEC* is using digital signature, authentication and aggregation schemes. *USS* which is more secure and efficient is not implemented in *MEC* yet. *USS* in *MEC* provides more secure transactions and eliminates the risk of fraudulent transactions. It also eliminates the chances of such fraudulence by imposter. Additionally, there is a protocol to check whether the sign is forged.

1.1 *Motivation*

There are two types of protocol namely verification and disavowal protocol avaliable in undeniable signature scheme whereas in digital signature scheme only verification protocol is present. But the disavowal protocol acts as a re-verification protocol to find out who forged the signature. This motivates us to choose the undeniable signature scheme. There is no existing undeniable signature scheme applied in MEC. Therefore we have applied undeniable signature scheme in MEC.

In this paper, an existing DLP is combined with hidden parameters to establish a new hard problem called Discrete Logarithm Problem With Factor Problem ($DLFP$). The complexity and security is explored. Then, utilizing the set of invertible matrices over semiring, a new USS is developed and the functioning of USS in MEC is explained. The hardness of the $DLFP$ decides the security of this scheme.

Mobile edge computing introduces challenges such as limited computational resources, intermittent connectivity, and the need for lightweight cryptographic solutions. The above challenges are solved by the proposed USS based on $DLFP$ over semiring.

Leveraging the discrete logarithm factor problem within a semiring framework for cryptographic protocols in mobile edge computing provides a promising avenue for achieving a balance between security and efficiency, catering to the unique challenges presented by the edge computing paradigm.

The notion used in the paper are listed in Table 2.

Table 2. List of notation

Notation	Description
$\mathbb{S}_{\mathbb{R}}$	Semiring
$Z_r \setminus \{0, 1\}$	Ring of integer modular and remove 0 and 1
H	Commutative semiring
H_1, H_2	Subsemiring
$C_G(g)$	Centralizer of g
G	Non commutative group
V	Commutative subgroup
$(0, 1)^*$	Set of all finite-length strings of 0 s and 1 s, including the empty string

1.2 *Solution to MEC-based challenges using USS*

Limited computational resources:

Provides a computationally lightweight solution for resource-constrained mobile devices in edge computing scenarios.

Secure key management on mobile devices:

Offers a robust mechanism for secure key management, crucial for maintaining the integrity of signatures on mobile devices within edge networks.

Efficient key distribution:

Optimizes key distribution processes, which is vital in edge environments by providing a secure and efficient means of distributing and managing keys.

1.3 *Advantage of MEC for proposed USS*

1. Efficiency in resource constrained devices.
2. Flexibility for diverse edge applications.
3. Efficient key distribution for edge networks.

The rest of the paper is arranged in the following order. Section 2 presents the essential concepts for understanding the work. Section 3 discusses about the first ever USS. Section 4 explains $DLFP$ and discusses its complexities. Section 5 key

exchange protocol based on *DLFP* is dicussed. Section 6 proposes a new *USS* based on *DLFP* over a non-commutative group and its application in *MEC*. Section 7 evaluates the security and complexity of the proposed scheme. Section 8 ends with conclusions.

2. Preliminaries

Definition 1 (Semiring $(\mathbb{S}_{\mathbb{R}})$) [35] A semiring $(\mathbb{S}_{\mathbb{R}})$ is a nonempty set on which the addition and multiplication operations are defined as

1. $(\mathbb{S}_{\mathbb{R}}, +)$ is a commutative monoid with identity element 0.
2. $(S_R, .)$ is a monoid with identity element 1.
3. Multiplication distributive over addition from either side.
4. $0r = 0 = r0$ for all $r \in R$.

Example 1 $(\mathbb{R}, +, \cdot)$, $(\mathbb{Z}, +, \cdot)$, $(\mathbb{C}, +, \cdot)$ and $(\mathbb{W}, +, \cdot)$ are semirings. Let $Z_o = Z^+\{0\}$. $(Z^o, +, \cdot)$ is a semiring of infinite cardinality.

Definition 2 (Discrete Logarithm Problem) [36] The *DLP* is to find an integer x, $0 \leq x \leq p-2$ such that $\alpha^x \equiv \beta \pmod{p}$ where p is a prime, $\mathbb{Z}_{\mid} \setminus \{0, 1\}$ is a cyclic group, a generator α of $Z_p \setminus \{0, 1\}$ and an element $\beta \in Z_p \setminus \{0, 1\}$.

Example 2 Let's say $g = 2$, $h = 8$, and $p = 11$. We want to find x such that $2^x \equiv 8 \pmod{11}$.

To solve this:

$2^1 \equiv 8 \pmod{11}$

$2^2 \equiv 8 \pmod{11}$

$2^3 \equiv 8 \pmod{11}$

Therefore, in this example, the solution to the discrete logarithm problem is $x = 3$ because $2^3 \equiv 8 \pmod{11}$.

Definition 3 (Factor Problem) [37] Let x be any arbitrary element in non-commutative semiring H and two subsemiring $H_1, H_2 \in H$. Factor problem (FP) is defined as finding any two element $x_1 \in H_1, x_2 \in H_2$ such that $x = x_1 \cdot x_2$.

Example 3 Let H be 3×3 matrix and H_1 Upper triangular 3×3 matrix, H_2 lower triangular 3×3 matrix. Find element is $A_1 \in H_1, A_2 \in H_2$ such that

$$A = H_1 \cdot H_2 = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 7 & 0 \\ 0 & 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 2 & 5 & 0 \\ 3 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 14 & 16 & 9 \\ 14 & 35 & 0 \\ 9 & 6 & 9 \end{bmatrix} = A.$$

Definition 4 (Centralizer) [38] The centralizer of an element g in a group is the set of all elements in G that commute with g. Formally, the centralizer $C_G(g)$ of an element g in group G is defined as: $C_G(g) = \{ h \in G : hg = gh \}$. This means that for every element h in the centralizer, the product hg is equal to gh, indicating that h commutes with g.

Example 4 Let G be the group of 2×2 matrices with real entries and matrix multiplication as the group operation:

$$G = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} | a, b, c, d \in \mathbb{S}_{\mathbb{R}} \right\}$$

Consider the matrix g in G: $g = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}$. Centralizer of g in G is denoted as which $C_G(g)$ consists of all matrices in G that commute with g. In this case, any diagonal matrix h commutes with g.

$$C_G(g) = \left\{ \begin{pmatrix} x & 0 \\ 0 & x \end{pmatrix} | x \in \mathbb{S}_{\mathbb{R}} \right\}.$$

This set of matrices is the centralizer because for any matrix h from $C_G(g)$, the product hg is equal to gh.

In the Definitions 1-4, the selection of H and p is determined by the level of security the designer desires in the cryptosystem.

2.1 *Framework of MEC*

Three layers are present in this scheme. The first layer is the Terminal Device (TD). The Edge Server (ES) is the second layer and the next layer is the Public Cloud Center (PCC). These three layers are are shown in Figure 2.

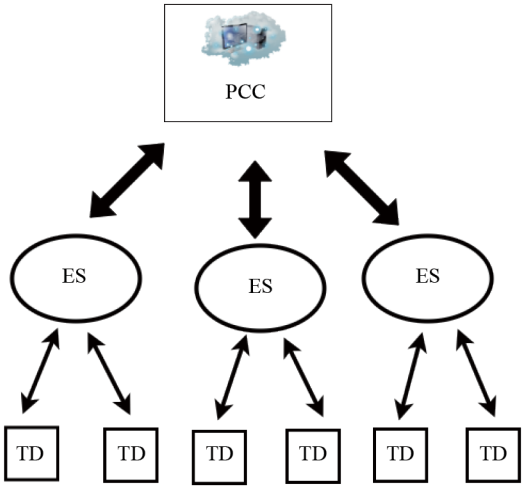

Figure 2. Framework of our scheme

Clients' devices, such as smartphones and IoT gadgets, are linked to the edge network to gather certain information are called TDs. These gadgets often have constrained computer and communication capabilities. ESs are managed by PCC and are situated close to TDs (near the network's edge). By offering processing and storage services for the TDs, ESs serves as an interface between TDs and PCC which expands the uses of cloud services. For clients with big storage capacities and powerful computers, PCC offers cloud-based storage and computing facilities. In PCC, data gathered from TDs and ESs can be processed and saved. PCC creates system parameters which initializes the system. Additionally, PCC is in charge of the registration of TDs and generating the private and public keys for ESs.

2.2 *Flow of data between three layers*

User input (terminal device):

A user interacts with a mobile device, such as a smartphone, to capture an image using the device's camera.

Local processing (edge server):

The image data is processed locally on the edge server located in proximity to the user. This processing may involve initial analysis or filtering to reduce the amount of data that needs to be sent to the cloud.

Edge-to-cloud communication (edge server to public cloud center):

Processed data or relevant information is then sent from the edge server to the public cloud center for further analysis or storage. This communication may occur over the internet.

Cloud processing (public cloud center):

The public cloud center performs more resource-intensive computations, such as complex image recognition or deep learning algorithms, leveraging the extensive computational resources available in the cloud.

3. DLP based USS

During the verification process, the verifier and signatory both cooperate in undeniable signature. But in *DS* there is no verification process.

3.1 *Setup choices*

Non-commutative matrix:

Choosing a non-commutative matrix introduces additional complexity to $DLFP$, enhancing its security.

Commutative subgroup:

Selecting a commutative subgroup within the non-commutative matrix is a strategic choice.

Hash function:

The hash function choice is crucial for $DLFP$ security. A cryptographic hash function ensures that the mapping from input to output is one-way and collision-resistant.

Multiplicative group:

Choosing a multiplicative group (e.g., modulo a prime) aligns with traditional $DLFP$ settings. The difficulty of solving the discrete logarithm factor problem in a multiplicative group is a cornerstone of $DLFP$ security.

3.2 *Key generation choices*

Public key:

The public key typically includes parameters such as the base of the group, the generator, and other public parameters. Making these public does not compromise security, and it facilitates key exchange and validation.

Private key:

The private key is kept secret and is chosen randomly within the chosen group. The randomness ensures that the discrete logarithm factor problem is hard to solve.

Parameter considerations:

Parameters such as matrix size, subgroup order, and hash function output size are chosen based on security considerations and the application's requirements.

Key lengths:

The lengths of public and private keys are chosen based on the security requirements of the application. Longer key lengths enhance security but may increase computational overhead.

The following is the definition of USS [8].

3.3 *Setup*

Let us take H of order b to be a subgroup of Z_p^* and $p = 2b + 1$ (prime), where b is also a prime.

3.4 *Key-gen*

Let us take r an element of order b in Z_p^*; $x \equiv r^a \pmod{p}$ where $a \in [1, b-1]$. The signatory's secret key $SK = a$ and public key $PK = (p, r, x)$.

3.5 *Sign-gen*

The signatory sends (s, z) to the verifier where $s \equiv z^a \pmod{p}$ is computed by the signatory to sign $z \in H$ which is a message.

3.6 *Verification protocol (VP)*

The following steps are done by the verifier.

1. Verifier takes arbitrary element $c_1, c_2 \in Z_p^*$, and finds $c_3 = s^{c_1} r^{c_2} \pmod{p}$.
2. $d_1^{'}$ is sent to the verifier where $d_1^{'} = c_3^{a^{-1} \bmod b} \pmod{p}$.
3. $d_1 = z^{c_1} r^{c_2} \pmod{p}$, d_1 is for verification and c_3 is sent to the signatory.
4. The verifier verifies d_1 and $d_1^{'}$ by checking whether $d_1, d_1^{'}$ are equal or not and accepts or denies, respectively.

3.7 *Disavowal protocol (DP)*

If the verifier sees that the signature is invalid during the verification process, the verifier can use the *DP* to determine whether the signatory is being dishonest at the time of verification or alternatively whether the sign is fabricated. This is how the protocol is written:

1. The verifier selects $e_1, e_2 \in Z_p^*$ at random and transmits $C = s^{e_1} x^{e_2}$ to the signatory.
2. The signatory calculates and sends $d_1 = C^{a^{-1} \pmod{b}} \pmod{p}$ to the person who verifies.
3. The verifier calculates $d_1^{'} = z^{e_1} r^{e_2} \pmod{p}$ and observes that $d_1 \neq d_1^{'}$.
4. The verifier selects f_1 and f_2 in Z_p^* at random, calculates $C = s^{f_1} x^{f_2} \pmod{p}$, and delivers it to the signatory.
5. The signatory calculate $D_1 = C^{a^{-1} \pmod{q}} \pmod{p}$. D_1 is sent to the verifier.
6. $D_1^{'} = z^{f_1} r^{f_2}$ is calculated by the verifier once more, and observes that $D_1 \neq D_1^{'}$.
7. Iff $(d_1 r^{-e_2})^{f_1} \equiv (D_1 r^{-f_2})^{e_1}$, the sign y is forged.

4. Discrete logarithm problem with factor problem (DLFP)

In this section, the *DLFP* problem is introduced. It's security is analyzed. The brute force complexity of this problem is also examined along with an algorithm. Then, a key exchange protocol based on *DLFP* is given.

Definition 5 The *DLFP* is defined as for given $g \in G$ be a non-commutative group and V be a commutative subgroup of G. Then the problem is $g = h_1^t h_2^s$ where $h_1 \in G$, $h_2 \in V$ and $t, s \in Z_p^*$. p is an prime. Here, finding h_2, t, s is the problem and h_1, g are public parameters.

4.1 *Security and complexity of DLFP*

If one of the result parameter h_1, h_2, t, s is provided the *DLFP* problem will be reduced to the Factorization Discrete Logarithm Problem (*FDLP*), *FP*, Double *DLP* and *DLP*, respectively. The following are the possible cases.

1. If the adversary is aware of s, then $g = h_1^t h_2^s$ is reduced to $g = h_1^t h_2$ which is *FDLP* and similarly for t as well [39].
2. Suppose t, s are known, the equation $g = h_1^t h_2^s$ is reduced to $g = h_1.h_2$ which is *FP* [37].
3. If h_1, h_2, t are known, the equation $g = h_1^t h_2^s$ is reduced to *DLP*. Similarly for h_1, h_2, s as well [40].
4. If h_1, h_2, t, s are known, the equation $g = h_1^t h_2^s$ is reduced to double *DLP*.

DLFP cannot be solved using well-known approaches as the determinant attack [41], eigenvalues attack and Cayley-Hamilton attack [9]. The determinant attack becomes invalid, due to the determinant property (non-commutativity) [41] and the knowledge of $det(A^{t_1} B^{t_2})$ and $det(A)$.

This indicates that even if

$$(detA)^{t_1} (detB)^{t_2} = det(A^{t_1} B^{t_2}) = det(P)$$

holds, this approach cannot be used to obtain t_1, t_2 since B is unknown. Even though the adversary knows $P = A^{t_1}B^{t_2}$ and A's eigenvalues, the anonymity of B's prevents the adversary from condensing the issue to a smaller group [41]. The adversary must also be aware of A, $B \in S_R$ such that $A^{t_1}B^{t_2} = f_1(A)f_2(B)$, for any $f_1(x)$, $f_2(x) \in (S_R)[x]$ with

$$f_1(A) = \sum_{i=0}^{m-1} a_i A^i, f_2(B) = \sum_{i=0}^{m-1} b_i B^i,$$

in order to use the Cayley-Hamilton attack. As $f_2(x)$ cannot be found because B is unknown. This approach cannot solve the $DLFP$. Our algorithm overcome the known message attack, chosen message attack or a total break attack. The algorithm is executed in the same runtime as the $FDLP$ based USS more secure.

4.2 *Brute force complexity of DLFP*

Suppose $Z_m = \{0, 1, 2, 3 \ldots m-1\}$ and $G = \{k_1, k_2 \ldots k_\beta\}$ is a non-abelian group of β members. Let $z \in G$ and $t_1, t_2 \in Z_m \setminus \{0, 1\}$ such that $x = y^{t_1}z^{t_2}$ holds for given x, $y \in G$. The variables G and $Z_m \setminus \{0, 1\}$, respectively, are used to choose the components z, t_1 and t_2.

As a result, there are β and $m-2$ total methods to choose z, t_1 and t_2, respectively. Therefore, the $DLFP$'s calculated complexity is $O(\beta m)$. It is exponential in β m's bit size. The $DLFP$ problem overcomes determinant attack, eigenvalue attack and Caylay-Hamilton attack. Since, the problem reduces to $FDLP$, FP, DLP or double DLP when various parameters are known to the adversary, the security is not compromised in anyway.

A brute force attack is a hacking method where an attacker systematically tries all possible combinations of passwords until the correct one is found. It's time-consuming but can be effective if passwords are weak or predictable. Employing strong, unique passwords and using additional security measures, like two-factor authentication, helps mitigate the risk of brute force attacks.

Algorithm 1 Brute force algorithm
Input: Let $a \in A$ such that $C_1 = a^i y^j$, $C_2 = a^k z^l$
Output: Secret parameters B, y, $z \in B \setminus C_A(a)$, i, j, k, $l \in F_r^*$, K_1 and K_2

```
for i ← 1 to r − 1 do
  for j ← 1 to r − 1 do
    K1 ← a^i y^j;
    for k ← 1 to r − 1 do
      for l ← 1 to r − 1 do
        K2 ← a^k z^l;
        if K1 = K2; then
          Return K1 = K2 & exit;
        else
          go to next step;
        end if
      end for
    end for
  end for
end for
```

An exhaustive search algorithm for the following $DLFP$ based protocol is given in Algorithm 1.

4.3 *Explanation of Algorithm 1*

Input:

The algorithm takes an input $'a'$ from the set A and generates two cipher texts, $C_1 = a^i \cdot y^j$ and $C_2 = a^k \cdot z^l$. The desired output includes secret parameters B, y, z in B but not in $C_A(a)$, along with indices i, j, k, and l.

Initialization:

Initialize a loop for $'i'$ from 1 to $r-1$. Inside the $'i'$ loop, initialize a loop for $'j'$ from 1 to $r-1$. Set K_1 to $a^i \cdot y^j$.

Nested Loop for K_1:

Within the $'i'$ and $'j'$ loops, calculate K_1 for each combination of $'i'$ and $'j'$. Move to the next step for comparison.

Nested Loop for K_2:

Set up a nested loop for $'k'$ and $'l'$, both ranging from 1 to $r-1$. Set K_2 to $a^k \cdot z^l$ for each combination.

Comparison:

Compare K_1 and K_2. If they match, return $K_1 = K_2$ and exit the algorithm. If K_1 is not equal to K_2, proceed to the next step.

Increment and repeat:

Increment $'l'$ and repeat the nested loop for K_2. Increment $'k'$ and repeat the nested loop for K_2. Increment $'j'$ and repeat the loop for K_1. Increment $'i'$ and repeat the loop for K_1.

5. The DLFP based key exchange protocol

Assume that G and H are finite non-commutative group and commutative subgroup of G, respectively. Assume an element of order r (large), $y \in G$ and $C_H(y)$ is the centralizers of y in H. The names of the groups G, H, y and r are public parameters.

1. Alice chooses a secret number as random $a_1, a_2 \in Z_r \setminus \{0, 1\}$ and an element z (secret) in $H \setminus C_H(y)$. She calculates $x_1 = y^{a_1} z^{a_2}$. x_1 is sent to Bob.

2. Bob chooses the secret parameters: the integers $b_1, b_2 \in Z_r \setminus \{0, 1\}$ and an element $w \in H \setminus C_H(y)$. He calculates $x_2 = y^{b_1} w^{b_2}$. Bob sends x_2 to Alice.

3. Alice computes $y^{a_1} x_2 z^{a_2}$ the secret key (shared) .

4. Bob computes $y^{b_1} x_1 w^{b_2}$ the secret key (shared).

$y^{a_1+b_1} z^{a_2} w^{b_2}$ is the secret key (common). Since, $zw = wz$ in H.

Example 5 (*DLFP* based key exchange protocol example)

1. Alice choose secret integer $a_1 = 2$, $a_2 = 3$. She calculate

$$x_1 = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}^2 \cdot \begin{pmatrix} 4 & 0 \\ 0 & 7 \end{pmatrix}^3 = \begin{pmatrix} 256 & 0 \\ 0 & 3037 \end{pmatrix}$$

2. Bob choose secret integer $b_1 = 5$, $b_2 = 9$. He calculate

$$x_2 = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}^5 \cdot \begin{pmatrix} 3 & 0 \\ 0 & 5 \end{pmatrix}^9 = \begin{pmatrix} 629856 & 0 \\ 0 & 474609375 \end{pmatrix}$$

3. Alice compute

$$K_1 = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}^2 \cdot \begin{pmatrix} 629856 & 0 \\ 0 & 474609375 \end{pmatrix} \cdot \begin{pmatrix} 4 & 0 \\ 0 & 7 \end{pmatrix}^3 = \begin{pmatrix} 161243136 & 0 \\ 0 & 1465119140625 \end{pmatrix}$$

4. Bob compute

$$K_2 = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}^5 \cdot \begin{pmatrix} 629856 & 0 \\ 0 & 474609375 \end{pmatrix} \cdot \begin{pmatrix} 3 & 0 \\ 0 & 5 \end{pmatrix}^9 = \begin{pmatrix} 161243136 & 0 \\ 0 & 1465119140625 \end{pmatrix}$$

6. USS based on DLFP over semiring

A novel *USS* based on *DLFP* is provided in this section. The working of this scheme in *MEC* is also explained. The *USS* is constructed as follows.

6.1 *Set-up*

Let $G = GL_n(S_R)$ be a group of order α, H be an commutative subgroup of G (order β) and $Z_r = \{0, 1...r-1\}$ (ring of integers modulo r).

6.2 *Key gen*

Let $P = A^{t_1}B^{t_2}$ and $A \in G$ of high order r, where $B \in H$ and $t_1, t_2 \in Z_r \setminus \{0, 1\}$ are secret parameter. The public key (signatory) is therefore $PK = (P, A)$ while the secret key is $SK = (t_1, t_2, B)$.

6.3 *Sign gen*

Let h be a hash function with the following syntax $h : (0, 1)^* \to G \setminus C_g(B)$. $S = (h(m))^{t_1t_2}Z$, where $Z = B^{-(t_2+1)}A^{-t_1}$; the hashed value $h(m) \in G \setminus C_g(B)$ is a sign on a message $m \in (0, 1)^*$.

6.4 *VP*

The following procedures are used to verify the sign S with the cooperation of the two parties:

1. After receiving the message m's sign, the verifier chooses integer $a_1, a_2 \in Z_r$ and a random matrix $T \in H$, calculates $C = (h(m))^{a_1a_2}SPT$. C is sent to the signatory.
2. The signatory computes $X = (h(m))^{-(t_1t_2)}CB$. The result is sent to the verifier.
3. The verifying party computes $X_1 = (h(m))^{a_1a_2}T$ and finds if $X = X_1$.
4. The sign is valid if and only if $X = X_1$.

6.5 *DP*

If the sign is found to be incorrect, that is $X \neq (h(m))^{a_1a_2}T = X_1$ the verifier performs an extra round with random elements W, b_1 and b_2 in the verification protocol. $C' = (h(m))^{b_1b_2}SPW$ is sent to the signatory then the verifier observes that $X' \neq (h(m))^{-(t_1t_2)}W = X_2$ after receiving $X' = (h(m))^{-(t_1t_2)}C'B$ from the signatory and come to the conclusion that $h(m)$ is forged iff XW is equal to $X'T$.

The following are the advantages of using *DLFP* in the proposed *USS* compared to other cryptography schemes.

1. Mathematical complexity.
2. Security foundations.

3. Asymmetric cryptography.
4. Difficulties in factorization.

Example 6 Setup

Let $A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 3 & 1 \\ 0 & 0 & 7 \end{pmatrix} \in GL_3(S_R)$ and $B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{pmatrix} \in 3 \times 3$ diagonal matrix.

Let $t_1 = 2$ and $t_2 = 3$ then the hash funch is defined as.

Let hash function is defined has

$$h : (0,1)^* \to G \setminus C_g(B).$$

$h(m) = [M] = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 7 \end{pmatrix}$ where the message $m \in \{0, 1\}^*$.

Key generation

Calculate

$$P = A^{t_1} \cdot B^{t_2} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 3 & 1 \\ 0 & 0 & 7 \end{pmatrix}^2 . \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{pmatrix}^3 = \begin{pmatrix} 1 & 1152 & 8918 \\ 0 & 576 & 3430 \\ 0 & 0 & 16807 \end{pmatrix}$$

Public key is (P, A) and private key is (t_1, t_2, B).

Signature generation

$$S = (h(m))^{t_1 t_2} B B^{-3} A^{-t_1}$$

$$= \begin{pmatrix} 5 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 7 \end{pmatrix}^{2.3} . \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{pmatrix}^{-3} . \begin{pmatrix} 1 & 2 & 3 \\ 0 & 3 & 1 \\ 0 & 0 & 7 \end{pmatrix}^{-2}$$

$$S = \begin{pmatrix} 15625 & 0 & 0 \\ 0 & 1249/4 & 893035/98 \\ 0 & 0 & 49 \end{pmatrix}$$

S is signature with message.

Verification protocol

Let $a_1 = 3$ and $a_2 = 4$ are integer.

$$T = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 9 \end{pmatrix}$$

$$C = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 7 \end{pmatrix}^{3.4} . \begin{pmatrix} 15625 & 0 & 0 \\ 0 & 1249/4 & 893035/98 \\ 0 & 0 & 49 \end{pmatrix} . \begin{pmatrix} 1 & 1152 & 8918 \\ 0 & 576 & 3430 \\ 0 & 0 & 16807 \end{pmatrix} . \begin{pmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 9 \end{pmatrix}$$

$$= \begin{pmatrix} 5 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 7 \end{pmatrix}^{3.4} . \begin{pmatrix} 46875 & 126000000 & 1254093750 \\ 0 & 1258992 & 1388038680 \\ 0 & 0 & 7411887 \end{pmatrix}$$

$$\text{Compute } X = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 7 \end{pmatrix}^{-(2.3)} . \begin{pmatrix} 5 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 7 \end{pmatrix}^{3.4} . \begin{pmatrix} 46875 & 126000000 & 1254093750 \\ 0 & 1258992 & 1388038680 \\ 0 & 0 & 7411887 \end{pmatrix}$$

$$. \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{pmatrix} = \begin{pmatrix} 1171875 & 0 & 0 \\ 0 & 115123680 & 0 \\ 0 & 0 & 51883209 \end{pmatrix} = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 7 \end{pmatrix}^{3.4} . \begin{pmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 9 \end{pmatrix}$$

$$= X_1$$

Suppose $X \neq X_1$ it goes to disavowal protocol.

6.6 *Application of the proposed USS in MEC*

The process of setup in sec 6 takes place in PCC. The generation of key and sign is done by TD. The two protocols VP and DP involves both ES and TD.

In the VP, both ES and TD has to cooperate. Step 1 is done by ES and the calculated value of C is sent to TD. In Step 2, TD calculates X and sends to ES. Then, ES calculates X_1 and checks if $X = X_1$, in Step 3. The validity of the sign is determined by the values of X and X_1. If the sign is not valid, i.e., $X \neq X_1$, then ES does the DP.

In the DP, ES performs an extra round by calculating $C^{'}$. $C^{'}$ is sent to TD. TD calculates $X^{'}$ and sends to ES. Now, ES concludes that $h(m)$ is forged iff XW is equal to $X^{'}T$.

The use of undeniable signature schemes over semiring based on the Discrete Logarithm Factor Problem $(DLFP)$ in MEC provides enhanced security features, including stronger non-repudiation, adaptability to dynamic networks, privacy-preserving protocols, resistance to quantum attacks, efficient key distribution, support for multiparty computation, robustness against tampering, and dynamic trust models. These advantages make semiring-based undeniable signatures a promising approach for securing transactions and communications in mobile edge computing environments.

7. Analysis of the proposed undeniable signature scheme's security and complexity

In this section, we assess the security of the suggested USS, as well as the complexity. The completeness and soundness of the proposed approach is explained in the subsections 7.1 and 7.2.

7.1 *The scheme's completeness*

The following theorems 1-4 are used to analyze the VP's completeness, DP's completeness. VP's soundness and DP's soundness respectively.

Theorem 1 (VP's completeness) If $X = X_1$ where $X_1 = (h(m))^{a_1 a_2} T$, then the verification process is deemed complete.

Proof. The signatory calculates X after receiving C from the verifier. The signatory delivers X to the verifying party. The verifier evaluates that $X = X_1$ is true or false. The equivalence $X = X_1$ is explained as follows: □

$$
\begin{aligned}
X &= (h(m))^{-(t_1 t_2)} CB \\
&= (h(m))^{-(t_1 t_2)} (h(m))^{a_1 a_2} SPTB \\
&= (h(m))^{-(t_1 t_2)} (h(m))^{a_1 a_2} (h(m))^{t_1 t_2} ZA^{t_1} B^{t_2} TB \\
&= (h(m))^{-(t_1 t_2)+(a_1 a_2)+(t_1 t_2)} ZA^{t_1} B^{t_2} TB \\
&= (h(m))^{a_1 a_2} B^{-(t_2+1)} A^{-t_1} A^{t_1} B^{t_2} TB \\
&= (h(m))^{a_1 a_2} B^{-(t_2+1)} B^{t_2} TB \\
&= (h(m))^{a_1 a_2} B^{-1} TB \\
&= (h(m))^{a_1 a_2} T = X_1.
\end{aligned}
$$

As a result, when the verifier receives X_1 and checks if $X = X_1$. If it is equal, he accepts the sign.

Theorem 2 (DP's cmpleteness) If $S \neq (h(m))^{a_1 a_2} BZ$ and the verifying party can deduce that $(h(m))^{b_1 b_2} XW = (h(m))^{a_1 a_2} X^{'} T$.

Proof. Let us consider the LHS of the equation, □

$$
\begin{aligned}
(h(m))^{b_1b_2}XW &= (h(m))^{b_1b_2}(h(m))^{-(t_1t_2)}CBW \\
&= (h(m))^{b_1b_2-(t_1t_2)}(h(m))^{a_1a_2}SPTBW \\
&= (h(m))^{b_1b_2-(t_1t_2)+(a_1a_2)}SPTBW \\
&= (h(m))^{a_1a_2-(t_1t_2)+(b_1b_2)}SPTBW.
\end{aligned} \tag{1}
$$

Similarly, let us consider the *RHS* of the equation,

$$
\begin{aligned}
(h(m))^{a_1a_2}X'W &= (h(m))^{a_1a_2}(h(m))^{-(t_1t_2)}CBW \\
&= (h(m))^{a_1a_2-(t_1t_2)}(h(m))^{b_1b_2}SPTBW \\
&= (h(m))^{a_1a_2-(t_1t_2)+(b_1b_2)}SPTBW
\end{aligned} \tag{2}
$$

Equating (1) and (2) we get,

$$
(h(m))^{b_1b_2}XW = (h(m))^{a_1a_2}X'T. \tag{3}
$$

7.2 Soundness of the scheme

If the likelihood of the signatory giving a valid answer for a forged sign is below a particular level, the *VP* is considered to be sound. Similar to this, A *DP* is considered sound if the signer has a low probability of convincing the verifier to mistake a formal identity for a forgery

Theorem 3 (*VP*'s soundness) The probability that a dishonest signer will provide a valid response to a false signature is no greater than the maximum $\left(\frac{1}{\beta r}, \frac{1}{\alpha-\beta}\right)$.

Proof. Assume the signatory tries to respond to an incorrect sign with a legitimate answer. From the verification protocol, following the receipt of C, the dishonest signatory will

(i) Attempt to educe the pair (a_1, a_2, T) in order to evaluate $X \ni X = X_1$.

(ii) Alternatively the deceptive signatory will just choose $\bar{X}$ (an element) $\ni \bar{X} = X$.

In the first case, the probability of choosing the correct pair is (a_1, a_2, T) not more than $\frac{1}{\beta r}$.

For the second case, the element $X_1(h(m))^{a_1a_2}T$ does not commute with B because $h(m) \in G \setminus C_H(B)$, this implies X_1 doesn't belong to H and hence $X_1 \in G \setminus H$. Hence the probability of selecting such an element $\bar{X}$ from $G \setminus H$ is not greater than $\frac{1}{\alpha-\beta}$. □

Theorem 4 (*DP*'s soundness) Assume $S = (h(m))^{t_1t_2}Z$; Bob complies to the *DP*. If $X \neq (h(m))^{a_1a_2}T$ and $X' \neq (h(m))^{b_1b_2}W$, then the probability of $(h(m))^{b_1b_2}XW \neq (h(m))^{a_1a_2}X'T$ is the minimum of

$$\left(1-\left(1-\frac{1}{\beta r}\right), 1-\left(1-\frac{1}{\alpha-\beta}\right)\right).$$

Proof. Assume that $S=(h(m))^{t_1t_2}Z$ is a legitimate sign for $h(m)$. Also, assume that the identities listed below are true.

$$X \neq (h(m))^{a_1a_2}T \tag{4}$$

$$X \neq (h(m))^{b_1b_2}W \tag{5}$$

$$(h(m))^{b_1b_2}XW = (h(m))^{a_1a_2}X'T \tag{6}$$

The equation (6) is expressed as

$$X' = (h(m))^{(b_1b_2)-(a_1a_2)}XWT^{-1}.$$

If $b_1, b_2 \in Z_r \setminus \{0, 1\}$ then putting

$$R = ((h(m))^{(b_1b_2)-(a_1a_2)}XT^{-1})^{b_1^{-1}b_2^{-1}},$$

we get $X' = R^{b_1b_2}W$. Theorem 3 leads us to the conclusion that S is a valid sign for R with a probability of $\left(1-\frac{1}{\beta r}, 1-\frac{1}{\alpha-\beta}\right)$. But, on $h(m)$, S is a valid sign. Thus $(h(m))^{t_1t_2}Z = R^{t_1t_2}Z$, $h(m) = R$ with a high probability, that is, the min $\left(1-\frac{1}{\beta r}, 1-\frac{1}{\alpha-\beta}\right)$. The fact that $X' = R^{b_1b_2}W$ gives $X' = (h(m))^{b_1b_2}W$, contradicts Equation (4). As a result, Alice may trick Bob with probability $\left(\frac{1}{\beta r}, \frac{1}{\alpha-\beta}\right)$, that is, with the probability that $(h(m))^{b_1b_2}XW \neq (h(m))^{a_1a_2}X'T$ is less than $\left(1-\frac{1}{\beta r}, 1-\frac{1}{\alpha-\beta}\right)$. □

7.3 *Classical security*

Here is a description of the security of the suggested *USS*. Let S stand for the collection of all message signs that could be used. The intruder will attempt to attack using the known message (*KM*) attack, a chosen message (*CM*) attack, or a total break (*TB*) attack to generate a valid sign message pair (m, s) at the very least, in this scenario.

***KM* attack:** Let S stand for the set of all feasible message signs. If an intruder chooses a pair (m, s) in S to fake the sign, the intruder can undertake the following two steps: (i) attempt to get the key (t_1, t_2, B) pair from S (ii) find a message m' not equal to m, such that, their hash value is equal. In this case, the intruder will try to use a *KM* attack, a *CM* attack, or a *TB* attack.

However, it is impractical to get the key pair due to *DLFP*'s challenges, as explained in Section 3. Additionally, the hash function's second pre-image resistant property restrains the intruder from determining m' such that $h(m)$ is equal to

$h(m')$. Even if the intruder discovers such m' and S is a legitimate sign for it, the intruder would still have to compute X with the help of the secret parameters (t_1, t_2, B). It is impractical.

***CM* attack:** The intruder will attempt to match (m, m') which has the same hash value same hash value, and then generate a fake sign S for m'. Once more, the opponent is prevented from doing so by the usage of the collision resistant function. The adversary will still encounter difficulties in solving $DLFP$ for the computation of X during the verification stage after receiving (m', s). Because $DLFP$ is a computationally difficult task, the system is safe against existential forgery through a relented message attack.

***TB* attack:** The attacker will attempt to pose as a legitimate signatory on a communication without being aware of the message sign pairs in the given scenario in order to successfully forge the sign. However, the technique is protected from the attack by the use of a pre-image resistant hash function.

Our proposed USS overcomes KM, CM and TB attacks. Also, the $DLFP$ problem reduces to $FDLP$, FP, DLP and double DLP when various parameters are known to the adversary. Therefore, our scheme is more secure.

7.4 *Analysis of time complexity*

Given the constraints established in Section 3, the total number of bit operations necessary for sign creation, VP, and DP for the proposed USS is outlined below.

Number of operations required in sign-generation:

For the sign-generation, we must first compute $P = A^{t_1}B^{t_2}$ and $S = (h(m))^{t_1 t_2}Z$, where the secret parameters are $A \in GL_m(S_R)$, $B \in H$ and $t_1, t_2 \in Z_r \setminus \{0, 1\}$. The number of operations in bits necessary to multiply two matrices of order m is $O(m^3)$. To calculate A^{t_1}, $m^3 log\ r$ bit operations are required.

As a result, the number of bit operations necessary to determine $A^{t_1}B^{t_2}$ is $2m^3 log\ r$, which is proportional to $O(m^3 logr)$. Thus, $m^3 log\ r$ number of procedures are needed to compute $S = (h(m))^{t_1 t_2}Z$. As a result, the total number of operations necessary to produce a sign is $O(m^3 log\ r)$.

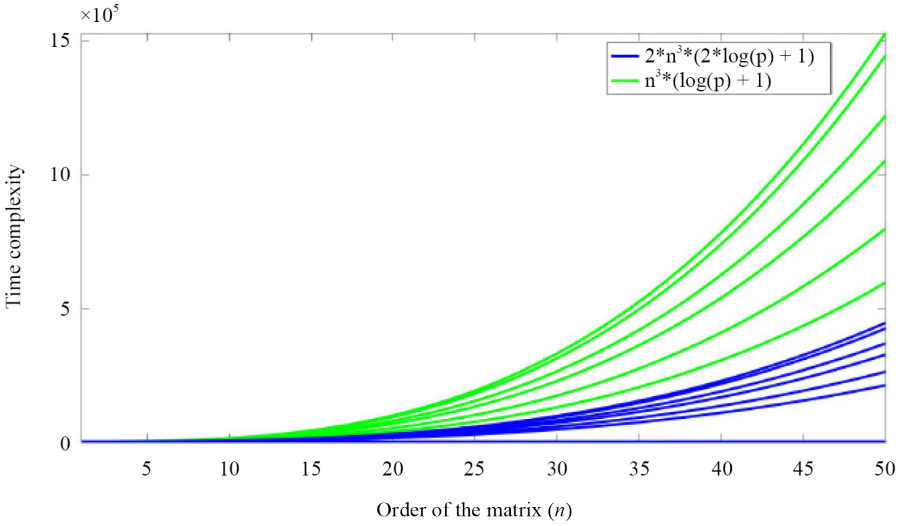

Figure 3. Number of operations in sign-generation

The comparison between the run time of the sign-generation of the USS based on $DLFP$ and $FDLP$ for various sizes of the order of the matrix is depicted in Figure 3. Here, the x-axis and y-axis corresponds to the order (n) of the matrix and the time complexities of the sign-generation of the USS based on $FDLP$ and $DLFP$, respectively.

When the value of the prime p is fixed as 15, the curve of the graph is as portrayed in the Figure 3. The green curves symbolize the run time of the sign-generation of the scheme based on $FDLP$ for different sizes of n. The blue curves

symbolize the run time of the sign-generation of the USS based on $DLFP$ for different sizes of n. When the connection between the green and blue curves is examined, it is evident that the order (n) and the time complexity are directly proportional to each other and this is the reason for the positive slope of the curve.

In both the schemes, the constant terms and the co-efficient terms in the time complexity of sign-generation makes the curves deviate. The time complexity of these USS is given by $(O(n^3 log\ p))$.

Numbers of operations required in *VP*:

The number of operations in bits necessary to do the calculation $C = (h(m))^{a_1 a_2} SPT$ are $6m^3(log\ r+1)$. To calculate $X = (h(m))^{-(t_1 t_2)} CB$, we require $m^3(8log\ r+9)$ operations and to calculate $X_1 = (h(m))^{a_1 a_2} T$, we require $2m^3(log\ r+1)$ operations.

The verification process, the comparison requires only one operation. Therefore, $16m^3 log\ r+17m^3+1$ total bits are needed for sign verification.

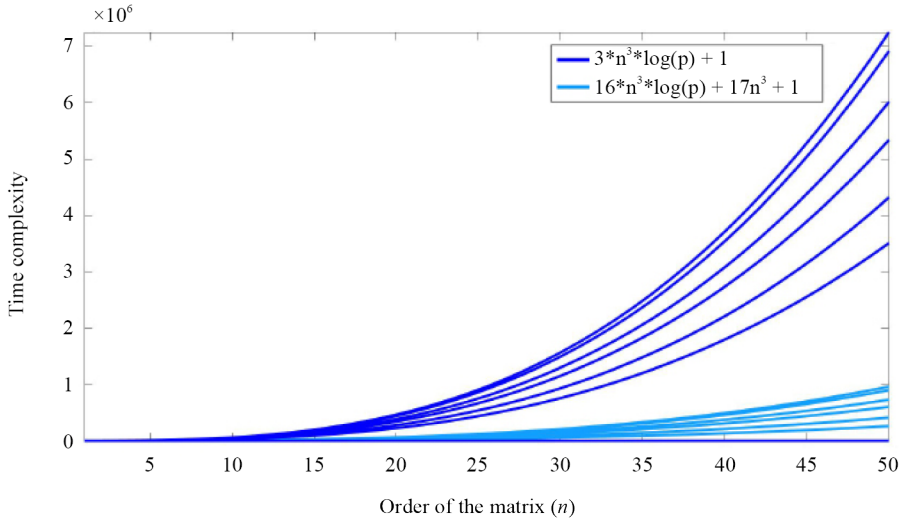

Figure 4. Numbers of operations in VP

Figure 4 displays the required number of operations (i.e., the time complexity) for the VP of the USS based on $FDLP$ and $DLFP$ for different sizes of n. The order of the matrix (n) is represented in x-axis and the number of operation required for the verification protocol of the USS based on $FDLP$ and $DLFP$ is represented in y-axis.

For fixed values of prime ($p = 15$ say), Figure 4 illustrates the curves in dark blue and light blue color. The amount of operations shown by the dark blue curves is needed to verify the USS based on $FDLP$ for different values of n. The amount of operations needed to verify the USS based on $DLFP$ is indicated by light blue curves for different values of n. The dark blue and light blue curves have positive slope and it is obvious when these curves are set side by side.

Since the constant and coefficient terms in the schemes differ, the slope of the dark blue curves and the slope of the light blue curves also differ. Additionally, the VP of the USS based on $DLFP$ and $FDLP$ requires $(O(n^3 log\ p))$ operations to be executed.

Number of operations required in *DP*:

The DP comprises two steps of the VP. Hence, the total number of DP operations is $2(16m^3 log\ r+17m^3+1)$. It is proportional to $O(m^3 \log e)$.

Figure 5 shows the time complexity of the $FDLP$ based USS and the time complexity of the DP of the $DLFP$ based USS. The x-axis shows the order (n) of the matrix. The y-axis represents the time complexity of the DP of the scheme based on $FDLP$ for some n values and the time complexity of the DP of the scheme based on $DLFP$ for some values of n in the graphs.

When the value of the prime p is 15, the graph's curves looks as in the Figure 5. For various values of n, the blue curves reflect the time complexity of the DP of the $FDLP$ based USS. The time complexity of the DP of the USS based on $DLFP$ is depicted in the figure with green color for various values of the order of the matrix. When looking at both the curves in the Figure 5, it is evident that as n grows, i.e., as the complexity of the matrix grows, so does the complexity of time. The curves have a positive slope as a result. When comparing the curves for the scheme based on $FDLP$ with the curves for the scheme based on $DLFP$, it is evident that the slopes of the curves differs a little. When looking at the graph, it is evident that as n grows, i.e., as the complexity of the matrix grows, so does the complexity of time. The curves have a positive slope as a result. When comparing the time complexity between $FDLP$ and $DLFP$, it is clear that the slopes of the curves differ a little. This discrepancy in positive slopes demonstrates that the number of operations required to run the procedure is affected by the coefficient and constant terms.

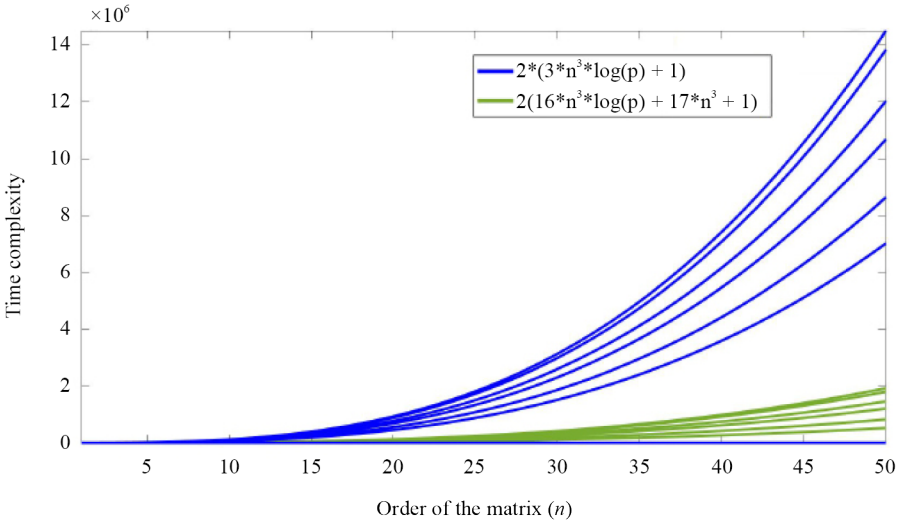

Figure 5. Numbers of operations in DP

The time complexities is proportional to $O(n^3 log\ p)$ in all the cases. When the blue and green curves in the figures are closely examined, it is concluded that the runtime of the USS based on $DLFP$ is less than that of the USS based on $FDLP$. The suggested protocol has the advantage of requiring fewer operations to run than the previous protocol, making it more efficient. The security of the proposed USS is improved by the introduction of the secret parameters t_1, t_2 and B and by concealing the commutative subgroup. As a result, attacking with established ways is impossible.

The number of operation required in sign-generation, verification protocol and disavowal protocol are the same when compared to the scheme based on $FDLP$. The overall time complexity is also not affected in our scheme.

8. Conclusion

In this paper, a new problem named $DLFP$ is proposed. The security of this problem is analyzed. Also, the complexity of this problem is examined by brute force algorithm. Then, a key exchange protocol based on $DLFP$ is proposed with numerical example. With semiring as the platform, an USS based on the $DLFP$ is proposed with illustration. This scheme's application in MEC and it's significance is given. The scheme's completeness and soundness given via theorems. The security of USS is provided by various attacks. The time complexity of each step involved in the USS is calculated and explained through graphical representations has the same time as the exciting scheme.

Acknowledgments

The author expresses gratitude to the anonymous reviewers, the associate editor, and the editor for their constructive feedback towards the improvement of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

[1] Diffie W, Martin EH. New directions in cryptography. In: Rebecca S. (ed.) *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*. Association for Computing Machinery, New York, NY, United States: ACM; 2022. p.365-390. Available from: https://doi.org/10.1145/3549993.3550007.

[2] Stinson DR. *Cryptography: Theory and Practice*. New York: Chapman and Hall/CRC; 2005.

[3] Rivest RL, Adi S, Leonard A. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*. 1978; 21(2): 120-126. Available from: https://doi.org/10.1145/359340.359342.

[4] Ye J, Li L. Group signature scheme based on verifiable random number. *Journal of Discrete Mathematical Sciences and Cryptography*. 2017; 20(2): 525-533. Available from: https://doi.org/10.1080/09720529.2016.1220090.

[5] Chande MK, Lee CC, Li CT. Cryptanalysis and improvement of a ECDLP based proxy blind signature scheme. *Journal of Discrete Mathematical Sciences and Cryptography*. 2018; 21(1): 23-34. Available from: https://doi.org/10.1080/09720529.2017.1390845.

[6] Chaum D, Van Antwerpen H. Undeniable signatures. In: Brassard G. (ed.) *Advances in Cryptology-CRYPTO' 89 Proceedings*. New York, NY: Springer; 1990. p.212-216. Available from: https://doi.org/10.1007/0-387-34805-0_20.

[7] Chaum D. Zero-knowledge undeniable signatures (extended abstract). In: Damgård IB. (ed.) *Advances in Cryptology-EUROCRYPT '90*. Berlin, Heidelberg: Springer; 1991. p.458-464. Available from: https://doi.org/10.1007/3-540-46877-3_41.

[8] Chen TS, Hsu ET, Yu YL. A new elliptic curve undeniable signature scheme. *International Mathematical Forum*. 2006; 1(29-32): 1529-1536. Available from: http://dx.doi.org/10.12988/imf.2006.06129.

[9] Micheli G. Cryptanalysis of a non-commutative key exchange protocol. *arXiv*. 2013; 2013: 1306.5326. Available from: https://doi.org/10.48550/arXiv.1306.5326.

[10] Pathak HK, Manju S. Public key cryptosystem and a key exchange protocol using tools of non-abelian group. *International Journal on Computer Science and Engineering*. 2010; 2(4): 1029-1033. Available from: https://europub.co.uk/articles/-A-150198.

[11] Goel N, Gupta I, Dubey MK, Dass BK. Undeniable signature scheme based over group ring. *Applicable Algebra in Engineering, Communication and Computing*. 2016; 27: 523-535. Available from: https://doi.org/10.1007/s00200-016-0293-8.

[12] Amir NAS, Wan AMO, Wong KB. A secure cryptosystem in group signature scheme based over group ring. *Communications in Combinatorics, Cryptography and Computer Science*. 2022; 2022(1): 62-70.

[13] Balasubramaniam P, Muthukumar P, Binti Mior Othman WA. Prime factorization without using any approximations. In: Balasubramaniam P, Uthayakumar R. (eds.) *Mathematical Modelling and Scientific Computation*. Berlin, Heidelberg: Springer; 2012. p.537-541. Available from: https://doi.org/10.1007/978-3-642-28926-2_61.

[14] Jalaja V, Anjaneyulu GSGN, Narendra Mohan L. New digital signature scheme on non-commutative rings using double conjugacy. *Journal of Integrated Science and Technology*. 2023; 11(2): 471.

[15] Pandey A, Indivar G, Dhiraj KS. On the security of DLCSP over GL n (F q [S r]). *Applicable Algebra in Engineering, Communication and Computing*. 2023; 34(4): 619-628. Available from: https://doi.org/10.1007/s00200-021-00523-6.

[16] Sunil K, Sandeep K, Gaurav M, Dharminder D, Shiv N. Non-singular transformation based encryption scheme. *International Journal of Mathematical Sciences and Computing*. 2021; 7(3): 32-40. Available from: https://doi.org/10.5815/ijmsc.2021.03.04.

[17] Pan J, Chen Q, Magnus R. Signed (group) diffie-hellman key exchange with tight security. *Journal of Cryptology*. 2022; 35(4): 26. Available from: https://doi.org/10.1007/s00145-022-09438-y.

[18] Wang H, Cao J, Zhang Y. A self-scalable anonymity payment approach in cloud environment. *Access Control Management in Cloud Environments*. Cham: Springer International Publishing; 2020. p.91-115. Available from: https://doi.org/10.1007/978-3-030-31729-4_5.

[19] Manasa C, Chandra K, Dhuli S, Enduri MK. A secure matrix inversion protocol for iot applications in smart home systems. *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. Kharagpur, India: IEEE; 2021. p.1-5. Available from: https://doi.org/10.1109/ICCCNT51525.2021.9580058.

[20] Rawal S, Padhye S, He D. Lattice-based undeniable signature scheme. *Annals of Telecommunications*. 2022; 77: 119-126. Available from: https://doi.org/10.1007/s12243-021-00843-1.

[21] Goel N, Gupta I, Dass B. Survey on SAP and its application in public-key cryptography. *Journal of Mathematical Cryptology*. 2020; 41(1): 144-152. Available from: https://doi.org/10.1515/jmc-2016-0004.

[22] Kaur R, Kaur A. Digital signature. *2012 International Conference on Computing Sciences*. Phagwara, India: IEEE; 2012. p.295-301. Available from: https://doi.org/10.1109/ICCS.2012.25.

[23] Hiral ST, Sankita JP. Homomorphic cryptosystem-based secure data processing model for edge-assisted IoT healthcare systems. *Internet of Things*. 2023; 22: 100693. Available from: https://doi.org/10.1016/j.iot.2023.100693.

[24] Yukun Z, Zhibin Y, Liang G, Dan F. Homomorphic cryptosystem-based secure data processing model for edge-assisted IoT healthcare systems. *Internet of Things*. 2023; 22: 100693. Available from: https://doi.org/10.1016/j.tbench.2022.100062.

[25] Ding X, Lv R, Pang X, Hu J, Wang J, Yang X, et al. Privacy-preserving task allocation for edge computing-based mobile crowdsensing. *Computers & Electrical Engineering*. 2022; 97: 107528. Available from: https://doi.org/10.1016/j.compeleceng.2021.107528.

[26] Ganjavi R, Ahmad RS. Edge-assisted public key homomorphic encryption for preserving privacy in mobile crowdsensing. *IEEE Transactions on Services Computing*. 2022; 16(2): 1107-1117. Available from: https://doi.org/10.1109/TSC.2022.3172136.

[27] Zhang J, Zhong H, Cui J, Tian M, Xu Y, Liu L. Edge computing-based privacy-preserving authentication framework and protocol for 5G-enabled vehicular networks. *IEEE Transactions on Vehicular Technology*. 2020; 69(7): 7940-7954. Available from: https://doi.org/10.1109/TVT.2020.2994144.

[28] Wang Y, Jia X, Xia Y, Khan MK, He D. A blockchain-based conditional privacy-preserving authentication scheme for edge computing services. *Journal of Information Security and Applications*. 2022; 70: 103334. Available from: https://doi.org/10.1016/j.jisa.2022.103334.

[29] Wang Y, Tian Y, Hei X, Zhu L, Ji W. A novel IoV block-streaming service awareness and trusted verification scheme in 6G. *IEEE Transactions on Vehicular Technology*. 2021; 70(6): 5197-5210. Available from: https://doi.org/10.1109/TVT.2021.3063783.

[30] Tu S, Waqas M, Huang F, Abbas G, Abbas ZH. A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing. *Computer Networks*. 2021; 195: 108196. Available from: https://doi.org/10.1016/j.comnet.2021.108196.

[31] Mishra D, Dharminder D, Yadav P, Rao YS, Vijayakumar P, Kumar N. A provably secure dynamic ID-based authenticated key agreement framework for mobile edge computing without a trusted party. *Journal of Information Security and Applications*. 2020; 55: 102648. Available from: https://doi.org/10.1016/j.jisa.2020.102648.

[32] Huang H, Wu Y, Xiao F, Malekian R. An efficient signature scheme based on mobile edge computing in the NDN-IoT environment. *IEEE Transactions on Computational Social Systems*. 2021; 8(5): 1108-1120. Available from: https://doi.org/10.1109/TCSS.2021.3076209.

[33] Li X, Liu S, Wu F, Kumari S, Rodrigues JJPC. Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications. *IEEE Internet of Things Journal*. 2019; 6(3): 4755-4763. Available from: https://doi.org/10.1109/JIOT.2018.2874473.

[34] Rakeei M, Moazami F. An efficient and provably secure authenticated key agreement scheme for mobile edge computing. *Wireless Networks*. 2022; 28(7): 2983-2999. Available from: https://doi.org/10.1007/s11276-022-03005-w.

[35] Nivetha S, Thiruveni V, Chandramouleeswaran M. Semiring actions for Public key cryptography. *Journal of Computer and Mathematical Sciences*. 2019; 10(1): 238-244. Available from: https://doi.org/10.29055/jcms/1002.

[36] Menezes AJ, Van Oorschot PC, Vanstone SA. *Handbook of Applied Cryptography*. Boca Raton: CRC Press; 1997.

[37] Muthukumaran V, Ezhilmaran D. New key agreement protocol based on factor problem in centralizer near-ring. *Journal of Science and Arts*. 2018; 18(2): 375-380. Available from: https://doi.org/10.1088/1757-899X/263/4/042137.

[38] Gallian JA. *Contemporary Abstract Algebra*. 9th ed. American: Cengage Learning; 2016.

[39] Pandey A, Gupta I. A new undeniable signature scheme on general linear group over group ring. *Journal of Discrete Mathematical Sciences and Cryptography*. 2022; 25(5): 1261-1273. Available from: https://doi.org/10.1080/09720529.2020.1744814.

[40] Gupta I, Pandey A, Dubey MK. A key exchange protocol using matrices over group ring. *Asian-European Journal of Mathematics*. 2019; 12(5): 1950075. Available from: https://doi.org/10.1142/S179355711950075X.

[41] Eftekhari M. A Diffie-Hellman key exchange protocol using matrices over noncommutative rings. *Groups Complexity Cryptology*. 2012; 4(1): 167-176. Available from: https://doi.org/10.1515/gcc-2012-0001.