DOI: 10.24996/ijs.2025.67.2.33

Iraqi Journal of Science

ISSN: 0067-2904

A Reliable and Effective Authentication Key Agreement Protocol Based on Complete Graph

Raghad K. Salih[1,2], Ali A. Aubad[1]*

[1]*Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq.*
[2]*College of Applied Sciences, University of Technology, Baghdad, Iraq.*

Received: 30/6/2024 Accepted: 27/1/2025 Published: 28/2/2026

Abstract

Protecting information systems from manipulation and unauthorized access is extremely important. The Diffie-Hellman (DH) protocol is essential for key exchange because it is based on the discrete logarithm problem. Although widely used, this protocol is vulnerable to various attacks like man-in-the-middle, replay, and brute force due to its lack of validation mechanisms for authorized parties. The novel proposed algorithm treats these vulnerabilities and enhances the security of the DH scheme by performing authentication between the two parties based on the complete graph. We have used simple mathematical operations to ease implementation. The proposed work enhances the security of the DH algorithm by using the properties of the complete graph K_n containing n vertices and $(n(n-1)/2)$ edges. These edges are used with a pair of passwords (n, v_1) to construct combined public keys Yc_A and Yc_B from the public keys Y_A and Y_B of the sender and receiver, respectively. These combined keys ensure that only authorized parties can decrypt the message and accurately reconstruct the graph. The proposed algorithm provides strong security enhancement to the traditional DH protocol during keys exchange and prevents tampering.

Keywords: Diffie Hellman, complete graph, Man in the middle, Key Agreement Protocol, and Authentication.

بروتوكول اتفاقية مصادقة مفتاح موثوق وفعال يعتمد على الرسم البياني الكامل

رغد كاظم صالح[1]و[2] , علي عبد عبيد[1]*

[1] قسم الرياضيات، كلية العلوم، جامعة بغداد، بغداد، العراق.
[2] كلية العلوم التطبيقية، الجامعة التكنولوجية، بغداد، العراق.

الخلاصة

إن حماية أنظمة المعلومات من التلاعب والوصول غير المصرح به أمر بالغ الأهمية. يعد بروتوكول ديفي هيلمان اساسيا لتبادل المفاتيح لأنه يعتمد على مشكلة اللوغاريتم المنفصل. وعلى الرغم من استخدامه على نطاق واسع، إلا أن هذا البروتوكول عرضة لهجمات مختلفة مثل الرجل في المنتصف والإعادة والقوة الغاشمة، بسبب افتقاره إلى آليات التحقق للأطراف المصرح لها. تعالج الخوارزمية الجديدة المقترحة هذه الثغرات وتعزز أمان مخطط ديفي هيلمان من خلال إجراء مصادقة بين الطرفين بالاعتماد على الرسم البياني الكامل. استخدمنا عمليات رياضية بسيطة لتسهيل التنفيذ. يعزز العمل المقترح امان خوارزمية ديفي هيلمان

*Email: ali.abd@sc.uobaghdad.edu.iq

من خلال استخدام خصائص الرسم البياني الكامل K_n، والذي يحتوي على n رأسًا و $\left(\frac{n(n-1)}{2}\right)$ حافة. يستخدم هذا النهج هذه الحواف مع زوج كلمة المرور (n, v_1) لإنشاء مفاتيح عامة مركبة Yc_A و Yc_B من المفاتيح العامة Y_A و Y_B للمرسل والمستقبل على التوالي. تضمن هذه المفاتيح المجمعة أن الأطراف المصرح لها فقط يمكنها فك تشفير الرسالة وإعادة بناء الرسم البياني بدقة. توفر الخوارزمية المقترحة تعزيزًا أمنيًا قويًا لبروتوكول ديفي هيلمان التقليدي أثناء تبادل المفاتيح وتمنع العبث.

1. Introduction

Cryptography maintains data confidentiality, integrity, and authentication, and it is one of the areas of information security. Since data security can be easily compromised, it is essential to use the most effective systems to secure information. Cryptographers have developed many systems to achieve this goal, and ongoing research focuses on evaluating the security levels of cryptographic systems to ensure the secure transmission of messages from authorized senders to intended recipients [1,2].

In 1976, Diffie and Hellman [3] addressed the key distribution problem in open networks by presenting an asymmetric key agreement scheme. This basic element of modern cryptography enables the secure exchange of keys over an unsecured channel. It's common use confirms its pivotal role in preserving confidentiality and integrity in digital communications. The main advantage of the Diffie-Hellman (DH) algorithm is the ability to create shared secret keys without prior arrangement, hence ensuring confidentiality. Several important applications used the DH scheme to ensure data integrity and secure communication, like secure web connections, blockchain techniques, and the Internet of Things [4,5]. However, this protocol is susceptible to many threats such as man-in-the-middle (MITM), modification, replay, and brute-force attacks because of the lack of authentication between communicating parties [6,7].

To ensure the DH algorithm remains robust against evolving cryptographic challenges, some researchers have proposed key authenticated agreement protocols to provide robust security and prevent various attacks by employing passwords [8,9,10] or combining them with cryptographic systems such as RSA or elliptic curve [11,12,13]. Other researchers suggested adding time stamps, hash functions, second shared secret key, and nonce numbers for additional security strata. Moreover, chaotic map algorithms [14], zero-knowledge proofs [15], Geffe generator random sequences [16], and digital signatures [12] have been used to mitigate vulnerabilities of DH protocol.

Our work presents a novel proposed algorithm for an authenticated key agreement to improve the security of the DH protocol employing the vertices $(v_i, i = 1,2, \dots, n)$, edges $(e_{ij}, i, j = 1,2 \dots, n, i < j)$ and the adjacency matrix of the complete graph K_n with a pair of passwords (n, v_1) and simple mathematical operations to reduce the cost and ease of implementation. It ensures that only authorized parties can interpret messages and rebuild the graph, thus verifying the integrity of the public key and preventing manipulation. It effectively mitigates common threats. This is a robust and efficient enhancement to the traditional DH protocol for secure key exchange.

This work is structured as follows: Section 2 reviews some previous work. Section 3 describes the Diffie-Hellman algorithm, while Section 4 covers the details of complete graph. The proposed method is offered in Section 5. Section 6 presents experimental results. Section

7 provides the security analysis of the suggested algorithm. Finally, Section 8 concludes the work.

2. Related Work

In 2018, Al-Maamori et al. [17] demonstrated that RSA and Diffie-Hellman encryption methods, traditionally reliant on selecting secure primes (p) and (q), can utilize an arithmetical function to generate these primes securely. It introduces a novel construction of RSA and DH key exchange using primes derived from a real number. Chen et al. [18] proposed a secure authentication and key establishment protocol for the Internet of Things, ensuring utilizer and sensor anonymity, forward secrecy, and cheap computing costs utilizing XOR, hash functions, and minimal asymmetric enciphering. Naher et al. [19] suggested a system that amalgamates a cryptographically safe cyclic redundancy check with the algorithm of Diffie-Hellman to identify potential MITM attacks. This involves employing a secure, randomly generated polynomial as a nonzero divisor. The approach demonstrates negligible overhead when contrasting the sizes of public keys with those in the traditional DH protocol. In 2020, Zhang et al. [20] proposed a lightweight authenticated key agreement scheme using secure hash functions and XOR operations, ensuring robust security and lower communication and computation costs. In 2021, Yadav et al. [21] proposed modifying the DH key exchange algorithm claims to thwart MITM attacks through encryption and Zero-Knowledge Proofs. However, analysis reveals that it remains vulnerable.

This paper develops an algorithm to detect these security flaws, incorporating essential authentication and enciphering functions, and demonstrates the MITM attack on the modified DH key exchange. In 2022, Gupta et al. [12] proposed a model for combining RSA with Diffie-Hellman to prevent MITM attacks, ensuring secure key exchange. The model's effectiveness is compared to standalone DH and RSA systems for enhanced security. In 2023, Chaudhary et al. [22] analyzed Zhang et al. [20] protocol, finding vulnerabilities like stolen smart card attacks and excessive data storage, and proposed a secure, efficient modified protocol for the Internet of Drones, and validated efficiency and security as well.

Xia et al. [23] proposed a novel two-round authenticated key exchange scheme for smart grids, combining Diffie-Hellman and secure digital signatures, offering efficient communication and computation with tight security. Nita et al. [24] suggested a low-cost authentication mechanism for Internet of Things devices using elliptic curves, integrating a blockchain network for secure identity verification before data transmission. This method ensures lightweight, secure authentication. In 2024, Hasan et al. [25] proposed a hybrid key agreement protocol using an elliptic curve Diffie-Hellman with a trusted authority. It ensures mutual authentication, forward secrecy, and anonymity with reduced computational overhead. Security assessments confirm its effectiveness, making it suitable for smart grid metering infrastructure. Our proposal enhances the Diffie-Hellman scheme using complete graph structures and simple math operations, ensuring only authorized parties interpret messages and reconstruct graphs. It prevents public key tampering and mitigates threats, with pair passwords ensuring adversaries gain no key information, providing a secure and efficient key exchange method.

3. Diffie-Hellman Algorithm

The Diffie–Hellman (DH) key exchange, developed by Whitfield Diffie and Martin Hellman in 1976, is a foundational method for securely exchanging cryptographic keys over an unsecure public channel [3]. It enables two parties with no prior acquaintance to establish a shared secret key, facilitating encrypted communication via a symmetric-key cipher. Unlike traditional methods that require secure physical key exchange, DH offers a practical solution for secure key exchange over insecure channels. This protocol is widely used in securing Internet services. However, in 2015, concerns emerged about the strength of DH parameters, revealing vulnerabilities to attackers and sophisticated adversaries [2, 26]. The DH scheme is inherently susceptible to various threats such as man-in-the-middle, replay, modification, brute-force, and discrete logarithm attacks. These vulnerabilities primarily arise from the lack of a mechanism for the communicating parties to authenticate each other [5,6,26]. Figure 1 shows the DH algorithm. The steps of DH key exchange process are [3,12]:

1. Public Parameters:
   - Prime Number (p): A large p.
   - Primitive Root (g): A number less than p that can generate all possible remainders ($1\ to\ p-1$) when raised to successive powers modulo p.
   - Key Generation: Alice and Bob both accept the public values p and g.
2. Private Keys:
   - Alice chooses a private key X_A (X_A random number).
   - Bob selects a private key X_B (another random number).
3. Public Keys:
   - Alice computes her public key as $Y_A = g^{X_A}\, mod\ p$
   - Bob computes his public key $Y_B = g^{X_B}\, mod\ p$.
   - Alice sends Y_A to Bob, and Bob sends Y_B to Alice.
4. Shared Secret Computation:
   - Alice privately computes $K = {Y_B}^{X_A}\, mod\ p$.
   - Bob privately computes $K = {Y_A}^{X_B}\, mod\ p$.
   - Both calculations result in the same shared secret K because $(g^{X_B}\, mod\ p)^{X_A} mod\ p = (g^{X_A}\, mod\ p)^{X_B}\, mod\ p$.

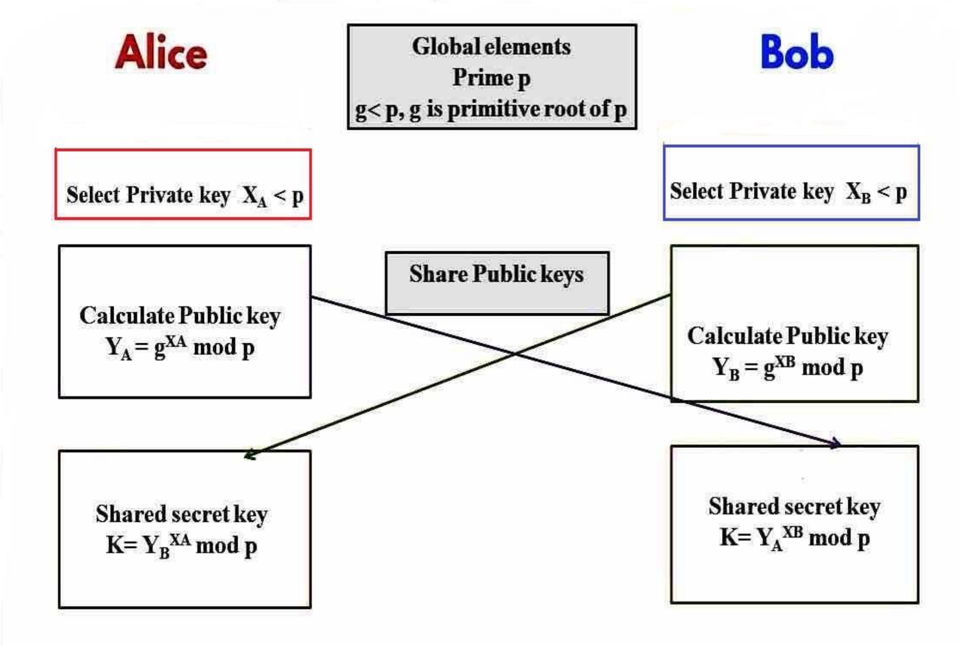

Figure 1: Diffie-Hellman key exchange process.

4. The Complete Graph

Many studies explore the connection between graph and group theories [27,28]. A complete graph K_n is an undirected graph where every pair of vertices is connected by exactly one distinct, containing $\left(\frac{n(n-1)}{2}\right)$edges [29, 30]. A weighted graph assigns numerical values (weights) to edges, representing attributes like costs, lengths, or capacities [31].

4.1 Adjacency Matrix

The graph G=(V, E) consists of a set of vertices $V = \{v_1, v_2, \dots, v_n\}$ and a set of edges E. The adjacency matrix A of G is $(n \times n)$ matrix defined as follows [27,32]:

$$A = (a_{ij})_{n\times n} \quad i = 1, \dots, n \text{ and } j = 1, \dots, n \qquad \dots(1)$$

$$a_{ij} = \begin{cases} w_{ij} & \text{If } v_i \text{ and } v_j \text{ are connected by an edge with weight } w_{ij} \\ 0 & \text{If there is no edge between } v_i \text{ and } v_j \end{cases} \qquad \dots(2)$$

The adjacency matrix A of a weighted complete graph K_n is defined in Eq. (3): In an undirected weighted complete graph, the weights $w_{ij} = w_{ji}$ [31].

$$A = a_{ij} = \begin{cases} w_{ij} & \text{If } i \neq j \\ 0 & \text{If } i = j \end{cases} \qquad \dots(3)$$

5. The Proposed Method

To address the vulnerability of Diffie-Hellman described in Section 3. The new suggested authentication key agreement algorithm is presented based on complete graph to address these vulnerabilities. Our method leverages simple mathematical operations to enhance security. Alice and Bob participate a joint pair of passwords (n, v_1) prior to the protocol starts, where n is the number of vertices in the complete graph $K_n, n > 3$ and (v_1) is the first vertex of K_n. To ensure security, both n and v_1 must be chosen suitably to ensure compatibility with the size of p and to provide adequate security for the combined public keys, as illustrated in Table 1. A larger n increases complexity, making it more difficult for adversaries to compromise the system. They also utilize the same public values (p) and (g) as in the original DH protocol. Alice randomly selects (n-1) remaining vertices of K_n such that $v_1 > v_i, i = 2,3, \dots, n$. She calculates $(w = n \times v_1)$ and computes n_e using Eq. (4). Then she computes the (n-1) weighted edges $e_{1i}, i = 2, \dots, n$ of K_n using Eq. (5). Alice sums the digits of w and determines whether the result is even or odd to create her combined public key Yc_A by appending the e_{1i} in her public key Y_A using either Eq. (6) or Eq. (7) as explained below. She then sends this combined key to Bob while keeping the remaining (n-1) vertices private. She ensures that each edge weight has a number of digits equal to the number of digits of the first vertex. This ensures that only authorized parties can reconstruct the graph and interpret the message, enhancing security by verifying the integrity of the public key during exchange, thereby preventing tampering.

$$n_e = \frac{3n(n-1)}{2} + 1 \qquad \dots (4)$$

$$e_{1i} = v_1 - v_i \;, v_1 > v_i \;, \; i = 2,3, \dots, n \qquad \dots (5)$$

Let u be the number of digits of v_1. To determine the combined Alice's public key Yc_A, there are two cases based on the sum of digits in w (Sw).

1. If Sw is an even number

Yc_A is constructed by appending the edge weights e_{1i} as described in formula (6). Ensure that e_{12} has exactly u digits to avoid adding zeros and pad $e_{1i}, i = 3,4, \dots, n$ and n_e with leading zeros as needed to match u.

$$Yc_A = e_{12} \parallel 1^{st} u \text{ digits of } Y_A \parallel e_{13} \parallel 2^{nd} u \text{ digits of } Y_A \parallel \cdots \parallel e_{1n} \parallel$$
$$\text{the remaining } u \text{ digits of } Y_A \text{ if any} \parallel n_e \parallel \text{the remainig digits of } Y_A \text{ if any} \qquad \dots (6)$$

2. If Sw is an odd number

Yc_A is constructed by inverting formula (6), which means reversing the order of its components and appending the edge weights e_{1i} in the following manner.

$$Yc_A = The\ remaining\ digits\ of\ Y_A\ if\ any \parallel n_e \parallel the\ remaining\ u\ digits\ of\ Y_A\ i\ any \parallel e_{1n} \parallel \cdots \parallel 2^{nd}\ u\ digits\ \text{of}\ Y_A \parallel e_{13} \parallel 1^{st} u\ digits\ of\ Y_A \parallel e_{12} \quad \ldots (7)$$

where ‖ means [33,34] combining the above values by placing them side by side to form a single sequence without spaces.

Bob receives combined Alice's public key Yc_A. He first finds the public key of Alice Y_A by calculating the number of digits of Y_A as described in Eq. (8). Then, he finds n_e and calculates $(n_e - n^2)$, verifying if it matches the value of the remaining $\left(\frac{n(n-1)}{2}\right)$ edges. If these values do not match, he stops and rejects the session as dishonest. Otherwise, he assumes the authenticity of the communicating party as being genuinely Alice and proceeds to compute the $\left(\frac{(n-1)(n-2)}{2}\right)$ weighted edges of K_n as in Eq.(9) and Eq.(10), using the password pair (n, v_1) and the (n-1) weighted edges from Alice.

$$Digits(\ Y_A) = Digits(Yc_A) - (n \times u) \quad \ldots (8)$$

$$v_i = v_1 - e_{1i} \quad \ldots (9)$$

$$e_{ij} = |v_i - v_j|\ ,\ i < j \quad \ldots (10)$$

where $i = 2,3, \ldots, n\ , j = 3,4, \ldots, n$ and $Digits(X)$ in Eq. (8) represents the number of digits in *X*, Bob then appends the weighted edges e_{ij} in his public key Y_B to create his combined public key Yc_B based on *Sw*. This is done using the method described below.

1. If *Sw* is an even number
   Yc_B is constructed by appending the edge weights e_{ij} in Eq.(10) as described in formula (11). Ensure that e_{23} has exactly *u* digits to avoid leading zeros and pad $e_{ij}\ , i = 2, \ldots, n-1,\ j = 3,4, \ldots, n\ , i < j$ with leading zeros as needed to match *u*.

$$Yc_B = e_{23} \parallel 1^{st} u\ digits\ of\ Y_B \parallel e_{24} \parallel 2^{nd} u\ digits\ of\ Y_B \parallel \cdots \parallel e_{n-2,n-1} \parallel\ the\ remaining\ \ u\ digits\ of\ Y_B\ if\ any \parallel e_{n-1,n} \parallel The\ remainig\ digits\ of\ Y_B\ if\ any \quad \ldots (11)$$

2. If *Sw* is an odd number
   Yc_B is constructed by reversing formula (11) as follows.

$$Yc_B = The\ remaining\ digits\ of\ Y_B\ if\ any \parallel e_{n-1,n} \parallel remaining\ u\ digits\ of\ Y_B\ if\ any \parallel e_{n-2,n-1} \parallel \cdots \parallel 2^{nd}\ u\ digits\ \text{of}\ Y_B \parallel e_{24} \parallel 1^{st} u\ digits\ of\ Y_A \parallel e_{23} \quad \ldots (12)$$

Bob then sends Yc_B to Alice, ensuring each edge weight has the same digit count as the first vertex to avoid confusion for Alice. Including the weighted edges of K_n is crucial, as it provides the necessary information for Alice to verify the integrity and completeness of the graph K_n. Upon receiving Bob's combined public key, Alice finds the public key of Bob Y_B by calculating the number of digits of Y_B as described in Eq. (13). Next, she extracts the weighted edges and computes the adjacency matrix as in Eq. (14) and verifies if it represents her complete graph, which is crucial for authenticating the session key. If verified, Alice proceeds with the authentication, and both Alice and Bob use the session key to compute the shared key for secure communication as shown in Figure 2. Otherwise, they reject the session due to suspected dishonesty.

$$Digits\ (Y_B) = Digits(\ Yc_B) - \left(u \times \left(\frac{(n-1)(n-2)}{2}\right)\right) \quad \ldots (13)$$

$$A=\begin{pmatrix} 0 & e_{12} & e_{13} & e_{14} & \cdots & e_{1n} \\ e_{21} & 0 & e_{23} & e_{24} & \cdots & e_{2n} \\ e_{31} & e_{32} & 0 & e_{34} & \cdots & e_{3n} \\ e_{41} & e_{42} & e_{43} & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & e_{(n-1)n} \\ e_{n1} & e_{n2} & e_{n3} & \cdots & e_{n(n-1)} & 0 \end{pmatrix} \quad , e_{ij} = e_{ji} \ , i \neq j \text{ and } i,j = 1,2,\dots n \qquad \dots (14)$$

where $Digits(X)$ in Eq. (13) is the number of digits in *X*. This method significantly enhances security by ensuring the integrity and authenticity of the public key during the exchange, preventing tampering and unauthorized access, as explained in the AKACG algorithm.

5.1 Authentication Key Agreement Based on Complete Graph (AKACG) Algorithm

The AKACG algorithm enhances the security of the DH protocol by protecting the public key exchange process against MITM, replay, brute force, and discrete logarithm attacks. It achieves this by verifying the integrity of exchanged data to ensure tampering is prevented. The algorithm utilizes properties of the complete graph K_n, as described below.

AKACG Algorithm

```
Input public values
•    $p$  ← A large prime
•    $g$  ← A primitive root
Output
•    $K$  ← A shared secret key between two parties.

Step 1: Start
Step 2: Select $(n, v_1)$ ← (a pair of passwords jointly agreed upon by Alice and Bob)
Step 3: Alice (1st party)

1.   Selects randomly:
    i.   $X_A < p$ ← (Her private key)
    ii.  $v_i, i = 2, \dots, n$ ← (The remaining vertices of $K_n$)

2.   Computes:
    i.   $w = n \times v_1$
    ii.  $n_e$ ← (Using Eq. (4)).
    iii. $e_{1i}, i = 2, \dots, n$ ← (The ($n$-1) edges of $K_n$ using Eq. (5))
    iv.  $Y_A = g^{X_A} \bmod p$ ← (Her public key)
    v.   $u$ ← (The number of digits in $v_1$)
    vi.  $Sw$ ← The sum digits of $w$

3.   If $Sw$ even
         Use formula (6) to compute $Yc_A$ ← The combined public key of Alice
     Else
         Use formula (7) to compute $Yc_A$

4.   Sends $Yc_A$ to Bob
```

```
Step 4: Bob (2nd party)
    1. Selects randomly $X_B < p$ ← (His private key)
    2. Receives $Yc_A$
    3. Finds
        i. $Y_A$ using Eq. (8).
        ii. $n_e$
    4. Calculates $n_e - n^2$
    5. If ( $n_e - n^2) = \frac{(n-1)(n-2)}{2}$
    go to Step 4(6)
Else
    go to step 7 ← Reject
    6. Computes
        i. $v_i = v_1 - e_{1i}$ , $i = 2,3,\dots,n$
        ii. $e_{ij} = |v_i - v_j|$ , $i < j$ , $i = 2,3,\dots,n-1$ , $j = 3,4,\dots,n$
        iii. $Y_B = g^{X_B} \, mod \, p$ ← Bob public key
        iv. $Sw$
    7.      If $Sw$ even
        Use formula (11) to compute $Yc_B$ ← The combined public key of Bob
                Else
        Use formula (12) to compute $Yc_B$
    8.          Sends $Yc_B$ to Alice.

Step 5: Alice
    1. Finds $Y_B$ ← Using Eq. (13)
    2. Extracts $e_{ij}$ in Step 4 (6)
    3. Computes $A$ ← The adjacency matrix in Eq. (14)
    •  If $A$ matches the expected adjacency matrix of $K_n$
        Go to Step 6
    •      Else
        Go to Step 7 ← Rejects the session as dishonest.

Step 6: Compute the shared secret key
    1. Alice: $K = {Y_B}^{X_A} \, mod \, p$
    2. Bob: $K = {Y_A}^{X_B} \, mod \, p$

Step 7: End
```

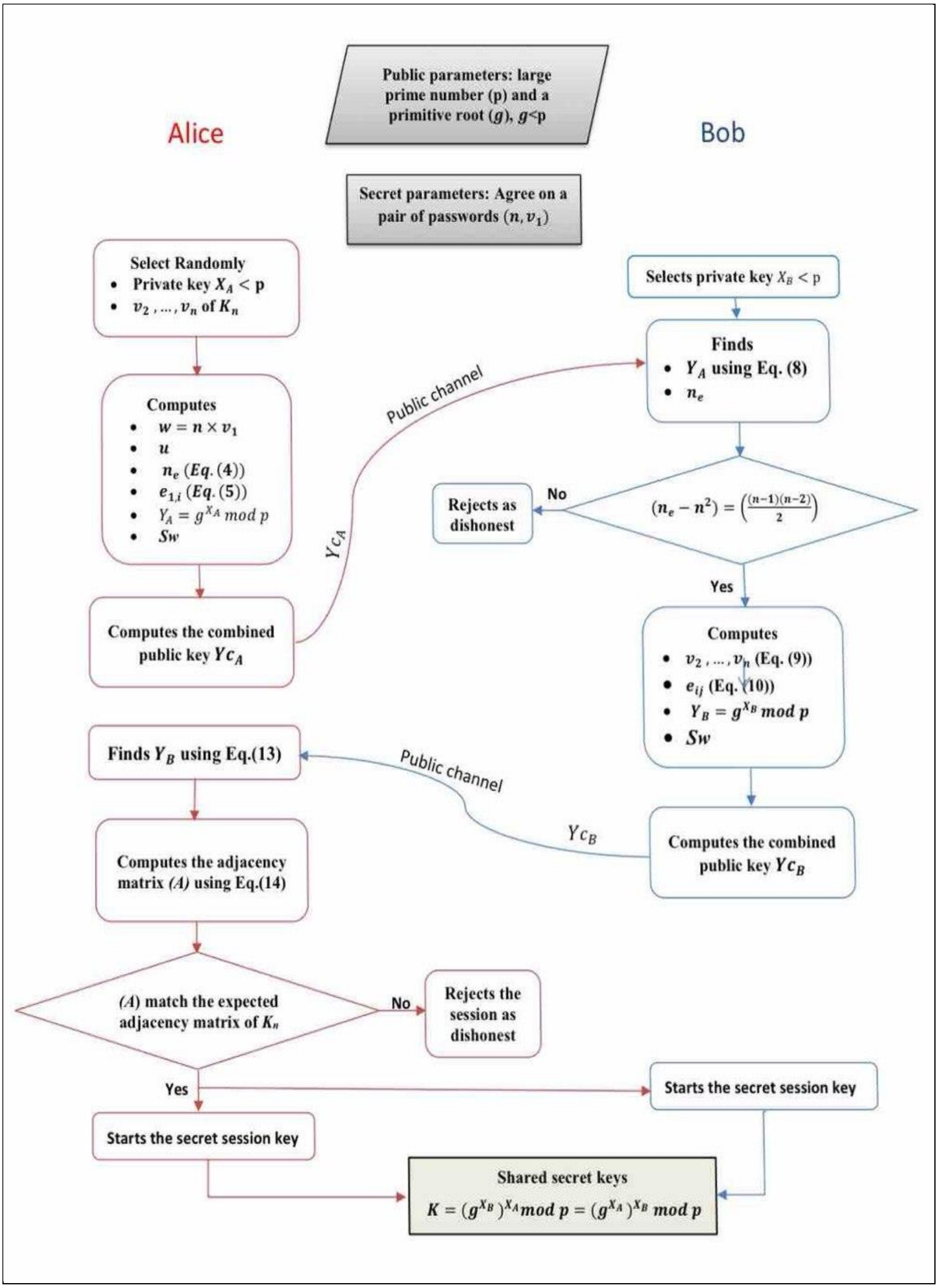
Public parameters: large prime number (p) and a primitive root (g), g<p
Alice
Bob
Secret parameters: Agree on a pair of passwords (n, v_1)
Select Randomly
Private key $X_A < p$
$v_2, \ldots, v_n$ of K_n
Selects private key $X_B < p$
Finds
Y_A using Eq. (8)
n_e
Computes
$w = n \times v_1$
u
n_e $(Eq. (4))$
$e_{1,i}$ $(Eq. (5))$
$Y_A = g^{X_A} \mod p$
Sw
Public channel
Yc_A
Rejects as dishonest
No
$(n_e - n^2) = \left(\frac{(n-1)(n-2)}{2}\right)$
Yes
Computes the combined public key Yc_A
Computes
$v_2, \ldots, v_n$ (Eq. (9))
e_{ij} (Eq. (10))
$Y_B = g^{X_B} \mod p$
Sw
Finds Y_B using Eq.(13)
Public channel
Yc_B
Computes the adjacency matrix (A) using Eq.(14)
Computes the combined public key Yc_B
(A) match the expected adjacency matrix of K_n
No
Rejects the session as dishonest
Yes
Starts the secret session key
Starts the secret session key
Shared secret keys
$K = (g^{X_B})^{X_A} \mod p = (g^{X_A})^{X_B} \mod p$

Figure 2: The proposed authentication key agreement protocol.

6. Experimental Results

The proposed AKACG algorithm was implemented, demonstrating remarkable improvements in security and efficiency, as only authorized parties can share the session key.

These enhancements were achieved with a slight increase in execution times compared to the original Diffie-Hellman algorithm. Table 1 presents the results of the algorithm when (*n*=5).

Table 1: Example of AKACG algorithm for n=5.

Alice	**Bob**
Agree a passwords $(n, v_1) = (5, 321)$	
Let p = 1593350922240001 and g = 101	
private key: $X_A = 76673$ $v_2 = 27, v_3 = 222,$ $v_4 = 105$ and $v_5 = 9$	private key: X_B =98659
$w = 1605 \xrightarrow[digit\ sum]{}$ even $n_e = 31$ $e_{12} = 294, e_{13} = 99$ $e_{14} = 216, e_{15} = 312$	Finds Y_A by using Eq. (8)
	Verification: $n_e - n^2 = \frac{(n-1)(n-2)}{2} = 6$ edges
Public key: $Y_A = 1269057120802484$ Combined public key by using formula (6): $Yc_A =$ $2941260999052167123120800312484$ Sends to Bob →	Uses Eq.(9) and Eq.(10) to calculate $e_{23} = 195, e_{24} = 78$ $e_{25} = 18, e_{34} = 117$ $e_{35} = 213, e_{45} = 96$
Finds Y_B by using Eq.(13)	Public key: $Y_B = 478882990385123$ Combined public key by using formula (11): $Yc_B =$ $195478078882018990117385213123096$ ← Sends to Alice
Computes A using Eq.(14): $A = \begin{pmatrix} 0 & 294 & 99 & 216 & 312 \\ 294 & 0 & 195 & 78 & 18 \\ 99 & 195 & 0 & 117 & 213 \\ 216 & 78 & 117 & 0 & 96 \\ 312 & 18 & 213 & 96 & 0 \end{pmatrix}$ and validates that it accurately represents Alice's adjacency matrix of K_5, as illustrated in Figure 3.	
Shared secret keys $K = (101^{98659})^{76637} mod$ 1593350922240001 $= (101^{76637})^{98659} mod$ 1593350922240001 =761689768780187	

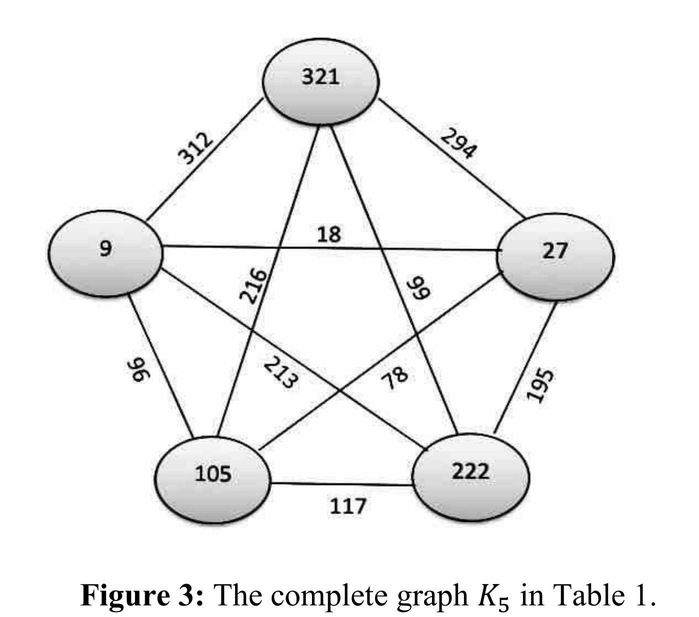

Figure 3: The complete graph K_5 in Table 1.

7. Security Analysis

In this section, we will demonstrate the effectiveness of the proposed algorithm in countering the attacks discussed in Section 3, thereby verifying its security and efficiency, as detailed below.

7.1 Man-in-the-Middle Attack

A Man-in-the-Middle (MITM) attack on DH key exchange happens when an attacker adversary eavesdrops on the public keys being exchanged between two parties. The attacker substitutes their own public keys, creating two separate keys: one with each party. This allows the attacker to decrypt, read, alter, and re-encrypt all messages, effectively becoming an intermediary who can access and manipulate the data without the parties realizing their communication has been compromised. The success of this attack depends on the attacker's ability to remain undetected while intercepting and altering the keys during the initial exchange process. [1,12]. The proposed AKACG algorithm prevents a MITM attack during DH key exchange. Alice and Bob using a mutual password represented by the pair (n, v_1) ensures that an attacker C cannot impersonate Alice or Bob.

a) Mutual Authentication: The mutual password (n, v_1) allows Alice and Bob to authenticate each other, preventing the attacker C from masquerading as either party. C lacks the mutual password and thus cannot authenticate.

b) Knowledge Limitations:

- Insufficient digit information: Even if C knows the value of n, without knowing the values of u (or v_1), The attacker C cannot determine (n-1) edges necessary to identify the vertices and the remaining edges of K_n.

- Vertex and edge ambiguity: Even if C manages to detect v_1, C still cannot determine n or the sum of digits of w. Consequently, C cannot ascertain the number of vertices, edges, or edge positions in K_n , making it impossible to reconstruct the adjacency matrix and the graph.

c) Key Security: The attacker C cannot access the components of the mutual password, w and n_e, preventing them from deriving the remaining edges of the graph or knowing Alice's or Bob's public key due to their combined public key. Even if an attacker predicts whether the sum of digits w is even or odd, they cannot determine the number of digits u, making it impossible to reach Y_A or Y_B.

Consequently, C cannot reach the key used for secure communication. Thus, using a mutual password effectively thwarts MITM attacks by ensuring that only Alice and Bob, who know the mutual password, can successfully authenticate and establish a secure communication channel.

7.2 Brute Force Attack

The Diffie-Hellman key exchange allows two parties to securely share a secret key over an insecure channel based on the difficulty of computing discrete logarithms. In a brute force attack, an attacker intercepts the public values $Y_A = g^{X_A} \, mod \, p$ and $Y_B = g^{X_B} \, mod \, p$, then exhaustively searches for the private keys X_A and X_B by trying all possible values of (x) from 1 to (p-1). This involves computing $g^x mod \, p$ for each (x) and checking against Y_A and Y_B [35].

Our algorithm presents two exceedingly difficult challenges for attackers. The first is the large size of the global prime number, which greatly increases the complexity of brute-force attempts, while the second involves the combined public keys Yc_A and Yc_B, where the large value of n adds additional complexity, making it exceptionally difficult to distinguish or extract the individual keys. In the proposed AKACG algorithm, the composition of the combined public keys depends on appropriately selecting the parameters (n, v_1, u) relative to the size of p. In the example in Table 1, p is set to 16 bits, with $(n, v_1, u) = (5{,}321{,}3)$, resulting in a 10-edge complete graph K_5. This yields a combined public key with a key space of 31 bits for Yc_A and a 33-bit for Yc_B. To break the system, an attacker must first identify the original public key Y_A and Y_B from the combined public key Yc_A and Yc_B and then apply the (p-1) attempts to find the private key which adds additional complexity to the attacker's calculations and strong security to the protocol. The proposed process is infeasible for attackers, thereby reinforcing the security of the Diffie-Hellman key exchange and preventing successful attacks or tampering. Consequently, this type of attack fails to compromise the system's security.

7.3 Discrete Logarithm Attack

Discrete Logarithm Attack aims to break cryptographic systems by solving the discrete logarithm problem. In this context, given g and $g^X mod\ p$, the goal is to find the integer (x). This problem is computationally challenging when (p) is a large prime, forming the basis for the security of the cryptographic algorithm of DH key exchange. Suppose an attacker can efficiently solve the discrete logarithm problem. In that case, they can derive private keys from public keys, compromising the security of the encrypted communications and allowing unauthorized decryption of messages [1,4]. The proposed algorithm introduces ambiguity in determining the value of g and $g^X mod\ p$ because the combined public key Yc_A or Yc_B includes additional numbers, obscuring the original public key. This confuses an attacker, making it impossible to ascertain the true public key without knowing the entire password. Partial knowledge, such as n or v_1, is insufficient for this purpose. The increased complexity of the combined public key ensures that only Alice or Bob, possessing the full password, can correctly extract Y_A or Y_B, as illustrated in the example in Table 1. Consequently, the security of the key exchange is maintained, and the attack is thwarted.

7.4 Replay Attack

A replay attack involves an attacker intercepting and retransmitting valid data to impersonate a legitimate user or gain unauthorized access. The attacker bypasses security measures by capturing and resending messages or authentication tokens without decryption. Countermeasures include using nonces, employing timestamps and implementing sequence numbers. Such attacks can compromise system integrity leading to unauthorized access, data manipulation, or risks associated with replay attacks in cybersecurity [36,4]. The proposed algorithm prevents replay attacks by ensuring message freshness through a password (n, v_1), and random remaining vertices in step 3(1) that generate the edges e_{1i} in step 3(2). Only Alice (or Bob) can embed the shared session key $K = (g^{X_B})^{X_A} mod\ p = (g^{X_A})^{X_B}\ mod\ p$ and the authentication code n_e of step 3(2) (or e_{ij} of step 4(6)), respectively. Additionally, the combined public key Yc_A or Yc_B based on w confuses the attacker and prevents replay attacks. Consequently, the proposed protocol effectively ensures security against replay attacks. Therefore, if an attacker attempts to replay or tamper with Alice, it will be flagged as a fraud because the graph lacks completeness without the remaining edge weights, resulting in session rejection. Similarly, any attempt to replay or tamper with Bob's constructed key will be identified as manipulation due to the absence of n_e, leading to session rejection. Complete authentication, achieved through constructing the full graph, effectively thwarts attacks and manipulation by ensuring that the session cannot proceed unless all components are verified.

8. Conclusion and Future work

The proposed protocol used the combined public keys Yc_A and Yc_B for Alice and Bob, respectively to ensure mutual authentication between two parties and effectively counters critical attacks that pose significant challenges in cryptography. Verifying the integrity of communication messages detects and prevents malicious modifications, tampering, and impersonation, safeguarding against major cryptographic threats. Our scheme can withstand many common attacks based on mutual passwords, authentication numbers n_e, w and e_{ij}, and the complete graph structure. For future work, the proposed protocol can be extended to three parties Diffie Hellman authenticated key agreement by modifying the combined public keys depending on complete graph properties. This extension can resist impersonation and replay attacks.

References

[1] W. Stallings, *Cryptography and Network Security, 4/E. Pearson Education India*, 2006.

[2] M. A. Cardona-López, J. C. Chimal-Eguía, V. M. Silva-García, and R. Flores-Carapia, "Key Exchange with Diffie-Hellman Protocol and Composite Hash-Functions," In *2024 12th International Symposium on Digital Forensics and Security (ISDFS),* IEEE, pp. 1-6, 2024.

[3] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no.6, pp. 644-654, 1976.

[4] W. Stallings and L. Brown, *Computer Security: Principles and Practice*, *Pearson, 7th Edition*, 2015, p.669.

[5] M. Soni and D. K. Singh, "LAKA: lightweight authentication and key agreement protocol for internet of things based wireless body area network," *Wireless Personal Communications*, vol.127, no.2, pp.1067-1084, 2022.

[6] H., Debiao, C. Jianhua and Z. Rui, "A More Secure Authentication Scheme for Telecare Medicine Information Systems," *Journal of Medical Systems*, vol. 36, pp. 1989-1995, 2012.

[7] W. Stallings, *Network security essentials: applications and standards*, *Pearson*, *6th Edition book,* 2016.

[8] D. H. Seo and P. Sweeney, "Simple Authenticated Key Agreement Algorithm" *Electronics Letters*, vol. 35, no.13, pp. 1073-1074, 1999.

[9] Z. N. Khudhair, A. Nidhal and N. K. El Abbadi, "Text Multilevel Encryption Using New Key Exchange Protocol," *Baghdad Science Journal*, vol.19, no.3, pp.619-630, 2022.

[10] F, Hazzaa, A, M. Shabut, N. H. M. Ali and M. Cirstea, " Security Scheme Enhancement for Voice Over Wireless Networks," *Journal of Information Security and Applications*, vol. 58, 102798, 2021.

[11] N.M. Khassaf and N.H.M. Ali," Multilevel Text Protection System Using AES and DWT-DCT-SVD Techniques," *Mesopotamian Journal of Cyber Security*, vol. 5, no.3, pp.913-926, 2025.

[12] C. Gupta and N. S. Reddy, "Enhancement of Security of Diffie-Hellman Key Exchange Protocol using RSA Cryptography," In *Journal of Physics: Conference Series,* vol. 2161, no. 1, p. 012014. IOP Publishing, 2022.

[13] H. H. Hadi and A. A. Neamah, "Diffie-Hellman Key Exchange Based on Block Matrices Combined with Elliptic Curves," *International Journal of Intelligent Systems and Applications in Engineering*, vol.11, no.5s, pp.353-360, 2023.

[14] X. Wang and J. Zhao, "An Improved Key Agreement Protocol Based on Chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol.15, no.12, pp. 4052-4057, 2010.

[15] M. K. Ibrahem, "Modification of Diffie-Hellman key exchange algorithm for Zero knowledge proof," *Eng. & Tech. Journal*, vol. 30, no.3, pp. 443-454, 2012.

[16] A. S. Khader and D. Lai, "Preventing Man-In-The-Middle Attack in Diffie-Hellman Key Exchange Protocol," In *2015 22nd International Conference on Telecommunications (ICT),* pp. 204-208, IEEE, 2015.

[17] F. A. AL-Maamori and M. S. Rashid, "Counting Functions to Generate the Primes in the RSA Algorithm and Diffie-Hellman Key Exchange," *Ibn AL-Haitham Journal for Pure and Applied Science*, Special issue, pp. 404-408, 2018.

[18] Y. Chen, L. López, J. F. Martínez, and P. Castillejo, "A Lightweight Privacy Protection User Authentication and Key Agreement Scheme Tailored for The Internet of Things Environment: LightPriAuth," *Journal of Sensors*, vol. 2018, no.1, 2018.

[19] N. Naher, N., Asaduzzaman and M. M. Haque, “Authentication of Diffie-Hellman Protocol Against Man-In-The-Middle Attack Using Cryptographically Secure CRC,” *In Proceedings of International Ethical Hacking Conference 2018: eHaCON 2018, Kolkata, India,* pp.139-150, Springer Singapore, 2018.

[20] Y. Zhang, D. He, L. Li, and B. Chen, “A Lightweight Authentication and Key Agreement Scheme for Internet of Drones,” *Computer Communications*, vol. 154, pp. 455-464, 2020.

[21] V. K. Yadav, R. K. Yadav, B. K. Chaurasia, S. Verma and S. Venkatesan, “MITM Attack on Modification of Diffie-Hellman Key Exchange Algorithm,” *In Communication, Networks and Computing: Second International Conference, CNC 2020, December 29–31, Gwalior, India, vol. 1502,* pp. 144-155, Springer Singapore, 2021.

[22] D. Chaudhary, T. Soni, K. L. Vasudev and K. Saleem, “A Modified Lightweight Authenticated Key Agreement Protocol for Internet of Drones,” *Internet of Things*, vol. 21, pp.100669, 2023.

[23] Z. Xia, T. Liu, J. Wang and S. Chen, “A Secure and Efficient Authenticated Key Exchange Scheme for Smart Grid,” *Heliyon*, vol.9, no.7, e17240, 2023.

[24] S. L. Nita and M. I. Mihailescu, “Elliptic Curve-Based Query Authentication Protocol for IoT Devices Aided by Blockchain,” *Sensors*, vol. 23, no.3, pp.1371, 2023.

[25] M. K. Hasan, M. M. Hasan, A. K. Budati, S. Islam, N. Safie, F. R. A. Ahmed, K. A. Abu Bakar, N. B. M. Babiker and T. M. Ghazal, “A Hybrid Key Agreement Scheme Utilized Elliptic Curve Diffie-Hellman for IoT Based Advanced Metering Environment,” *Earth Science Informatics*, pp.1-14, 2024. available on: https://doi.org/10.1007/s12145-024-01292-9.

[26] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thome, L. Valenta, B. V. Sloot, E. Wustrow, S. Z. Beguelin, and P. Zimmermann, "Imperfect forward secrecy: How Diffie-Hellman Fails in Practice," In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 5-17. 2015.

[27] N.K. Shawkat and M.A. Ahmed," On Antimagic Labeling for Some Families of Graphs." *Ibn AL-Haitham Journal for Pure and Applied Sciences*, vol.36, no.1, pp.284-291, 2023.

[28] A. Aubad and P.Rowley, “Commuting Involution Graphs for Certain Exceptional Groups of Lie Type,” *Graphs and Combinatorics*, vol.37, no.4, pp.1345-1355, 2021.

[29] J. L. Gross, J. Yellen and M. Anderson, *Graph Theory and Its Applications*. *Chapman and Hall/CRC*, 2018, p.48.

[30] G. Chartrand and P. Zhang, *A First Course in Graph Theory*. Courier Corporation, 2013.

[31] J. A. Bondy and U. S. R. Murty, *Graph theory with applications,* vol. 290, *London: Macmillan*, 1976.

[32] G. Ringel, “Theory of Graphs and Its Applications,” In *Proceedings of the Symposium Smolenice*, vol.1, p.162, 1963.

[33] Federal information processing standard (fips), " Sucre Hash Standard,"180-2. National Institute of Science and Technology, 2002.

[34] J. Wang, G. Liu, Y. Chen and S. Wang, "Construction and Analysis of SHA-256 Compression Function Based on Chaos S-Box," *IEEE Access*, vol.9, pp.61768-61777, 2021.

[35] C. Paar, and J. Pelzl, *Understanding Cryptography: a textbook for students and practitioners*, *Springer Science & Business Media*, 2009.

[36] W. N. Flayyih, “Adding Perfect Forward Secrecy to Kerberos,” *Journal of Engineering*, vol.16, no.01, pp.4593-4605, 2010.