Received: April 1, 2022. Revised: May 14 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

New Key Agreement Protocol and Cryptosystem over ECC under SET Protocol Environment in E-Commerce

Narendra Mohan Lingamgunta[1]* **Anjaneyulu Gubbala SGN[1]**

[1]*Vellore Institute of Technology, India*
* Corresponding author's Email: mohan.narendra3@gmail.com

Abstract: Cryptographic technologies are the most cost-effective and practical way to secure data transfer over an open public network. Although ECC (Elliptic Curve Cryptography) has a much smaller key size than standard cryptography methods and most of the research uses the cryptography technique to offer information authentication and integrity. Additionally, the computing effort required by EC- Diffie-Hellman (EC-DH) for equivalent key lengths is comparable and provides additional security benefits. This study provides a New Key Agreement Protocol with EC-Diffie-Hellman in the SET (Secure Electronic Transaction) environment (NKAP-ECDH-SET) to improve security. This NKAP-ECDH-SET protocol successfully implemented the key in the SET environment for digital payments and was used to encrypt and decrypt the confirmation evidence. This NKAP-ECDH-SET Protocol is used in an E-Commerce payment gateway to provide high-level security analysis, quick authentication, and security for the proposed cryptosystem. The proposed NKAP-ECDH-SET produces better security enhancement in terms of communication cost (1251 bits), computational time (4.27 ms), and data sizes (165 bytes), when compared to existing Password-Based Key Agreement and American Standard Code for Information Interchange-ECC protocols.

Keywords: Elliptic curve cryptography, E-commerce, Security, Secure electronic transaction, Diffie-hellman.

1. Introduction

For the past few years, the Internet is rapidly gaining popularity due to its accessibility for a variety of collaborative installation tool applications such as electronic messaging, electronic games, e-learning, e-commerce distributed simulation, and so on [1]. These programs offer dependable services and provide a guarantee for the messages which were delivered on time [2]. Basic security elements such as security and confidentiality, integrity, communication, and entity identification can be included in a reliable group communications system [3]. These services were not efficient to provide a secure and effective group key management solution. As a result, group key management is required for effective group key agreement [4]. The key agreement protocol allows all members of a group to agree on a single session key by allowing them to communicate securely over an insecure network [5]. There are three sorts of key management schemes for groups: (1) centralized, (2) distributed, and (3) contributory. A centralised group key management system [6, 7] is a simple key management protocol in which a single entity (or a small number of elements) produces and distributes keys to team members over a secure communication channel.

In many cases, the key agreement process is acceptable because most scenarios allow for continued secure operation on a single-source platform [8]. In a distributed form of group key agreement protocol, each group member can construct the group key independently. On the other hand, in a participatory group key agreement approach, each group member contributes equally to the development of the group key [9]. A key cryptography approach for safeguarding communication in cooperative applications through open networks is group key agreement [10]. Over the last few years, many groupware collaboration solutions have been presented, and not all of them can easily meet security and performance demands [11].

As a result, a unique key agreement protocol for network service groups was established. Because group key production takes the least amount of time in calculations, the suggested protocol is built utilizing elliptic curve cryptography techniques [12], [13]. The ECC protocol overcomes the previous equivalent approaches [14] because it consumes fewer computer resources and is resistant to various attacks. As a result, ECC with a reduced key length outperforms the existing RSA in terms of processing efficiency [15]. The following are the major contributions of this study,

- The proposed NKAP-ECDH-SET provides a high level of security while being relatively quick in terms of authentication, communication, and computing cost.
- This study provides a new EC-DH technique for validating and ensuring the integrity of data.
- Authenticated key agreement procedures are critical for assuring the security of transactions across a shared network with large amounts of data volumes being exchanged.

The organization of this research is given as follows; Section 2 represents the literature review of encryption and decryption work related to security. Section 3 stated the working operation of the SET environment. Section 4 represents the mathematical equations for the proposed NKAP-ECDH-SET with its working operations. Section 5 analyzes the result and discussion of the key authentical protocol. Finally, the conclusion is stated in Section 6.

2. Literature review

Zirui Qiao [16] developed the Forking lemma in the random oracle to show a new Improved Secure Transaction Scheme With Certificateless Signature (CLS) Cryptographic scheme for improving the security Primitives. Here, an improved CLS technique was proposed to accomplish security payments of wearable devices in open networks to increase security. In addition, the key techniques for developing a secure CLS system are demonstrated. But this CLS system cannot maintain its stated security for malicious users and fabricate effective initial on some fresh messages through a public key auxiliary attack.

Loïc D. Tsobdjou [17] developed New Mutual Authentication and Key Agreement Protocol with ECC for Mobile Client-Server Environment (MAKA-ECC). This is developed for resource-constrained mobile devices which was intended to be lightweight. Also provides a formal and informal security analysis of the proposed protocol. Session key security, perfect forward secrecy, user anonymity, replay and insider attacks are some of the security features measured in this research. As a result, the suggested protocol is safe, efficient, and appropriate for use in mobile situations. While at the time of decision making, user privacy is not guaranteed by this protocol.

Ping Liu [18] developed a Secured Password-dependent Key Agreement Scheme for the Edge Cloud for security improvement. There have been no bilinear reduction procedures in pKAS, a password-dependent key agreement system that used ECC. The system beginning phase, registration stage, authentication, and key agreement step, and offline key reset stage are the 4 stages of pKAS. The suggested pKAS simply requires delivering the message repeatedly, resulting in significant transmission bandwidth savings. To secure the confidentiality and reliability of the message, as well as the anonymity of identification, signcryption, identity authentication, and hash operations are used. However, it is unable to get the previously used session key without resolving the elliptic curve discrete logarithm problem (ECDLP) which increases the communication cost.

Yuting Li, Qingfeng Cheng, and Xinghua Li [19] showed the Assessment and development of key exchange and authorization protocols in a client-server context to improve security enhancement. The HEEL standard was investigated from a security standpoint, and various insecure vulnerabilities were discovered. Then make some minor adjustments to the HEEL protocol, creating the iHEEL protocol, which is a second key exchange and user authentication mechanism. Based on the CDH hypothesis, the system is an ID-based strategy. It protects against key compromise impersonation and the exposure of ephemeral private keys. The random oracle model illustrates its integrity. The authentication value is identical to the session key. However, the value was sent via a public channel, sometimes, there is a risk in that session which consumes large amount of data volumes.

Vivek Kumar [20] proposed a Pairing Free Identity dependent Two Party Authenticated Key Agreement Protocol through Hexadecimal Extended ASCII Elliptic Curve Cryptography to increase security. (ASCII-ECC). The projected design has been built suitably, with enhanced security and lower costs. The enhanced ASCII code description of the user's identity strengthens the protocol's security. When the effectiveness of this method was evaluated in the Bin and Balls Verification system, the ASCII-ECC strategy has a clear route to enhance security.

This procedure fails to overcome the problem of minimizing the computational time and consequently increases the priority-based data response times, because of the nonlinear combinations.

3. Working operation of SET environment

E-Commerce is demonstrated by the widespread use of the internet and contains an infinite impression on commercial facilities and industry. Also presented the SET technology for handling online MasterCard securing with considerable interest. Both cardholders and dealers should register first with the CA (Certificate Authority) for buying or selling on the internet. Once the registration process is complete, the cardholder and merchant can begin attempting transactions, which require nine fundamental steps during this protocol's simplified version [21].

1. The customer looks through the website and determines what to buy.
2. Customer delivers purchase and payment details in one message:

a. Purchase Order - This section is for the merchant to fill out.
b. Card Data - This section is just for the merchant's bank.

3. The merchant sends the card details to their bank (part b).
4. The merchant's bank verifies payment authorization with the issuer.
5. The Issuer issues a letter of permission to the Merchant's bank.
6. The merchant's bank issues permission.
7. The merchant completes the order and emails the consumer a confirmation.
8. The merchant obtains a copy of the transaction from their bank.
9. The MasterCard bill (invoice) is printed by the issuer and delivered to the consumer.

4. Proposed NKAP-ECDH-SET protocol

In SET, the ECC approach is employed for all public-key encryption and digital signatures, as earlier mentioned. Rivest-Shamir-Adleman encryption (RSA) has a high computational cost and a large message overhead. For PReq generation, single public-key encryption and a cryptographic verification synthesis are required, hence the processing cost is fractional matrix multiplication. As a result, this study established A New Key Authentication Protocol with ECDH under the SET Environment entitled NKAP-ECDH-SET to improve the security of E-Commerce systems.

4.1 New proposed hard mathematical problem over ECC

Assume, E be an elliptic curve over Z_P, where P is a prime number which is demarcated by the equation $y^2 = x^3 + ax + b$, where $a, b \in Z_P$ are curve constraints and G is a prime number defined by the equation. Taking place at an elliptic curve, n is the order of G, and h is the cofactor, with the constraint $4a^3 + 27b^2 \neq 0$. Adopt the created point on the elliptic curve $Z_P \; as \; G = (x_1, y_1)$ in an elliptic curve with a big n order. Under elliptic curves, key is defined as $N = n_k G$ where n_k is an integer with $n_k < n$ and $nG = 0$is the smallest positive integer, the order n of a point G is the lowest optimistic digit. N is a point on an elliptical curve area, and this is a challenging mathematical issue. Even if $N \; and \; G$ are identified, it won't be able to determine n_k since scalar multiplication in ECC is based on multiple addition, which is built using a mix of linear and nonlinear calculations.

The inverse of point multiplication of Elliptic curves cannot be calculated under ECC. Scalar point multiplication is done in ECC using a non-invertible non-linear equation structure. The key structure is built in this research in a SET context utilizing a key agreement mechanism and an elliptic curve cryptosystem, which poses difficult mathematical challenges.

4.2 ECDH (key agreement)

The technological classes of Verifications are validated through different key agreement methods based on public key infrastructure, password, Group and Identity based connections. Ensure the key values are swapped amongst the sender and receiver, then execute encryption and decryption to prevent an unauthorized person from learning the key are all difficulties linked with key exchange and key agreement. Provide proof to the receiver that communication was secured by the person claiming to be the sender. Flow charts for the Key Agreement Protocol and an example are proposed in this article. The DHKA Protocol is unique when compared to existing protocols.

4.3 ECDH (key exchange)

Consider dual public-private key pairs were created, then, swap the public keys, and figure out the shared secret key. In earlier stages, the process of encryption and decryption using EC-dependent public-key encryption and decryption has been demonstrated. This is a difficult task that usually

demands the development of a combined encryption approach that combines ECC cryptography, ECDH key exchange, and asymmetric encryption protocol. Both private and public keys will be used to encrypt and decode data. Asymmetric encryption works like this: after encrypting data with a private key, the ciphertext can be decrypted using the equivalent public key. The aforementioned approach can be used to directly attack the RSA cryptosystem, except ECC. Because, it cannot provide an immediate encryption mechanism. As an alternative, construct a combined encryption approach through the ECDH key exchange mechanism to generate a mutual secret key for encryption and decryption. In a SET context, the shared key agreement protocol between consumer and dealer in E-commerce applications is defined as follows, assuming C is a client and D is a merchant.

Initially, choose binary private keys $n_\alpha < n$, $n_\beta < n$ of the customer and merchant through settings as

$$1 < n_\alpha \leq n-1, 1 \leq n_\beta \leq n-1$$

- Initially generate the Customer Private Key is set to n_α.
- Then produces a public key $N_A = n_\alpha G$ and transfer it to the merchant.
- Choosing the Merchant's Private Key as n_β
- The merchant produces a public key through the subsequent arrangement: $N_B = n_\beta G$ and sends it to the customer
- The produced keys are replaced in the process mentioned below.
- Acknowledged the merchant's key from the purchaser: $N_B = (x_B, y_B)$
- Merchant Created Secret Key = $K_M = n_\alpha N_B = n_\alpha n_\beta G$
- The consumer collects a key from the dealer: $N_A = (x_A, y_A)$
- Customer Produced Secret Key = $K_C = n_\beta N_A = n_\beta n_\alpha G$ $K_C = K_M = K$ might be established mathematically. $E_P(a,b)$ which contains a mutual secret key.

4.4 Flow chart of ECDH key agreement protocol

The flowchart for ECDH Key Agreement Protocol is given in Fig. 1. The steps are mentioned as follows,

- Initially, the Purchaser Private Key was Nominated as n_α.

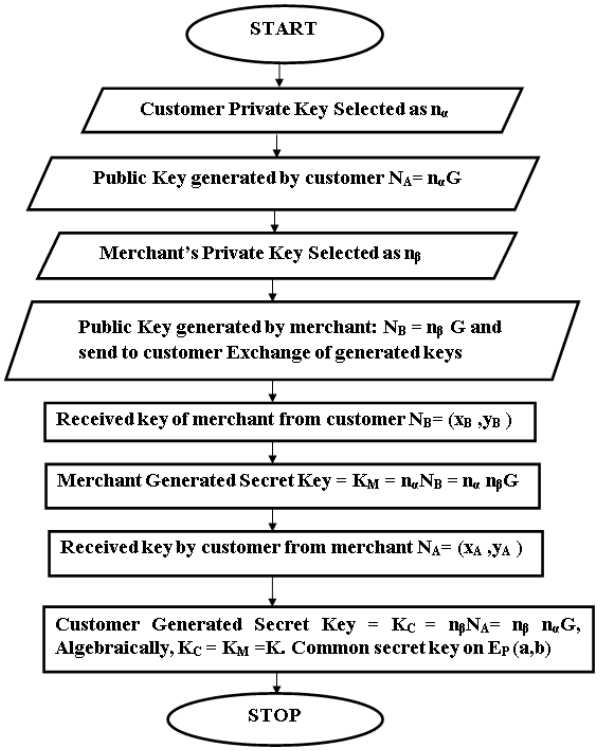

Figure. 1 Flowchart for ECDH key agreement protocol

- Then consumer-produced the Public Key: $N_A = n_\alpha G$ and transfer to the dealer
- Dealer's Private Key Designated as n_β.
- Dealer produced the Public Key: $N_B = n_\beta G$ and share to the consumer
- Interchange of generated keys proceeds in the subsequent stages
- Established key of dealer from consumer: $N_B = (x_B, y_B)$
- Merchant Generated Secret Key = $K_M = n_\alpha N_B = n_\alpha n_\beta G$. Received key through the consumer from a dealer: $N_A = (x_A, y_A)$
- Consumer Created Secret Key = $K_C = n_\beta N_A = n_\beta n_\alpha G$.
- Mathematically, it revealed that $K_C = K_M = K$. Mutual secret key on $E_P(a,b)$.

4.5 Example of ECDH key exchange protocol

Key generation: The common procedure of Elliptic curve is represented as Eqs. (1) and (2):

$$y^2 = x^3 + ax + b \quad (1)$$

where $4a^3 + 27b^2 \neq 0$

1. Select $p = 17, n = 2, a = 2, b = 2$ and Set

$$p = 17, y^2 mod_p = x^3 + 2x + mod_p \quad (2)$$

where $(4a^3 + 27b^2)mod_{17} = 4 \neq 0$

2. Take $G = (x_1, y_1) = (5,1)$
3. In group G , then calculate Eq. (3) as,

$$2G = G + G = (x_3, y_3)$$
and
$$S = \frac{3_{x1^{2+a}}}{2_{y_1}} \, mod_p \, x_3$$
$$= S_2 - 2x_1(mod_p), y_3$$
$$= S(x_1 - x_3) - y_1(mod_p) \qquad (3)$$

4. Finally, $S = {}^{(3.52+2)}/_{2.1} = ({}^{77}/_{2})mod_{17} = 9.9, 13. x^3 = 13^2 - 2.5 = 159 \, mod_{17} = 6, y^3 = 13(5-6) - 1 = -14mod_{17} = 3.$
5. Then $2G = (6,3)$ likewise the calculation is mentioned as follows, $3G = (10,6), 4G = (3,1), 5G = (9,16), 6G = (16,13), 7G = (0,6), 8G = (13,7), 9G = (7,6), 10G = (7,11).$

The process of ECDH Key exchange are described as follows:

- The Private Key of the consumer is designated as $n_\alpha = 3$.
- The public key is created by the consumer: $N_A = 3, G = (10,6$ and transferred to the dealer.
- Dealer's Private Key ($n_\beta = 9$) is designated.
- The dealer produces a public key through the subsequent constraints: $N_B = 9, G = (7,6)$ and directs it to the customer. The produced keys are replaced in the following process
- Established the merchant's key from the consumer: $N_B = (7,6)$
- Dealer Created Secret Key$= K_M = 3.9G = 27G = 8G = (13,7)$
- Consumer obtains key from merchant: $N_A = (10,6)$
- Consumer Produced Secret Key $= K_C = 9.3G = 27G = 8G = (13,7), K_C = K_M = (13,7)$ can be established algebraically. $E_{17}(2,2)$ has a mutual secret key.

The aforesaid algorithm's flow chart is presented in Fig. 2.

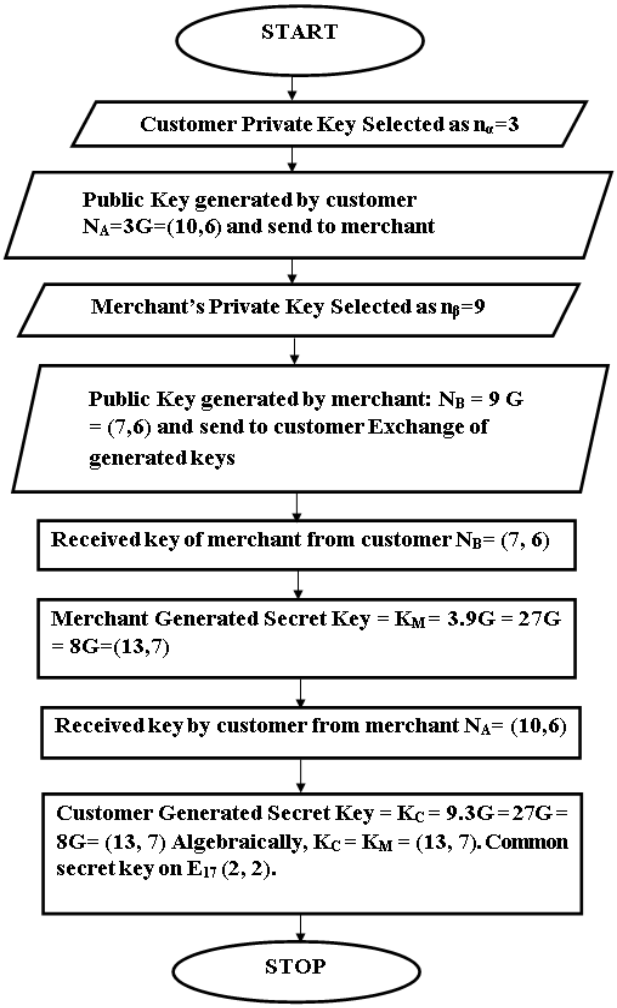

Figure. 2 Flowchart for ECDH key exchange protocol example

4.6 Elliptic curve cryptography

The key agreement protocol was discussed in the previous section. Now, SET Environment is used to create a new cryptosystem that completely depends on ECC. The text message m is encoded by a point in this case $T_m = (x, y)$. Under SET, this is encrypted as plain text by the public key and decrypted as elliptic curve domain by the private key in point creation

Key Generation: Assume that E is an elliptic curve over Z_P, where P is a prime number. Assume that the generator point on the elliptic curve $Z_P \; as \; G = (x_1, y_1)$ in an elliptic curve with a big n order. Under elliptic curves, define the key as $N = n_k G$ where n_k is an integer with $n_k < n$, and the order n of a point G on an elliptic curve is the smallest positive integer such that $nG = 0$. In the elliptic curve $E_P, a, b \; and \; P$ are global public elements (a, b). Here, G represents a point on a set of $E_P(a, b)$ parameters. Customer public key N_A , private key n_α, and merchant public key N_B, private key n_β are used here.

Encryption: The consumer selects the private key as n_α and Generates a public key as $N_A = n_\alpha G$. At that time encrypts the message Tm through

preferred numeral k and estimates Ciphertext as Eq. (4).

$$\text{Ciphertext } C_A = \{T_m + kN_B + kG\} \quad (4)$$

And transferred to the dealer for the process of decrypting. Here consumer utilized the dealer's public key N_B correspondingly.

Decryption: Once receiving the encrypted message from consumer, dealer decrypts the code through private key n_β that is reproduced by means of second module of cipher text and deducted from first section, Eq. (5) is written as

$$T_m + kN_B - n_\beta kG = T_m + kn_\beta G - kn_\beta G = T_m \quad (5)$$

At this point consumer covered the message by totaling kN_B as extra safety. Here consumer recognizes the k value, although N_B is the public key, anyone can eliminate the cover kN_B. If an attacker identifies private key n_β to modify the message, initially the attacker needs to compute integer k and kG, that is too difficult to agree.

4.7 EC cryptography example

Key Generation: The global public essentials are $P = 17\ in\ E_{17}(2,2)$ for elliptic curve $y^2 = x^3 + 2x + 2$, and $G = (5,1)$. Consumer needs to send a message to dealer that is programmed in elliptic point $T_m = (16,13) = 6G$) ,customer private key is assigned as $n_\alpha = 3$ and public key is assigned as $N_A = 3G$, merchant private key $n_\beta = 9$ and public key is $N_B = 9G$.

Encryption: If $G = (5,1)$consumer translates the message in elliptic point $T_m = (16,13)$

At that point, choose a number $k = 5$ and $kG = 5G = 5(5,1) = (9,16)$, evaluate the Eq. (6) as

$$T_m + kN_B = 6(5,1) + 5(7,6) = 6G + 5.9G = 51G = 13G = (16,4) \quad (6)$$

During encryption process, encrypt a ciphertext $C_A = 13G, 5G = (16,4), (9,16)\ with\ C_A = 13G, 5G = (16,4), (9,16)$. This should be sent to the merchant to be decrypted.

In this case, the consumer utilized the merchant's public key, which is $N_B = 9G = (7,6)$.

Decryption: Later the converted message was received, the merchant decrypts it using private key $n_\beta = 9$ by multiplying the additional element of the ciphertext by a private key and subtracting it from the first element which is described as Eq. (7):

$$T_m + kN_B - n_\beta kG = 6G + 5.9G - 9.5G = 6G = (16,13) = T_m \quad (7)$$

As a final point, the messages are exposed at the end.

In SET environment, the proposed NKAP-ECDH has a leading dynamism in the security of automated communications and is used to identify and authenticate the integrity of the information. Because it offers slighter key sizes and quicker computation over conventional RSA, NKAP-ECDH-SET is used as the foundation for validation and confirming the data reliability in SET. As a result, SET offers a high level of efficiency in authentication and security. In addition, the proposed framework was strengthened by evaluating security attacks such as high security and effective authentication.

5. Result and discussion

This segment compares the proposed protocol with conventional authentication in terms of key creation, verification, and computation time. The Matlab tool R2018a is exploited on a Windows 10 system with the recommended NKAP-ECDH-SET methodology. The NKAP-ECDH-SET scheme's security analysis and demonstration are covered in this section. Furthermore, the suggested NKAP-ECDH-SET has been demonstrated to be resistant to some attacks. To increase security, the proposed work uses an NKAP-ECDH-SET dependent authentication technique for wireless sensor networks. Table 1 and 2 illustrate the simulation effects of the proposed NKAP-ECDH-SET and conventional ASCII-ECC [20] in terms of key verification and generation time, energy usage which varied through message length and various key sizes. Fig. 3 depicts a plot of key verification time.

Table 1. Performance of key verification time

Key Size	Key Verification Time (ms)	
	Existing ASCII-ECC [20]	**Proposed NKAP-ECDH-SET**
8	1824	1504
16	1950	1678
32	2000	1748
64	2284	1996
128	2748	2066
256	2990	2402
512	3158	2886
1024	3324	3024

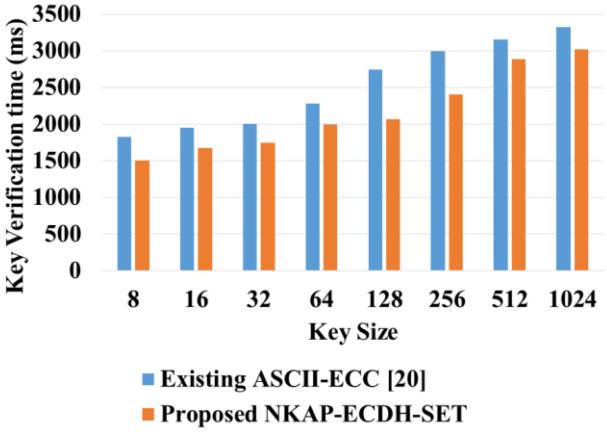

Figure. 3 Performance analysis of key verification time

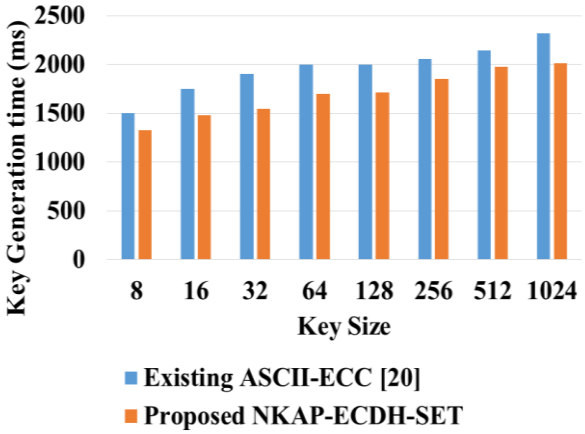

Figure. 4 Performance analysis of key generation time

Table 2. Performance of key generation time

Key Size	Key Generation Time (ms)	
	Existing ASCII-ECC [20]	**Proposed NKAP-ECDH-SET**
8	1500	1330
16	1750	1478
32	1900	1548
64	2000	1696
128	2000	1716
256	2056	1850
512	2146	1976
1024	2320	2016

Table 3. Performance of energy consumption

Key Size	Energy Consumption (J)	
	Existing ASCII-ECC [20]	**Proposed NKAP-ECDH-SET**
32	51	37
64	55	40
128	60	42
256	67	47
512	75	52
1024	71	56
2048	87	62
4096	93	74

The key can be up to 1024 bits in length. A key with 1024 bits takes 7900 milliseconds to verify, which is much faster than current approaches. The BIBA method verifies an 8-bit key in 8000 milliseconds and a loss-tolerant key in 9000 milliseconds. When the suggested NKAP-ECDH-SET is compared to the current ASCII-ECC [20], it is clear that the proposed NKAP-ECDH-SET generates keys faster.

Fig. 4 shows the key generation of the proposed technique as a function of key size variation. Whenever the key number is larger in bits, the graphs show a linear increase in the period. Although the suggested solution takes longer to generate keys and verify them, it takes less time overall. Fig. 5 shows the energy consumption of the proposed NKAP-ECDH-SET with existing ASCII-ECC [20]. The calculation for performance assessment of energy consumed when sending data along with the key size is shown in Table 3. Table 3, clearly shows that the proposed NKAP-ECDH-SET consumes less energy consumption when compared to the existing ASCII-ECC [20].

Fig. 3, 4, and 5 showed an analysis of the proposed scheme's effectiveness with existing technologies ASCII-ECC [20] in terms of key

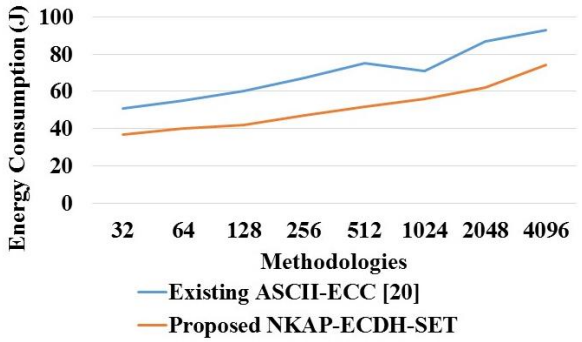

Figure. 5 Performance analysis of energy consumption

verification time, generation time, and energy consumption. When comparing the proposed NKAP-ECDH-SET to the existing ASCII-ECC [20] protocol, the aforementioned numbers clearly illustrate that the proposed NKAP-ECDH-SET obtains superior performance.

5.1 Comparative analysis

Table 4 compares the performances of communication costs, data volumes and total computational cost of the proposed NKAP-ECDH-SET with existing methods such as MAKA-ECC [17], pKAS [18], iHEEL [19] and ASCII-ECC [20]. All the existing methods are analysed with performance

Table 4. Comparative analysis

Protocols	Communication Cost (bits)	Data Volumes (bytes)	Total Computational Time (ms)
MAKA-ECC [17]	1365	192	4.91
pKAS [18]	1472	186	4.78
iHEEL [19]	1498	180	5.13
ASCII-ECC [20]	1302	174	5.00
Proposed NKAP-ECDH-SET	1251	165	4.27

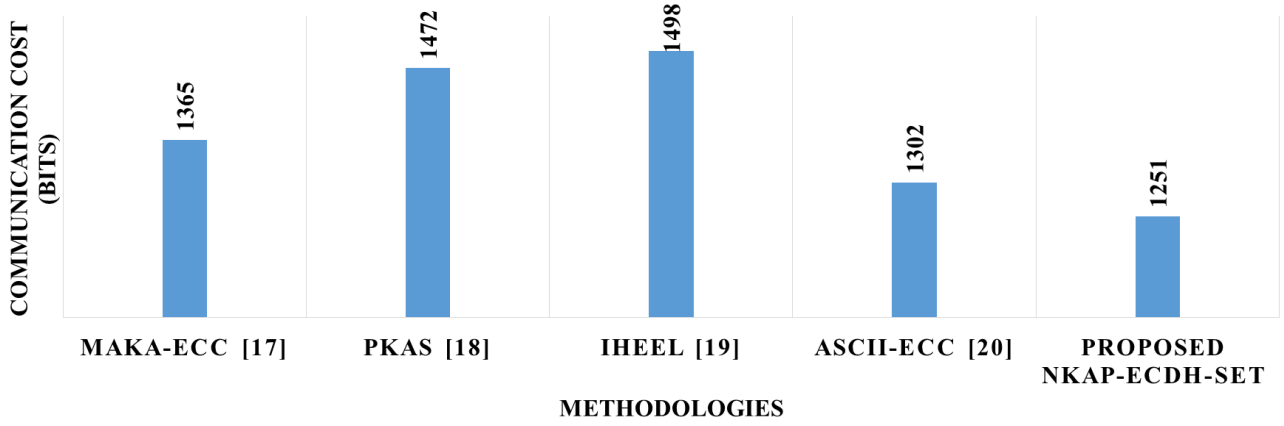

Figure. 6 Performance analysis of communication cost

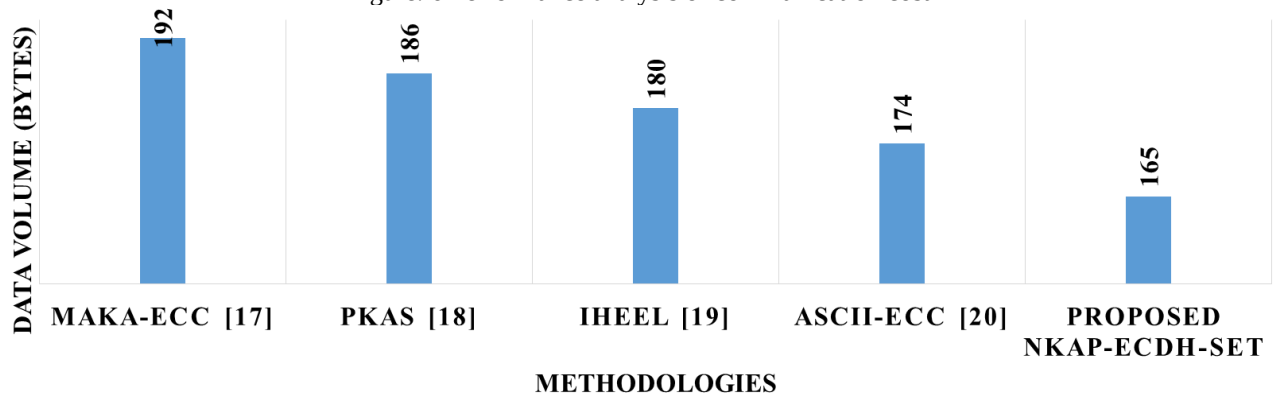

Figure. 7 Performance analysis of data volume

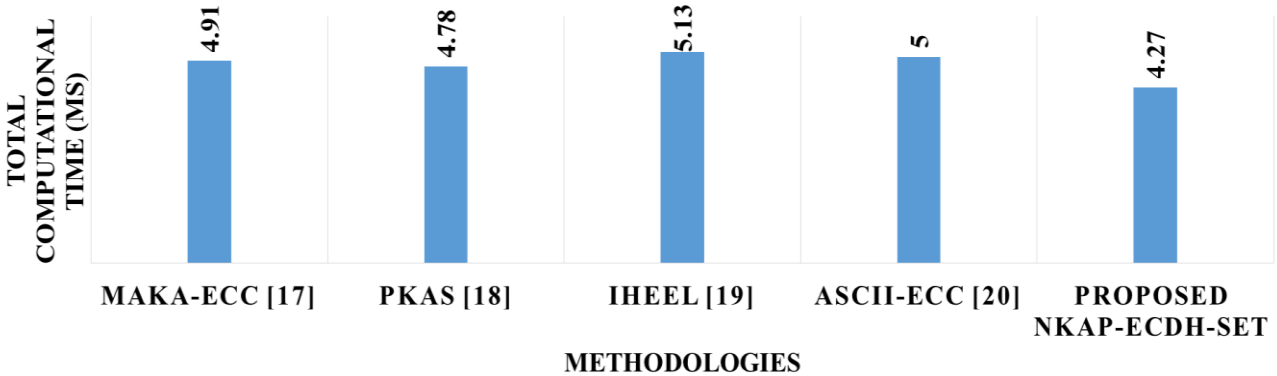

Figure. 8 Performance analysis of total computational cost

parameters such as communication cost, data volumes and total computational time to prove that proposed NKAP-ECDH-SET is a superior one which is discussed briefly in below section.

5.1.1. Comparison of communication cost

While considering the performance of communication cost, the conventional methods are

subject to copy, insider attacks and requires a lot of bandwidth. Furthermore, the proposed NKAP-ECDH-SET outperforms the existing methods. As a consequence, the NKAP-ECDH-SET recommended for user and server verification is preferable. Table 4 tabulated the comparative study of the proposed NKAP-ECDH-SET, which achieved the communication cost of 1251 bits, which is fewer than the existing methods which attains the message count of higher amount of communication cost. Figure 6 shows the performance analysis of communication cost with different methods.

5.1.2. Comparison of data volumes

While considering the data volumes, there are two sections to the proposed NKAP-ECDH-SET system. Initially, establish security features for the system calculations which are accurate or not. Then communication between participants, which entails choosing a session key and identifying one another. From the Table 4, it clearly illustrates that the proposed NKAP-ECDH-SET achieves the data volume of 165 bytes, which is much better than the existing methods that is shown in Fig. 7.

5.1.3. Comparison of total computational time

While considering the total computational time, the proposed NKAP-ECDH-SET protocol contains a higher security standard with less computational time as shown in Table 4. Fig. 8 shows the accomplishment of proposed NKAP-ECDH-SET which achieved the better computational time of 4.27 ms, whereas the existing method accomplishes at certain intervals which is not superior to the proposed NKAP-ECDH-SET protocol.

6. Conclusion

In this research, the NKAP-ECDH-SET technique is offered as a new key exchange and user authentication scheme to undertake a security analysis of corresponding improvements to the security problems. While operating in the SET context, the proposed technique will increase the successful functioning of the key agreement protocol, ECC, which has been successfully used in E-commerce applications. Finally, compare the security aspects of the proposed NKAP-ECDH-SET scheme to those of numerous other protocols, such as existing MAKA-ECC, ASCII-ECC, pKAS, and iHEEL, to assess communication costs, total computational time and data volumes. In terms of security qualities and data quantities, the suggested NKAP-ECDH-SET protocol reaches 165 bytes, whereas the existing protocol consumes large amount of data volumes. While considering the performance of communication cost, the proposed NKAP-ECDH-SET achieves less cost of 1251 bits, whereas existing methods accomplishes at very high rate. By deliberating the computational time, the proposed NKAP-ECDH-SET achieves the result in 4.27 ms, which is better than existing protocols. All the results, clearly show that the proposed NKAP-ECDH-SET attains the best results to improve security while related to conventional protocols. In the future, the suggested key authentication procedure will be technologically advanced and evaluated for wireless communication performance and other security needs.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

References

[1] A. O. Sharif, D. A. Mood, and M. Nikooghadam, “An enhanced anonymous and unlinkable user authentication and key agreement protocol for TMIS by utilization of ECC”, *International Journal of Communication Systems*, Vol. 32, No. 5, p. e3913, 2019.

[2] M. Qi, J. Chen, and Y. Chen, “A secure authentication with key agreement scheme using ECC for satellite communication systems”, *International Journal of Satellite Communications and Networking*, Vol. 37, No. 3, pp. 234-244, 2019.

[3] M. Qi and J. Chen, “Anonymous biometrics-based authentication with key agreement scheme for multi-server environment using ECC”, *Multimedia Tools and Applications*, Vol. 78, No. 19, pp. 27553-27568, 2019.

[4] M. Ouaissa, M. Ouaissa, and A. Rhattoy, “An Efficient and Secure Authentication and Key Agreement Protocol of LTE Mobile Network for an IoT System”, *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 4, pp. 212-222, 2019, doi: 10.22266/ijies2019.0831.20.

[5] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous Lightweight Chaotic Map-Based Authenticated Key Agreement Protocol for Industrial Internet of Things", *IEEE Transactions on Dependable and Secure Computing*, Vol. 17, No. 6, pp. 1133-1146, 2020.

[6] S. Qiu, X. Guosheng, A. Haseeb, X. Guoai, Q. Xinping, and X. Hong, "An Improved Lightweight Two-Factor Authentication and Key Agreement Protocol with Dynamic Identity Based on Elliptic Curve Cryptography", *KSII Transactions on Internet and Information Systems*, Vol. 13, No. 2, p. 2019.

[7] C. M. Chen, B. Xiang, K. H. Wang, K. H. Yeh, and T. Y. Wu, "A Robust Mutual Authentication with a Key Agreement Scheme for Session Initiation Protocol", *Applied Sciences*, Vol. 8, No. 10, p. 1789, 2018.

[8] J. Mo, Z. Hu, H. Chen, and W. Shen, "An Efficient and Provably Secure Anonymous User Authentication and Key Agreement for Mobile Cloud Computing", *Wireless Communications and Mobile Computing*, Vol. 2019, pp. 1-12, 2019.

[9] D. Xu, S. Zhang, J. Chen, and M. Ma, "A provably secure anonymous mutual authentication scheme with key agreement for SIP using ECC", *Peer-to-Peer Networking and Applications*, Vol. 11, No. 5, pp. 837-847, 2018.

[10] A. Hassan, A. A. Omala, M. Ali, C. Jin, and F. Li, "Identity-Based User Authenticated Key Agreement Protocol for Multi-Server Environment with Anonymity", *Mobile Networks and Applications*, Vol. 24, No. 3, pp. 890-902, 2019.

[11] A. G. Reddy, A. K. Das, V. Odelu, A. Ahmad, and J. S. Shin, "A Privacy Preserving three-factor authenticated key agreement protocol for client-server environment", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, No. 2, pp. 661-680, 2019.

[12] J. Mo and H. Chen, "A Lightweight Secure User Authentication and Key Agreement Protocol for Wireless Sensor Networks", *Security and Communication Networks*, Vol. 2019, pp. 1-17, 2019.

[13] A. A. Alamr, F. Kausar, J. Kim, and C. Seo, "A secure ECC-based RFID mutual authentication protocol for internet of things", *The Journal of Supercomputing*, Vol. 74, No. 9, pp. 4281-4294, 2018.

[14] Y. Lu, G. Xu, L. Li, and Y. Yang, "Anonymous three-factor authenticated key agreement for wireless sensor networks", *Wireless Networks*, Vol. 25, No. 4, pp. 1461-1475, 2019.

[15] S. Khatoon, S. M. M. Rahman, M. Alrubaian, and A. Alamri, "Privacy-Preserved, Provable Secure, Mutually Authenticated Key Agreement Protocol for Healthcare in a Smart City Environment", *IEEE Access*, Vol. 7, pp. 47962-47971, 2019.

[16] Z. Qiao, Q. Yang, Y. Zhou, and M. Zhang, "Improved Secure Transaction Scheme With Certificateless Cryptographic Primitives for IoT-Based Mobile Payments", *IEEE Systems Journal*, pp. 1-9, 2021.

[17] L. D. Tsobdjou, S. Pierre, and A. Quintero, "A New Mutual Authentication and Key Agreement Protocol for Mobile Client—Server Environment", *IEEE Transactions on Network and Service Management*, Vol. 18, No. 2, pp. 1275-1286, 2021.

[18] P. Liu, S. H. Shirazi, W. Liu, and Y. Xie, "pKAS: A Secure Password-Based Key Agreement Scheme for the Edge Cloud", *Security and Communication Networks*, Vol. 2021, pp. 1-10, 2021.

[19] Y. Li, Q. Cheng, and X. Li, "Analysis and improvement of a key exchange and authentication protocol in client-server environment", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 11, No. 9, pp. 3787-3799, 2020.

[20] V. Kumar, S. Ray, M. Dasgupta, and M. K. Khan, "A Pairing Free Identity Based Two Party Authenticated Key Agreement Protocol Using Hexadecimal Extended ASCII Elliptic Curve Cryptography", *Wireless Personal Communications*, Vol. 118, No. 4, pp. 3045-3061, 2021.

[21] R. Oruganti, S. Shah, Y. Pavri, N. Prasad, and P. Churi, "JSSecure: A Secured Encryption Strategy for Payment Gateways in E-Commerce", *Circulation in Computer Science*, Vol. 2, No. 5, pp. 13-17, 2017.