DOI: 10.24996/ijs.2023.64.8.39

ISSN: 0067-2904

IoT-Key Agreement Protocol Based on The Lowest Work-Load Versions of The Elliptic Curve Diffie-Hellman

Mohammed Sh. Oudah*, Abeer T. Maolood

Computer science department, university of technology, Baghdad, Iraq

Received: 20/6/2022 Accepted: 13/10/2022 Published: 30/8/2023

Abstract

A key agreement protocol (KAP) is a fundamental block in any cryptosystem since it ensures secure communication between two parties. Furthermore, KAP should include advanced features in limited-resource environments such as IoT, in which this protocol must be lightweight and efficient in consuming resources. Despite the Elliptic Curve Diffie-Hellman (ECDH) algorithm having often been considered efficient in providing an acceptable security degree with less resource consumption, it has suffered from weakness against Man-In-The-Middle Attacks (MITMA). This paper presents two versions of the Variant Elliptic Curve Diffie-Hellman (VECDH) algorithms as a key agreement protocol. The security analysis indicates that the proposed algorithm could be more robust compared to the MITMA, in addition to several security features. The proposed algorithms scale down the computation complexity by decreasing the arithmetic operations, to make the algorithms the lowest workload and suitable for application in restricted resource environments.

Keywords: IoT, Elliptic Curve Cryptography, Key Agreement

إتفاقية تبادل المفاتيح خاص بإنترنت الاشياء المتعمدة على نسخ اقل عبء لخوارزمية ديفي هيلمان المتعمدة على المنحني الاهليجي

محمد شاكر عودة * , عبير طارق مولود

قسم علوم الحاسبات, الجامعة التكنولوجية, محافظة بغداد, العراق

الخلاصة

ان اتفاقية تبادل مفاتيح التشفير من اهم اجزاء نظام التشفير لأنه يمكن اتصال آمن بين الاطراف، وعلاوة على هذا، يجب ان تتضمن هذه الاتفاقية خصائص متقدمة في البيئات التي تكون فيها المصادر محدودة مثل بيئة انترنت الاشياء. بمعنى اخر، يجب ان تكون هذه الاتفاقية خفيفة وكفؤة في استهلاك المصادر الحاسوبية. برغم من اعتبار خوارزمية ديفي هيلمان لتبادل المفاتيح المعتمدة على المنحنى الاهليجي كفوء في توفير مستوى أمني مقبول في قبال استهلاك مصادر جيدو ولكنها تعاني من ضعف امام الهجمات. هذا البحث يقدم مقترحين لتطوير هذه الخوارزمية يأخذ بعين الاعتبار جميع نقاط الضعف الخاصة بالخوارزمية الاصل. ومنها فان التحليل الامني للخوارزميتين المقترحتين يشير الى انهما أكثر مقاومة للهجمات بالإضافة الى خصائص امنية اخرى. وكذلك تمتع هذه الخوارزميات بأداء جيد يشير الى تقليل استهلاك المصادر, جاء هذا الامر من خلال تقليل عدد العمليات الرياضية المستعملة وكذلك تقليل تعقيد الخوارزمية لتصبح مناسبة للعمل في البيئات المحدودة المصادر .

*Email: cs.20.42@grad.uotechnology.edu.iq

1. Introduction

Secure communication between any two parties is achieved when each of them has the capability to ensure the legality of the other. In this context, several protocols were established. One of them, a KAP, is a security protocol used to provide a shared session key between two communication parties. This key is more important in integrating contact and boosting confidence in communication security [1].

An Elliptic Curve Cryptosystem (ECC) is an alternative model to popular public key security models such as RSA and the Diffie-Helman key exchange algorithm. ECC provides the same security level with a small key size, lower resource consumption, and faster computation [2]. ECC offers multiple security solutions, such as public key encryption and decryption algorithms; digital signature certificates by the Elliptic Curve Digital Signature Algorithm (ECDSA); and (what is in the scope of the paper) a key agreement protocol by the ECDH algorithm [3].

However, the standard ECDH algorithm suffers from a vulnerability against Man-in-the-middle attacks when all exchanged messages can be readable and modifiable by the impersonating attacker without giving any attention to legitimate users. Therefore, the EC parameters should be chosen carefully. Furthermore, the researchers suggested two solutions to make the ECDH algorithm strong enough against the mentioned attack [4]:

1. Authentication of the user’s public key: validating the user’s public key is required when it is static.
2. Temporal public key: both communication sides can produce new public keys for each communication session. This solution enables the Perfect Forward Secrecy Protocol (PFS) and reduces the algorithm’s complexity, which means it does not require extra authentication computation.

Yooni and Yoo [5] proposed a new two-party key agreement protocol (EECKE-1N) as a modification to ECKE-1N [6]. This protocol combines public key authentication and ECDH key exchange. The most important aspect is that this protocol has reduced the number of arithmetic operations in a single key-round to make the protocol usable on the lowest-cost network. It also achieved an efficient security feature such as known-key security, forward secrecy, unknown key-share resilience, and key control. In addition, EECKE-1N has the same security features that ECKE-1N enjoyed.

As a different improvement idea, Kaur and Paraste [7] proposed two enhancements for ECDH. The first one, the secret key, is a product of the multiplication between the secret key coordinates. The second improvement is exponentiation of the coordinates to encrypt a message, and the receiver computes the inverse to decrypt the cipher. That multiplication and exponential operations add strength to the algorithm, but at the same time, more execution time and resources are required to accommodate the complexity of the algorithm.

Mehibel and Hamadouche [8] proposed a new integrated algorithm that used ECDSA to authenticate the secret session key depending on two random variables. The proposed algorithm resolves the weakness of the previous integrated algorithm [9] that used a single random variable. The proposed algorithm achieves multiple security features such as mutual authentication, PFS, and more crucially, it is more immune against the man-in-the-middle attack. Also, the authors claimed that the proposed algorithm is lightweight and suitable for application in restricted resource environments.

Ripon Patgiri and Senior Member (2021) [10], proposed a new protocol called "PrivateDH" to manipulate the Man-in-the-middle weakness that standard ECDH suffers from. This protocol used the AES algorithm to encrypt the public shareable parameters of ECDH and used the RSA algorithm to retrieve the public key. The performance analysis shows that the privateDH can report a MITM attack to the receiver when he/she breaks the public key. Although the protocol has an obvious computation overhead, the protocol achieved better communication overhead, relatively. But still, this protocol does not look efficient to apply in restricted resource environments, such as the IoT.

Dar et al. [11] proposed an incorporated common shared key as an authentication procedure to make ECDH more secure and reliable against MITM attacks. But the performance analysis shows the modified algorithm consumes more memory since it has more computation overhead compared with standard ECDH. Thus, the analytic results demonstrated that the proposed algorithm cannot run efficiently with limited memory and processor.

The proposed algorithms are aimed at further mitigation of computations to make them more suitable for application in limited-resource environments, such as the IoT. The algorithms add an authentication scheme to prepare for the sharing of a secret key between legitimate parties. At the same time, the VECDH algorithms are bidirectional authentication, which means the calculation of the secret key depends on the communication direction. This feature allows both parties to change their parameters, thus changing the encryption key for every new session to enable the PFS protocol.

2. Preliminaries:

2.1 Elliptic Curve:

ECC has initial parameters over $\mathbb{F}_p$ both communication sides should synchronize these parameters. These parameters are called Elliptic Curve Domain parameters:

$$T = \{\mathbb{F}_p, a, b, G, n, h\}$$

Where : is a large prime number, $a, b \in \mathbb{F}_p$ specify the $E_p(a, b)$ equation:

$$E: y^2 = x^3 + ax + b \ (mod\ p) \qquad (1)$$

$G(x, y)$ is the base point on $E_p(a, b)$, n is the order of G, and h is cofactor, i.e., $h = E_p(a, b)/n$.

ECC over a finite field $\mathbb{F}_p$ security is depending on the Elliptic Curve Discrete Logarithm Problem (ECDLP), in which no successful subexponentially algorithm can solve the ECDLP problem in polynomial time. The hardness of (ECDLP) involves the computation of retrieving the multiplier point and multiplicand integer from a known product point [12]. Therefore, the EC parameters should be chosen carefully to make the algorithm immune against attacks on the ECDLP.

2.3 Elliptic Curve Diffie-Hellman (ECDH) Key exchange:

Both sides of communication have the same ECC parameters and generate different multiplicand private keys. Let's say: A generates d_A as the private key $and\ B$ generates d_B, too. Then both compute their $public\ keys\ Q_A(x, y), Q_B(x, y)$ that is a product of [3]:

$$Q_A = d_A G \qquad (2)$$

$$Q_B = d_B G \qquad (3)$$

Then B can encrypt a message using a Symmetric Secret Key S_i defined as follow:

$$S_B = d_B Q_A \tag{4}$$

And A can decrypt B's encrypted message using the same Symmetric Secret Key S_i, when computes the following:

$$S_A = d_A Q_B \tag{5}$$

The proven of $S_i(at\ A) = S_i(at\ B)$ comes from the scalar multiplies of $Eq(2)(3)$:

$$\begin{cases} S_A = d_A Q_B \\ \quad = d_A d_B G \end{cases} \tag{6}$$

2.3 The weakness of ECDH algorithm against Man-in-the-middle attack:

Figure 1 shows how a man-in-the-middle attack can threaten the ECDH. In which the adversary can intercept the traffic and expose the exchanged messages without any attention from the communication participants [13].

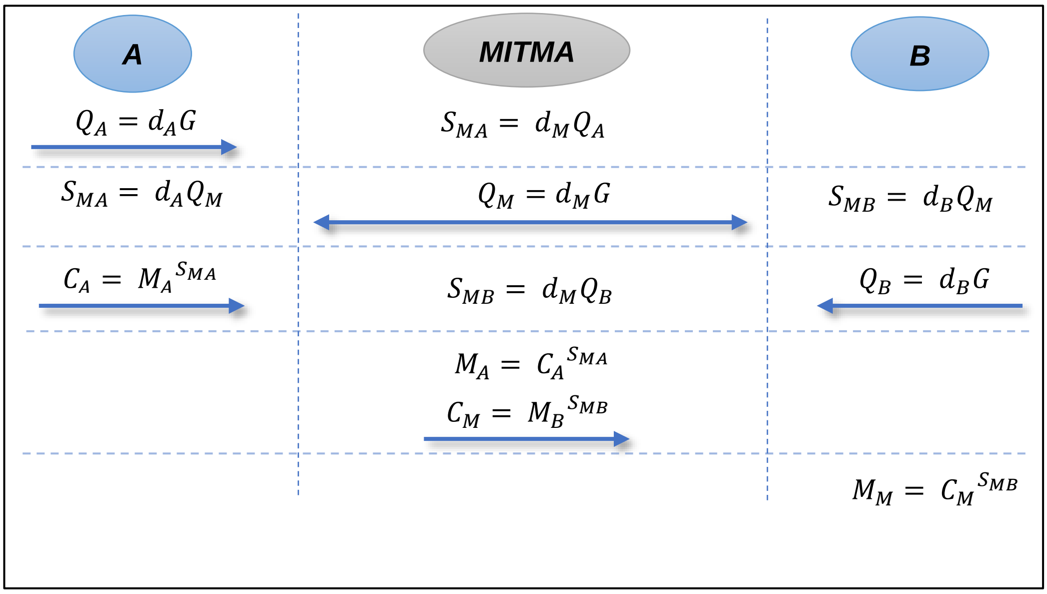

Figure 1: Man-in-The-Middle attack against the standard Elliptic Curve Diffie-Hellman algorithm

3. The Proposed system design:

The description of the algorithms of VECDHs is as follows:

3.1 VECDH version1 algorithm:

In the first stage of $VECDH_{v1}$, the centralized server should be responsible for registering the entities within the local network. The registration is vital in authenticating the registered entities. The server chooses a nonce private key (global certificate) K and divides it into (local certificates) k_i, based on Shamir's secret sharing algorithm. Thus, all plugged devices (PCs, laptops, IoTs,.. etc.) would gain a specified k_i, and each entity validates the communicated side by reconstructing the secret shared K, from their own local certificate and the communicated side's local certificate. The earlier registration stage was proposed in the previous work that was published in [14]. Figure 2 shows the $VECDH_{v1}$ model, which steps as follows:

1. Both sides compute their public key by $Eq(2)(3)$.
2. Both A and B compute K, to authenticate each other.

3. A computes $R = KG$ as the authentication point. Analogously, B computes the same point $R = KG$.
4. Both sides compute the secret hash value using a secret function:

$$e = H\{r_A, id_1, id_2, cinf\} \tag{7}$$

Where id_1 *and* id_2 are identities of the sender and destination, respectively. $cinf$ is immediate information comprising a query time and other changeable according to the time
5. When A wants to send a response message, he/she computes the session secret key SSK for a given i session, as follows:

$$SSK_i = d_A Q_B + Re \tag{8}$$

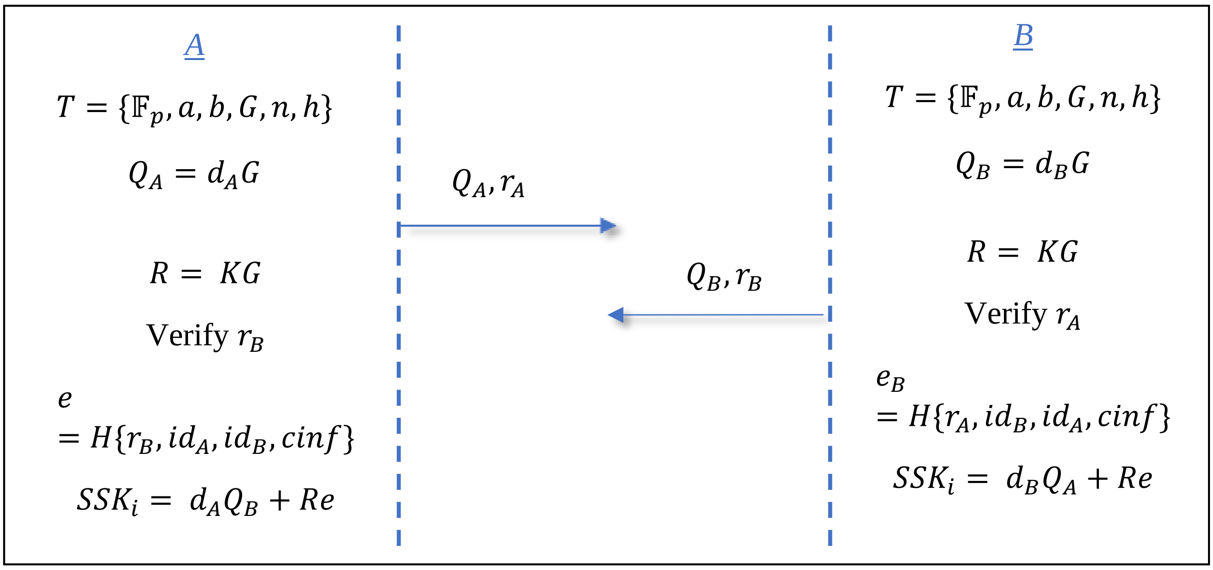
A
$T = \{\mathbb{F}_p, a, b, G, n, h\}$
$Q_A = d_A G$
$R = KG$
Verify r_B
$e = H\{r_B, id_A, id_B, cinf\}$
$SSK_i = d_A Q_B + Re$
Q_A, r_A
Q_B, r_B
B
$T = \{\mathbb{F}_p, a, b, G, n, h\}$
$Q_B = d_B G$
$R = KG$
Verify r_A
$e_B = H\{r_A, id_B, id_A, cinf\}$
$SSK_i = d_B Q_A + Re$

Figure 2: steps of $VECDH_{v1}$ to generate authenticated SSK_i

As below, it is proven that computation of SSK_i at both sides is equal. Let's assume that A wants to compute:

$$\left\{\begin{aligned} SSK_i &= d_A Q_B + Re \\ &= d_A d_B G + Re \\ &= d_A d_B G + KGe \\ &= d_A d_B G + KGH\{r_A, id_1, id_2, cinf\} \end{aligned}\right. \tag{9}$$

And, in such a session, after, B verifies the A's signature (r_A), he/she computes the decryption key as follows:

$$\left\{\begin{aligned} SSK_i &= d_B Q_A + Re \\ &= d_B d_A G + Re \\ &= d_B d_A G + KGe \\ &= d_B d_A G + KGH\{r_A, id_1, id_2, cinf\} \end{aligned}\right. \tag{10}$$

Ultimately, SSK_i at B and A are the same.

3.2 VECDH Version2 algorithm:

Figure 3 shows the $VECDH_{v2}$ model, which steps as follows:
1. Both sides compute their public key by $Eq(2)(3)$.
2. A chooses a random number $k \in [1, n-1]$ and computes $R_A = k_A G$ as the authentication point, Analogously, B chooses a random number $k \in [1, n-1]$ and computes $R_B = k_B G$.

3. Both sides compute a signature:

$$r = x_R (mod\ n) \tag{11}$$

4. Whenever one of them needs to send a message, he/she verifies r of the destination, which means it resides on the E_p, and computes the session secret key SSK_i as the following:

$$SSK_i = d_s Q_s + k_s^{-1} R_s^{-1} e_s \tag{12}$$

$H\{r_s, id_1, id_2\}$ is a secure hash function, and id_1 *and* id_2 are identities of the sender and destination sides.

The computation of $Eq(12)$ depends on the communication direction, in which the SSK_{Ai} can be computed by A to encrypt his/her message, and computed at B to form a decryption key. When B want to send his/her message, it computes a new SSK_{Bi} as an encryption key, whereas A computes it as a decryption key. At any time, both sides can generate new (encryption and decryption) keys as desired using $Eq(12)$. Noteworthy, both sides used the same secure hash function to compute the e value. As a result, this scheme achieves the perfect forward secrecy protocol.

As below, a proven that computation of SSK_i at both sides is equal. Let's assume that A wants to compute:

$$\left\{ \begin{aligned} SSK_i &= d_A Q_B + k_A^{-1} R_A^{-1} e_A \\ &= d_A d_B G + k_A^{-1} (k_A G^{-1}) e_A \\ &= d_A d_B G + G^{-1} e_A \end{aligned} \right. \tag{13}$$

And, in such session, B computes decryption key as follow:

$$\left\{ \begin{aligned} SSK_i &= d_B Q_A + k_B^{-1} R_B^{-1} e_A \\ &= d_B d_A G + k_B^{-1} (k_B G^{-1}) e_A \\ &= d_B d_A G + G^{-1} e_A \end{aligned} \right. \tag{14}$$

So, SSK_i at B and A are the same.

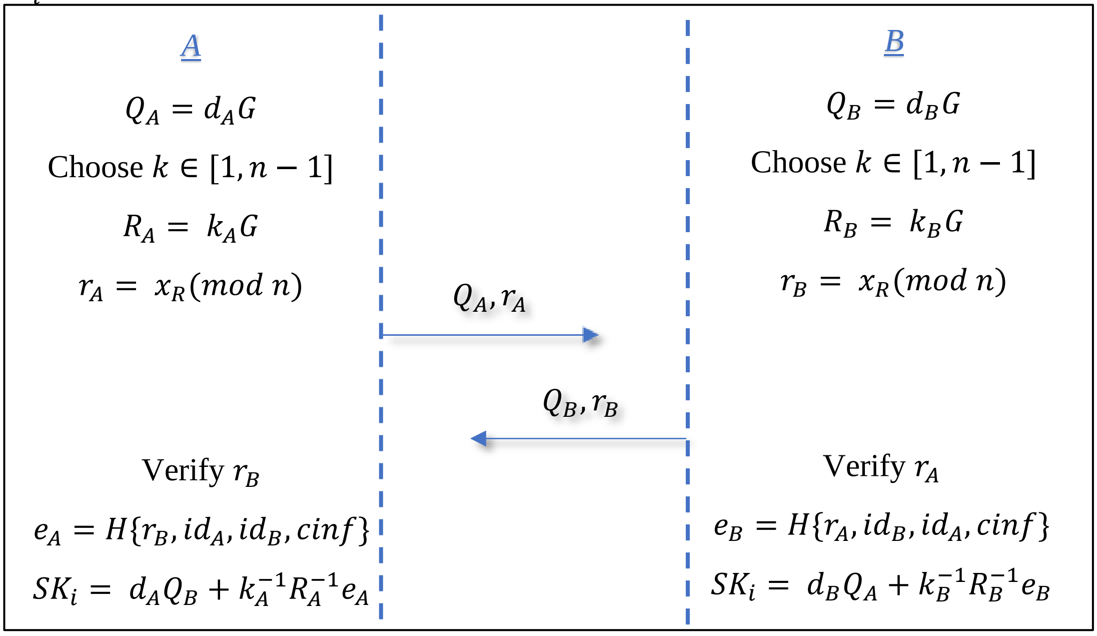

Figure 3: steps of VECDH-v2 to generate authenticated SSK_i

4. Security Analysis:

The proposed algorithms can satisfy multiple security features, as the following:

1. Resistant the Man-in-the-middle: $VECDH_{v1}$ algorithm can avoid the man-in-the-middle attack when run in a real-time state, since the adversary needs to gain the valid local certificate, k_i from the centralized server in an earlier (registration) state, to become authenticated on the remote side, so he can't calculate the SSK_i, and if he/she could forge a local certification $k_i{}^{!}$, the validation step (step-2) could detect the forged certification, in which:
When C (adversary) computes a forge $k_i{}^{!}$, this local certification cannot fulfill the following validation:

$$K^{!} = Builder\left(k_{ci}{}^{!}, k_{Ai}\right) \neq K$$

The global certification cannot be built because the $k_i{}^{!}$ was not generated by the cartelized server.
On the other hand, in the $VECDH_{v2}$, it is up to you to use a secret hash function and the secret parameters that are distributed during the installation stage.
2. Mutual authentication: in $VECDH_{v1}$ the communication participants can authenticate each other using the certification parameters that are obtained in registration phase. In turn, the validation phase involves building the global certification K using the local authentication parameters k_i and the remote one k_j. So, the adversary can detect when you forge illegal parameters.
3. Perfect Forward Secrecy protocol (PFS): $VECDH_{v1}$ and $VECDH_{v2}$ enable the PFS protocol since the global certificate K can be changeable at each new session by the centralized server, or even at each new plugging device, since a new set of local certificates should be calculated and distributed among the network's entities. According to the e and r_A calculation, SSK_i would be volatile, in which the key in the current session would not be used in further sessions. For example, the $VECDH_{v1}$ enables the PFS protocol, as follows:
When: $e_i = H\{r_{Ai}, id_1, id_2, cinf_i\}$
Then, $SSK_i = d_A Q_B + Re_i$
But when, $e_j = H\{r_{Aj}, id_1, id_2, cinf_j\}$
So, $SSK_j = d_A Q_B + Re_i$
4. Key privacy: the attacker cannot retrieve a session secret key SSK_i that established by honest parties, because the underlying computation of $VECDH_{v1}$ and $VECDH_{v2}$ algorithms depend on the intractability of ECDLP, when compute the d_A and d_B in $Eq(2)(3)$.
5. Key independence: in $VECDH_{v1}$, the calculation of SSK_i is independent of previous and subsequent session keys because the centralized server can generate a new collection of k_i, this step can lead to computing new K, thus new R, resulting in each individual session having fresh parameters. Hence, the revealed keys of a specific session or multi-sessions do not help in deducing the key of the current session. The proof of this step can be deduced from the one demonstrated at point 3. Also, this feature prevents key-compromise impersonation attacks from being able to impersonate one of the legal communication parts since there are fresh certification parameters that are generated at every new session. In addition, this prevention can be confirmed by the impossibility of deducing the private key according to point 4.
6. Hash function immunity: suppose an adversary can reveal the secret key of a specific legitimate user d_A and gain the authentication parameters k_i, and tries to deceive the communicated parties, but here the hash function's role comes to immunize the secrecy of the system, depending on the collision resistance and deterministic primitives of the hash function**.** Thus, the secret hash function reports the potential man-in-the-middle attack or other malicious interceptions.

5. Performance Analysis

Table 1 reports a comparison with regard to the computation effort requirements of the VECDH algorithms and other proposed algorithms. The first column refers to the count of scalar point multiply operations. The second column refers to the number of fields multiplied.

The third one refers to hash computation operations. The last refers to the number of field inversion operations. This aspect can affect memory and processor overload. As shown, the proposed $VECDH_{v1}$ adapted its performance to limited-resource environments, and it is better than $VECDH_{v2}$. But with respect to the remaining methods, they both achieve low workload and suitable computation efforts, so these are important features present in the VECDH versions.

Table 1: Arithmetic operations comparisons

	Point Mult.	Field Mult.	Hash	Field Inversion
[8]	3	1	0	1
[5]	1	1	0	0
[7]	1	1	0	1
$VECDH_{v1}$	1	0	1	0
$VECDH_{v2}$	1	0	1	1

From another aspect, the complexity of the proposed algorithm has been compared with [8] with respect to the evaluation of the execution time of the algorithm's phases (certification generation T_{CG}, certification validation T_{CV}, session key generation T_{SSk}). Table 2 depicts this comparison. The comparison illustrated that the $VECDH_{v1}$ can be the lowest complexity, which leads to the lowest workload. This characteristic makes the proposed algorithm run faster with limited processor and memory, and most importantly, it has the quickest response when working with a real-time system. That is what is aimed at most IoT networks.

Table 2: Complexity comparison

	T_{CG}	T_{CV}	T_{SSK}
[8]	331 ms	61.1 ms	298 ms
$VECDH_{v1}$	0.1739 ms	9.958 ms	9.2513 ms

Table 3 shows the security capabilities of the proposed algorithms $VECDH_{v1}$ and $VECDH_{v2}$ and compare them with other research works. The comparison showed that the performance of the proposed algorithms is efficient across achieving more security features.

Table 3: Security capabilities

	[5]	[8]	$VECDH_{v1}$	$VECDH_{v1}$
Resistant the Man-in-the-middle	✓	✓	✓	✓
Mutual authentication	✗	✓	✓	✓
PFS	✓	✓	✓	✓
Key privacy	✓	✓	✓	✓
Key independence	✓	✗	✓	✓
Hash function immunity	✗	✗	✓	✓

6. Conclusion

The VECDH algorithms enhance the security level of the standard algorithm by improving its immunity against various attacks, such as man-in-the-middle attacks, from which the original algorithm has suffered. On the other hand, the appropriate workload effort allows for running the algorithm with the lowest resource consumption. This aspect was confirmed through the time execution evaluation. These features make VECDH algorithms more suitable for applying in restricted resource environments, such as the IoT, especially those that are running in real-time fashion. The future work, depending on the $VECDH_{v1}$ and $VECDH_{v2}$, can develop a novel pseudo-random key generator as a further security level, for encryption of the sensor information and captured pictures and videos, to send real-time information across a hostile network in highly secure coding.

7. Disclosure and conflict of interest

Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing-original draft preparation, writing-review and editing, and visualization have been implemented by the first author. Supervision and project administration have been implemented by the second author.

References

[1] K. H. Moussa, A. H. El-Sakka, S. Shaaban, and H. N. Kheirallah, "Group Security Authentication and Key Agreement Protocol Built by Elliptic Curve Diffie Hellman Key Exchange for LTE Military Grade Communication," *IEEE Access*, vol. 10, no. June, pp. 80352–80364, 2022, doi: 10.1109/ACCESS.2022.3195304.

[2] M. Rana, Q. Mamun, and R. Islam, "Lightweight cryptography in IoT networks: A survey," *Futur. Gener. Comput. Syst.*, vol. 129, pp. 77–89, 2022, doi: 10.1016/j.future.2021.11.011.

[3] P. Zhang, Y. Li, and H. Chi, "An Elliptic Curve Signcryption Scheme and Its Application," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/7499836.

[4] R. Haakegaard and J. Lang, "The elliptic curve diffie-hellman (ECDH)," *Retrieved Febr. 10, 2020, from http//koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Haakegaard+Lang.pdf*, no. December, p. 4, 2015.

[5] E. J. Yooni and K. Y. Yoo, "A new elliptic curve Diffie-Hellman two-party key agreement protocol," *2010 7th Int. Conf. Serv. Syst. Serv. Manag. Proc. ICSSSM' 10*, pp. 549–552, 2010, doi: 10.1109/ICSSSM.2010.5530179.

[6] S. Wang, Z. Cao, M. A. Strangio, and L. Wang, "Cryptanalysis and improvement of an elliptic curve Diffie-Hellman key agreement protocol," *IEEE Commun. Lett.*, vol. 12, no. 2, pp. 149–151, 2008, doi: 10.1109/LCOMM.2008.071307.

[7] L. Kaur Pahal Singh Paraste Assistant Professor, "Enhanced Elliptic Curve Diffie-Hellman Key Exchange Algorithm for Ornamental Security based on Signature and Authentication Algorithm," *GRD J. Eng.*, vol. 2, no. 7, pp. 134–141, 2017, [Online]. Available: www.grdjournals.com.

[8] N. Mehibel and M. Hamadouche, "Authenticated secret session key using elliptic curve digital signature algorithm," *Secur. Priv.*, vol. 4, no. 2, pp. 1–15, 2021, doi: 10.1002/spy2.148.

[9] L. Harn and M. Mehta, "Integrating Diffie-Hellman key exchange into the digital signature algorithm (DSA)," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 198–200, 2004, doi: 10.1109/LCOMM .2004.825705.

[10] R. Patgiri, "privateDH : An Enhanced Diffie-Hellman Key-Exchange Protocol using RSA and AES Algorithm," *Woodstock '18 ACM Symp. Neural Gaze Detect. June 03â•fi05, 2018, Woodstock, NY*, vol. 1, no. 1, pp. 1–6, 2021.

[11] M. A. Dar, A. Askar, D. Alyahya, and S. A. Bhat, "Lightweight and Secure Elliptical Curve Cryptography (ECC) Key Exchange for Mobile Phones," *Int. J. Interact. Mob. Technol.*, vol. 15, no. 23, pp. 89–103, 2021, doi: 10.3991/ijim.v15i23.26337.

[12] A. Abdullah, A. Mahalanobis, and V. M. Mallick, "A new method for solving the elliptic curve discrete logarithm problem," *J. Groups, complexity, Cryptol.*, vol. Volume 12, Issue 2, no. 2, pp. 1–2, 2021, doi: 10.46298/jgcc.2020.12.2.6649.

[13] Aryan, C. Kumar, and P. M. Durai Raj Vincent, “Enhanced diffie-hellman algorithm for reliable key exchange,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 263, no. 4, 2017, doi: 10.1088/1757-899X/263/4/042015.

[14] M. S. Oudah and A. T. Maolood, “Lightweight Authentication Model for IoT Environments Based on Enhanced Elliptic Curve Digital Signature and Shamir Secret Share,” *Int. J. Intell. Eng. Syst.*, vol. 15, no. 5, pp. 81–90, 2022, doi: 10.22266/ijies2022.1031.08.