Iraqi Journal of Intelligent Computing and Informatics (IJICI)
Vol. 2, 2, December 2023, pp. 90~100
ISSN: 2791-2868, DOI: 10.52940/ijici.v2i2.39

Elliptic Curve Implementation and its Applications: A Review

Mohamed Wameedh Abdulnabi[1], Raad A. Muhajjar[1] and Mishall Al-Zubaidie[2]
[1]Department of Computer Science, Faculty of Computer Science and Information Technology, University of Basrah, Basrah, Iraq
[2]Department of Computer Science, Education College for Pure Science, University of Thi-Qar, Nasiriyah, Iraq,

Article Info

Article history:

Received August 15, 2023
Revised September 10, 2023
Accepted September 11, 2023

Keywords:

Public_key
Private_key
ECDH
ECDSA
ECC

ABSTRACT

Encryption is regarded as essential within the safety measures used to protect data via unsafe transport methods. Elliptic Curve encryption (ECC), one of the several asymmetric encryption algorithms currently in use, has become popular because of its high level of security and small key sizes. In contrast to Rivest, Shamir and Adleman (RSA), for instance, ECC can keep security levels at a certain level with a smaller key. The primary purpose of ECC is elliptic curve point multiply (ECPM), which also has the largest hardware cost. Numerous hardware applications have been made to accelerate the calculus of the ECPM. This paper discusses the idea behind the cryptography of elliptic curves (ECC's) and the benefits it quicker, moreover believed way of encoding as compared to the RSA public_key encryption techniques that are currently used as standards. ECC's specifications encompass all relevant asymmetry cryptography elements, including electronic signatures & key negotiation processes. The scalar multiplication Key Point (K.P) function, which is the fundamental operation of ECCs, is utilized to accomplish this objective. Where P is a value and k represents an integer and is exists on an elliptic curve. This article explains how ECC contributes to system safety and applications of it in the field of security.

Corresponding Author:

Mohamed Wameedh Abdulnabi
Department of Computer Science, College of Computer Science and Information Technology, Basrah, Iraq
Email: itpg.mohammed.wameedh@uobasrah.edu.iq

1. INTRODUCTION

The use of a technological tool called an electronic signature is spreading more widely to secure the data. Its primary responsibilities include locating and stopping any unwanted data modifications, in addition to confirming the legitimacy of the signing [1].

Digital signatures, also known as E-signatures, are used to verify the authenticity of electronic transactions and documents. E-signatures are often used in a variety of applications, such as e-banking, where they are used to confirm the user's identity and the legitimacy of the transaction.

The E-signature includes a number of distinctive features. A digital signature will belong to a particular individual. A user's electronic signature is his property and cannot be stolen from him. A right cannot be converted or in any other way alienated. A digital signature is regarded as a significant component in this situation and might be displayed on a physical object, a token (portable data resource), or a device that holds information.

There is a substantial constraint on utilizing procedure. Utilizing starts when the authorized certifying authority provides access [2]. The use of lightweight e-signature schemes that are specifically designed for use in deferent applications that can be use on light duty CPU. These schemes are typically based on simple and efficient algorithms that can be easily implemented on a wide range of devices, including smartphones and tablets. One example of a lightweight e-signature scheme is the digital signature algorithm (DSA), which is based on the use of modular arithmetic and hash functions. DSA is widely used in various applications, including e-banking, and is considered to be both secure and efficient [3] [4].

Each user has been provided a key pair by him. When a public_key becomes accessible to every individual, a private_key is maintained secretly. Anyone who has copy of the public_key may then use it for safeguarding data that could be decoded from the individual who owns it only of the associated private_key [5]. Elliptic curve and RSA cryptography are the two subcategories under which public_key cryptography

systems fall. The RSA algorithm claims that there is a factorization problem, or difficulty factoring large integers. Elliptic curve cryptography is based on elliptic curves over finite fields.

2. CRYPTOGRAPHY ALGORITHMS AND APPLICATIONS

The usage of a secret shared key between two parties and a key change for each symbol in the message was proposed as a new, efficient implementation approach for symmetric encryption over ECC in this work. This method offers non-repudiation, confidentiality, and authenticity [6].

The discipline of cryptography, which offers safety measures for data transported over internet connections, is particularly essential. By transforming the raw data into an unrecognizable format [7].

Data security is safeguarded throughout its transmission and storage using the process of encryption. Storage technicians commonly deal with storage systems and technologies that support encryption, such as cryptography. Memory or the hard drive cryptography [8].

According to the most recent security standards that improve email security, many solutions and standards levels have been developed. Some of the most recent updates focus on maintaining the integrity and security of email-based data sharing. Client will not retract from his message while the others are focused on verifying the sender's authenticity. This essay will outline how email functions, go through several email security measures, and examine email communication dangers. This paper offers a number of models and methods that are used to improve the security of email systems [9].

One technique for preventing data theft and unauthorized access is cryptography. Asymmetric and symmetric cryptosystems are the two different types of cryptosystems. The Sender and recipient share the same key in a symmetric cryptosystem. It implies that both encryption and decryption utilize the same key. Asymmetric cryptosystems employ several keys [10].

The study of mathematical methods connected to information security components including non-repudiation, entity authentication, data integrity, secrecy, and entity authentication is known as cryptography. The goal of the cryptanalyst is to undermine the work of the cryptographer by deciphering a cipher or creating fake coded signals that will pass muster as genuine [11].

Cryptography is heavily used in network security. Cryptography is essentially the technique of concealing data by encrypting the transmission. Cipher text is the practice of encrypting information into an unintelligible form (encrypted text). The communication can only be deciphered (decrypted) into plain text by someone who has the secret key. A cryptographic system is one that encrypts initial data (normal text) at the transmitter and decrypts it back into plain text at the receiver using a specific key or formula. Cryptanalysis, often known as code-breaking, can occasionally be used to decrypt encrypted messages [12].

The three basic subcategories of cryptography are hash functions, public_keys, and secret keys. An identical key is utilized for the two processes in symmetrical encryption is sometimes called private_key encryption. However, in asymmetrical encryption, often named simply public_key cryptography, two keys are utilized: a public_key that encrypts and decrypts communicated information with a private_key that the key owner uses to decrypt messages [13].

According to recent research findings and analyses that have been published, symmetric and asymmetric ciphering techniques are used to protect data. They do this by converting data into new forms that are difficult for unauthorized parties to break or decipher [14] [15].

Asymmetric encryption employs two keys public and private to cipher and decipher any message. Examples of these include RSA, ECC and DSA. Although the private_key remains private and is utilized to decode the message, anybody having the public_key is able to utilize it to transmit a message. This method improves security. Overall, they all require converting ordinary text into ciphertext and back again [16].

2.1 Secret Key or Symmetric Key Algorithm

Secret key cryptography, also known as symmetric encryption, uses one key to perform encryption and decryption. Used mostly for secrecy and privacy [17]. Figure 1 shows the symmetric key algorithm.

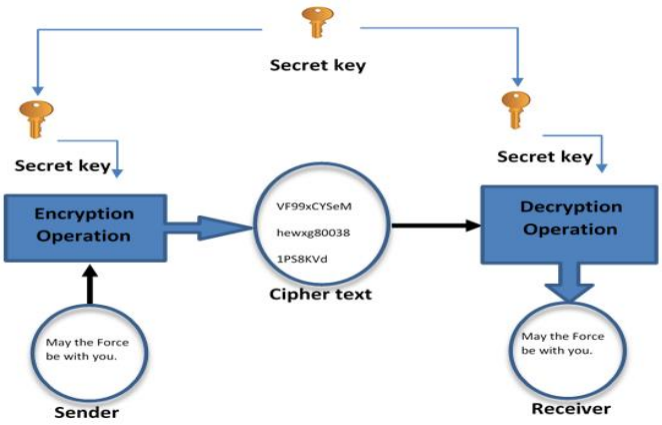

Figure 1. Shows the Symmetric key Algorithm.

Ref. [5]-[7] explaining research chronological, including research design, research procedure (in the form of algorithms, Pseudocode or other), how to test and data acquisition. Table 1 presentes comparison of various algorithms on the basis of different parameters like the year of development, key length/bit, block size/bit.. and so on [5], [8]–[13].

2.2 Public_key or Asymmetric key Algorithm

One key is utilized for encrypted and another key is utilized for decryption in public_key cryptography. Asymmetric encryption the name given to it. Mostly employed for key exchange and authentication [18]. Figure 2 shows the asymmetric key algorithm.

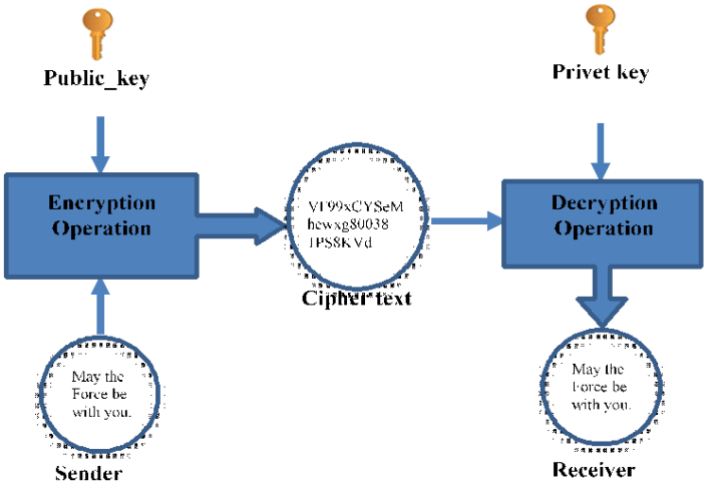

Figure 2. Shows Asymmetric key Algorithm.

Table 1. Comparison of Various Algorithms on the basis of Different Parameters

Paeameters	DES	3DES	AES	RSA	ECC
DEVELOPMENT	In early 1970 by IBM and Published in 1977.	IBM in 1978.	Vincent Rijmen, Joan Daeman in 2001	Ron Rivest, Shamir & Leonard Adleman in 1978	Victor Miller from IBM and Neil Koblitz in 1985
KEY LENGTH (Bits)	64 (56 usable)	168,112	128,192, 256	Key length depends on no. of bits in the module	Smaller but effective key
ROUNDS	16	48	10,12,14	1	1
BLOCK SIZE (Bits)	64	64	18	Variable block size	Stream size is variable
ATTACKS FOUND	Exclusive Key search, Linear cryptanalysis, Differential analysis	Related Key attack	Key recovery attack, Side channel attack	Brute force attack, timing attack	Doubling attack
LEVEL OF SECURITY	Adequate security	Adequate security	Excellent security	Good level of security	Highly secure
ENCRYPTION SPEED	Very slow	Very slow	Faster	Average	Very Fast

3. Hash functions

Provides a digital fingerprint by using a mathematical change to "Hash" information in an irreversible manner [19]. Figure 3 shows the hash function algorithm.

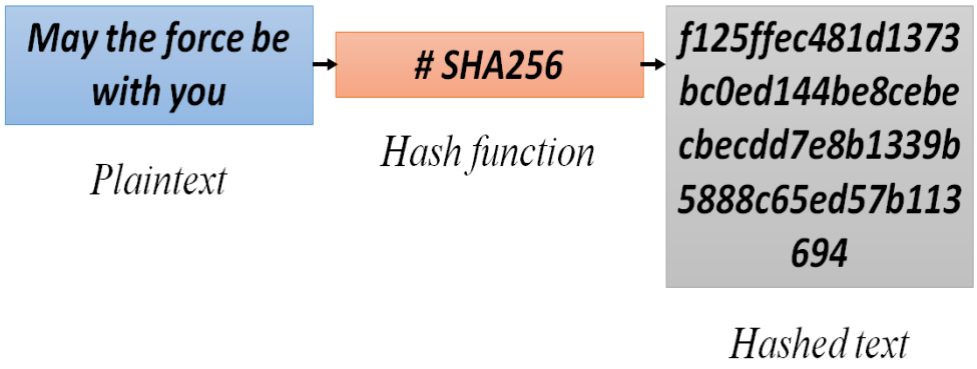

Figure 3. Shows the hash function.

3. A BACKGROUND ON: ELLIPTIC CURVE CRYPTOGRAPHY

V. Miller and N. Koblitz researcher introduced ECC at 1995, using a public or asymmetrical key encryption, which is simply dependent on mathematical operations [20] [21].

The distinctive quality of ECC is that it provides a difficult exponential time challenge of modest scale to the attacker in order to settle-the network or communications. precisely a consequence, ECC has small keys, high security, and quick encryption operations, and it offers small hardware and software [22].

Key generation, encrption, and decryption algorithms are three necessary components of the ECC cryptosystem, and they are all verified using the elliptic curve's domain parameters [23].

Figure 4 shows the equation curve of the ECC. The general equation, which gives the elliptic curve its name, is shows in Eq(1).

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

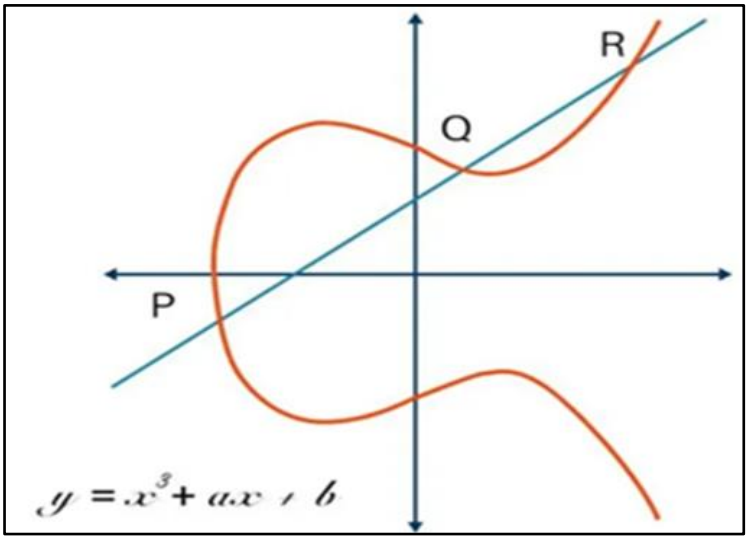

Figure 4. Elliptic Curve illustrates the operation P+Q=R [11]

4. MATHEMATICAL ECC OVER FINITE FIELDS

All solutions are represented by the elliptic curve E on the finite field F_q, (x, y) ∈ $\bar{F}_q * \bar{F}_q$ with the general equation (1) [24].

Wherein, a, b ∈ Fq and $4a^3 + 27b^2 \neq 0$. Additionally, this curve has a unique point \infty, known as the infinitely distant point denoted via letter O.

The curve group's functioning [25]:

Permit $P = (x, y) \in E$, subsequently specify $-P = (x, -y)$ and $P + (-P) = O$.

Considering *P* and *Q* as a pair of points over an elliptic curve.

P = (x, y) and *Q* = (x_2, y_2) where $P \neq -Q$ thereafter *P*+ = (x_3, y_3).

Where:

$$x_3 = \lambda^2 - x_1 - x_2 \quad (2)$$

$$y_3 = \lambda^2(x_1 - x_3) - y_1 \quad (3)$$

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{If } P \neq Q \\ \frac{3x_1^2 + a}{2y_1} & \text{If } P = Q \end{cases} \quad (4)$$

4.1. Adding Points in Elliptic Curve Cryptography

A finite field that can only be operated on using (mod p) can be defined as one that is confined by the prime integer p. ECC normally starts with an initial value (P), which is then multiplied by n times to produce nP. Our private_key is value of n, while our public_key is value of nP. To put two points together, select two locations within the elliptic curve *(R=P+Q)* [12] [26] Figure 5 shows the adding points in ECC.

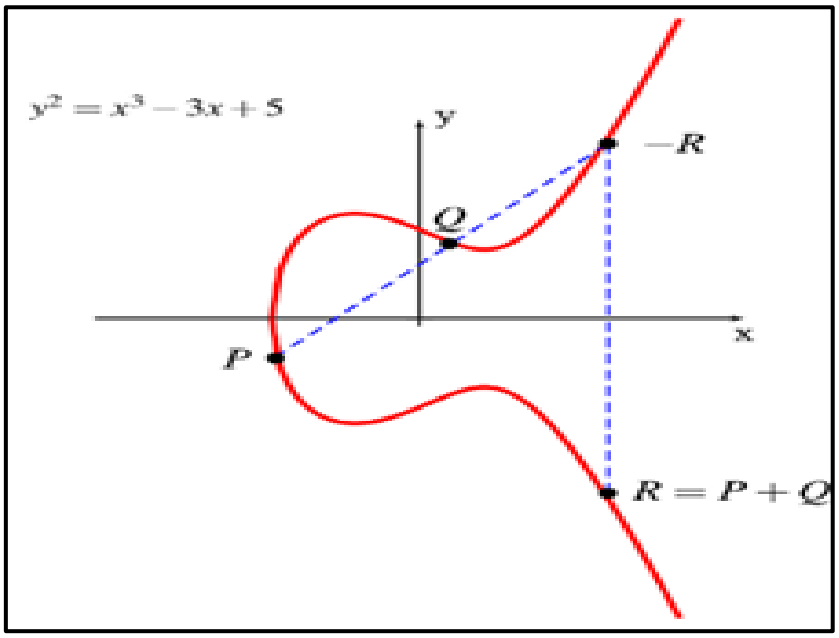

Figure 5. Shows the Elliptic Curve Point addition

Therefore, Let's adding two points $P(x_1, y_1)$, $Q(x_2, y_2)$ and we use x_3+ax+b (mod p), we compute the slope between the locations as follows [27]:

$$n = (y_1 - y_2) / (x_1 - x_2) \quad (5)$$

Next, we employ the following to find the new point $R(x_3, y_3)$:

$$x_3 = n^2 - x_1 - x_2 \quad (6)$$

$$y_3 = n(x_1 - x_2) - y_1 \quad (7)$$

4.2. Subtraction Points in Elliptic Curve Cryptography

The elliptic curve subtraction procedure. If point Q is the outcome of subtracting point *P*, then *P - R = P + (−Q)*. Can quickly describe the subtraction process on an elliptic curve as follows when the negation operation is introduced. If (–Q) is Q's negative procedure, then $R = P - Q = P + (-P)$ [28].

4.3. Multiplication Point in Elliptic Curve Cryptography

Repetition of the fundamental coordinate point's addition is known as multiplication also called (Elliptic curve scalar multiplication). Numerous algorithms have been developed to quickly multiply points. On an elliptic curve *E*, could do the multiplication, indicated by $Q = mP$, based on point addition. Where, $m \in \mathbb{Z}^+$ and $(P, Q) \in E^2$. Scalar multiplication is in fact a sequence of addition of points [29].

$$mP = \underbrace{P + P + \cdots + P}_{m \text{ times}} \quad (8)$$

4.4 Doubling Point in Elliptic Curve Cryptography

At point *P* using a tangential relationship to the curve, we may determination multiple of *P*. Undoubtedly, at some time, this line will come to cross the curve. Then, $R(x_3, y_3) = 2p(x_1, y_1)$ is this point's reflection as it relates to the *x*-axis [30]. Figure 6 shows the doubling point in ECC.

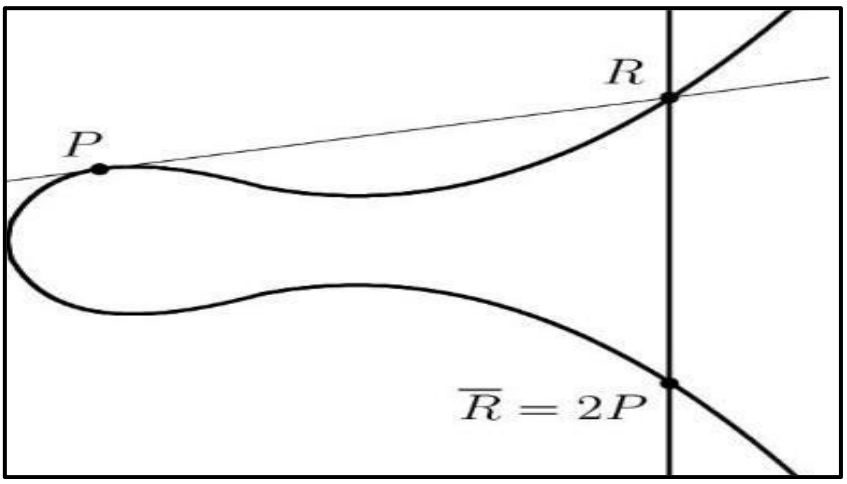

Figure 6. Shows Elliptic Curve Doubling point [20].

4.5 infinity Point in ECC

It is stated that the points meet in infinity, symbolized it through O, if $(x_1 = x_2, y_1 = y_2) = 0$ or $(x_1 = x_2, y_1) = y2$ [31]. Figure 7 shows the infinity point in ECC.

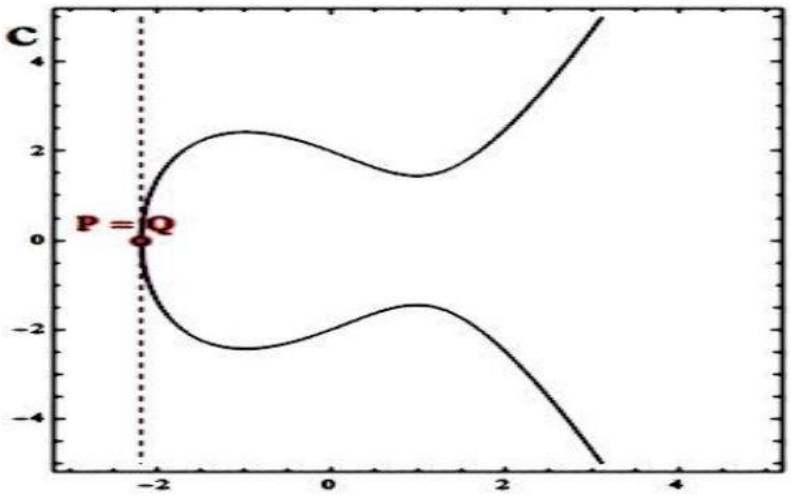

Figure 7. Shows Elliptic Curve infinity a point.

4.6 A Processor for Elliptic Curves

The ECC processor's architecture. It is made up of an ECC unit for arithmetic operations, an ECC ADD with Double unit, and a main control unit. Curves should be used for implementing Elliptic curve protocols, according to the SEC-2 proposal [31]. Figure 8 shows the processor of ECC.

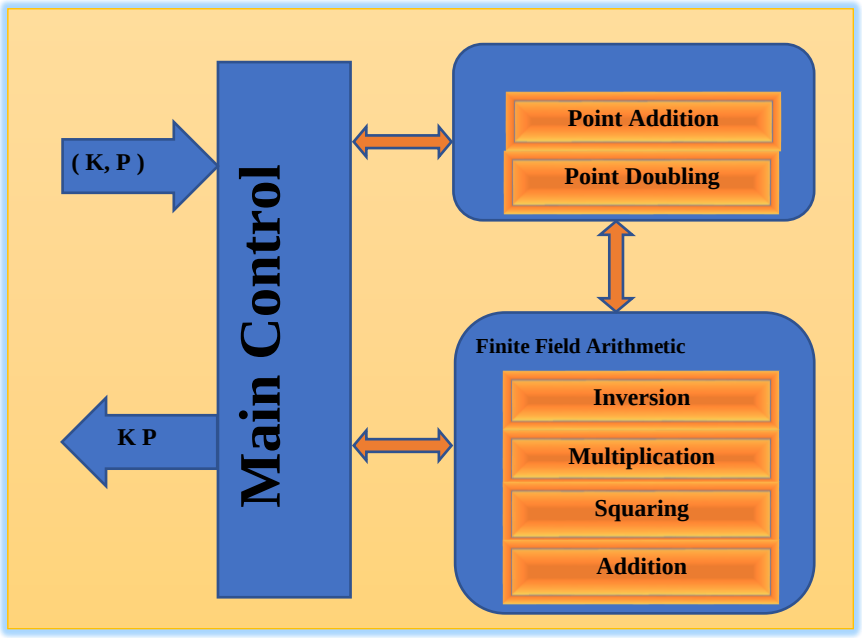

Figure 8. Processor for elliptic curve point multiplication.

5. Types of public key Algorithms

This section presents some algorithms that can be utilized in key transfer and electronic signatures. This types can be explain as follow.

5.1. Elliptic curve-Diffie-Hellman (ECDH)

The fundamental Diffie-Hellman process is a public-key cryptosystem that is recommended for secret key sharing. First A (Sender) and B (Receiver) correspond to employ certain curves, fields dimension, type of mathematical.

A secret key is then sent along using the following procedure. The required for implementing the Diffie-Hellman protocol is scalar multiplication [32].

The algorithm's steps are [33]:
Step1: A base g and a prime integer p are chosen by Sender and Receiver.
Step2: Sender selects the code number m and sends Receiver (g^m mod p).
Step 3: Receiver sends Sender at ((g^n mod p) using the secret number n.
Step4: $((g^n \ mod \ p)^m \ mod \ p)$ is computed by Sender.
Step5: $((g^m \ mod \ p)^n \ mod \ p)$is computed by Receiver.
This number can be used as a key by both Sender and Receiver. Note that p and g do not require protection.

5.2 Elliptic curve-digital signature algorithm (ECDSA)

The EC-Digital Signature Algorithm it's an DSA's elliptic curve counterpart; This protocol additionally needs scalar multiplication, area multiplying, and field inverse operation in addition to elliptic curve operations like integers multiplying, reverse operations, and modularity operation.

A (Sender) creates the signature using secret key in the ECDSA, B (Receiver) validates utilizing signature A. A sign the message using algorithm's ECDSA protocol, B validates A's signing [34].

The algorithm's steps are [35] [36]:

Step1: Key Generation
a) private_key "integer" ⇒ privkey.
b) public_key "EC point" ⇒ pubkey = privkey * *G*.
The [0...n-1] range of random integers is used to create the private_key.
The private_key multiplied by the generator point *G* yields the public_key (pubkey), which it points at EC derived from EC point multiplying: pubkey = privkey * *G*.
Step 2: generating Signature

1- Encryption hash function should be used to determine the message's hash. Like SHA 256: h = hash(msg).
2- Securely generate the range [1...n-1] of the random number k.
3- In The value k in the deterministic-ECDSA example is HMAC-derived from "h + privkey".
4- Locating the randomly chosen location (R = k * G) and obtaining its x-axis (r = R.x).
The modular inverse k^{-1} (mod n) is an integer, such that $k * k^{-1} \equiv 1$(mod n), when locating the sign proof:

$$s = k^{-1} * (h + r * \text{privkey}) \text{ (mod n)}.$$

5- Send the signature back {r, s}.

Step 3: Utilize ECDSA in order to verify Signature

1- Use the same cryptographic hashing algorithm used for signing to calculate the message hash: h = hash (msg).
2- Calculate the modular invert of the signing verification: $s_1 = s^{-1}$(mod n).
Regain the random point associated to signature: $R' = (h * s_1) * G + (r * s_1) *$ pubkey.
From R' Take it's x-axes: $r' = R'.x$.
To ascertain the outcome of the signature validation, evaluate if $r' == r$.

5.3 RSA Cryptography algorithm

One of the earliest public-key encryption systems that is frequently used for safe transmission of information is RSA (Rivest, Shamir, Adleman). The safety of networks makes extensive use of it. This uses two large prime numbers. The RSA Challenge and Integer Factor analysis Challenge RSA Challenge are the two main open issues with RSA [37].

The decryption key, sometimes referred to as a private_key in this encryption systems, kept it like a private or secret also is different from the decode key in that it is public. This asymmetric encryption method makes use of two unique but connected keys. The Private_key is used for decryption, whereas the Public_key is used for encryption. To ensure that only the message's authenticated recipient can decode it, the Private_key must be kept secret. A person creates and distributes a public_key based on two important prime numbers plus an additional value. The produced prime numbers must be kept a secret. Anyone with the public_key can encrypt information, but only the person having the prime numbers can decode it [38].
The algorithm's steps are [39]:
Step1: Key Generation

1- Create two huge, randomized primes, p along with q, that are roughly similar in length so that the outcome that they form, n = p.q, has the necessary bit length, for example 1024 bits.
2- Calculate: (φ) phi = (p-1)*(q-1) , n = p.q.
3- 1 < e < phi to select an integer (e), Conversely gcd (e, phi) = 1.
4- Now confidential exponential will be Calculating: d, 1 < d < phi, like that ed ≡ 1 (mod phi).
5- (n, e) its public_key, where private_key (n, d). Maintain all of value's d, p, q and phi confidential.

Where:

- modulus is referred to as n.
- e: is also referred to as public exponent, encryption exponent, or simply the exponent.
- d: referred to as the secret exponent or decryption exponent.

Step2: Cryptography Algorithm
Sender A carries out the following:

1- Obtains the public_key (n, e) of the receiver B.
2- Uses a positive integer m to represent the plaintext message.
3- determines the cipher text, which is c = me mod n.
4- Delivers to B the encrypted text c.

Step3: Decryption Algorithm
Receiver B carries out the following:

1- Calculates the formula m = cd mod n utilizing a private_key (n, d).
2- From message representative m Extract the plaintext.

6. Conclusion

ECC algorithm is very important and must be study carefully to display its power in encryption world. Public key is an interesting field of cryptography that enhance the security, privacy and support the tired of security (i.e. confidentiality, integrity and availability). This research provides a straightforward explanation of elliptic curve cryptography, including its description, mathematical foundations, a quick comparison of ECC and RSA, benefits of ECC, and various ECC-based applications like ECDSA and ECDH. Finally, a discussion of ECC's security, applications, comparision with other methods and performance can be conclude as follow.

The size of the signatures and keys; ECC keys are more effective for usage in devices with limited resources because they are smaller than RSA keys. The algorithms' computational complexity; ECC algorithms are faster than RSA methods because they require less computing. The algorithms' level of security; ECC is thought to be more effective while being just as secure as RSA. The algorithms work; in the majority of applications, ECC is quicker than RSA. The algorithms' appropriateness for various applications; Small key sizes and quick performance are critical in applications like mobile devices and the Internet of Things, where ECC excels. Applications requiring high security, including those in financial transactions, are well suited for RSA.

References

[1] K. Abdullah, S. A. Bakar, N. H. Kamis, and H. Aliamis, (2017) "RSA cryptosystem with Fuzzy Set Theory for Encryption and Decryption", In Proceeding of the 13th IMT-GT International Conference on Mathematics, Statistics and their Applications (ICMSA2017), Kedah, Malaysia, 4–7 December, pp. 1–6.

[2] V. V. Zubov, (2019) "An Electronic Signature Within the Digital Economy", European Publisher, Vol. 79, 1st Edition, pp. 1-1576, II International Scientific Conference GCPMED 2019, https://doi.org/10.15405/epsbs.2020.03.89.

[3] N. Saxena, N. S. Chaudhari, (2012) "Secure Encryption with Digital Signature Approach for Short Message Service", IEEE, pp. 803-806, Conference: 2012 World Congress on Information and Communication Technologies, DOI: 10.1109/WICT30467.

[4] P. K. Shukla, A. Aljaedi, P. K. Pareek, A. R. Alharbi, S. S. Jamal, (2022) "AES Based White Box Cryptography in Digital Signature Verification", Sensors, Vol. 22, Issue 23, DOI: 10.3390/s22239444,.

[5] S. Chauhan, N. Gulati, (2016) "Secure Elliptic Curve Digital Signature Algorithm", RESEARCH JOURNAL OF SCIENCE ENGINEERING AND TECHNOLOGY, vol. 6, no. 1, pp. 4-11.

[6] H. M. Al-Mashhadi, A.A. Khalf, (2018) "Hybrid Homomorphic Cryptosystem for Secure Transfer of Color Image on Public Cloud", IJCSNS International Journal of Computer Science and Network Security, Vol. 18, No. 3, pp. 48-55.

[7] A. Mohammed, N. Varol, (2019)"A Review Paper on Cryptography", IEEE, Conference, 2019 7th International Symposium on Digital Forensics and Security (ISDFS), pp. 1-6.

[8] W. Z. M. Al-Humadi, (2020) "Cryptography in Cloud Computing for Data Security and Network Security", Solid State Technology, vol. 63, no. 4, pp. 6965- 6973.

[9] H. M. Al-Mashhadi, I. Q. Abduljaleel, (2017) "Hybrid Homomorphic Cryptosystem for Secure Transfer of Color Image on Public Cloud", 26-27 April 2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT), pp. 1-6.

[10] G. Sahni, G. Singh, (2014) "Modern Cryptographic Technique – A Literature Review", International Journal for Scientific Research & Development, vol. 1, no.12, pp. 2657-2658.

[11] C. G. Desai1, M. B. Patil, B W Gawali, (2010) " A Review on digital Signature Schemes", International Journal of Mathematics, Computer Sciences and Information Technology, vol. 3, no. 2, pp. 387-402.

[12] R. Bhanot, R. Hans, (2015) "A Review and Comparative Analysis of Various Encryption Algorithms", International Journal of Security and Its Applications, vol. 13, no. 3, pp.289-306.

[13] M. Al Saadi, B. Kumar, (2020) "A Review on Elliptic Curve Cryptography", International Journal of Future Generation Communication and Networking, vol.9, no.4, pp.1597–1601.

[14] Md. Ismail Jabiullah, Kanij Nahar Arifa, "An ECSDA-based Security Approach on Blockchain for Cryptocurrency-based Online Transactions", Recent Innovations in Wireless Network Security, Vol 2, No.2, p. 1-12, 2020.

[15] Hemant B. Mahajan, Aparna A. Junnarkar, " Smart Healthcare System Using Integrated and Lightweight ECC with Private Blockchain for Multimedia Medical Data Processing", Multimedia Tools and Applications, p. 1-24, 2023.

[16] M. A. Al-Shabi, (2019) "A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security", International Journal of Scientific and Research Publications, Vol. 9, No.3, p. 576-589.

[17] N. Sharma, B. Kumar, Er. H. kaur, (2017) "A Review of Information Security using Cryptography Technique", International Journal of Advanced Research in Computer Science, vol.8, no.4, pp.1597–1601.

[18] E. Swathi, G. Vivek, G. S. Rani, (2016) "Role of Hash Function in Cryptography", International Journal of Advanced Engineering Research and Science (IJAERS), National Conference on Computer Security, Image Processing, Graphics, Mobility and Analytics (NCCSIGMA), pp.10–13.

[19] A. K. Singh, (2014) "A Review of Elliptic Curve based Signcryption Schemes", International Journal of Computer Applications, vol. 102, no. 6, pp. 26-30.

[20] K.L. Vasundhara, Y. V S Sai Pragathi, Y. Sai Krishna Vaideek, (2018) "A Comparative Study of RSA and ECC", Int. Journal of Engineering Research and Application www.ijera.com, Vol. 8, No. 1. p. 49–52.
[21] T. N. Shankar, G. Sahoo, (2009) "Cryptography with elliptic curves", International Journal Of Computer Science And Applications, Vol. 2, No. 1, p. 38–42.
[22] M. Arif. A. Habib, I. Rufat, S. Azer, (2015) "Study and Implementation of Elliptic Curve Encryption Algorithm for Azerbaijan E-ID", International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, no.5, pp. 3708–3713.
[23] N. Dinarvand, H. Barati, (2017) "An efficient and secure RFID authentication protocol using elliptic curve cryptography", Wireless Netw, vol. 25, no.1, pp.415–428.
[24] N. Koblitz, A. Menezes, S. Vanstone, (2000) "The state of elliptic curve cryptography", Designs, Codes and Cryptography, vol.19, pp.173–193.
[25] W. El Sobky, S. Hamdy, M. H. Mohamed, (2021) "Elliptic Curve Digital Signature Algorithm Challenges and Development Stages", International Journal of Innovative Technology and Exploring Engineering, vol. 10, no.10, pp.121–128.
[26] S. Banerjee, A. Patil, (2020) "ECC Based Encryption Algorithm for Lightweight Cryptography", International Conference on Intelligent Systems Design and Applications, Springer Nature Switzerland AG 2020, vol. 940, pp. 600–609.
[27] A. V. Lucca, G. A. Mariano Sborz, Valderi Reis Quietinho Leithardt, Marko Beko, Cesar Albenes Zeferino, Wemerson Delcio Parreira, (2018) "A Review of Techniques for Implementing Elliptic Curve Point Multiplication on Hardware", Journal of Sensor and Actuator Networks, vol. 10, no. 1, pp. 1–17.
[28] L. D. Singh, K. M. Singh, (2015) "Image Encryption using Elliptic Curve Cryptography", Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015), vol. 54, pp. 472-481.
[29] S. R. Singh, A. K. Khan, S. R. Singh, (2016) "Performance Evaluation of RSA and Elliptic Curve Cryptography", 2016 2nd International Conference on Contemporary Computing and Informatics (ic3i), pp. 302–306, 10.1109/IC3I.2016.7917979.
[30] S. Vasundhara, (2017) "A Review on Elliptic Curve", Global Journal of Pure and Applied Mathematics, vol. 13, no. 9, pp. 995-5011.
[31] M. Amara, A. Siad, (2011) "Elliptic Curve Cryptography and its Applications", 2011 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA), pp. 247- 250.
[32] M. R. Mishra, J. Kar, (2017) "A Study on Diffie-Hellman Key Exchange Protocols", International Journal of Pure and Applied Mathematics, vol. 114, no. 2, pp. 179-189.
[33] M. Yuliana, G. Awaludinsyah, A. Pratiars, A. Sudarsono, (2015) "Design and Implementation of A Secured Personal Identity Based ECC and ECDSA: An Inpatient System", European Scientific Journal, vol. 11, no. 21, pp. 473-483.
[34] M. Al-Zubaidie, Z. Zhang, J. Zhang, (2019) "Efficient and Secure ECDSA Algorithm and its Applications: A Survey", International Journal of Communication Networks and Information Security (IJCNIS), vol. 11, no. 1, pp. 7-35.
[35] K. Ravikumar, A. Udhayakumar, (2013) "A Detailed Study of Elliptic Curve Cryptography Algorithm and Its Performance", International Journal of Engineering Sciences & Research Technology (IJCNIS), vol. 2, no. 10, pp. 2960-2964.
[36] D. Mahto, D. K. Yadav, (2017) "RSA and ECC: A Comparative Analysis", International Journal of Applied Engineering Research, vol. 12, no. 19, pp. 9053-9061.
[37] A. Saini, Vandana, (2022) "A Study on Modifid RSA Algorithm IN Network Swcurity", International Research Journal of Modernization in Engineering Technology and Science, vol. 4, no. 4, pp. 1461 - 1465.
[38] K. N. Kishore, S. Chhetri, (2020) "RSA Algorithm: A Theoretical Study and Implementation", International Research Journal of Modernization in Engineering Technology and Science, vol. 2, no. 5, pp. 834 - 840.
[39] I. Jahan, M. Asif, L. J. Rozario, (2015) "Improved RSA cryptosystem based on the study of number theory and public_key cryptosystems", American Journal of Engineering Research (AJER), vol. 4, no. 1, pp. 143-149.