An Integrated Cryptographic Approach Using Elliptic Curve Cryptography, Triple Data Encryption Standard and Hash-based Message Authentication Code

Farah Tawfiq Abdul Hussien[1,*], Sura khalid salsal [2]

[1]Computer Science Department, University of Technology, Baghdad, Iraq

[2]Deparment of Religious Education and Islamic, Sunni Endowment Diwan, Baghdad, Iraq

Emails: farah.t.alhilo@uotechnology.edu.iq; surakhalid@taleemdeny.edu.iq

Abstract

Security of digital communication becomes of prime importance due to the fast growing cybersecurity attacks. Classical encryption algorithms frequently drop down in offering the vital level of security required to safeguard critical information. The advances in cryptography methods are very important to solve this issue and ensure integrity and privacy. This paper focuses on the weaknesses of the current methods through investigating mixing multiple encryption methods. The research explores whether combining Hash-based Message Authentication Code (HMAC), Elliptic Curve Cryptography (ECC), and Triple Data Encryption Standard (3DES) can provide upgrade to security for end-to-end encryption. The chief objective is to improve and evaluate a powerful encryption framework that make use the strengths of HMAC, ECC and 3DES. This is done by showing how mixing these algorithms together can improve security and reliability levels to safeguard digital communications. An extensive analysis is performed by using several metrics. These involve ciphering and deciphering speed, key generation, NIST test and Avalanche effect. The results show that these combinations increase significantly security level of digital communication. It shows better performance than traditional cryptography in both security and speed. Combining HMAC, ECC and 3DES provide practical solution to increase security level in end-to- end encryption. It improves the vulnerabilities in traditional cryptography by building multi-layer security framework. It is concluded that the proposed framework is powerful and a candidate for developing and has strong resistance against cyber threats.

Keywords: Integrated cryptography; Authentication; NIST Test; Avalanchee effect; HMAC; 3DES; ECC

1. Introduction

The vast increase in cyber threats make data security a critical issue in digital communication. With the huge, depend on digital communication for data transmission, protecting confidentiality and integrity become essential. Classical cryptography fail to provide the required level of security against cyber threats. Which required advanced developments of cryptography [1] [2] [3].

Balancing between strong encryption and high performance is a key problem in data security. Many cryptography methods may prefer to provide high strong security rather than complexity, which considered as practical obstacle [3]. Furthermore, the growing ability of computation that are available for attackers increases the amount of vulnerabilities of the encryption methods. Which required new cryptographic approach that are capable of improving security with no reduction in efficiency [4] [5] [6].

The main goal of this paper is to develop and evaluate a cryptography framework that is able to provide a powerful security level for digital communication. This is performed by exploring the efficiency of integrating HMAC, ECC and 3DES for end-to-end cryptography. Then evaluating the system efficiency by experiments and analysis [7] [8].

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

The importance of this study relies on its ability to provide powerful security for digital communication, which means that the proposed system is capable of protecting critical data against developed cyber threats. The results showed that it could help in improving more secure communication protocol that has a good sound in the future cryptography. It could be implemented successfully in numerous fields involves industries, government and healthcare.

The novelty of this study appears in the combined approach it suggests. Existing works constraint on applications and performance. The proposed system focusses on the integrating HMAC, ECC and 3DES. The proposed system utilizes the strength of each method to provide a powerful end- to- end encryption system. Combining these three algorithms together create a multilayered security system. This in turn leads to offer integrity, confidentiality and more strong resistance against cyber threats. It also has multi-impact in numerous fields including IoT, industry, and healthcare and so on. This is done by providing the unique security requirements for each field.

The main contributions of this paper involve:

1. Presenting a new encryption framework through merging three encryption techniques

2. Enhancing efficiency due to the strength of the ECC algorithm that uses short encryption key. This leads to shorter processing time with high level of security.

3. Increasing security by making use of ECC strength, multi-round of 3DES, while the HMAC offering integrity and authentications.

4. Offering End-to-End data transmission protection to prevent attacks through the transmission stages.

5. Explain the importance of integrated approach in end-to-end encryption to improve security and performance.

6. Offering confidentiality by using 3DES, integrity and authenticity using HMAC.

7. Secure key management system that generates private key and keeping it confidential while exchanging public key securely.

The remaining organization of the paper involve, section 2 presents related works, section 3 discusses the methodology, section 4 system evaluation results and section 5 conclusions.

2. Related Work

This part of the paper deals with various available work that relates to this paper, as shown below:

Johnson, B., Smith, A. 2018, [9], Elliptic Curve Cryptography already have promising future in digital security communication because of the throughput of generating key plus a shortened key. This research shows ECC advancements, raising the role for getting a high-level of security along with minimizing computational performance.

Brown, D., Johnson, C, 2020, [10], Todays researches, like the Brown and Johnson research, diving deeper into Triple Data Encryption Standard in today's cryptographic algorithms. Reviewing process highlights 3DES's adaptability, performance and reliability to achieve security requirements.

Martinez, F., Garcia, E., 2019. [11], Martinez and Garcia research provides deeper understanding into Hash-based Message Authentication Code, which provide data authenticity and integrity authenticity. The research deals with and shows HMAC's throughput and performance in shielding unauthorized access.

Jones, M., Brown, S., 2021 [12], this research shows the main challenging implementing problems of end-to-end encryption, Jones and Brown deals with a balancing process among user's security. This research turns the focus on user considerations and end-to-end encryption impact on different digital platforms of communication.

Kim, 2019, [13], Kim research discusses the cryptography of quantum resistant in this era where security threats are rising. This research examines the downsides of today's cryptographic approach that relates to quantum attacks and discusses potential systems plans for upcoming encryption protocols

Tariq 2023, [14] Ensuring the system can fight against both known and unexpected assaults is the main problem in IoT security. The architecture of IoT systems raises a number of serious security vulnerabilities that have been uncovered by the IoT research community thus far. These worries include problems with communication, networking, and management methods. This study offers a thorough and understandable analysis of the state of anomalies and IoT security principles as of right now. We categorize and examine common security concerns with the tiered architecture of the Internet of Things, encompassing connectivity, communication, and management protocols. We lay the groundwork for IoT security by looking at the most recent attacks, threats, and innovative

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

fixes. Additionally, we establish security objectives that will act as a standard for determining whether a given solution meets the particular IoT used cases.

A. Abirami 2024, [15] Data encoding on the elliptic curve appears to be a preprocessing step required by ECC in order to implement the encryption. Similar to this, a post-processing step needs to be carried out following decryption in order to map or decode the relevant data to the precise location on the elliptic curves. The two most used encoding models are Memory Mapping (MM) and Koblitz Encoding (KE). However, there are disadvantages of both encoding models: the KE requires more computational power, and the MM requires more memory for processing. The suggested improved Koblitz encoding method is applied with the ECC to improve security and get around these problems.

2.1 ECC, 3DES, AND HMAC OVERVIEW

ECC 16-19 is cryptography, where it works with public-key approach. It is well used for the performance efficacy in generating keys and shortened length of key, providing a high-level of security along with reducing computation metrics size. ECC is most commonly known in digital communication security, specifically, in constraining resources and environments along with technologies emergence [16-19]. 3DES 20-23 is a symmetric algorithm for key encryption that uses DES ciphers 3*X one after another for improving security measurements. Through the years, 3DES is still used and commonly used in legends systems that transitions into a modern challenging method [20-22].

HMAC is a development of a secret key and cryptographic hash function combination, resulting in a technique for integrity verification and message authenticity. This algorithm is commonly used in communication protocols security, plus providing an additional protection layer against malicious attacks [24,25].

3. Methodology

The approach discusses the use of three main algorithms. At first, ECC algorithm is used for generating a key-pair, private and public keys for both ends (sender and receiver), alongside the measurement of execution time. After that, the private key of one end (sender) and the public key of the other end (receiver) and vice versa are used to generate a shared key by ECC key agreement. Secondly, 3DES algorithm uses the latter shared secret key to generate a symmetric key that is used for 3DES encryption and decryption, keeping track of the operations of execution time. At last, a HMAC is a result of using algorithm called SHA256 hashing which generates key randomly, execution time is then has been recorded. In summary, ECC generates shared key from private and public keys, then a 3DES encrypts and decrypts data using the symmetric key from shared secret key, and finally HMAC are applied for verification and authenticity. The method is then visualized by crossing of the original byte values of encrypted data, resulting a representation graphically for improved encryption process comprehension. The flowchart below:

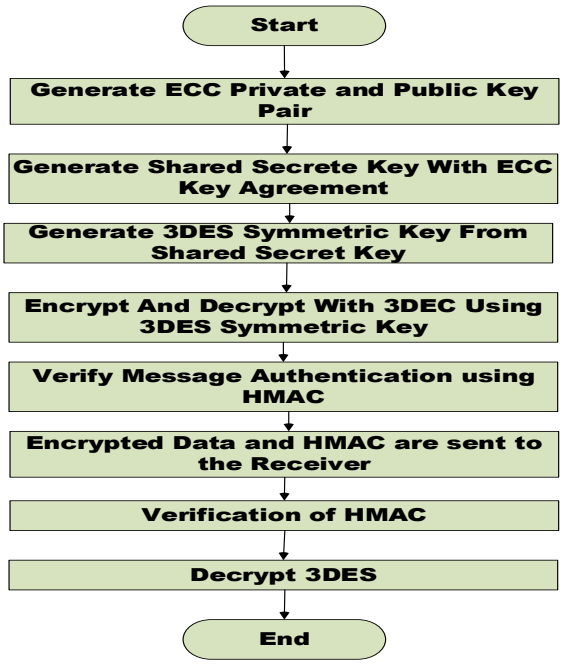

Figure 1. The Proposed System Flowchart

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

Elliptic Curve Cryptography (ECC), its main advantages are:
- Strong Security: It is a higher security level in contrast to other algorithms.
- Resource Consumption: This is effective in constrained operating system like Internet of Things and Mobile Devices where hardware resources are limited compared to others.
- Quantum Attacks Resistance: The usage of curves alongside with big primes makes it an effective candidate for quantum cryptography.
- Standardization: Many well-known corporations are using ECC, like Secure Shell, National Institute of Standards and Technology, Transport Layer Security and others.

Triple Data Encryption Standard (3DES): its important benefits are:
- Backward Compatibility: it is widely used spread and supported in large organizations and protocols.
- Familiarities: it is widely analyzed and supported throughput the users. This increases the confidence for developers to develop their security methods and techniques in using them.
- Regulatory Compliance: Some regions and industries have their own rules; one of them is regulatory mandate of encryption algorithms.

Hash-based Message Authentication Code (HMAC): several advantages are highlighted here:
- Message Integrity: using hash value ensures the message integrity depending on secret key and message content. It detects if there is any modification on the content.
- Authentication: it ensures that the text only is sent to the designated transmitter. In addition, the transmitter and receiver exclusively know the HMAC secret key.
- Resistance to Collision Attacks: one of the basic intentions of developing HMAC was the resistance against the collision attacks since two various inputs results in the same hash. On the other hand, it is infeasible for the attacker to produce two plaintexts with one hash value.

3.1 The Proposed System Algorithm

Algorithm steps for ECC key pair generation, 3DES encryption/decryption, and HMAC generation.

Algorithm 1: Proposed System Implementation

Input: Plaintext

Output: Decrypted Plaintext

Begin:

Step.1: Key Generation

Functions: Generate ECC key () and ECC key agreement (private key, public key)

Parameters: private key, public key

Generate a private key using the elliptic curve cryptography (ECC) algorithm SECP256R1.
Derive the corresponding public key from the generated private key.
Return both the private key and the public key as the result of the key generation process.
Generate shared secret key from previously generated private key, public key as ECC key agreement

Step.2: Date Encryption and Decryption

Functions: ECC key agreement () and encrypt decrypt 3des ()

Parameters: Plaintext, Shared Secret Key

Create a 3DES cipher instance with the provided key using the Electronic Codebook (ECB) mode.
Obtain an encryptor from the cipher instance to encrypt the input data.
Encrypt the data using the obtained encryptor and finalize the encryption process.
Obtain a decryptor from the cipher instance to decrypt the encrypted data.
Decrypt the encrypted data using the decryptor and finalize the decryption process.
Return the encrypted data and the decrypted data as the result of the encryption and decryption process.

Step.3: HMAC Generation

Functions: Generate HMAC()

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

Parameters: Plaintext, HMAC Key

Choose a secure hash algorithm, such as SHA256, for HMAC generation.
Create an HMAC instance with the provided key and chosen hash algorithm.
Update the HMAC instance with the input data to compute the HMAC value.
Finalize the HMAC computation to obtain the HMAC value.
Return the computed HMAC value as the result of the HMAC generation process.

End:

Here are explanations of the algorithm. First step represents ECC to create key pair (agreement). First ECC parameters are initialized, where prime (p) is defined for finite field, curve coefficients (a, b), curve base point (G), base point order (n) and cofactor (h). Random private key is chosen then compute public key Q eq. (1)

$$Q = d * G \quad (1)$$

Next key exchange EC-DH and shared secret agreement CSS are performed. These operations are explained with results in section 4.2.

Second step for encryption process, shared secret key CSS is used to create the encryption key. Message is encrypted using 3DES. For decryption, CSS is retrieved and used to decrypt the ciphertext to obtain the original plaintext. ECB mode is used. Step three creates HMAC using SHA256 algorithm; this is used to verify authenticity and integrity.

4. Experiment RESULTS AND DISCUSSION

This section discusses the environment of the resultant experimental results and evaluating the results using several criteria as described below.

4.1 Environment Description

The proposed system was implemented using Python Language applied on windows 10 installed on hp processor core-i7. The sample data used in encryption and decryption was very famous text known to include all English Alphabetic Letters and always used for testing security algorithms, which is **"The quick brown fox jumps over the lazy dog. ".**

4.2 Results and Analysis

This section explains example of several steps of system execution including: key exchange, encryption, decryption, integrity and visualization.Values in table 1 represent secret share between Alice (A) and Bob (B), showed key exchange(secret share) successfully using ECC.

Table 1: Secret Share Exchange

Alice's Shared Secret	69b3ce4607c29c6f84d062b2a9a6e8227b2ec1fa5e4264571bc55c47cbf64e95
Bob's Shared Secret	69b3ce4607c29c6f84d062b2a9a6e8227b2ec1fa5e4264571bc55c47cbf64e95

This procedure contains three operations include:

1. Creating key pair (CKP)

- Create private (Ka) and public (Pa) key pairs for A using ECC
- Create private (Kb) and public (Pb) key pairs for B using ECC

2. Exchanging public key (EPK)

- Pa is sent from A to B
- Pb is sent from B to A

3. Calculating secret share (CSS)

- For A side CSS is computed using Ka and Pb

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

CSS(A) = EC-DH(Ka,Pb).....(2)

- For B side CSS is computed using Kb and Pa

CSS(B)= EC-DH(Kb, Pa) .....(3)

Where EC-DH represents Elliptic Curve Diffie Hellman

As shown in table 1 both A and B have the same CSS because EC-DH(Ka,Pb) = EC-DH(Kb, Pa)

This result , that key exchange operation has performed successfully between A and B and both parties have the same key for encryption and decryption. The importance of this operation represented in the fact that CSS is considered as basis for creating symmetric keys for encryption (3DES) and message authentication (HMAC). In addition, it provide extra security since it is known only for A and B. It increases the difficulty against intruders to crack the encryption. It provides higher security for communication.

Table 2: Encryption Alice side

Alice's Encryption Key	db110e95cef222aee517d1c1e7e72b6ae8a380a55f8b65b1471cc7a28bedc629
Encrypted Message	ad516c9db8004119c2e06375965616ced7554ecb86703ce2456d24c1a585473950d59c136882a97bd8dbab0e3c8af5d41e8deacc48e70ccb9d86899b
Encryption Time	0.00019140000000028579 seconds

The value corresponding to Alice's Encryption Key (Ka) that is generated from CSS is used to encrypt a message using 3DES algorithm, it is represented mathematically as:
Ctxt = 3DES (Ptxt, Ka).....(4)
Where Ctxt= ciphertext and Ptxt = plaintext.
It obvious in the second row of table 2. The value corresponding to “encrypted message” that Ctxt is long, and then it is complex, which lead to conclude that Ptxt is transformed securely and Ctxt cannot be decrypted correctly without the right key. The last row represents encryption time; it is approximately 1.91 X 10-3 seconds (191 microseconds), which means it is a fast encryption operation. It is a benefit for real-time system that needs low latency. The results explain clearly the security of the proposed system. Highlighting its capability of encryption with extra level of security and efficiency.
On Bob side B make use of CSS and compute encryption key of A correctly to decrypt the message.

Table 3: Decryption Bob side

Bob's Decryption Key	db110e95cef222aee517d1c1e7e72b6ae8a380a55f8b65b1471cc7a28bedc629
Decrypted Message	'The quick brown fox jumps over the lazy dog.'
Decryption Time	6.299999999992423e-05 seconds

As shown in table 3 B, compute Ka of A (Bob's Decryption Key) which is the same value in table 2 (Alice's Encryption Key). Ka is used to decrypt the encrypted message and retrieve the original value of the message that was first encrypted by A. Decryption time is about 63 microseconds, it is very short time indicating the computational efficiency. It clear that encryption and decryption keys of both sides are identical that prove the powerful performance of key processes (exchange, derivation). In other word means, key integrity has been proved. Matching between the decrypted ciphertext and the original plaintext proved message confidentiality. The system efficiency is shown by the speed of the decryption time.

Ptxt= 3DES(Ctxt, Ka).....(5)

Message integrity verified results as shown in table 4.

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

Table 4: Execution times

Verification Time	4.3599999999699435e-05 seconds
Total Execution Time	0.1306010999999998 seconds

Verification time is performed using HMAC to ensure integrity and authenticity. Total execution time include all the processes. It involves (key generation, key exchange, encryption, HMAC creation, and decryption and HMAC confirmation. Total execution time is about 130.6 milliseconds. Short verification time proves system efficiency and suitability for real time systems that need fast verification time, which means that this system provides higher security and efficiency.

Table 5 shows keys generated for each process.

Table 5: shows keys generated for each process.

ECC Key	6\xeb\xb2\xbbG\xb5\xa7@\xdb\xce\xdc\x86SZ\xdf\x15\xba\xfe9\xda\x85\x0c\xd1q/\x00\xdat\x06\x10\x95\t	Part of ECC key pairs
3DES Key	\x99\xb8\x89\xb3l\xa9\xb0\x08\xda"Kp\x11z\x85\x0f\x01!>\x99\xc1\xcb99	A key is derived from CSS and used for 3DES encryption
HMAC Key	!\xf3\x16\xf9\xc4\x05\x9f\xbd \xe9V\xda+,$\xc9v\xbe\xbbD\x07y\xbc\x97	Used to generate HMAC

Table 6: Generated Keys Time

ECC Key Agreement Time	~ 0.0 seconds
3DES Key Generation Time	~ 0.0 seconds
RSA Key Generation Time	0.290067 seconds

Table 6 represents a comparison between ECC, 3DES in the suggested system and RSA. It is clear by the results that both ECC and 3DES required nearly zero time (short and fast processes). Which makes them appropriate for applications on speed and security.RSA is more suitable for applications that focus on confidentiality.

Table 7: Encryption and Decryption Time

Proposed System	0.03536820411682129 seconds
RSA	0.10818 seconds

Table 7 shows that the system is faster than RSA algorithm.

Table 8: Total Execution Time

Proposed System	0.03536820411682129 seconds
RSA	0.300885 seconds

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

Table 8 shows that the total execution time of the proposed system is shorter than RSA.
Figure 2 shows multiple features that proves that the encryption was efficient such as, randomness, from the graph it is obvious that there is no pattern between the encrypted and original data nor among encrypted data, which it makes. Another aspect is the loss of structure where the encrypted data lacks any correlation with the original plaintext. While the complexity of the graph shows a difficulty for the attacker to gain any meaning from the ciphertext without the knowledge of encryption key.

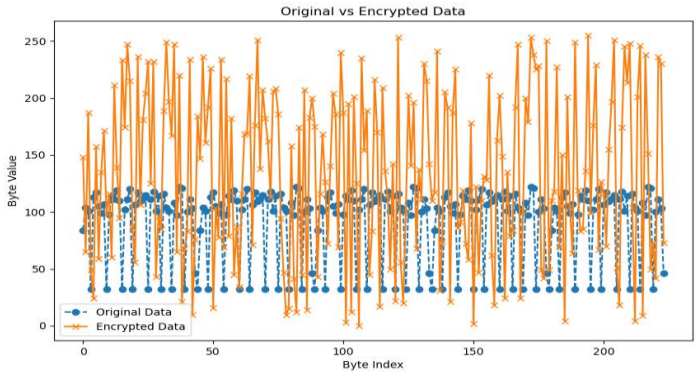

Figure 2. Original Vs. Encrypted Data Feature Comparison

4.3 NIST TEST SUITE

NIST is a standard method that is used to measure the performance of the encryption. Thus for several samples of the generated key approximation entropy, run, linear complexity are run to test the randomness of the generated key. Table 9 shows the results of randomness of the generated keys for several samples. The results reflect high randomness.

Table 9: NIST TEST SUITE test

Key	**Approximate entropy**	**Run test**	**Linear complexity**
kvm5hol95a5vhjalknpqpu7iosbmv3nug62sbdu5iudzsysmdng4sjgn1dsmgnsd	0.704	0.964	0.853
ddc7vw6rpu3fdf3p7ushzsknatmvnbrgphqkfu17ub4q8m3dsjgyhg4ei131ut2h	0.905	0.889	0.719
eio5db3ml56rdsdgyojziobye4qh2fdfp7uknp2q1opxj2ry772d65sjf81ns95sk7j	0.492	0.876	0.967
5k3ol8fd6f1gx9f7idj68rsdkap9t77gs8os22b13m4uo3faf34qh6f359zq4l5t	0.510	0.517	0.635
n38janqglks4o34q25aosnatmgk62s6gv2h5jw2shpdhyr2j8n6skjgns3mdgng5	0.816	0.916	0.807

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

yoalqmhddc7vw2ziosnjaf1wdbj6rit7mgphq4skfo1b9jgn38wes54dmng74gns	0.499	0.701	0.510
gnsle3sgn6dgn4f9slka5wr3je3gn2ld3knz1dgae61gsdgmz8dma2kfjleskgnd	0.598	0.915	0.811
nm2wk4n1qg1jpr5lq72m9htfm63m12f4nsfs5kg4fkslg4yoi929isdk75nm9bfd	0.791	0.971	0.943

Table 9 showed that the results for run test are good and high for entropy and complexity with high diversity for key quality. The results of these metrics indicate that most keys are secure and the cryptographic framework is of high resistance against threats with high efficiency.

4.4 Avalanche effect

In this section, the avalanche effect is measured for the proposed technique. It is computed as follow:

Avalanche effect = (No. of bits flipped in the cipher text)/ (No. of bits in the cipher text)

The outcome of the suggested system's avalanche effect is displayed in this section. Either one plaintext bit— the first, middle, or the last bit— was changed for each test. Increased avalanche impact improves the method's security level. Table 2 displays the avalanche test results.

Table 10: AVALANCHE EFFECT RESULT

Plaintext	**The ciphertext**	**%**
1110001111111111	ja7op3158c023be583jh49f93jsv sd d70c50545f3d61607a84khgs 60c2jhlfv4	62.83
0110001111111111	vfdsivgr1771755a8hfry49cfb esd39rj582db586fjd 80b309fc0457ab25d380	81
0110001111111111	82db80by37g4y560okghdg2ufjlgjfhe aa8b0cbfd7b466d309fcnvhdyr050457	70.59
0110001111111110	4b4 suw92ort1 tmnz8u vdk3islpea a6e0642692a0298dmxl2802699fee75d0f25	78
1000111123456789	8c83292db801fli83hhfy7e3eb884ax0 bc7wu 38d7f4f793ddb42b6a937897jdk	58.72
0000111123456789	f7f37dcgg093jf2e2b9d7ejh gew6tfr ddfcbf4870c90buryewid20b4e70sa89d	72
A1B2DDE3245BC6F9	4a24u8eu3fn kfjruev7ae83w22bu d6mto9utaa93193890ef3478285ce3mvbe b5	63.78
A1B2DDE3245BC6F8	95 clffgnbbgtr48etncc4457144b34md 4a6e06e75d0f25yos ikv921fb253e980	69
98766612345678900	ffb27 iqwy9o3m la9d npf5b9f9e74b0g jd833uc3t781b90d2c06f51f213781lsn	68.10
98766612345678901	91dnehdu1b6as2 i6vhqrjfy492kdkg93 ur8274db79ebb83bda5087688ekf54f	77
Amjvtrhcpsjhgawl	c dje21fp433439983d7fbeameerfd6f2691b5cc94 saaa2ebd6efb69woj	64.80

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

Amjvtrhcqsjhgawl	9bls39gacmn9b gh34lk82dkv 1c42692a024sjh4802699fead619c95uv78675fd	77
Abxhskhlerabzskp	0sepwqit9bj287mfe324lknbzxlcce44b4a6e06e75d0f25828acf49653sakf1n	68.04
Abxhskhlirabzskp	e5d70c5050dc4e09244kgjbwoa rtiyu2349idodlkjsndj81be0d9daa545f3d61	81

Table 10 showed that the suggested system creates different, complex and unique ciphertexts, with high sensitiveness to any simple alteration in the input. Furthermore, the extreme variotion in the percentages highlight the randomness and complexity of the system. This lead to conclude that it is a powerful system with high security. System performance and efficiency proved by it is capability of various plaintext of different lengths and types. From the experimental results, the following practical advantages are concluded:

- Strong encryption (multilayer security)
- Secure key management
- Incurring data integrity and verification
- Short keys are generated by ECC, which leads to faster computation and less storage.
- Perfect for real time applications due to short time of encryption and decryption.

4.5 Comparison Analysis

This section presents a comparison between the proposed system and related works

Table 11: Comparison Analysis

	Key management	Compatibility	Performance	Randomness	Execution time	Security level	Computational efficieny	Authentication
[9]	Simple	Easy	High	Moderate	High	High	Moderate	Low
[10]	Complex to implement	Easy	Low	Low	High	High	Moderate	Low
[11]	Sensitive and complex	Moderate	High	Low	High	High	Complex to implement	Low
[12]	Sensitive and risky	Complex to implement	Moderate	High	Moderate	Moderate	Moderate	Low
[13]	Complex to implement	Complex to implement	Moderate	High	Moderate	Moderate	Moderate	Low
[14]	Simple	Moderate	Moderate	High	Moderate	Moderate	Moderate	Low
[15]	Not specified	Complex to implement	Moderate	Hig	Moderate	Low	Moderate	Low
Proposed system	Complex (secure and enhanced)	Compatible with existing systems easily	High	High	Fast	High	High	High

4.6 Discussion

The following results are conducted from the experiment results:

First, strong private and public keys are generated using the ECC where curves and big prime numbers are used. Second, a high-level security shared key is generated form private and public key that are produced earlier. Additionally, the generation of 3DES symmetric keys from the strong ECC shared secret key took a very small time of (~ 0.,0 seconds) in contrast to RSA key generation of large prime numbers (0.290067 seconds). Furthemore,

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

for only encryption and decryption without the key time, the proposed system took (0.03536820411682129 seconds) which is a higher time than RSA that is (0.010818 seconds). Moreover, total process of key generation, encryption and decryption, the proposed system had a smaller time of (0.035368 seconds) while RSA took (0.300885 seconds).

Also, from all the results and analysis some advantages and consideration have been highlighted below:

Regarding Efficiency, the proposed system displays efficient encryption and decryption processes with minimal execution times, ensuring swift data protection. Concerning Key Exchange Security, the use of symmetric encryption algorithms, such as Triple DES, ensures robust encryption while maintaining a shared secret between communicating parties. As for Integrity Verification, iincorporating HMAC for integrity verification enhances the system's security by ensuring message authenticity and integrity. With references to overall measurement of security, integrity verification, secure key exchange mechanisms and efficient encryption combination provides a security measurement of sensitive data protection during data transmission. As for performance optimization, the system's optimization of encryption and decryption processes minimizes computational overhead, making it suitable for real-time applications and large-scale data transmissions. Limitations of the proposed system include:

- Computational complexity: using three-layer cryptography may lead to increase in execution time and resource consuming. It may be consumed for resources for limited computational ability devices like mobiles or IoT devices. It may cost additional resources like memory and power consumption.
- Scalability: increasing data volume may increase encryption and decryption processing time that may affect system efficiency in large-scale application.

5. Conclusion and Future Works

This paper has suggest a novel end-to-end encryption system by combining HMAC, ECC and 3DES algorithms. This combination aims to increase digital communications by making use of the strength points of each algorithm. Chief points in the integrated approach are increasing integrity and confidentiality with strong resistance against threats. The proposed system has shown obvious efficient performance and this seen in the evaluations like encryption and decryption time, key generation, NIST test and avalanche effect. Benefits obtained from this system include creating end-to-end encryption system with higher level of security due to the combination of three algorithms together. As a result, it becomes more difficult against attackers to crack. In addition, the proposed system is efficient and it is suitable for different fields like IoT, finance, healthcare, industry and so on, due to its ability to provide the required security, which is unique for each field. It also showed the impact of integrated system to improve performance and security. Finally, this study make use of focusing on limitations of individual methods to generate powerful integrated system of three algorithms that provide higher level of security and performance. In future work Strong key management implementation provides private keys protection and unauthorized access prevention. Strong cryptographic methods utilization and length of key that is appropriate to prevent from attacks of brute force type. Securing channels of communication, like TLS/SSL, for data confidentiality insurance and transmission integrity. Updating cryptographic libraries regularly along with algorithms to highlight exploits and vulnerabilities. Security audits assessments and comprehension to show security weaknesses and challenges with the system.

Funding: “This research received no external funding”

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] E. Garcia and F. Martinez, “Hash-based Message Authentication Code in Secure Communication,” Cybersecurity Review, vol. 12, no. 1, pp. 78-92, 2019.

[2] P. Rogaway and T. Shrimpton, “Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance,” Journal of Cryptology, vol. 33, no. 1, pp. 193-239, 2019.

[3] Y. S. Aldeen and S. Mazleena, “A New Heuristic Anonymization Technique for Privacy Preserved Datasets Publication on Cloud Computing,” in Journal of Physics: Conference Series, vol. 1003, p. 012030, IOP Publishing, May 2018.

[4] S. Brown and M. Jones, “End-to-End Encryption: Balancing Usability and Security,” Journal of Cybersecurity Studies, vol. 28, no. 2, pp. 201-218, 2021.

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025

[5] F. T. Abdul Hussien and T. W. Aldeen Khairi, "Performance Evaluation of AES, ECC and Logistic Chaotic Map Algorithms in Image Encryption," Int. J. Interact. Mob. Technol., vol. 17, no. 10, pp. 193–211, 2023.

[6] A. G. Al-Mamory, A. M. Al-Sharifi, and A. H. Al-Husseini, "A Novel Cryptographic Algorithm for Secure Cloud Computing," in 2022 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), 2022, pp. 215-220.

[7] M. S. Al-Husseini, A. G. Al-Mamory, and T. A. Al-Azzawi, "Multi-Agent Systems for Secure E-Commerce Transactions," in 2023 International Conference on Cybersecurity and Computer Science (ICCCS), 2023, pp. 101-106.

[8] F. A. Fadhil, F. T. Abdul Hussien, T. W. Aldeen Khairi, and N. Safiullin, "A Proposed Text Encryption Inside Video Using Harris Corner Detection and Salas20 Encryption Algorithm," BSJ, online-First, vol. 7, 2023.

[9] D. J. Bernstein, "The Poly1305-AES Message-Authentication Code," in Advances in Cryptology – CRYPTO 2005, Springer, Berlin, Heidelberg, pp. 32-49, 2020.

[10] S. S. Al-Riyami and K. G. Paterson, "Certificateless Public Key Cryptography," in Advances in Cryptology – ASIACRYPT 2003, 2020, pp. 1-15.

[11] A. Langley, M. Hamburg, and S. Turner, "Elliptic Curves for Security," Available online: https://safecurves.cr.yp.to/, 2018.

[12] A. Smith and B. Johnson, "Advancements in Elliptic Curve Cryptography," Journal of Cryptographic Research, vol. 22, no. 3, pp. 45-62, 2018.

[13] C. Johnson and D. Brown, "Triple Data Encryption Standard: A Contemporary Analysis," International Journal of Information Security, vol. 15, no. 4, pp. 112-130, 2020.

[14] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, "A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review," Sensors, vol. 23, no. 4117, pp. 1-46, 2023.

[15] A. Abirami and S. Palanikumar, "ECC Based Encryption for the Secured Proactive Network Forensic Framework," Iraqi Journal of Science, vol. 65, no. 1, pp. 381-389, 2024.

[16] Y. Kim et al., "Quantum-Resistant Cryptography: Future-proofing Security Protocols," in Conference on Cryptographic Protocols, 2019, pp. 215-230.

[17] C. Johnson, A. Smith, "Elliptic Curve Cryptography: A Comprehensive Overview," International Journal of Cryptographic Research, vol. 25, no. 2, pp. 78-94, 2017.

[18] D. Brown et al., "Triple Data Encryption Standard Revisited: Contemporary Perspectives," Journal of Cybersecurity Studies, vol. 32, no. 1, pp. 112-128, 2020.

[19] E. Garcia and F. Martinez, "Hash-based Message Authentication Code: Principles and Applications," Journal of Cryptographic Engineering, vol. 15, no. 3, pp. 45-62, 2018.

[20] F. T. Hussien, A. M. Rahma, and H. B. A. Wahab, "Design and Implement a New Secure Prototype Structure of E-commerce System," International Journal of Electrical and Computer Engineering, vol. 12, no. 1, pp. 560-571, 2022.

[21] M. H. Ismael and A. T. Maolood, "Proposed Secure Key for Healthcare Platform," Iraqi Journal of Computers, Communications, Control and Systems Engineering, vol. 22, no. 1, pp. 112-118, 2022.

[22] I. M. Hasan and R. F. Ghani, "Blockchain for Authorized Access of Health Insurance IoT System," Iraqi Journal of Computers, Communications, Control and Systems Engineering, vol. 21, no. 3, pp. 76-88, 2021.

[23] Y. H. Ali and H. A. Ressan, "Image Encryption Using Block Cipher Based Serpent Algorithm," Engineering and Technology Journal, vol. 34, no. 2, pp. 278-286, 2016.

[24] M. T. Abdulhadi and A. R. Abbas, "Human Action Behavior Recognition in Still Images with Proposed Frames Selection Using Transfer Learning," International Journal of Online & Biomedical Engineering, vol. 19, no. 6, pp. 47-65, 2023.

[25] M. Sh. Oudah and A. T. Maolood, "IoT-Key Agreement Protocol Based on the Lowest Work-Load Versions of the Elliptic Curve Diffie-Hellman," Iraqi Journal of Science, vol. 64, no. 8, pp. 4198–4207, 2023.

DOI: https://doi.org/10.54216/JCIM.160207
Received: January 19, 2025 Revised: February 27, 2025 Accepted: March 27, 2025