Encryption Algorithms Modeling in Detecting Man in the Middle Attack in Medical Organizations

Sulaiman Alnasser, Raed Alsaqour
Department of Information Technology, College of Computing and Informatics, Saudi Electronic University, Riyadh 93499, Saudi Arabia

***Abstract*—A cyberattack is a serious crime that could affect medical organizations. These attacks could affect medical organization sensitive data disclosure, loss of organization data, or the business's continuity. The Man-in-The-Middle (MITM) attack is one of the threats that could impact medical organizations. It happens when unapproved outsiders break into the traffic between two parties that think they are conversing directly. At the same time, the adversary can access, read, and change secret information. Because of that, medical organizations lose confidentiality, integrity, and availability. Data encryption is a solution that changes vital information to unreadable by unauthorized and unintended parties. It could involve protecting data with cryptography, usually by leveraging a scrambled code. Only the individuals with the decoding key can read the information. There is no full protection due to the variety of MITM attacks. Each encryption algorithm has its advantages and disadvantages, like the speed of encryption and decryption, strength of the algorithm, and the cipher type. This research investigates the MITM attacks and comprehensively compares the Rivest Shamir Adleman algorithm and the Elliptic Curve Cryptography algorithm.**

Keywords*—*Cyberattack; medical organization; man in the middle attack; encryption algorithm; rivest shamir adleman algorithm; elliptic curve cryptography algorithm

I. Introduction

In the contemporary organizational environment, governments, businesses, medical, and individuals all store data in electronic form. Electronic data storage is more effective than the previous physical storage forms since it is more compact, allows instantaneous transfer, and is easier to access information via databases [1]. Over time, the value of data increases, and organizations and individuals widely recognize stored data as among the most valuable items that must be protected against all potential threats. However, with such a notable electronic revolution, effective data storage and management face multiple new security threats that are potentially more damaging [2]. For example, electronic data has a high risk of being copied, leaving the original unaltered, or stolen, and has a high vulnerability for interceptions and alterations. Therefore, an effective data security measure must enhance secrecy, integrity, and availability. Part of the technical services crucial for optimizing data protection include data authentication and encryption [3].

A Man-in-The-Middle (MITM) attack is one of the threats that could impact medical organizations [4]. It happens when unapproved outsiders break into the traffic between two parties that think they are conversing directly. At the same time, the adversary can access, read, and change secret information. Because of that, organizations lose confidentiality, integrity, and availability. Data encryption is a powerful solution to eliminate the MITM attack [5]. It encompasses translating or encoding data into another form or a code to ensure that it is only accessible to persons with access to the secret key. It is a robust approach to protecting private information and sensitive data. It enhances communication security between different parties and servers. Encrypted data is largely depicted as ciphertext. The process is the most effective and popular information security method [6].

Data encryption is central in enhancing and maintaining the confidentiality of sensitive and private information, and the technology also increases data safety among remote workers. Therefore, it is an essential security safeguard for corporations, and in the long term, it positively impacts consumer trust and overall profitability [7].

Asymmetric encryption algorithms can be complicated, especially; most businesses and individuals rely on this type of encryption since they are strong and hard to break. Unfortunately, studies reveal that in the contemporary technology-infiltrated market setting, a wide range of cybersecurity issues and threats negatively affect entities' effective functionality [7]. However, with data encryption being done properly, for instance, by leveraging a high enough level of encryption and adequate safeguarding of the respective encryption key, the security and privacy of various features can be enhanced. This is vital in eliminating potential threats that could ultimately compromise data safety and security. Furthermore, the Internet of things (IoTs) started to become a valid solution in the medical field. Currently, many surgeries are done remotely [8]. So, strong, accurate, and speed algorithms are primary conditions that medical sectors cannot abandon.

This study presents a critical analysis of encryption. It provides cases of MITM attacks to reduce the risk of MITM attacks in corporations and urges organizations to encrypt their data. Additionally, this research aims to better understand the Rivest Shamir Adleman (RSA) and Elliptic Curve Cryptography (ECC) algorithms. Based on that, the medical organizations' systems may select the most appropriate ones for their needs. The proposed innovative method involves evaluating both algorithms using four performance measures on five distinct security level bits, as suggested by National Institute of Standards and Technology (NIST) [9]. The result will enhance the existing medical organizations' systems and

help the medical organizations' engineers to choose the optimal encryption algorithm.

The body of the article has the following structure: Section II includes the background and related work. Section III presents the research methodology. Section IV explains the results and discussion. Finally, the conclusions and possible guidelines for further work are presented in Section V.

II. Background and Related Work

A. Background

Modern society relies on communication networks and the Internet for almost every facet of everyday activities. Like online home banking, social media networks, and online shopping, most applications need cellular networks or the Internet. This is the major target of hackers since it involves transmitting sensitive information. Hackers prey on businesses and organizations, causing enormous financial damage [10]. The MITM attacks are the most effective method of controlling sensitive end-user information being sent. Therefore, it is one of the most serious risks to the security of wireless networks. A typical MITM attack scenario includes the victims, the two endpoints, and the perpetrators, a third party [11]

During a security breach, an attacker gets into a communication system and changes messages between the two endpoints. Third-party attackers can intercept, alter, replace, or alter information being carried across the communication channel between two endpoints when they conduct MITM attacks. Due to their lack of knowledge, victims feel that their communication channels are secure. Global System for Mobile (GSM), Universal Mobile Telecommunications System (UMTS), Bluetooth, and Wi-Fi are a few of the communication channels that may be used to execute such MITM attacks [11]. The hackers also compromise the data's security by targeting the actual data sent between the endpoints.

An adversary may tamper with the secrecy and integrity of communication [3]. Alternatively, an adversary may stop communication between the two parties and weaken the availability issue by intercepting, modifying, or destroying the messages. As shown in Fig. 1, the authors explain in how user one and user two do not have a trusted connection with the MITM.

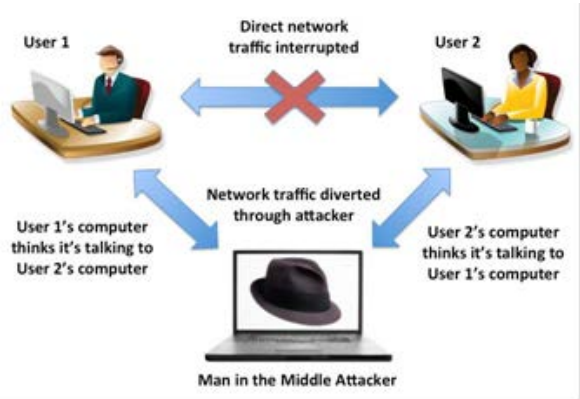

Fig. 1. Traditional MITM Attack [3].

A malicious attacker can intercept and decrypt the data passed between sensors and the Local Processing Unit (LPU) [4]. Consequently, an attacker may access confidential information and assess the recorded data to discover significant changes or clinical concerns. For example, MITM may change incorrect data and communicate normal readings to the LPU, preventing the monitoring system from sounding an alert when a patient asks for help. The author demonstrated the same malign spirit by using the Medtronic infusion pump to block it from administering insulin or overloading diabetic patients with insulin. The sensor's data is usually normal, with just a few exceptions. In [4], the authors note that the LPU analyzes the data to look for significant shifts in measurements before sounding the alert. Because the MITM cannot access personal details, the sensor merely transmits a digital signature of what it has collected. This interval between readings is preserved by the change detection mechanism in the LPU. Researchers employ Locality Sensitive Hashing (LSH) to create an irreversible information fingerprint that makes it impossible for an adversary to deduce or access confidential data.

In contrast, sending signatures rather than measurements greatly decreases the size of the data packet and, as a result, the amount of energy needed to transmit it. IMD is a trusted healthcare platform that has sensitive patient information. The authors in [11] explain that attackers can listen, change, and drop the messages when the medical unit loses authentication with the patient's IMD (Fig. 2).

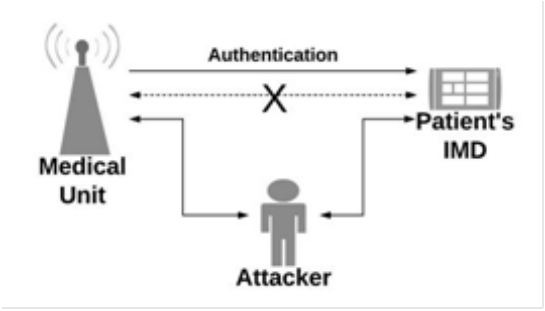

Fig. 2. Example of MITM Attacks Scheme in a Medical Unit [12].

B. Related Work

In [4], the authors reviewed aims to mitigate MITM attacks on the Internet of medical systems. Specifically, the attack happens by identifying the respective monitored individuals' healthcare emergency and replaying normal physiological data to prevent the system from raising the alarm. The authors depend on Locality Sensitive Hashing (LSH) signature as transmitted instead of physiological value. To prevent modification, replay, and black hole attacks, a Hash-Based Message Authentication Code (HMAC) is used with a key based on the Received Signal Strength Indicator (RSSI) value measured on both sensors and (Local Processing Unit) LPU. Also, propose a system that could be leveraged to prevent the devastating aftermath effects of the alarms of the remote healthcare monitoring system.

In [13], the authors proposed an efficient scheme to help design a generalized yet robust authentication protocol in medical systems. It is a countermeasure against medical

facilities' potential man-in-the-middle attacks and impersonation attacks. Specifically, the countermeasure involves mutual authentication between users, their devices, and the system's cloud server. It also involves standardizing a key agreement scheme with Elliptic Curve Cryptography (ECC). With the model in place, the authors opine that the keys are thoroughly secured, hence not copy-able; therefore, it is pivotal in enhancing security robustness.

In [14], the authors examined and proposed using a lightweight cross-layer trust computation algorithm for the MITM attacker detection, known as IC-MADS. IC-MADS are identified to have two notable contributions that the others, such as the trust-based and cryptography-based solutions, failed to possess, which relates to energy-efficient clustering and cross-layer attack detection. According to the authors, simulation results identify IC-MADS as efficient in achieving better protection against potential MIMA attacks with minimum energy consumption.

In [15], the authors have proposed a biometric-based authentication scheme that would help ensure secure access to patient's electronic health records virtually from any location. There has been a notable trend in Healthcare 4.0-based diagnostics systems globally. However, the authors often identify that patient records are continually stored in Electronic Health Records (EHR) repositories. Therefore, they use RSA encryption to protect patient data security and privacy risks. Nevertheless, results attribute the scheme as superior to the previously used state-of-the-art schemes.

In [16], the authors discussed the aspects of big data, especially in the modern-day context where it has been most impactful across industries. For example, benefits include driving health research, enhancing knowledge discovery, and improving personal health management in the healthcare domain. Primary identified big data challenges include technical challenges and privacy and security issues. The authors recommend incorporating encryption and anonymization as the best practices in enhancing big data security and privacy.

In [17], the authors identified the potential risk of the Internet of Things (IoT) era, especially with the continued advent of technology. Therefore, the security and protection of IoT depend on various factors, ranging from the producer of the device and their respective perception of device protection to the end-user and their probable awareness of the associated risks. Furthermore, in [17], the authors noted that attackers are often at an advantage concerning their inherent knowledge and technology. Therefore, despite the apparent great potential of IoT, it is faced with considerable risks that stem from insufficient protection. Therefore, advancing prevention and reactiveness is the best approach to managing the situation.

Table I shows the advantages and disadvantages of exploring related work.

TABLE I. ADVANTAGES AND DISADVANTAGES OF THE RESPECTIVE STUDIES

Related Studies	Advantages	Disadvantages
[4]	The authors successfully proposed an effective mitigation strategy to lessen the negative impact of MITM attacks on the Internet of Medical Things (IoMT). Furthermore, the approach successfully addressed critical domains, such as the privacy of the physiological data and energy consumption.	Using a classification model increases the risk of failure of the strategy.
[13]	The proposed scheme is a robust authentication protocol and can fulfill its scope within the medical infrastructure.	Mutual authentication can be disastrous, especially when parties fail to honor their pledges.
[14]	Simulation results prove that IC-MADS are integral in better protection against MITM attacks and leverage minimum overhead and energy consumption.	It is associated with a limited power rating as it is usually impossible to manufacture higher power.
[15]	The proposed biometric-based authentication method proved superior in its computational and communication costs, especially when compared to conventional schemes.	The proposed solution took more time than expected. The author suggests decreasing the encryption key.
[16]	Encryption and anonymization are unique solutions to the presenting data privacy and security challenges.	Encryption could be disadvantageous as it consumes significant resources and has issues with data compatibility.
[17]	IoT has great potential in present-day society due to increased technology and expertise. Providing certificates to identify each device will mitigate the MITM attack.	Certificates might be an insufficient solution due to the high cost. In addition, insufficient protection of the users results in increased privacy concerns.

III. RESEARCH METHODOLOGY

This chapter will debate the proposed research methodology used to achieve the research objective. Assimilation is the most popular experimentation technique in the network field. This chapter further discusses the proposed authentication scheme aimed at helping to overcome the impersonation and the MITM attacks during the user login and data storing phases, respectively. First, it introduces the authentication scheme and details how the attacks occur. It then proposes a solution for the attacks.

This research proposes a new authentication scheme for cloud computing for mobile users. The proposal is motivated by the rising levels of attacks on wireless channels. The research views that authentication and verification are critical elements that can help enhance the security channels between mobile users and cloud computing. The prosed solution in this research involves implementing two-layer security with a crucial agreement scheme. Cryptography, a well-known approach for securing communication, is proposed to be the baseline for the proposed solution.

The methodology of this research is demonstrated in Fig. 3. The figure provides an overview of the different phases of this research methodology. It begins with studying previously done literature reviews related to the research area. The focus is mainly on the literature reviews completed in recent years, whose primary study is how health care systems' phases work. After that, the focus is on the MITM attack on the health care systems. Then, the research will implement a solution that mitigates the MITM attacks. The performance of the proposed solution is then analyzed. Finally, the results are generated and discussed.

In addition, the experiment was done on a hardware server. A real server chassis model Quanta S5HF-1U was rented. To conduct the experiments, the processor is 1× AMD EPYC 7281. CPU - 16C/32T - 2.1 GHz. Storage 2 × 1 TB NVMe. 96 GB DDR4 ECC. The operating system is Ubuntu 22.04 LTS.

Furthermore, This research used python language to experiment. Python was chosen because it is a dynamically semantic, interpreted, object-oriented high-level programming language. Its high-level built-in data structures, combined with dynamic typing and binding, make it ideal for Rapid Application Development and scripting or glue language for connecting existing components.

A. Health Care System

Two phases form a basic wireless and healthcare structure. The first phase is the login phase, while the second phase is called the information storing stage. The assumption is that an impersonation attack occurs at the user-long stage. An assumption is that an attack on the man in the middle is usually in the information storing stage [13] (See Fig. 4).

The absence of authentication between a sender and receiver is one of the elements leading to an impersonation attack in a medical center. An impersonation attack involves the success of external adversaries in stealing the identity of one authorized system user or stealing a communication protocol. Therefore, an impersonation attack is bound to occur when one of the user accounts is stolen. In most instances, these attacks are usually through emails attempting to impersonate someone trustworthy. The attacks also attempt to mimic an organization to access company information and finances [18].

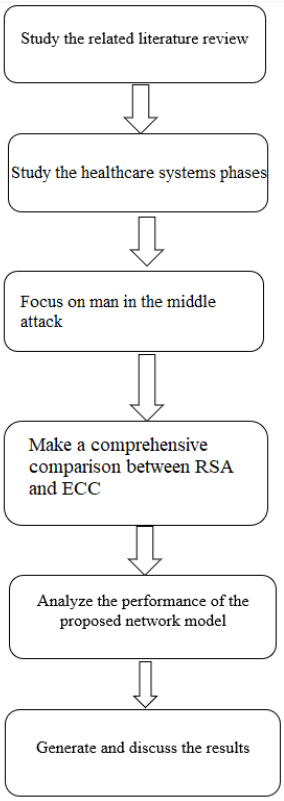

Fig. 3. Research Methodology.

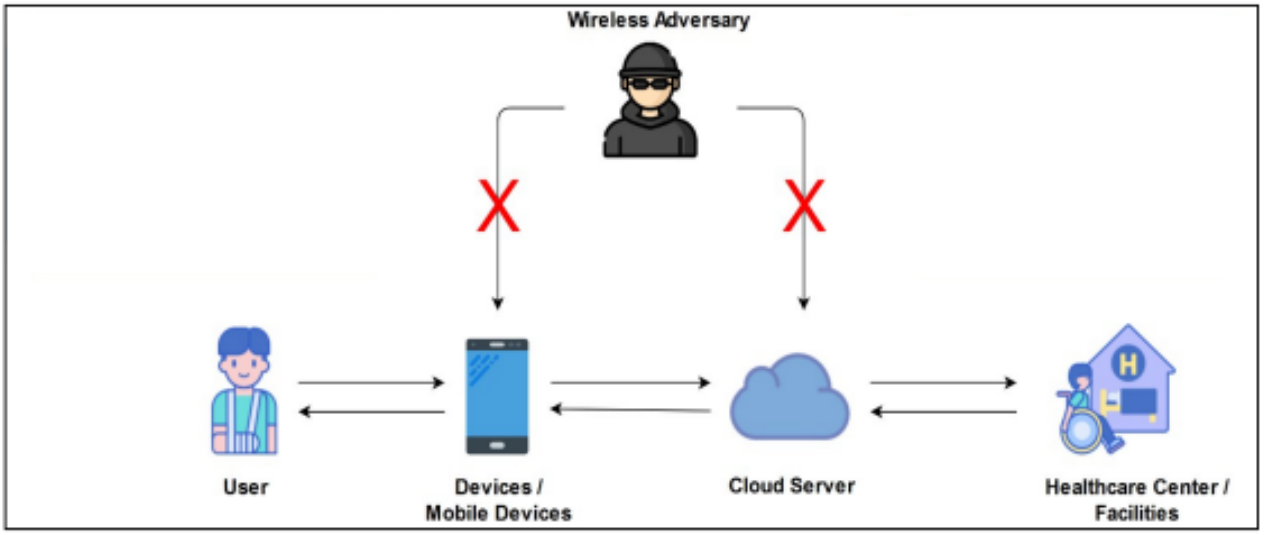

Fig. 4. Wireless Adversary Attack [13].

B. Asymmetric Keys

Asymmetric key cryptography, often known as public-key cryptography, is a type of encryption that uses asymmetric keys. In cryptography, keys are divided into two types: the first is a public key used for encryption, and the second is a private key used for decryption, as shown in Fig. 5 and Fig. 6. A certain user or device can only access the private key. Nevertheless, on the other hand, the public key is disseminated to all users and devices participating [19].

The speed and security strength are the most significant shortcomings of asymmetric ciphers; they are significantly slower than symmetric algorithms and more prone to intruder attacks, making the key exchange process more difficult. The advantage of using an asymmetric key technique is that it eliminates the need to distribute the encryption key between parties. Private keys are kept secret, and only public keys are made available to the public [19].

In addition, Digital signatures are possible with public key encryption, allowing the communication recipient to verify that the message came from the sender who specified the digital signature. With digital signatures in public-key encryption, the receiver can determine whether or not the message has been altered during transit. No changes can be made to a digitally signed communication without invalidating the signature [19].

If part A wants to communicate with part B confidentially, it should encrypt a message using B's publicly available key. Because only B has access to the associated private key, such communication can only be deciphered by B.

If part A wants to send an authenticated message to party B, as shown in Fig. 7, part A should encrypt the message using A's private key. Because this message can only be deciphered using A's public key, which may be used to verify the message's authenticity, A is indeed the message's source [20].

At the same time, public-key cryptography may support message authentication and confidentiality. For example, Fig. 8 shows how public-key cryptography ensures authentication.

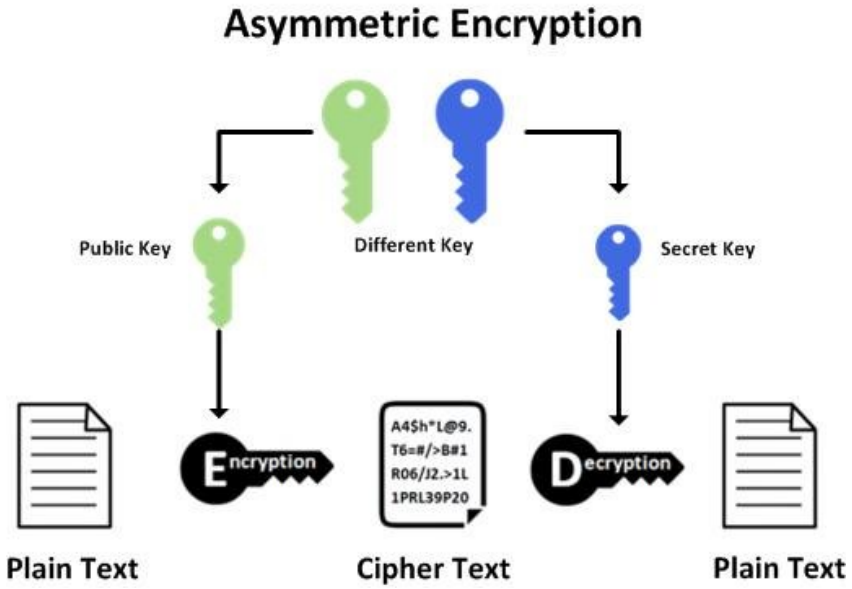

Fig. 5. Asymmetric Encryption [19].

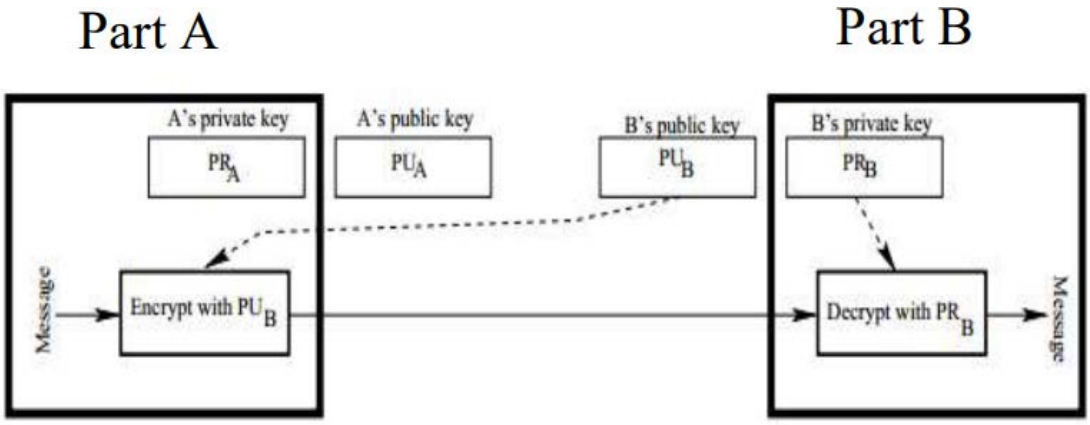

Fig. 6. Confidential Communication in Public-Key Cryptography [20].

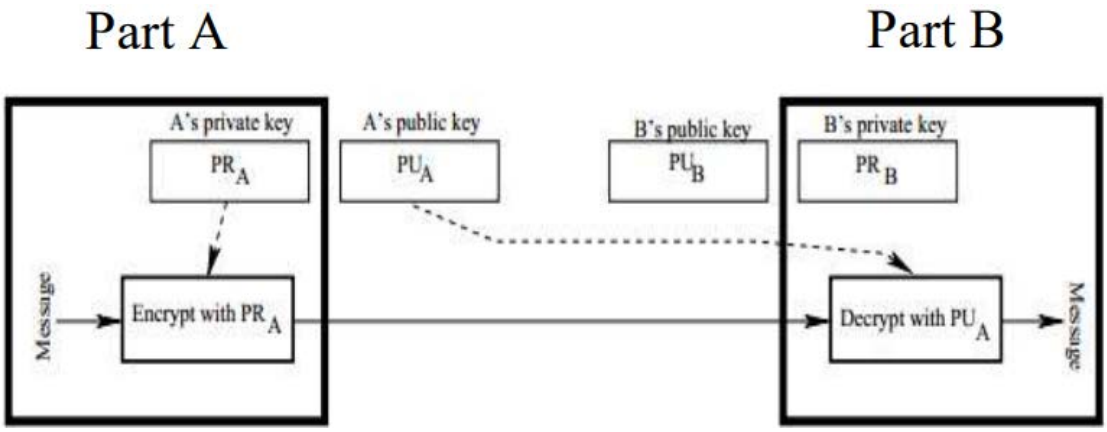

Fig. 7. Authenticated Communication in Public-Key Cryptography [20].

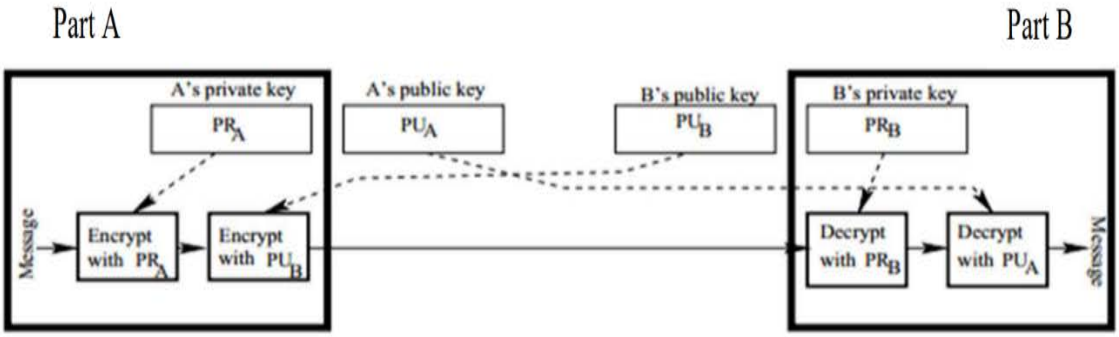

Fig. 8. Confidentiality and Authentication [20].

Fig. 8 illustrates how public-key cryptography can be used for confidentiality and authentication, including digital signatures. RSA and ECC can provide security services such as Confidentiality, Integrity, Authentication, and Authorization. Authors in [20] defined them as below as;

1) Confidentiality: Any illegal connection to the data via this security service is refused.

2) Integrity: Ensuring that messages sent to a destination have not been tampered.

3) Authentication: Any anonymous/malicious node wishes to interact with network nodes. It needs the authorized node's public key pair.

4) Authorization: This service assigns each node a unique key pair (private and public) for decryption and encryption.

C. RSA

The RSA algorithm, named after its creators, is the first method used for data encryption and digital signatures simultaneously. It is the most widely used today. The RSA algorithm's security depends on how difficult it is to decompose large integers. The public and private keys are created using two huge prime integers employed to generate the public and private keys. A rough estimate of how difficult it is to deduce the plaintext from the signal key and the ciphertext is the decomposition of the product of two large prime numbers [21].

In the Internet Key Exchange (IKE) architecture, the RSA algorithm has been proposed as a potential authentication technique. The Diffie-Hellman key exchange method is critical to the framework's security architecture. Participants interact using the Diffie-Hellman algorithm and create shared keys at the start of a key agreement session. These shared keys will be utilized for the key agreement protocol of the next steps [21].

Encrypting with private or public keys provides RSA users with many services. If the public key is used for encryption, the data must be decrypted with the private key. This is ideal for delivering sensitive data over a network or over the Internet, where the data recipient sends the data sender their public key. The data sender then encrypts the sensitive information with the recipient's public key and sends it to them. The private key owner can only decrypt the sensitive data because the public key encrypts it. Thus, even if the data is intercepted in transit, only the intended recipient can decode it. Fig. 9 explains how RSA encryption works [21].

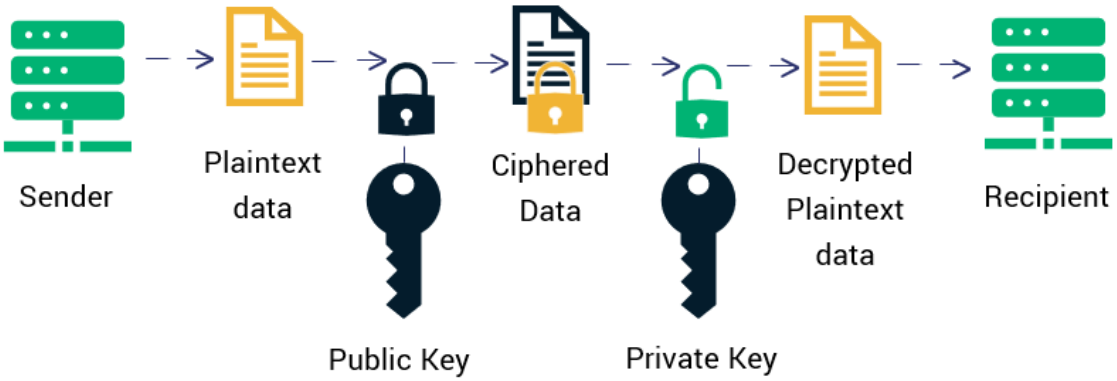

Fig. 9. How RSA Works [21].

Encrypting a message with a private key is the other asymmetric encryption method with RSA. In this case, the data sender encrypts the data with their private key and sends the encrypted data together with their public key to the data recipient. The recipient can then decrypt the data using the sender's public key, proving that the sender is whom they say they are. The data could be stolen and read in transit using this method. However, the primary purpose of encryption is to prove the sender's identity. The public key would be unable to decrypt the new message if the data was stolen and modified in route, and the recipient would be aware that the data had been altered in transit [21].

The technical aspects of RSA are based on the premise that it is simple to construct a number by multiplying two sufficiently large numbers together. Still, it is incredibly difficult to factorize that number back into the original prime numbers. For example, two numbers are used to construct the public and private keys, one of which is a product of two huge prime numbers. To calculate their value, they both use the same two prime numbers [21].

RSA is widely regarded as the first real-world asymmetric-key cryptosystem. For public-key cryptography, it becomes the de-facto standard. The integer factorization problem guarantees its safety. However, the decryption technique used by RSA is less efficient than the encrypting process. Many scholars have advocated using the Chinese Remainder Theorem (CRT) to improve the efficiency of RSA decryption. Authors in [21] suggested a CRT model improves RSA decryption time. They also advocated using a small matrix order to obtain big modulus and cryptographic keys. Larger key sizes are required for better and stronger data security, which involves higher overhead on computer systems. Small gadgets are becoming increasingly vital in today's digital world, with less memory but need security to meet market demand. RSA becomes a secondary consideration in this case.

RSA Algorithm

Key Generation
Step 1. Select p, q where p and q both are primes, $p \neq q$
Step 2. Calculate $n = pq$
Step 3. Calculate $\Phi(n) = (p-1)(q-1)$
Step 4. Select integer e $gcd(\Phi(n), e) = 1$; where $1 < e < \Phi(n)$
Step 5. Calculate d ; $d \equiv e^{-1} (mod\ \Phi(n))$
Step 6. Public key = {e, n}
Step 7. Private key = {d, n}

Encryption
Step 1. Plaintext: M<n
Step 2. Ciphertext: $C=M^e \bmod n$

Decryption
Step 1. Ciphertext: C
Step 2. Plaintext: $M=C^d \bmod n$

Each party must generate its keys to communicate safely with one another. First, the value of e in the RSA algorithm for encryption should be chosen so that gcd((n), e) equals 1. Once e has been chosen, the appropriate 'd' for decryption should be constructed by determining the inverse of 'e' mod n. During the encryption process, a sender must encrypt the message, i.e., in decimal digits, using the receiver's public key, i.e., e and n. The recipient must decrypt the ciphertext using his private key, represented by the letters d and n.

D. ECC

The ECC algorithm is public-key cryptography (PKC) with public and private keys for authentication. ECC is known as a sort of PKC built upon the algebraic structure of the elliptic curve over finite fields. The difficulty of the elliptic curve discrete logarithm problem (ECDLP) plays a major role in the security of ECC, and this problem can be resolved exponentially. Meanwhile, it has to be added that the performance of this algorithm is mainly intertwined with the efficiency of its scalar multiplication algorithm. Hamming weight of the private key is a determinant factor in algorithm efficacy regarding the scalar arithmetic level of the computation. Hamming weight measures the number of non-zero digits in a scalar representation. As the extent of Hamming's weight lowers, the speed of scalar multiplication performance rises. Accordingly, the scalar recoding method can be used to lessen the Hamming weight of the private key's scalar representation [22].

Because of its lower-key size and capacity to preserve security, ECC has gradually gained popularity over the last several years. Due to the increasing size of keys and the increasing desire for devices to remain secure, this trend will likely continue as mobile resources become more precious and the demand for devices to remain secure increases. To fully

comprehend elliptic curve cryptography, it is necessary to understand it in context. It also makes sense to use ECC to maintain high performance and security levels [22].

The ECC is becoming increasingly popular as businesses attempt to improve the online security of client data and the mobile optimization of their sites simultaneously. As the number of sites that use elliptic curve cryptography to secure data grows, the demand for brief guides to elliptic curve cryptography also grows. For the current ECC, an elliptic curve is a plane curve over a finite field composed of the points meeting the following equation: $y^2=x^3 + ax + b$. as shown in Fig. 10. It is possible to mirror any point on this elliptic curve cryptography example over the x-axis and yet have the curve retain its shape in this example. Any non-vertical line will intersect the curve three times or less if it is not vertical [22].

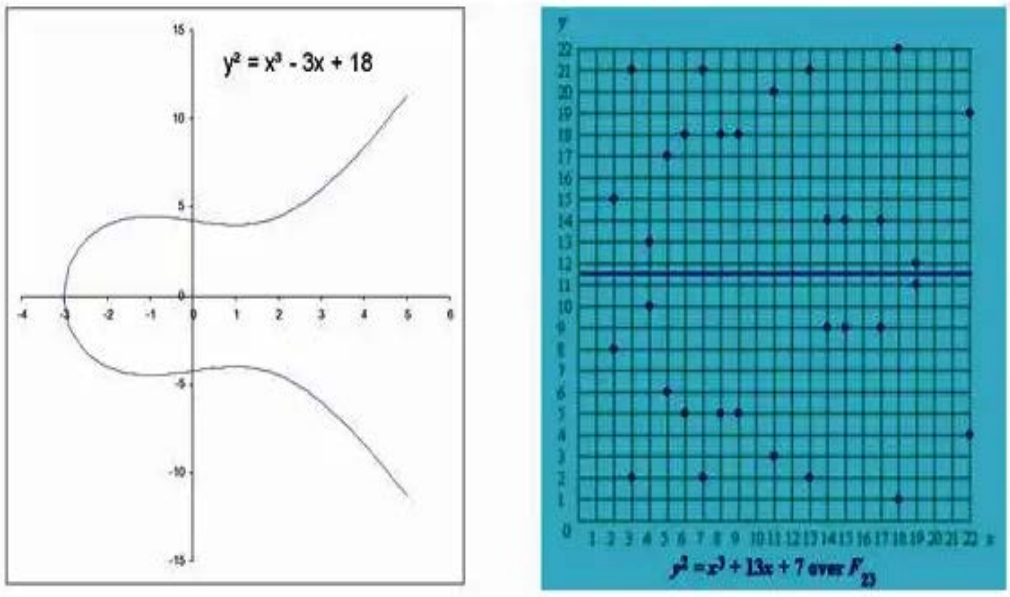

Fig. 10. 3rd-Degree Elliptic Curves [22]

Authors in [22] coined ECC as another potential asymmetric key cryptosystem in the late 1980s. This type of technology is best suited for devices with limited memory, such as Palmtops, Smartphones, and Smartcards. An ECC requires fewer or smaller parameters than RSA for encryption and decryption but with equal degrees of security.

ECC Algorithm

Global Public Elements
Step 1. $E_q(a, b)$ elliptic curve with parameters a, b, and q, where q is a prime or integer of from 2^m.
Step 2. G point on the elliptic curve whose order is large value n, where n is the mod.

User Alice Key Generation
Step 1. Select private key n_A; where $n_A < n$
Step 2. Calculate public key P_A
Step 3. $P_A = n_A G$

User Bob Key Generation
Step 1. Select private key n_B; where $n_B < n$
Step 2. Calculate public key P_B
Step 3. $P_B = n_B G$

Calculation of Secret Key by User Alice
Step 1. $K = n_A P_B$
Calculation of Secret Key by User Bob
Step I. $K = n_B P_A$

Encryption by Alice using Bob's Public Key
Step 1. Alice chooses the message P_m and a random positive integer k.
Step 2. Ciphertext: $C_m = \{ kG, P_m + kP_B \}$

Decryption by Bob using his own Private Key
Step 1. Ciphertext: C_m
Step 2. Plaintext: $P_m = P_m + kP_B - n_B (kG) = P_m + k(n_B G) - n_B (kG)$
P_m is a (x,y) point encoded with the plaintext message m in this case. Encryption and decoding take place at the P_m.

E. Performance Metrics

This section of the paper will determine the performance metrics that have been used to base our comparison of the RSA and ECC algorithms on. Performance metrics can take on various forms, but the focus is on just four types for this research.

1) Memory utilization

Memory is an integral component of the entire computer system and essentially consists of a system of devices that helps in data storage on electronic digital computers. Computer memory can either be temporary or permanent, although this depends largely on the frequency of data retrieval [23]

Memory utilization is calculated by storing the resident set size before the encryption or decryption functions. After running the encryption or decryption functions, the system time is stored in another variable. Now the difference between these two variables is memory utilization.

2) Signature generation time

When sending data, for instance, through a document, it is paramount to identify the authenticity of the senders, for optimal security and safety, for instance, against the distinct forms of cyber theft [24].

Signature generation time is calculated by storing the system time value in a variable, then running the generation function. After that, store the system time in another variable. The difference between these two variables is the signature generation time.

3) Signature verification time

The use of signature verification by algorithms means an effort to unearth the identity of the parties involved in sending and receiving messages and is integral in facilitating timely identification and aversion of potential threats that could negatively affect data security and integrity [24].

Signature verification time is calculated by storing the system time value in a variable, then running the verification function. After that, store the system time in another variable. The difference between these two variables is the signature verification time.

4) Encryption and decryption time

Encryption time is required to convert plaintext to ciphertext, while decryption time is required to convert ciphertext to plaintext [25].

Encryption and decryption time is calculated by storing the system time value in a variable before the encryption. Then, running the encryption or decryption function. After that, store the system time in another variable. Now, the difference between these two variables, encryption or decryption time

F. NIST Recommendation

The comparable key-size classes addressed in this section are based on estimations generated using currently available methodologies as of the publishing of this Recommendation. Future advancements in factoring algorithms, general discrete-logarithm assaults, elliptic-curve discrete logarithm attacks, and quantum computing may impact these equivalencies. In addition, new or improved attacks or technologies may emerge, rendering some of the current methods utterly insecure. For example, if quantum attacks become realistic, asymmetric approaches may no longer be secure. Periodic reviews will be conducted to see if the stated equivalencies need to be altered. For example, key sizes need to be increased or if the algorithms are no longer secure. Other than brute-force cryptographic attacks, strong cryptographic algorithms may be able to mitigate security vulnerabilities. For example, the algorithms may be built, so those small quantities of information about the key are unintentionally leaked. In this situation, the larger key may lower the chances of a compromised key due to the disclosed information [26]. Table II shows equivalent maximum-security strengths for the accepted algorithms and key lengths.

TABLE II. NIST RECOMMENDED SECURITY BIT LEVEL (BARKER, 2020)

Security Bit Level	RSA	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Security bit level is a cryptographic primitive's security level measures its strength, such as a cipher or hash function. The security level is commonly stated in bits, with n-bit security implying that breaking it would take 2n operations [26].

IV. RESEARCH RESULTS AND DISCUSSIONS

This chapter contains the analysis comparison parts of RSA and ECC algorithms based on the security bit-level suggested by NIST. Memory utilization is in bytes, signature generation time, signature verification, encryption, and decryption time are in milliseconds.

A. Memory Utilization

Based on Fig. 11 and Table. III, ECC shows better than RSA in memory utilization at all security bit levels; ECC needs less memory usage than RSA [20]. A massive spike after 192-bit level in RSA was observed. That makes RSA worst in memory handling, especially in the large keys.

TABLE III. MEMORY UTILIZATION COMPARISON BETWEEN RSA AND ECC

Security Bit Level	Memory Utilization in bytes	
	RSA	ECC
80	160	109
112	239	119
128	315	127
192	620	144
256	1777	220

B. Signature Generation Time

Fig. 12 and Table. IV show that ECC and RSA are close to each other in the signature generation time. RSA is better at 80, 112, and 128 security bit levels. In the 192-security bit-level, a small RSA latency compared to ECC was observed. Also, great latency in RSA at the 256-security level was noticed. RSA needs 3 ECC times to generate the signature [27].

TABLE IV. SIGNATURE GENERATION TIME COMPARISON BETWEEN RSA AND ECC

Security Bit Level	Signature Generation Time in Milliseconds	
	RSA	ECC
80	0.0102	0.1530
112	0.1533	0.3411
128	0.2119	0.5912
192	1.5322	1.1897
256	9.2152	3.087

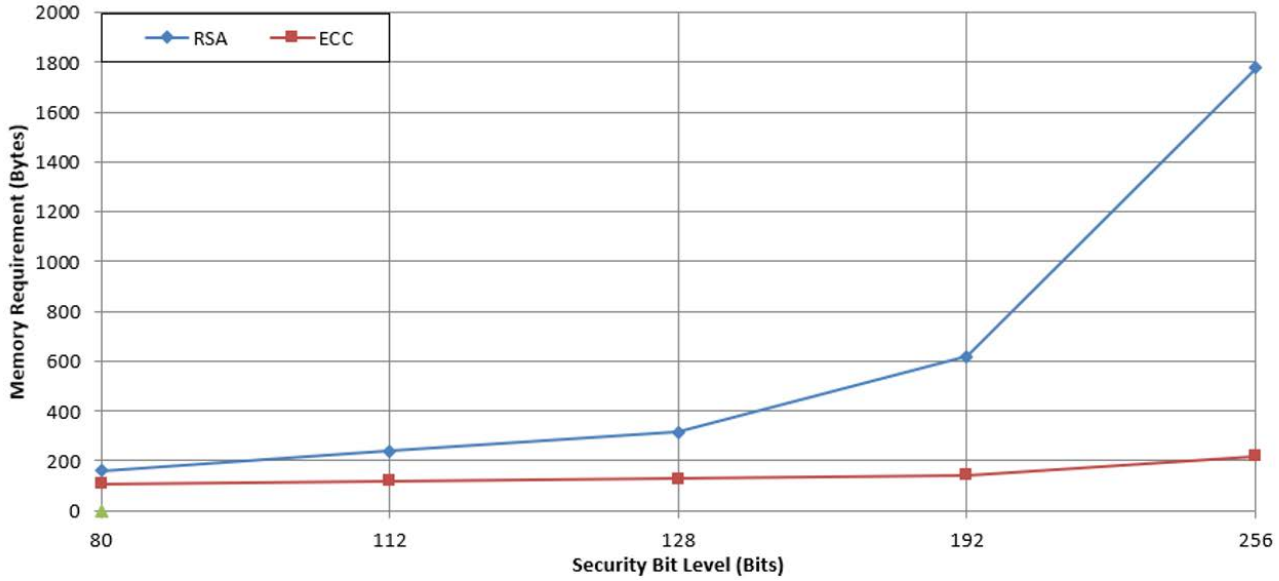

Fig. 11. Memory Utilization Comparison Graph between RSA and ECC

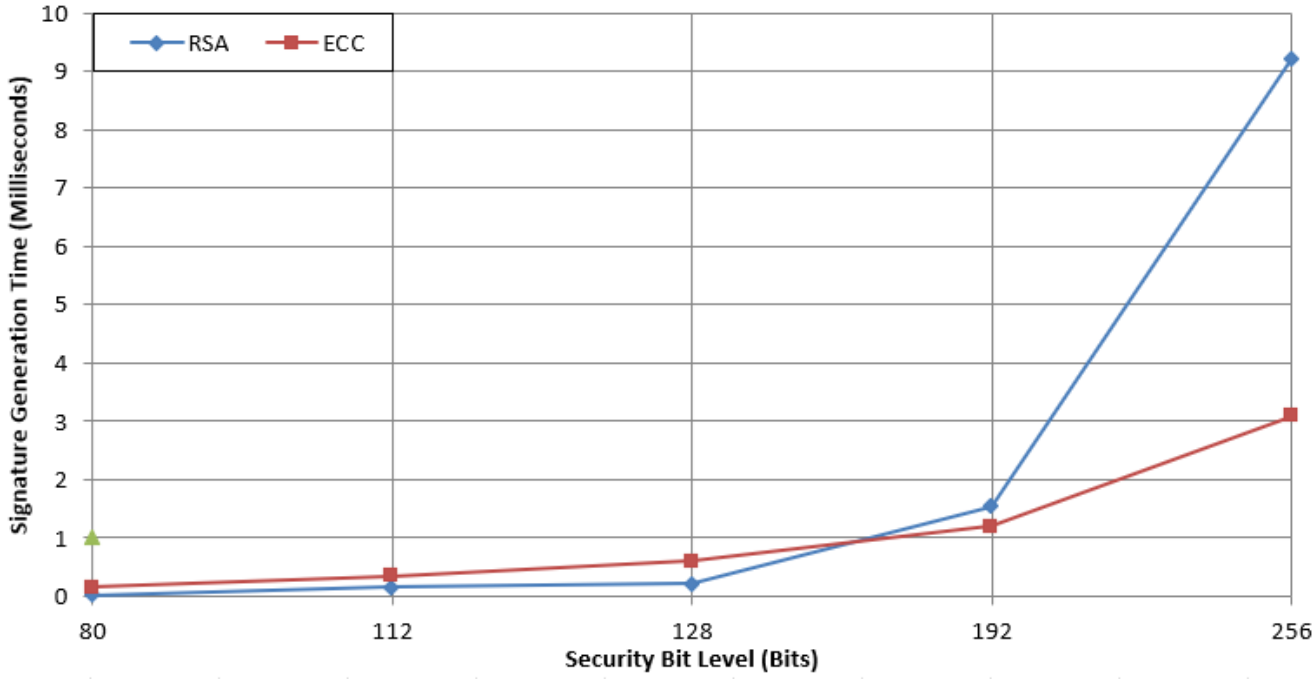

Fig. 12. Signature Generation Time Comparison Graph between RSA and ECC

C. Signature Verification Time

As shown in Fig. 13 and Table. V, RSA trumps the ECC in all security bit levels regarding signature verification. In RSA, the time required to verify a signed message is trivial for the key length employed. However, ECC is significantly slower to perform in each key range and exhibits an almost linear rise in performance with increasing the security bit level [20]. The reason because in RSA, the cost of verification can be controlled to be minimal.

TABLE V. SIGNATURE VERIFICATION COMPARISON BETWEEN RSA AND ECC

Security Bit Level	Signature Verification Time in a millisecond	
	RSA	ECC
80	0.0110	0.2310
112	0.0116	0.5231
128	0.0124	0.8622
192	0.0130	1.8100
256	0.0310	4.5410

D. Encryption and Decryption Time

Fig. 14 and Table VI show that RSA is very fast compared to ECC. in all security bit levels. Even with 256 bits, RSA needs around 1.03 seconds for encryption [28].

TABLE VI. ENCRYPTION TIME COMPARISON BETWEEN RSA AND ECC

Security Bit Level	Encryption Time in seconds	
	RSA	ECC
80	0.0306	0.4886
112	0.0310	2.2030
128	0.0360	3.8763
192	0.0489	5.2113
256	1.0310	8.5441

Fig. 15 and Table VII show a noticeable massive RSA change when the security bit level increases. Only on 80 bit of security level than RSA is better in decryption time. When the security bit level increments, a high time increment in RSA was observed. After 80 security level bit, ECC becomes better than RSA.

TABLE VII. DECRYPTION TIME COMPARISON BETWEEN RSA AND ECC

Security Bit Level	Decryption Time in seconds	
	RSA	ECC
80	0.7634	1.3376
112	2.7165	1.6012
128	7.1022	1.7770
192	14.002	2.0031
256	22.120	4.1194

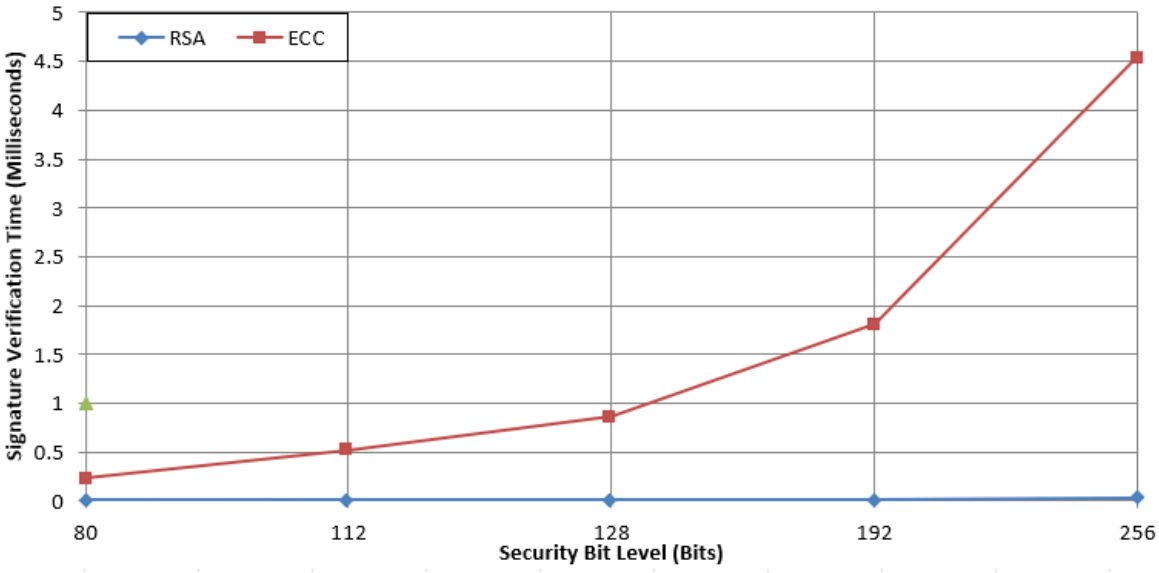

Fig. 13. Signature Verification Comparison Graph between RSA and ECC

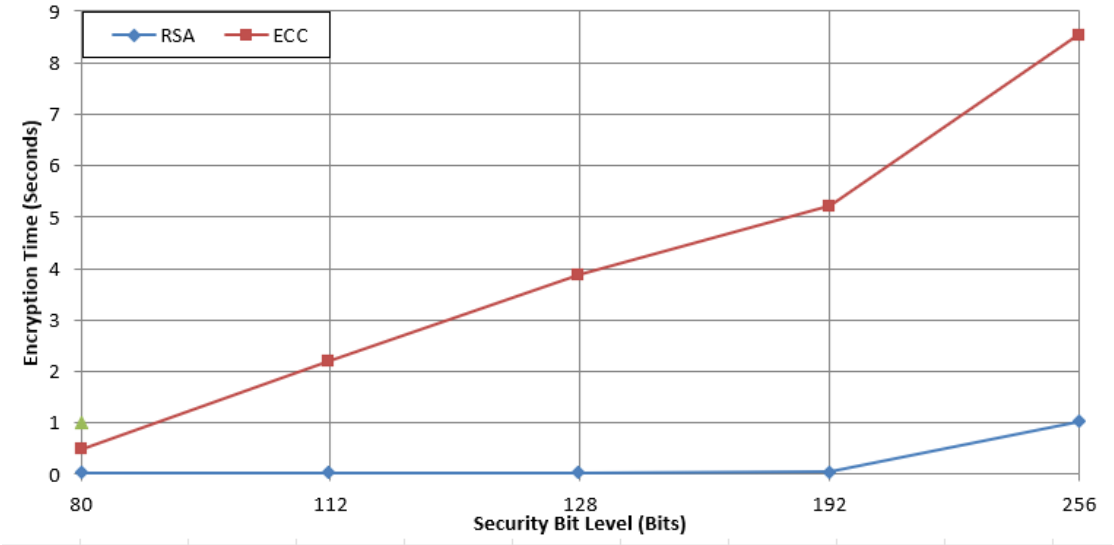

Fig. 14. Encryption Time Comparison Graph between RSA and ECC

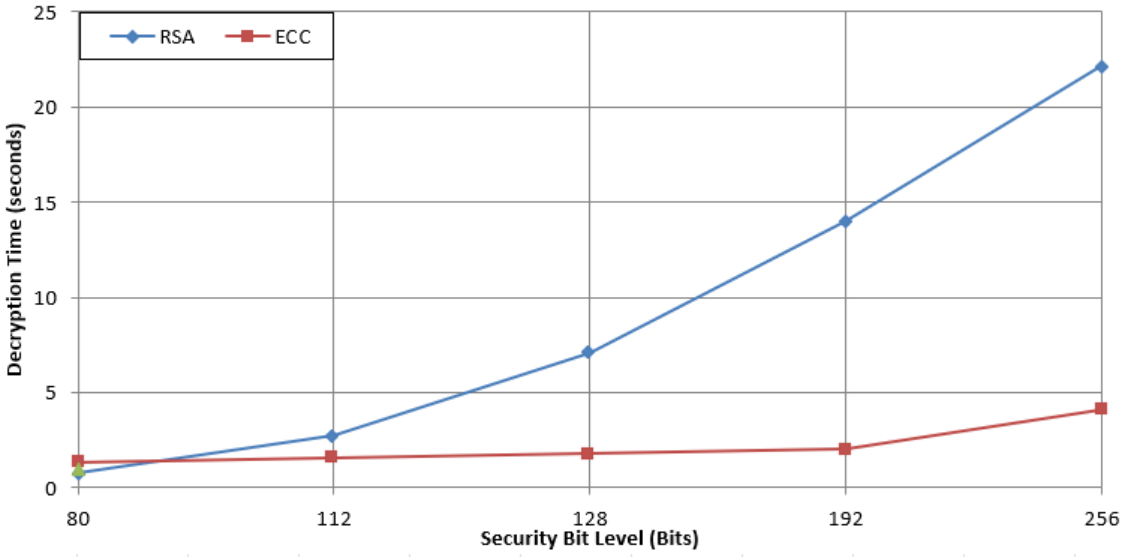

Fig. 15. Decryption Time Comparison Graph between RSA and ECC

V. Conclusion and Future Work

The majority of organizations and individuals rely on asymmetric encryption algorithms despite their complexity since they are safe and tough to break. Both RSA and ECC are instances of powerful asymmetric algorithms. This research analyzed the similarities and differences in both algorithms. Memory utilization, signature generation time, signature verification time, encryption time, and decryption time were used as performance metrics. The findings of this research show that ECC is more successful in memory use across all of the security bit-levels recommended by NIST. In addition, regarding the time required to generate signatures, RSA is more efficient than ECC when the security level is 80 or 112. On the other hand, when there is a rise in the security bit level, ECC becomes faster than RSA. When it comes to signature verification time, RSA is outstandingly fast, but ECC takes more than ten times as long as RSA, at the very least were used as a performance metric. RSA maintains its encryption time speed even when the security bit-level increases in encryption time. However, regarding decryption time, RSA becomes faster only when the security bill level increases by more than 80 security level bits. Although this experiment is done in a dedicated physical server, that service provider has limitations. One of them is in displaying the server's power consumption. That limits us from calculating the power consumption comparison between RSA and ECC in the five-bit security levels. Since electricity has become a primary factor in operational costs, adding a power-consuming as a new comparison parameter between RSA and ECC is the potential for future work.

References

[1] Petrov, I. Beliak, A. Kryuchyn, and A. Shikhovets, "Analysis of methods for creating media for long-term data storage," in 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT), 2020, pp. 238-241.

[2] Tabrizchi and M. Kuchaki Rafsanjani, "A survey on security challenges in cloud computing: issues, threats, and solutions," The journal of supercomputing, vol. 76, pp. 9493-9532, 2020.

[3] S. Akter, S. Chellappan, T. Chakraborty, T. A. Khan, A. Rahman, and A. A. Al Islam, "Man-in-the-middle attack on contactless payment over NFC communications: design, implementation, experiments and detection," IEEE Transactions on Dependable and Secure Computing, vol. 18, pp. 3012-3023, 2020.

[4] O. Salem, K. Alsubhi, A. Shaafi, M. Gheryani, A. Mehaoua, and R. Boutaba, "Man-in-the-Middle attack mitigation in internet of medical things," IEEE Transactions on Industrial Informatics, vol. 18, pp. 2053-2062, 2021.

[5] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: Threats and potential solutions," Computer networks, vol. 169, p. 107094, 2020.

[6] T. Hidayat and R. Mahardiko, "A Systematic literature review method on aes algorithm for data sharing encryption on cloud computing," International Journal of Artificial Intelligence Research, vol. 4, pp. 49-57, 2020.

[7] P. Brandão, "The importance of authentication and encryption in cloud computing framework security," International Journal on Data Science and Technology, vol. 4, pp. 1-5, 2018.

[8] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," Journal of Big data, vol. 6, pp. 1-21, 2019.

[9] S. Greenberg, L. P. Mason, S. O. Sadjadi, and D. A. Reynolds, "Two decades of speaker recognition evaluation at the national institute of standards and technology," Computer Speech & Language, vol. 60, p. 101032, 2020.

[10] M. Knežević, S. Tomović, and M. J. Mihaljević, "Man-in-the-middle attack against certain authentication protocols revisited: Insights into the approach and performances re-evaluation," Electronics, vol. 9, p. 1296, 2020.

[11] B. Bhushan, G. Sahoo, and A. K. Rai, "Man-in-the-middle attack in wireless and computer networking—A review," in 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall), 2017, pp. 1-6.

[12] T. Belkhouja, A. Mohamed, A. K. Al-Ali, X. Du, and M. Guizani, "Light-weight solution to defend implantable medical devices against man-in-the-middle attack," in 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1-5.

[13] S. O. Maikol, A. S. Khan, Y. Javed, A. L. A. Bunsu, C. Petrus, H. George, et al., "A novel authentication and key agreement scheme for countering MITM and impersonation attack in medical facilities," International Journal of Integrated Engineering, vol. 13, pp. 127-135, 2021.

[14] Kore and S. Patil, "IC-MADS: IoT enabled cross layer man-in-middle attack detection system for smart healthcare application," Wireless Personal Communications, vol. 113, pp. 727-746, 2020.

[15] J. J. Hathaliya, S. Tanwar, S. Tyagi, and N. Kumar, "Securing electronics healthcare records in healthcare 4.0: A biometric-based approach," Computers & Electrical Engineering, vol. 76, pp. 398-410, 2019.

[16] K. Abouelmehdi, A. Beni-Hessane, and H. Khaloufi, "Big healthcare data: preserving security and privacy," Journal of big data, vol. 5, pp. 1-18, 2018.

[17] Z. Cekerevac, Z. Dvorak, L. Prigoda, and P. Cekerevac, "Internet of things and the man-in-the-middle attacks–security and economic risks," MEST Journal, vol. 5, pp. 15-25, 2017.

[18] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," IEEE Communications Surveys & Tutorials, vol. 22, pp. 1909-1941, 2020.

[19] M. B. Yassein, S. Aljawarneh, E. Qawasmeh, W. Mardini, and Y. Khamayseh, "Comprehensive study of symmetric key and asymmetric key encryption algorithms," in 2017 international conference on engineering and technology (ICET), 2017, pp. 1-7.

[20] Z. Vahdati, S. Yasin, A. Ghasempour, and M. Salehi, "Comparison of ECC and RSA algorithms in IoT devices," Journal of Theoretical and Applied Information Technology, vol. 97, 2019.

[21] G. Amalarethinam and H. Leena, "Enhanced RSA algorithm with varying key sizes for data security in cloud," in 2017 World Congress on Computing and Communication Technologies (WCCCT), 2017, pp. 172-175.

[22] M. A. Khan, M. T. Quasim, N. S. Alghamdi, and M. Y. Khan, "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data," IEEE Access, vol. 8, pp. 52018-52027, 2020.

[23] N. Markham and G. Pereira, "Experimenting with algorithms and memory-making: Lived experience and future-oriented ethics in critical data science," Frontiers in big Data, vol. 2, p. 35, 2019.

[24] S. Abd Elminaam, H. M. Abdual-Kader, and M. M. Hadhoud, "Evaluating The Performance of Symmetric Encryption Algorithms," International Journal of Network Security,, vol. 10, pp. 213-319, 2010.

[25] O. P. Verma, R. Agarwal, D. Dafouti, and S. Tyagi, "Notice of Violation of IEEE Publication Principles: Peformance analysis of data encryption algorithms," in 2011 3rd International Conference on Electronics Computer Technology, 2011, pp. 399-403.

[26] Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for key management part 1: General (revision 5)," NIST special publication, pp. 800-57, 2020.

[27] Pharkkavi and D. Maruthanayagam, "TIME COMPLEXITY ANALYSIS OF RSA AND ECC BASED SECURITY ALGORITHMS IN CLOUD DATA," International Journal of Advanced Research in Computer Science, vol. 9, 2018.

[28] Mallouli, A. Hellal, N. S. Saeed, and F. A. Alzahrani, "A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms," in 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2019, pp. 173-176.