Global Scientific JOURNALS

GSJ: Volume 12, Issue 5, May 2024, Online: ISSN 2320-9186

www.globalscientificjournal.com

SECURITY DESIGN CRITERIA FOR INTERNET OF THINGS (IOT) DEVICES

ATALOR Osaroboh Daniel, OKOSODO Richard Odiase, ODUBULU Kingsley, OBAJAJA Courage

KeyWords

Security, Cryptography, Internet of Things, Microsoft Message Queue, Network devices.

Abstract

The prominent challenge faced by enterprises and investors in the parlance of IoT is the issue of security of the data in transit of the IoT device. Curated data from IoT right from the sensor source of data to its destination have been known to face serious vulnerability that poses challenges to IoT manufacturers. Unwell-secured IoT devices and applications mark IoT as a potential goal of cyberattacks. Manufacturers and vendors of IoT strive yearly and invest capital to market IoT products to users by providing a framework and domain that builds the confidence of the users of the system that the data exchange between IoT devices is secure and safe. Due to people's dependency on the internet for social and online transactions, the phenomenon of IoT security can never be overemphasized. This project aims to find an improved security algorithm mechanism to fit modern-day IoT data protection.

The methodology adopted in this project is an improved Cryptography algorithm where a dynamic secret key is shared among sending and receiving IoT devices. The data on transit in the form of ciphertext is encoded and decoded between devices using a shared secret key that can be altered dynamically. A prototype in the form of a device was designed to simulate live IoT devices communicating with each other in this research project. To solve the problem of connection and disconnection of the devices in the form of short-circuiting, the publish and subscribe messaging scheme of Microsoft Messaging Queue (MSMQ) was utilized. In this case, a sending device can continue to send encrypted messages to a listening device even if a listening device is not connected yet. C# as a sophisticated programming language was used to implement the end-to-end system algorithm design using Microsoft Visual Studio Integrated development environment.

The result of the finding was that the simulated devices were able to send a secure data sample in the form of plaintext to each other even in a connected and disconnected state in a safe mode. This research project is recommended to be used to secure ATM Card transaction data in a financial institution and sensitive sensor data that transmits its data via a public cloud computing environment for data analytics and machine learning sciences.

INTRODUCTION

1.1 Background to the Study

IoT is a trending topic in information technology. It has the potential to alter the planet even over the web has done. Connecting electronic objects into the digital network has been ongoing for a very long term. This success is due in part to radio frequency identification and wireless sensor network technologies. The Internet of Things (IoT) combines several devices with numerous platforms, computing capabilities, and functions. The heterogeneity of the network and the ubiquity of IoT devices place increased demands on security and privacy protection (Ramaswamy, 2015).

IoT provides a network infrastructure with interoperable communication protocols and software tools to enable connectivity to the internet for handheld smart devices (Liyanage, and Chen, 2018) such as smartphones, personal digital assistants (PDA) and tabs), good home equipment (smart TV, AC, intelligent lighting systems, good refrigerator, etc.), vehicles, and sensory acquisition systems). However, the improved connectivity and accessibility of devices present major concerns for the security of all the parties connected to the network regardless of whether they are humans or machines.

The Internet of Things, like any fast-evolving technology, has several security issues. The greater the number of devices linked to the network, the greater the risk of attackers gaining and exploiting others (Aljarah, et al, 2020). As IoT systems are increasingly entrusted with sensing and managing highly complex ecosystems, concerns about the security and reliability of data transmitted to and from IoT devices are quickly becoming a major concern (Alaba, & Othman, 2017). As the number of linked items grows, so will the number of breaches the system is exposed to. These IoT platforms include sensitive or proprietary content that could be leaked (Temirbekova and Pyrkova, 2020). Also, the growth of connected devices will continue to increase in hundreds of millions. This shows that society will increasingly rely on such devices in business and other personal-related dealings. The Internet has become second nature to billions of people already because it allows enormous gains in productivity, efficiency, and communication; however, security loopholes inherent in it can endanger business and personal lives. Creating a channel whereby hackers or third parties cannot have access to personal messages (Shankar, and Jenifer, 2017) sent out through IoT devices.

A heterogeneous system such as IoT creates a deployment problem. This heterogeneity poses many security challenges. On the other hand, the extent systems are vulnerable to infiltration from third parties has also increased. The complexity of cloud IoT architecture design, security architecture, and their unique deployment models make security a challenge for computing systems. Attacks on IoT can be physical internetwork and encryption programs. The effect of these attacks can be seen in the 2016 devastating attacks on the network and finance of companies by DNS. The attack has shown that more research is needed in IoT security systems (Lakshmi, and Srikanh, 2018).

IoT devices can be secured with encryption but still suffer from a brute-force attack. A wrongdoer or agency that doesn't grasp how it works might create measures abundant and computer program to hack into the system. This is why encoding needs to be extraordinarily robust and complicated. Another problem with attempting to secure a device is that making a device most important doesn’t necessarily mean it is the best approach or best or most practical approach. Increasing the key size or iterating over the algorithm could make algorithms much safer, but these ultra-secure algorithms would be extremely sluggish and resource-intensive, making them practically unworkable. Algorithms that are recommended aim to reach the optimal balance between security and usefulness.

In the encryption world, some elements are becoming outdated. 3DES is still in use but it looks might be retired shortly. Utilizing science New Year's Ev algorithm, going secret writing with different government agency steerage results in 3DES. Nowadays, many researchers have proposed many encryption and decryption algorithms: Rivest Shamir Adleman, etc. However, the major algorithms proposed had drawbacks such as unreliability and requiring considerable amounts of time to increase the delay of packets for the communication channel, which impedes the communication process. The study therefore aims to develop an encryption program that secures messages sent through IoT devices from a sender to a receiver.

1.2 Statement of the Problem

The proliferation of Internet of Things (IoT) devices has fundamentally transformed connectivity and automation across various sectors. While these advancements offer considerable benefits in terms of efficiency and convenience, they also introduce significant security vulnerabilities. The primary concern is that the increased number and heterogeneity of devices on a network amplify the potential entry points for attackers, thereby heightening the risk of unauthorized access and data breaches. Current encryption methods, although integral for security, face challenges in balancing robustness with performance. Traditional algorithms like 3DES are becoming outdated and may soon be inadequate against sophisticated cyber threats, suggesting a need for more advanced solutions. Additionally, the diverse nature of IoT devices complicates the deployment of uniform security measures, making comprehensive protection strategies difficult to implement. This study identifies an urgent need to develop an encryption program tailored for IoT communications that not only enhances security but also addresses the limitations of existing technologies in terms of speed and resource consumption. The goal is to ensure that sensitive data transmitted between IoT devices remains inaccessible to unauthorized parties, thereby safeguarding both personal and business information from potential cyber threats. This problem statement underscores the importance of advancing IoT security to keep pace with the rapid growth of connected devices and the evolving landscape of cyber threats.

1.3 Objectives of this Study

To Build an encryption algorithm program that will prevent IoT devices from being vulnerable to attackers desperate to tamper with the integrity and authenticity of data on transit across the network.

1.4 Expected Contribution to Knowledge

This study is expected to contribute significantly to the field of IoT security by developing an advanced encryption program specifically designed for IoT communications. The proposed solution aims to address the critical balance between high-level security and operational efficiency, which is a major gap in existing cryptographic methods. By focusing on the optimization of encryption algorithms to enhance security without compromising performance, this research will provide a viable security framework that can be implemented across various IoT platforms and devices. Furthermore, the study will contribute to academic and practical understandings by evaluating the performance of the new encryption method in real-world IoT scenarios. This includes assessments of algorithmic complexity, resource consumption, and resistance to common and emerging cyber threats such as brute-force attacks. The outcomes of this research are expected to offer substantial improvements over traditional encryption techniques, providing a robust defense mechanism that can adapt to the evolving threats in the IoT landscape. These contributions will not only advance the theoretical framework of IoT security but also provide practical guidelines for developers and policymakers in enhancing the resilience of IoT ecosystems against cyber intrusions.

LITERATURE REVIEW

2.1 Concept of IoT Devices

The Internet of Things (IoT) allows devices to communicate with one another. It is extensively used in industrial production and social applications, as well as a nice home, healthcare, and industrial automation. While providing unparalleled convenience, accessibility, and efficiency, IoT has recently posed serious security and privacy risks (Zhou, Jia, Peng, Zhang, & Liu, 2019). The Internet of Things (IoT) refers to a new generation of technological gadgets. In the most basic sense, they are ordinary items that connect and interact over the internet. There are more things linked to the internet than humans in the world today (Evans, 2011). The term "Internet of Things" first appeared in a highly technical study by ITU (2005). IoT devices are commonly referred to as "smart gadgets," and they have a wide range of purposes and functionality (Leo & Battisti et al., 2014).

The Internet of Things is defined in a variety of ways, but it generally refers to a global network of networked devices that use established communication standards (Bassi, and Horn, 2008). The idea of free information exchange via wireless communication between various embedded computing devices connected to the internet, using the internet as a communication medium, is defined as "the idea of free information exchange via wireless communication between various embedded computing devices connected to the internet, using the internet as a communication medium." According to Chen (2014)'s essay on IoT can devices collect and analyze data? Many network security solutions exist, but they must be adapted to the Internet of Things (IoT) and the devices that comprise it (Siegel, 2017). It entails the same level of security preparation for traditional systems as it does for Internet of Things (IoT) applications (Catherine, & Dabbura, 2018). Trust management, authorization, authentication, identity, access control, network security, standardization, and interoperability are only a few of the different security characteristics. The relevance of the issue becomes clear with the personal data exchanged in periods of increased attack vectors and attack routes.

2.2 Vulnerabilities in IoT Security

The possibility or danger that a device is exposed to harm that would result, and the time and resources required to accomplish a degree of protection are all factors in determining a device's security (Mulani, and Pingle, 2016). Organizations should focus on and raise awareness of information security concerns. Security challenges in IoT devices stem from various factors. Firstly, the extensive range of devices and operations in IoT designs introduces complexities in providing adequate security measures (Roman et al., 2013). The inherently interconnected nature of IoT, allowing ubiquitous sensors to process independently, increases vulnerability to cyber-attacks (Leo, 2014). Moreover, the interconnectedness of IoT devices facilitates the rapid spread of attacks globally, surpassing the reach of the internet (Atzori, 2010; Rose, 2015).

Resource limitations pose a significant hurdle, with IoT devices typically having constrained resources while necessitating robust security features such as cryptographic primitives and security protocols (Skarmeta & Moreno, 2013). The constrained user interfaces of many IoT devices, lacking keyboards or screen displays, make passwords a weak link in security, especially when default or weak passwords are prevalent (Desai, 2016). Data integrity emerges as a critical concern, encompassing issues of authentication, access control, and secure connectivity in IoT (Abie, 2012). Trusting transmitted data and identifying the entity sending it becomes pivotal questions in ensuring data integrity. While encryption is a common tool for securing wireless communication, its implementation in IoT requires more effective and energy-saving algorithms, coupled with efficient key distribution systems (Whitmore et al., 2015; Bandyopadhyay et al., 2011). Addressing these challenges is imperative for enhancing the overall security posture of IoT ecosystems.

2.3 Encryption

Encryption techniques convert a plaintext message (or keep data) into ciphertext in such a way that the ciphertext exposes little or no information about the original plaintext (Kirst, 2015). Coding schemes are made up of three parts: a key generation rule, an associate coding rule, and a coding rule. The secret writing rule accepts plaintext and secret writing keys and returns the needed text in cipher (Metcalf, 2016). Today, coding protects people's and organizations' communications from naive and complicated criminals as well as undemocratic nations. It ensures the security of electronic commerce transactions over the internet, for example, by enabling the transmission of MasterCard numbers. It safeguards data stored on cell phones, computers, and other devices. Encrypted communication capabilities are built into key computer platforms in a variety of electronic messaging apps used by many people (Salama, 2017).

In the realm of cryptography, various applications serve crucial purposes for users across computer software, apps, and hardware. A primary application involves safeguarding stored files through parallel cryptography, where the key may be user-entered, derived from a password, or protected by an asymmetric cryptography system. Another vital encryption application is the encryption of the whole disk, preventing exposure of both user data

and system applications. This method requires the validation of digital signatures on key components, user authentication, and subsequent decoding processes. Device security, particularly for mobile devices, is ensured through cryptographic passcode security and full disk encryption, with the unlocking key being a combination of the user password and hardware key. Virtual Private Networks (VPNs) employ symmetric cryptography to establish encrypted connections between remote users and organizational networks, facilitating secure remote access. Additionally, encrypted messaging applications, such as those utilizing the Signal protocol, employ end-to-end encryption to protect communications from third-party access. Lastly, confidentiality protection in cloud or third-party computing involves encoding data to ensure secrecy, although access to encryption keys may vary based on service design and provider policies. These diverse applications highlight the multifaceted role of encryption in addressing various security needs across different technological contexts.

2.4 Review and Gaps in Past Studies

Previous studies on IoT security have primarily focused on the application of traditional encryption technologies like RSA and AES to IoT scenarios. However, these studies often overlook the unique challenges posed by the vast scale and heterogeneity of IoT devices. While these encryption methods offer proven security, their resource-intensive nature can hinder performance and feasibility in the diverse and resource-constrained environment of IoT. Additionally, much of the existing literature has concentrated on theoretical aspects of security without fully addressing practical deployment issues, such as the integration of secure algorithms across different IoT platforms and devices.

There is also a notable gap in the research regarding the adaptation of encryption technologies to the specific demands of IoT communication, where efficiency and low latency are critical. The need for real-time processing in IoT applications makes many traditional encryption methods impractical. Furthermore, studies often fail to consider the emerging threats that specifically target IoT devices, such as side channel attacks and advanced persistent threats, leaving a knowledge gap in defensive strategies tailored to these new forms of cyberattacks.

This study aims to bridge these gaps by developing a tailored encryption solution that meets the specific security and operational demands of IoT systems, thus contributing both theoretical insights and practical applications to the field.

3.0 METHODOLOGY

Research Design

The study uses asymmetric encryption design as an approach to solving the problem of password protection sent through the internet. Symmetric encryption of data employs a key, the secret code or phrase that you've chosen which may be an integer, a word, or a random string of characters. The conventional text of a message is combined with other text to change the content in a specific way. A public and private key will be generated when using asymmetric encryption.

3.1 Algorithm

The study made use of cryptography encryption algorithm.
The art of safe communication is the science of cryptography. The goal is to communicate the information securely, without a third party getting the message. This is a significant concern because it opens options for improving the security of devices. The entire linked gadget serves as an endpoint that is vulnerable to hackers. Protecting what you have even if you don't have millions to lose is critically important. It is widely believed that every device user should know how to use encryption. Secret key (symmetric): Both encryption and decryption are done using the same key. Asymmetric (public key): It uses two separate keys for encryption and decryption, one of which is a public key. The Steps employed are:

- First an exchanging of public keys between entities (sender and receiver).

- Sensitive documents are encrypted and sent with the sender's public key.
- Receivers decrypt the document using its private key to unlock the document.
- The strength and security of asymmetric encryption rely on the sender and receiver to keep their private key well-protected.

3.2 Source of Data on Transition

Data in the form of messages will be moved in real-time from communicating device A to listening device B to interpret and communicate. To achieve the objectives of this study, we introduce encryption.

3.3 Repository Assist of Message on Transit

Microsoft Message Queue will be used as a repository should a listening device not be available to access the message in transit in real-time should there be no internet connection at the time of sending the message.
Message Queuing (MSMQ) technology permits applications running at completely different times to speak across heterogeneous networks and systems that will be briefly offline. The applications will send messages to the queues and might conjointly browse messages from queues. Below is an associated illustration showing how a queue will hold messages that are generated by many sending applications and read by numerous receiving applications.

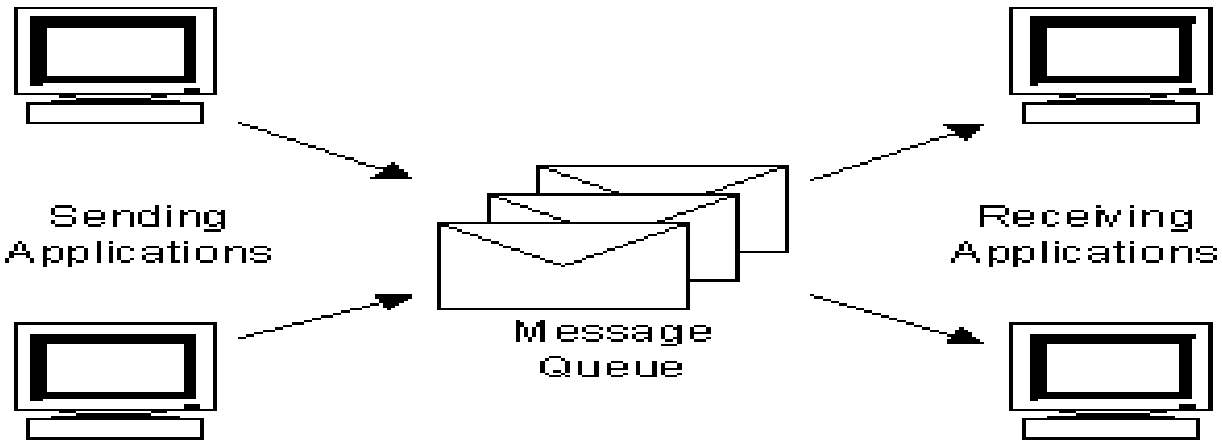

Figure 3.1 The Message Queue (**Anush, 2017**)

3.4 Pseudocode

In the encryption and decryption process, a cipher object is employed with a specific cryptographic algorithm, such as rhombohedral key encryption, in this study. The chosen encryption type, rhombohedral, utilizes asymmetric algorithms like Advanced Encryption Standard (AES), known for its security. AES offers three block ciphers AES-128, AES-192, and AES-256 with key lengths of 128, 192, and 256 bits, respectively, for encrypting and decrypting messages. Symmetric ciphers, also known as secret key ciphers, require both sender and receiver to use the same keys for encryption and decryption. The government classifies information into Confidential, Secret, and Top-Secret categories and various key lengths can be employed for different security levels. Top Secret information necessitates 192 or 256-bit key sizes, with 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys in the AES process, involving steps like substitution, transposition, and combination to transform plaintext into ciphertext.

AES Design

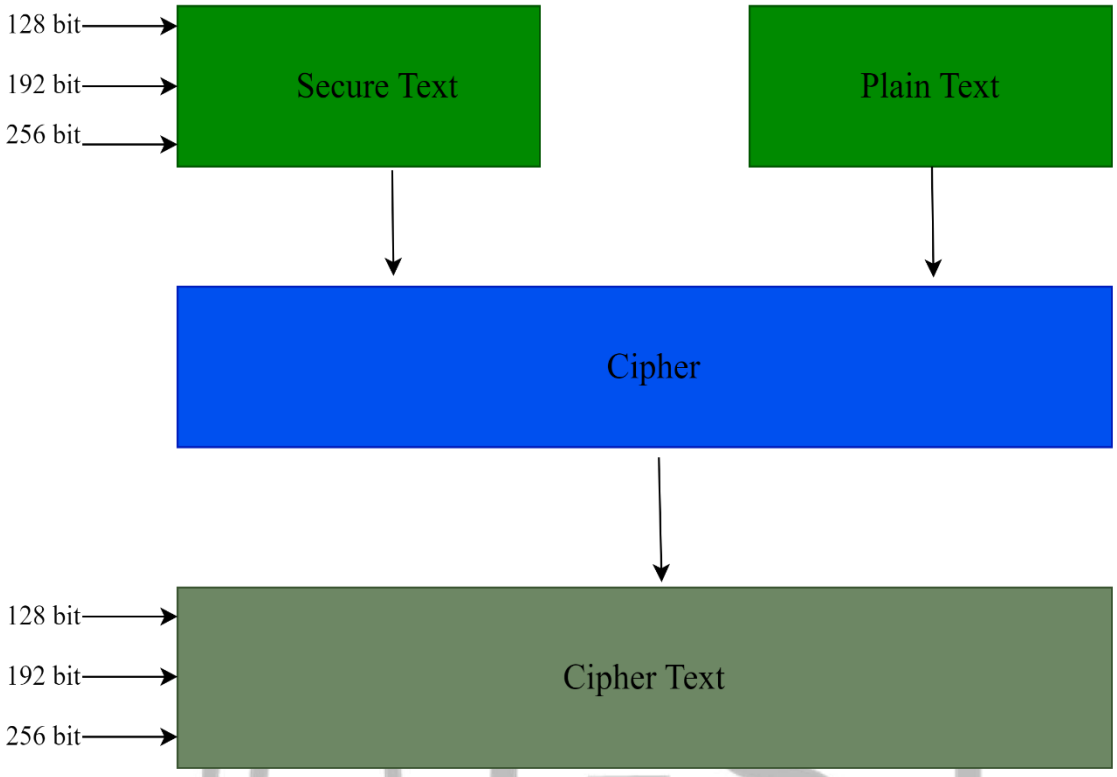

Figure 3.2, AES design operation (Researcher, 2024)

3.5 Encryption Operation

The diagram below displays the connection between the Secret key, Cipher, Plaintext, and Cipher Text, all of which are elements of the AES encryption method. The initial stage in the encryption process is to place the data into an array. Then, the encryption transformations are performed several times throughout many rounds of encryption. In the first step of the AES encryption cipher, the substitution of data using a substitution table is carried out; the second step is the shifting of data rows, and the third step is the mixing of columns. Every column utilizes a different piece of the cryptographic key for the last change. Longer keys would like additional rounds to finish.

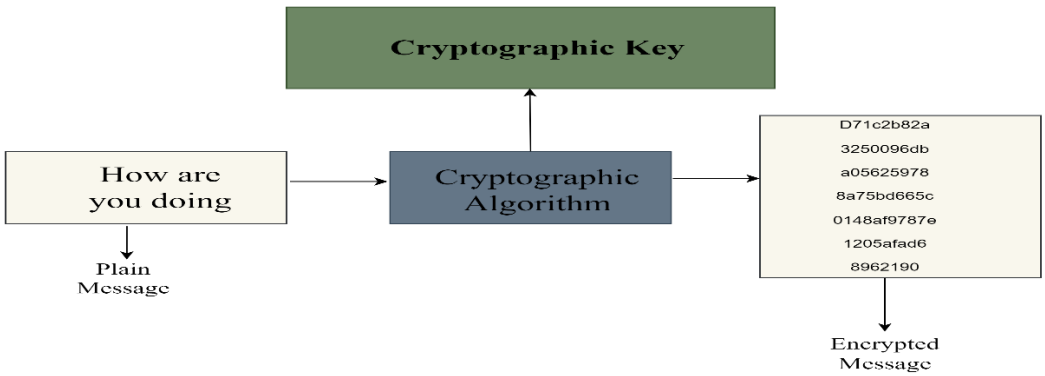

Figure 3.3, Encryption operation (Researcher, 2024**)**

3.6 Development Tools

The research employed Microsoft Visual Studio IDE for web development, leveraging its comprehensive features such as IntelliSense, code completion, and support for both native and managed code. The Microsoft DotNet framework, including components like Common Language Runtime, was utilized to create applications for Windows, Web, and mobile platforms, with a focus on the C# programming language. The study justified the choice of C# for its operational speed, object-oriented nature, cross-platform capability, automatic garbage collection, and strong support from Microsoft. Additionally, a secure messaging system module, developed as a command-line interface using a C# console application, played a crucial role, offering a simple user interface and facilitating automatic testing, thereby reducing automation implementation resources.

3.7 Design Technique

Various encryption technologies differ in key size and strength. Triple DES, with three 56-bit keys (168-bit total), is reliable for hardware encryption, especially in finance. RSA, a public-key algorithm, uses key pairs for encryption and decryption, making it resource-intensive to break. Blowfish, known for speed and effectiveness, remains unbeaten and is widely used as an open-source algorithm. Twofish, the sequel to Blowfish, supports up to 256-bit keys and finds applications in diverse environments. AES, the U.S. Government's choice, is cost-effective in 128-bit mode and resistant to decryption attempts. Despite this, AES is expected to become the standard for private-sector data encryption (Salama, 2017).

4.0 FINDINGS

The result analyses encrypted and decrypted data on transit between simulated IoT devices. Details of secret key encryption cryptography to achieve transmitted data confidentiality and integrity were demonstrated to ensure that encrypted data originating from a simulated IoT device arrived at the destination IoT device to achieve the aims and objectives of this research work.

4.1 Result of the IoT Messaging Protocol

Messaging Protocols

In this research project, MQTT with plain text message was used to model two IoT devices to communicate with each other and make the security of data mission critical. There are many protocols used in an IoT ecosystem at different layers of an OSI Model.

4.2 Establishing a Communication Link

Demonstration of the encryption, the program was carried out using 2 devices: device1 & devoice 2. The sender (Daniel in device 1) and the receiver (James device2). Communication was established by sending a message through the prompt. In this presentation, a visual approach is taken to highlight the process by which the connections were established.

Figure 4.1 shows Device 1 prompting the user to send a message to establish a connection for further message encryption. it also shows the inputted message from the terminal using the message is presented as:

Device 1: Prompt

Hello James, good afternoon, Sir. Are you at the office?

Device 2 responses and willingness to receive further messages. The prompt version was.

Device 2 plain text:

Hi Daniel, good afternoon. Yes, I'm at the office.

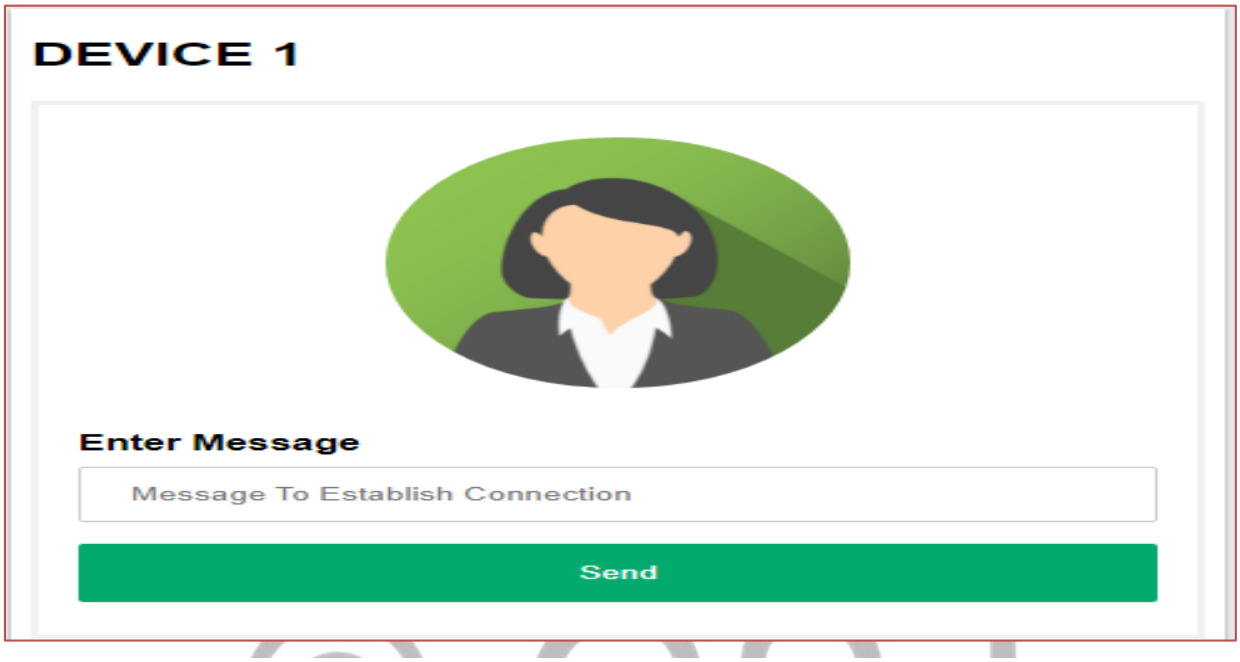

Figure 4.1 Device 1 Prompt User to Establish Connection

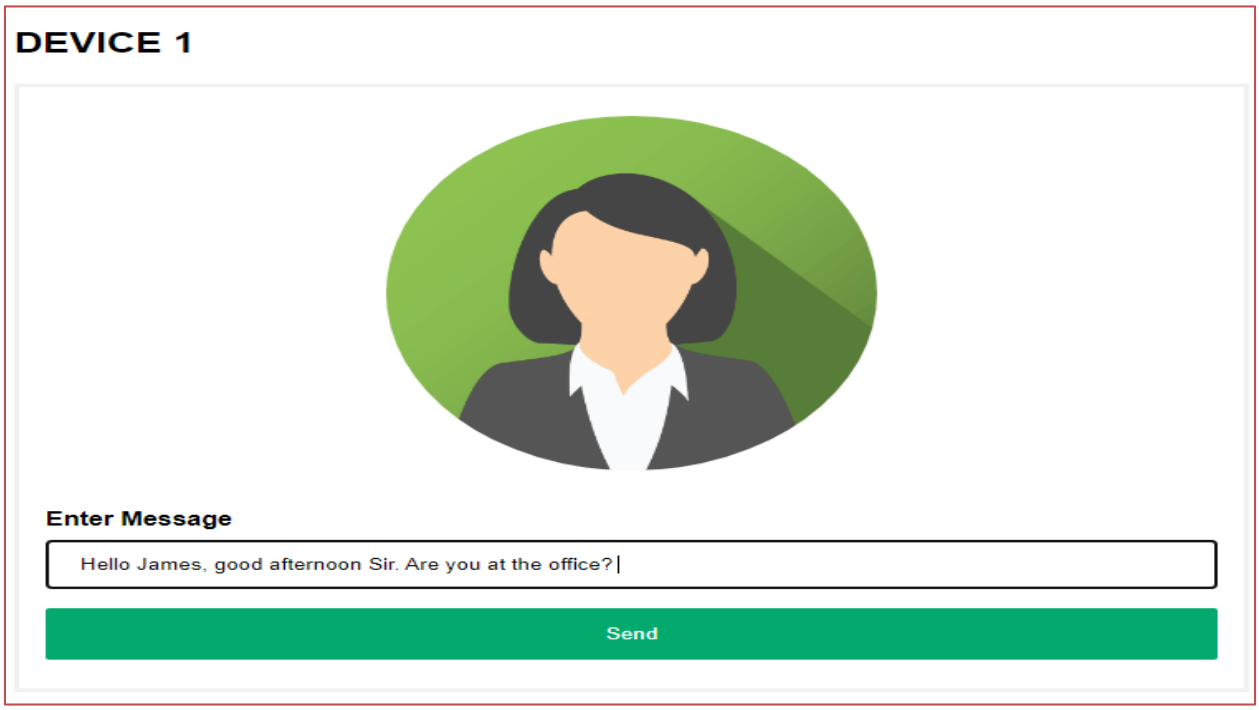

Figure 4.2 Device1 sent message To Device 2 for connection security.

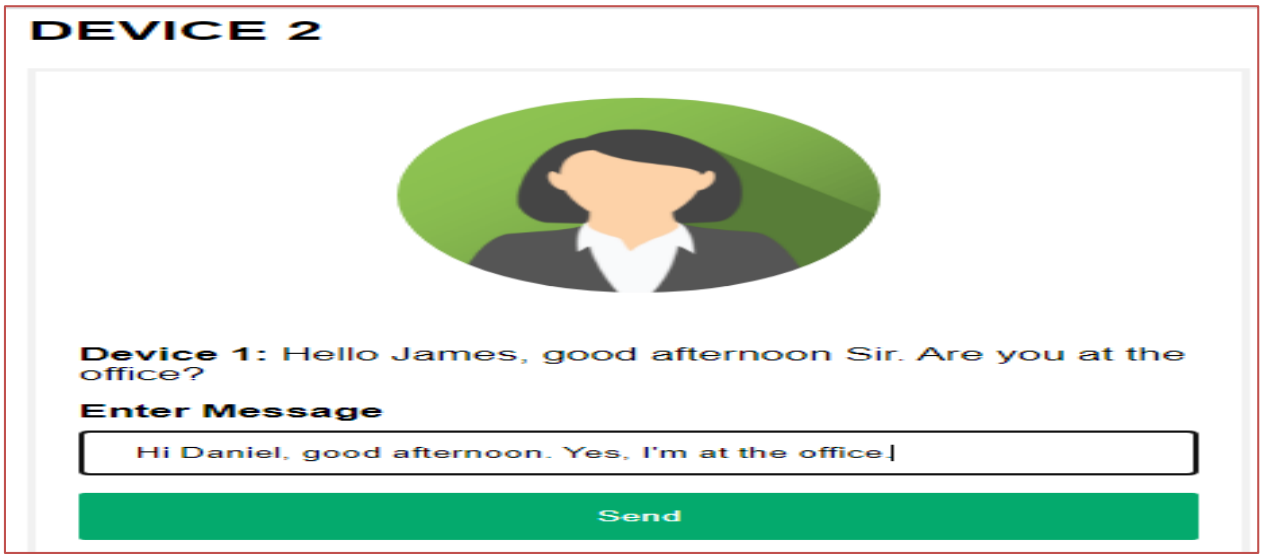

Figure 4.3 Device2 receive the message and agreed to establish a connection for further messages.

4.3 Encryption Phase

This project leverages the Microsoft message queue protocol as an instance of AMQP that allow plain text to be communicated from two simulated IoT device that communicates with each other with messages encrypted end to end using AES (Advanced Encryption Standard) Encryption instances. This involved Advanced Message Queuing Protocol, used for communication in servers. Advanced Message Queuing Protocol prevents message loss owing to the use of TCP and UDP. The transmission control protocol connects the two endpoints. Additionally, endpoints must admit the acceptance of each message. This standard also defines a possible transaction method with a formal multiphase commit classification. True to its roots in the banking business, Advanced Message Queuing Protocols traces information and ensures it reaches the desired ends.

In plain text, the message to be encrypted reads:

Plain test Message
James, I have sent N9, 000, 000 to the Engineer Daniel to begin the Code Contract

To demonstrate the encryption capability of the program the Device Message flow and interaction is shown in Figure 4.2 shows the prompt to the user for the message to be inputted, Figure 4.2 shows the message inputted, and Figure 4.2 shows the encrypted message. Figure 4.2 shows the encrypted message that was sent to Device 2. In prompt display, the plain text (James, I have sent N9, 000, 000 to Engineer Daniel to begin the Code Contract) was encrypted to:

c8c51560c992fd0db4e7159a45c31f01954cc171bbd70b3c100f6c8f857166c1818ab1514fb52efb54e2b38db4a83
3387fc0a47cd2edbb5e61470ac74380900d

Figure 4.2 shows the ciphertext encrypted with AES. The encrypted message has become a jumbled mess of random characters. The encryption was applied, and the plaintext was transformed into ciphertext using the key.

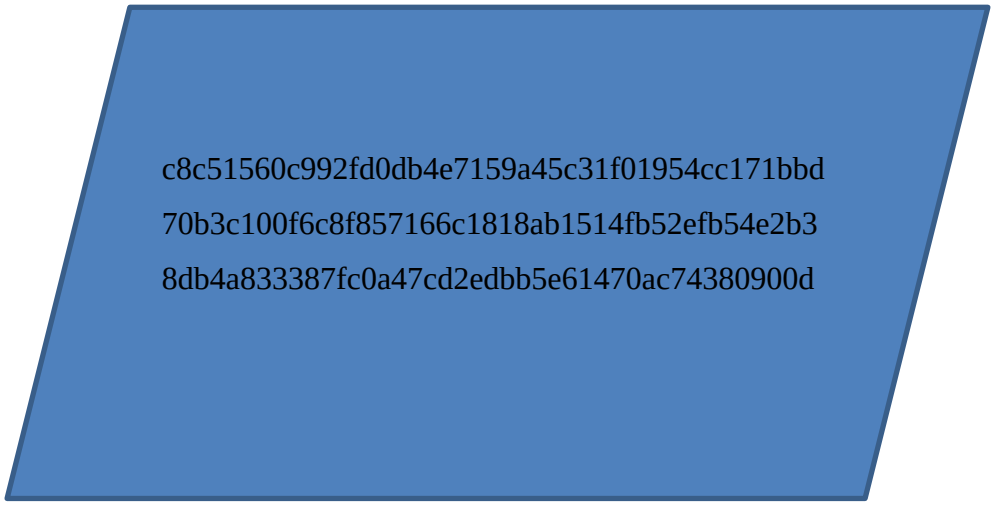

Fig 4.4 Encrypted message that was sent to Device 2

Encryption Algorithm Used

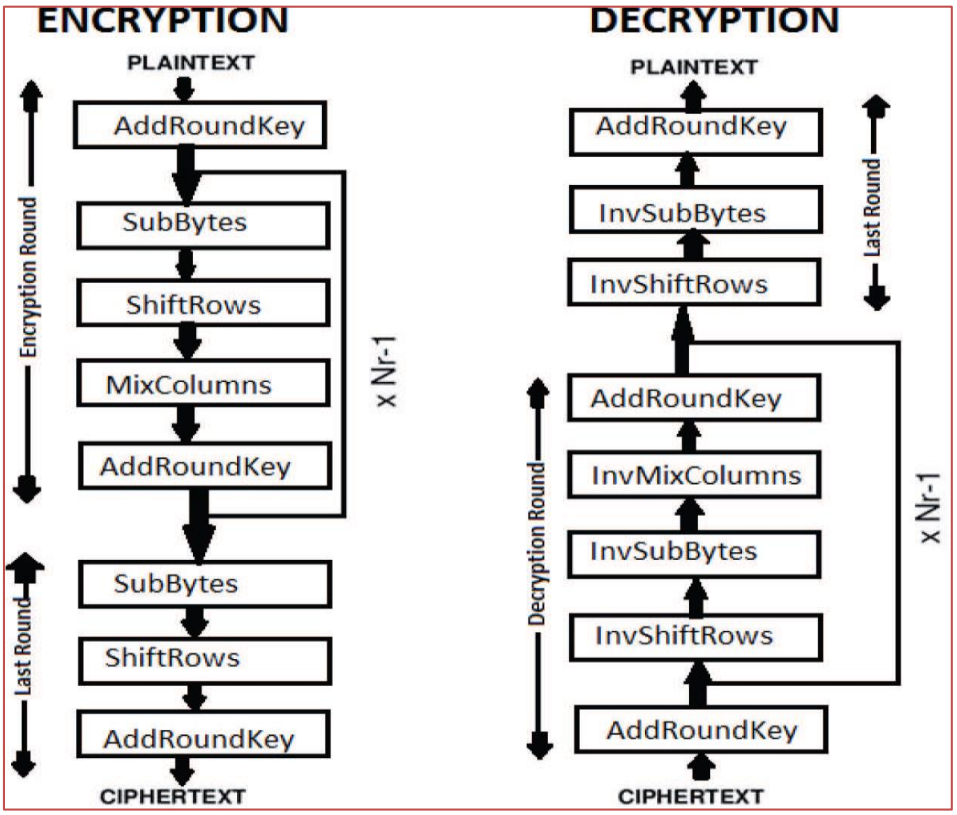

Fig 4.3 Overall structure of AES algorithm Source from Beg (2018)

4.4 Decryption Phase

At Device **2** the Receiver receives the Encrypted messages as

Device 2 encrypted the outgoing Message as:
d2d881bea43850ae0e6f0fc7b50068228f1cf5719478ec7419da48bca86bbc247b9736c0bd1681b40764484c96dc18e3f85fe146d2e379db605fe391e4ed891f

After applying decryption algorithm, that is using the private key the encrypted test becomes plain text.

Plain test Message
James, I have sent N9, 000, 000 to the Engineer Daniel to begin the Code Contract

Sub Bytes/Substitute Bytes:

The 16 input bytes are substituted by using a fixed lookup table known as S-box. Figure 3 shows the S-box of the AES algorithm. This s-box consists of all possible combinations of 8-bit sequences. The first four bytes of a 128bit input block occupy the first column in the 4×4 matrix of bytes. The next four bytes occupy the second column and so on.

The Sub Bytes stage in AES algorithm. The encryption process involves mapping each byte of state to a new byte using row and column values. The leftmost 4 bits determine the row, and the rightmost 4 bits determine the column, serving as indexes into the S-box for an 8-bit output value. For instance, if the hexadecimal value (EA) corresponds to row 9, column 5, the S-box yields the value (87), resulting in the mapping of (EA) to (87). In the shift row step, rows from the matrix generated in the byte substitution are regularly shifted to the left. Any bits dropped off are reinserted on the right side. The first row remains unchanged, the second row shifts one position to the left, and the third row shifts two positions to the left.

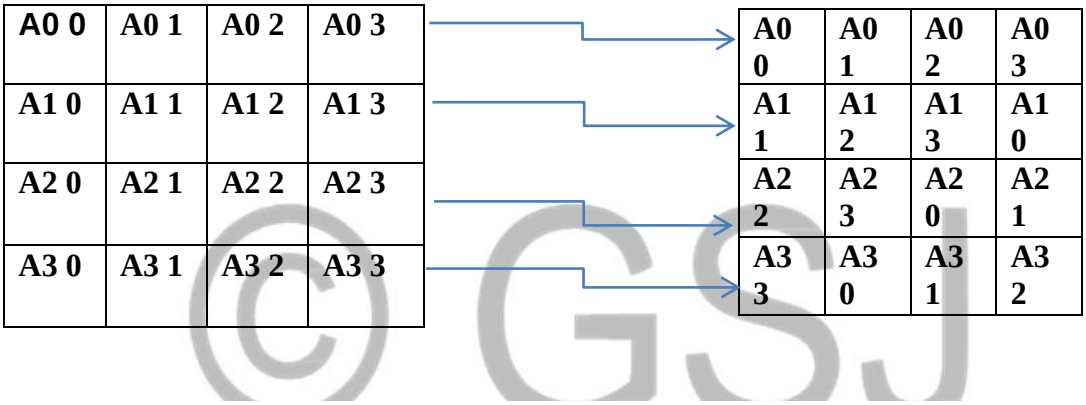

Fig 4.4: showing shift row stage in AES algorithm (Researcher, 2024)

Rules of shifting rows:

- Row 1, no shifting
- Row 2, 1 byte let the shift.
- Row 3, 2 bytes left shift.
- Row 4, 3 bytes left shift.

Mix column:

The mixed columns operate on each column individually. This task takes four bytes of the column as input and outputs, which is a completely new four bytes that substitute the unique four bytes. Each byte of a column is plotted into a new value which is a function of all four bytes in that column.

In the AES encryption process, the mix column stage produces a resultant matrix, treating the 16 bytes as 128 bits. The add round key stage involves bitwise XORing 128 bits of state with 128 bits of the round key. If the result pertains to the last round, the ciphertext output is 128 bits (16 bytes), initiating a new round with byte substitution. This column-wise operation between state columns and round keywords does not include the mix column step in the final round. AES decryption reverses the encryption stages, starting with an inverse initial round and continuing through nine rounds with processes like inverse shift rows, add round key (with keys in reverse order), inverse mix columns, and inverse byte substitution. The decryption stages, including an inverse

shift of rows, use an inverse S-box, and the inverse mix column employs polynomials over the Galois field (GF) 28 with coefficients from the state column. The entire decryption process aims to recover the original plaintext from the ciphertext, completing the cryptographic cycle.

4.5 Performance Test

Component Testing is a type of software testing in which individual components or units of the software are put to test. Component Tests separate a section of code and confirm its accuracy. This study tested the encryption and decryption phase of the code. In testing the code, the study used the infrastructure from Microsoft C # and net. The result showed that the code was fast, there's was no time to waste. The code coverage was high, although not an indicator of success.

The snippet for encryption shown was proven to be bug free.

```
public void Decrypt_should_return_plaintext_when_passing_a_ciphered_value()

{

   const string PLAINTEXT_VALUE = "anonymous@provider.com";

   string –ciphered String = "sjkalsdfjasdljs"; // captured ciphered value

   string plaintextString = _cryptoDummy.Decrypt(cipheredString);

   Assert.IsTrue(plaintextString == PLAINTEXT_VALUE);

}
```

Also, the Puran File Recovery free file recovery tool Windows 10 was not able to detect and recover the ciphertext.

5.0 SUMMARY AND CONCLUSION

5.1 Summary

The research explores the integration of cryptographic solutions to enhance security on the Internet of Things (IoT) landscape, where networked devices ubiquitously connect and communicate. It underscores the pivotal role of IoT in boosting productivity and simplifying business operations yet highlights the inherent security vulnerabilities that pose risks to both personal and business data. The study proposes the use of asymmetric encryption to safeguard communications between IoT devices. This cryptographic method involves generating a pair of keys, a public key for encryption, and a private key for decryption thus securing the data transmission against unauthorized access and ensuring that even if the data is intercepted, it remains indecipherable without the corresponding private key. Utilizing C# and the .NET framework within the Visual Studio Code environment, the research leverages widely adopted software tools to develop a model that addresses the complexities of encryption in a manageable and effective manner. The encryption-decryption model employs a block cipher approach, integrating symmetric and asymmetric key algorithms to robustly protect data while facilitating data interaction across heterogeneous networks through Message Queuing (MSMQ) technology. This ensures not only the security of data

in transit but also supports system reliability and data integrity in intermittently connected environments. However, the study acknowledges limitations related to the physical security of device IDs and the constraints of using C# for encryption tasks, which may affect scalability and performance across different IoT devices. It recommends regular updates to the encryption algorithms and continuous enhancement of the software frameworks to adapt to emerging security threats. Furthermore, future research directions suggest a focus on developing lightweight encryption algorithms to optimize memory and processing resources, aiming to advance secure and efficient data handling in IoT applications. This comprehensive approach aims to fortify the digital infrastructure against potential cyber threats while supporting the expansive growth of IoT implementations in various sectors.

5.2 Conclusion

The study encapsulates the necessity and implementation of enhanced cryptographic measures to secure communications in the rapidly expanding domain of Internet of Things (IoT) devices. Through the application of asymmetric encryption, the research targets the prevalent issue of data breaches and unauthorized access, which are significant threats in the IoT landscape. Asymmetric encryption, involving public and private keys, ensures that sensitive information transmitted across networks is safeguarded against interception by malicious entities. The use of the C# programming language and the .NET framework has been pivotal in the development of an encryption-decryption model that integrates both symmetric and asymmetric algorithms, employing a block cipher for robust security. This model is further reinforced by leveraging Message Queuing (MSMQ) technology, which facilitates reliable communication across inconsistent network connections, enhancing data integrity and availability.

However, the study also acknowledges the inherent limitations and challenges associated with the cryptographic solutions it proposes. The dependency on the physical security of device IDs and authentication keys remains a critical vulnerability that could undermine the encryption measures if not adequately managed. Additionally, the use of C# and .NET, while providing a solid foundation for development, presents constraints regarding versatility and performance across diverse IoT devices and platforms. The conclusion emphasizes the importance of continual updates and refinements to the encryption algorithms to address emerging security threats and vulnerabilities. It advocates for future research to focus on developing more resource-efficient cryptographic algorithms, particularly for image encryption and decryption, to better accommodate the constraints of IoT environments and enhance overall security efficacy. This forward-looking perspective underscores cybersecurity's dynamic and evolving nature in the IoT space, highlighting the ongoing need for innovation and adaptability in security strategies.

Acknowledgment

The authors wish to express their profound gratitude to the almighty God, the sustainer and protector of life, who grants boundless knowledge.

References

1. Abdel, K. A. T. (2017). “Performance analysis of data encryption algorithms,” Retrieved from http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf.pdf
2. Abdul, E., Kader, A., & Mohie, H. (2008). “Performance evaluation of symmetric encryption algorithms,” IJCSNS International Journal of Computer Science and Network Security.
3. Abdul, M., Kader, D. S., Abdul, H. M., & Hadhoud, M. M. (2001). “Analysis of performance for symmetric cryptography.”
4. Abdullah A. (2017). "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data Cryptography and Network Security,” Ryptography and Network Security, 16, 1-11.
5. Abd Zaid M. and Hassan S. (2019). “Modification Advanced Encryption Standard for Design Lightweight Algorithms,” Journal of Kufa for Mathematics and Computer, 6(1), 21-27.

6. Agrawal, M., & Pradeep, M. (2012). “A comparative survey on symmetric key encryption techniques,” Retrieved from http://www.enggjournals.com/ij cse/doc/IJCSE 12-04-05-pdf.
7. Ajay, K., Singh, M. L., and Bansal, P. K. (2012). “Comparison of various encryption algorithms and techniques for secured data communication in the multi-node network”, IJETInternational Journal of Engineering and Technology.
8. Akash, M., Chandra, P., & Archana, T. (2016). “Performance evaluation of cryptographic algorithms: DES and AES,” IEEE Students ’Conference on Electrical, Electronics and Computer Science.
9. Alanazi, O., Zaidan, B. B., Zaidan, A. A., Jalan, A., Shabbir, M., & Al-Nabhani, Y. (2015). “New comparative study between DES, 3DES, and AES within nine factors,” Journal of Computing.
10. Alese, B. K., Philemon E. D., Falaki (2012). “Comparative Analysis of Public-Key Encryption Schemes.”
11. Ali Ahmad Milad, Hjh Zaiton Muda, Zul Azri Bin Muhamad Noh, and Mustafa Almahdi Algaet (2012). “Comparative Study of Performance in Crypto”graphy Algorithms.”
12. Ali Makhmali, Hajar Mat Jani (2013). “Comparative Study on Encryption Algorithms and Proposing, A Data Management Structure.”
13. Ankita Umale,, Ms. Priyanka Fulare (2014). “Comparative Study of Symmetric Encryption Techniques for Mobile Data Caching in WMN.”
14. Apoorva, Yogesh Kumar (2013). “Comparative Study of Different Symmetric Key Cryptography Algorithms.”
15. Chinmoy Ghosh and SatyendraNath Mandal (2014). “A Combined Method for Image Encryption.”
16. Choi I. and Kim J. (2016). "Area-Optimized Multi-Standard AES-CCM Security Engine for IEEE 802.15. 4/802.15,” Journal of Semiconductor Technology and Science, 16(3), 293-299.
17. Daniel Wichs (2012). Barriers in Cryptography with Weak, Correlated and Leaky Sources.
18. Er. Satish Kumar, Amritsar Mr. Amit Puri (2012). “Comparative analysis of the various cryptographic algorithms.”
19. Espressif Systems. (2020). "ESP-IDF Programming Guide,” https://docs.espressif.com/projects/esp-idf/en/latest/esp32/.
20. Feng Bao Pierangela Samarati Jianying Zhou (2012). “Applied Cryptography and Network Security,” Pranay Meshram, Pratibha Bhais
21. G. Ramesh and Dr. R. Umarani (2012). “Performance Analysis of Most Common Symmetrical Encryption Algorithms.”
22. Govinda Rao, D. Siva Prasad, and M. Eswara Rao (2013). “Universal Session-Based Symmetric Cryptographic Technique to Strength the Security.”
23. Hamzah H., Ahmad N., and Ruslan S. (2020). “The 128-Bit AES Design by Using FPGA,” Journal of Physics: Conference Series, 1529(2), 022059.
24. Hussein N. and Shujaa M. (2020). “DNA Computing Based Stream Cipher for Internet of Things Using MQTT Protocol,” International Journal of Electrical and Computer Engineering, 10(1), 1035.
25. Inamdar A. (2020). "ESP32-S2-Security Features,” The ESP Journal. https://medium.com/the-esp-journal/esp32-s2-security-improvements-5e5453f98590.
26. Jitendra Shetland and Harsh Gupta (2014). “Comparative Study of a New Variable-Length Key Block Cipher Technique with DES for Network Security.”

27. Johannes Blömer, Peter Günther, and Gennadij Liske (2012). “Improved Side-Channel Attacks on Pairing Based Cryptography.”

28. K.Brindha, Ritika Sharma, Sapanna Saini (2014). “Use of Symmetric Algorithm for Image Encryption.”

29. Khoa T., Nhu L., Son H., Trong N., Phuc C., Phuong N., Dung N., Nam N., Chau D., and Duc D. (2020). "Designing Efficient Smart Home Management with IoT Smart Lighting: A Case Study." Wireless Communications and Mobile Computing,” 2020, 1-18.

30. Kodali R. and Soratkal S. (2016). "MQTT Based Home Automation System Using ESP8266,” Proceedings of IEEE Region 10 Humanitarian Technology Conference, Agra, 1-5.

31. Kouicem D., Bouabdallah A., and Lakhlef H. (2018). "Internet of Things Security: A Top-Down Survey,” Computer Networks, 141, 199-221.

32. Mansoor Ebrahim, Shujaat Khan and Umer Bin Khalid (2013). “Symmetric Algorithm Survey Comparative Analysis.”

33. Ms.Pallavi H.Dixit, Dr.Uttam L. Bombale, and Mr. Vinayak B. (2013). “Comparative Implementation of Cryptographic Algorithms on ARM Platform.”

34. Nandhini P. and Vanitha V. (2017). “A Study of Lightweight Cryptographic Algorithms for IoT," International Journal of Innovations and Advancement in Computer Science, 6(1), 26-35.

35. Narender Tyagi and Anita Ganpati (2014). “Comparative Analysis of Symmetric Key Encryption Algorithms.”

36. Parida D., Behera A., Naik J., Pattanaik S., and Nanda R. (2021). “Real-time Environment Monitoring System Using ESP8266 and Thing Speak on Internet of Things Platform.”

37. Rajinder Kaur, Er. Kanwalpreet Singh (2013). “Comparative Analysis and Implementation of Image Encryption Algorithms.”

38. Ritu Tripathi and Sanjay Agrawal (2014). “Comparative Study of Symmetric and Asymmetric Cryptography Techniques.”

39. Sheetal Charbathia and Sandeep Sharma (2014). “A Comparative Study of Rivest Cipher Algorithms.”

40. Swati Paliwal, Ravindra Gupta (2013). “A Review of Some Popular Encryption Techniques.”

41. T.Gunasundari and Dr. K.Elangovan (2014). “A Comparative Survey on Symmetric Key Encryption Algorithms.”

42. Veerpal Kaur, Aman Singh (2013). “Review of Various Algorithms Used in Hybrid Cryptography.”

43. Vishwa Gupta, Gajendra Singh, and Ravindra Gupta (2014). “Advance cryptography algorithm for Improving data security.”