International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/223423 Volume 10, Issue 5, September – October (2023)

RESEARCH ARTICLE

A Hybrid Cryptographic Cipher Solution for Secure Communication in Smart Cities

Mohammed Naif Alatawi
Information Technology Department, Faculty of Computers and Information Technology, University of Tabuk, Kingdom of Saudi Arabia (KSA).
alatawimn@ut.edu.sa

Received: 21 July 2023 / Revised: 29 September 2023 / Accepted: 15 October 2023 / Published: 30 October 2023

Abstract – The proliferation of sensor networks and other Internet of Things devices has prompted growing privacy and safety concerns. These devices have very little memory, computing power, and storage space. Security for low-powered IoT devices, such as RFID tags, nodes in wireless sensor networks (WSNs), etc., has become increasingly difficult. So, enough security for these devices was achieved by the development of lightweight cryptographic algorithms. In recent years, "smart cities" have emerged to improve contemporary lifestyles and further social development. These are enabled by developments in ICT and may open up new avenues for social and economic development. However, not everything is as secure and private as we hope it will be. The effects of the Internet of Things on IoT-based data transmission networks have been the subject of extensive study over the past few decades. Due to this flaw in the authentication process, verifying the identification of such people safely is extremely difficult. The study's goal is to provide a safe authentication technique for IoT that makes use of Hybrid and Adaptive Cryptography (HAC). In this study, we focus on authentication as a potential security risk in IoT data transmission networks. The study proposes a hybrid and adaptive cryptography (HAC) approach to authentication for the Internet of Things as a means of resolving this issue. The proposed technique of cryptographic protection makes use of the exclusive-or (Ex-or) operation, a hashing function, and a hybrid encryption strategy based on the Rivest Shamir Adleman (RSA) and the Advanced Encryption Standard (AES) algorithms. The proposed solution is simple to implement while effectively overcoming the cryptographic system's constraints via a hybrid encryption mechanism. Using the Diffie-Hellman key exchange protocol, the RSA algorithm for privacy, and the SHA-1 algorithm for authenticity, this study aims to provide a unified security architecture for modern networks.

Index Terms – Smart City, Authentication, IoT, RSA, AES, Cryptography, SHA, HAC, WSN, GPS, RFID.

1. INTRODUCTION

When it comes to tackling urban sustainability issues, "smart cities" are information- and technology-driven approaches [1]. The health, safety, quality of life, and environment of citizens in many major cities are all being improved because to information and technology developments [2]. Today's "smart cities" are an essential component of the international infrastructure [3]. Quality of life and ease of access to essential services are two of the main focuses of smart city initiatives [4].

Smart cities rely heavily on IoT devices and other forms of information and communication technologies [5]. Communication privacy issues have arisen in the context of smart cities due to the Internet of Things and other kinds of digital technology. The purpose of the IoT is to unify existing computer networks [6]. Even beyond logistics, "smart neighborhoods," "intelligent transportation," and "smart banking" are all finding applications in this fast-expanding industry [7] [8]. However, ensuring stable and successful services requires addressing security flaws in the design, which includes the management and deployment of devices [9].

An intruder with access to a compromised system can do anything from use a legitimate login token to launch a malicious program at boot time [10]. Furthermore, the attacker can monitor the victim's or another's activity on their own devices. Security for Internet of Things (IoT) data in transit, where an attacker could pose as the sending device while stealing critical information. The aforementioned privacy and security risks are only going to escalate, therefore it's crucial to take preventative measures now. Because attacks on the underlying IoT infrastructure could lead to loss of control of the application(s), denial of service, loss of user privacy, and many other security issues, researchers believe a cryptographic approach is one of the fundamental countermeasures for securing IoT applications and their components. Fewer cryptographic techniques are combined for greater security. Hybrid cryptography combines the best aspects of different algorithms in order to get around the drawbacks of the current state of the art in encryption. Among the many potential solutions to IoT security issues, this one stands out because of the privacy, integrity, authentication, and non-repudiation it offers for IoT data [11, 12].

If you encode something, you end up with a clear ascending script that can only be read by the one person who knows the

RESEARCH ARTICLE

code [13][14]. Using an always-on internet connection, it will manage the online servers and make purchases, ensuring the security of IoT-based data. Information about you will be stored in a database that is accessible by various companies if you deal with any firm, no matter how small [15]. Knowing how to protect information collected through the Internet of Things is essential. Encryption is necessary for the current task. Symmetric key encryption and asymmetric key encryption are the two main types of encryption.

Inverse Key a single key is used for encryption and decryption of data. Two separate keys are used in asymmetric encryption, one for encoding and one for decoding. A public key is a type of encryption key that can be shared across a group of people. A one-of-a-kind decryption key that is never shared [16].

1.1. Cryptography

When we talk about cryptography, or secret writing, we mean altering IoT Based Data Security so that no one except the intended recipient can read it. Digital encryption has evolved into a highly specialized research area in the field of information security and assurance since it not only secures data but also verifies the identities of those who access it. Due to rapid technical breakthroughs and the ever-increasing volume of IoT Based Data Security being transmitted across open channels like D2D communication, digital cryptography has garnered a lot of attention in the modern digital age. What we're looking for is a way to protect the data being exchanged while still using an unencrypted channel for communication. The Cryptographic System Model is depicted in Figure 1. A brief history of cryptography (till 2001) is given in the table below in Table 1.

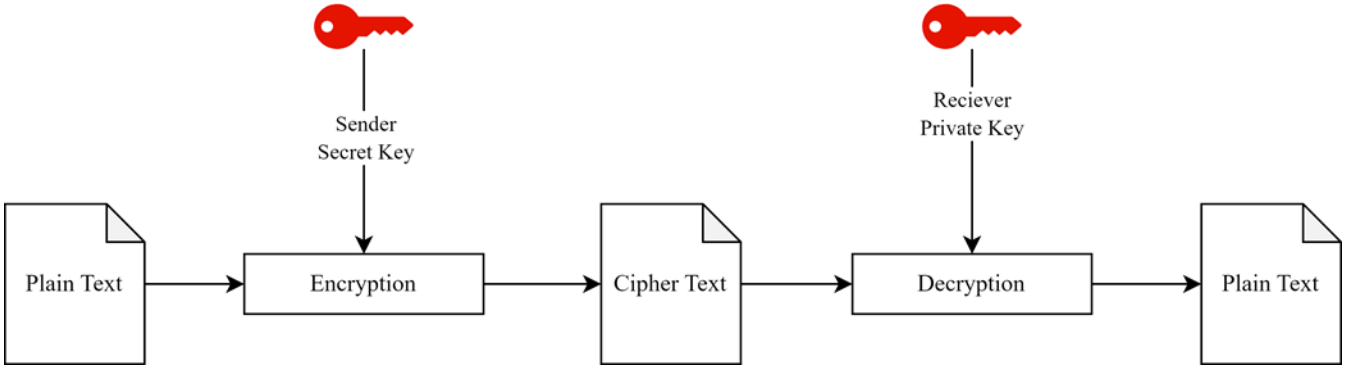

Figure 1 Model of Cryptographic System

Table 1 History of Cryptography

100 A.D	Ceasar Cipher
1553	Invention of Vigenère cipher
1835	Morse code developed by Samuel Morse
1854	Invention of Play fair Cipher by Charles Wheatstone
1883	Auguste Kerckhoffs publishes his book on Laws of Cryptography, La Cryptographie military
1932	Enigma Machine Broken
11948	Claude Shannon publishes his Paper on Information Theory
1974	Block cipher, Feistel-network designed by Horst Feistel
1976	DES-Data Encryption Standard, a Symmetric-Key Cipher based on Feistel Networks, was published as an Information Processing standard by the NBS in the United States. Diffie–Hellman key exchange Published
1977	Asymmetric-Key Encryption Algorithm, RSA invented by Adi Shamir, Ron Rivest, and Leonard Adelman
1994	Netscape releases SSL or Secure Sockets Layer encryption protocol.
1995	SHA (Secured Hashing Algorithm) is published
2001	Rijndael replaces DES to become the new Information Security Standard, the Advanced Encryption Standard, by the NIST.

RESEARCH ARTICLE

1.2. Secret-Key Cryptography

In the case of secret-key cryptography, there is only ever one key. It may be used for both encrypting and decrypting, making it ideal for IoT-based data security. The term "key" is used to describe any code that can be used to decipher encrypted text. The key can be used by both the sender and the receiver. If the key is made public, IoT-based data security will be compromised. Using a secret key will not safeguard the sender if someone else sends a message purporting to come from them. Longer keys are more secure and less likely to be discovered through brute force attacks. Since the encryption key and decryption key are the same, the entire operation takes very little time. However, if the key is compromised, the consequences could be catastrophic. If you have a symmetric key, anyone in possession of it can decrypt your encrypted data. Because symmetric encryption is used for two-way communication, the security of IoT-based data at the sender and receiver ends is jeopardized.

1.3. Public-Key Cryptography

Asymmetric cryptography uses two keys—a public key that is publicly known and a private key that is known only to the intended recipient—to encrypt and decode communications and verify signatures (or produce signatures). The word "asymmetric" comes from the fact that the key used to encrypt messages or verify signatures cannot be used to decrypt messages or create signatures, respectively. Because the private keys are never revealed, asymmetric key ciphers are more secure. Access code Symmetric encryption is faster, but encryption is more difficult. If an unauthorized third party attempts to intervene in a transaction, this is known as a "man-in-the-middle" assault. The implications of losing a private key are usually disastrous. A digital signature, like to a handwritten signature on a physical document, can verify the authenticity of a message and its origin. Figure 2: The Digital Signature Verification System.

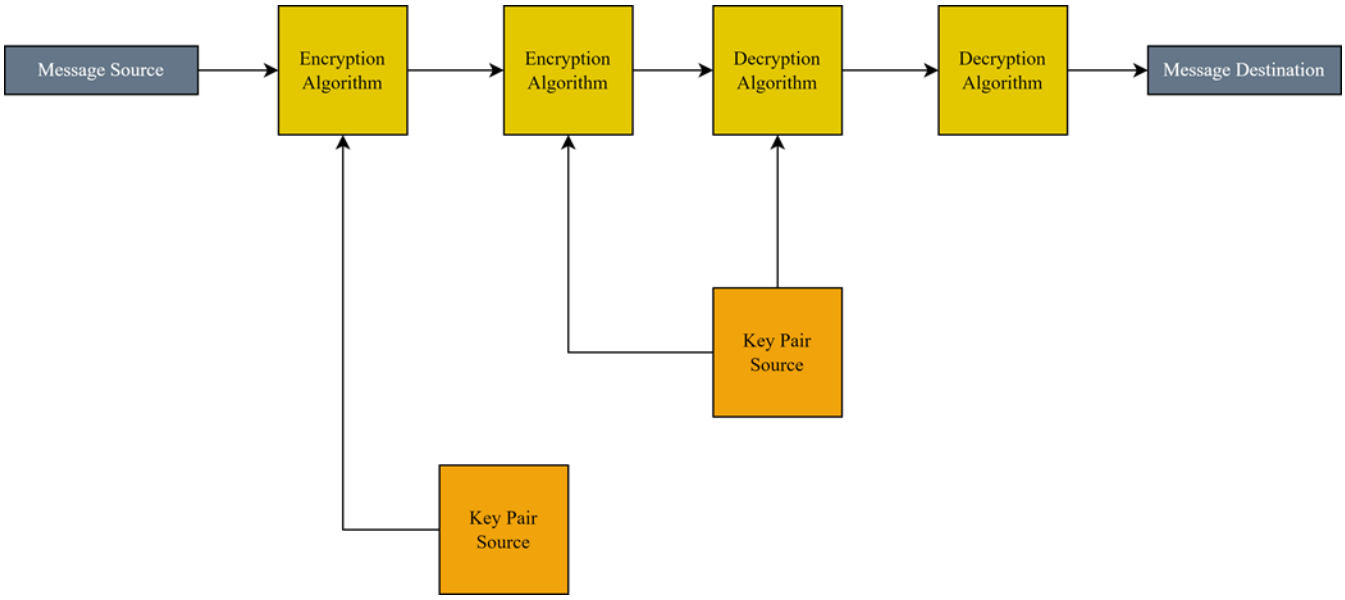

Figure 2 Digital Signature Verification System

A mathematical change known as a hash function is used to "encrypt" data security for the Internet of Things in a way that cannot be decrypted. IoT-based data can be encrypted and decrypted with this technology without the need for a key. A fixed-length hash value is generated from the plaintext, and neither its contents nor its length can be recovered from this value. It is difficult for an attacker or virus to change the contents of a file once it has been digitally fingerprinted using one of these methods. In order to prevent unauthorized access to private information, several OSes employ hash algorithms to encrypt passwords. Until the key interchange problem is resolved, using symmetric approaches can feel like a classic catch-22. When staking a secret symmetric key between two people, trust becomes a critical success factor that must be established. Trust issues can arise when encryption is used to verify something's authenticity. Asymmetric cryptography (RSA, ECC) is strong but requires a lot more processing power than symmetric methods (DES, AES), making it inefficient if used to encrypt and decrypt every packet sent over the internet.

This paper proposes a safe authentication solution for IoT, including the use of an exclusive-or (Ex-or) operation, a hashing function, and a hybrid encryption approach. The proposed fix efficiently addresses the weaknesses in the

RESEARCH ARTICLE

cryptographic system by combining symmetric and asymmetric Encryption using the Rivest Shamir Adleman (RSA) and the Advanced Encryption Standard (AES) algorithms.

The paper is divided into sections. Section 1 discusses the state of IoT security and why a solid authentication method is required. The literature survey on cryptographic techniques and IoT security in general, touching on symmetric and asymmetric Encryption are provided in section 2. The section 3, talks about the proposed authentication solution for the Internet of Things, which uses the Ex-or operation, a hashing function, and a hybrid encryption scheme, and is based on Hybrid and adaptive cryptography. Section 4, examines the evaluation of the proposed method, which includes a thorough examination of its performance and a comparison to alternative approaches. The significance and contribution of the proposed method, as well as possible future study routes, are discussed, together with a summary of the paper's findings, in section 5.

2. RELATED WORK

The creation of secure and efficient cryptographic methods to protect sensitive information is receiving increased attention in several domains, including the Internet of Things (IoT), cloud computing, and network security. Engineers are continually exploring and developing novel approaches to data security in such settings. By describing the most significant recent breakthroughs in cryptographic algorithms and their applications, this literature review aims to achieve just that.

Kundu et al. [1] introduced a novel implementation of the Vigenere cryptographic algorithm using Quantum-Dot Cellular Automata (QCA). This work explores the potential of leveraging quantum-dot-based technology for secure data encryption, offering a glimpse into the future of quantum cryptography.

Dorobantu and Apostol [2] examine the use of poly-alphabetic substitution ciphers in the Internet of Things. In this study, we look into the viability of these ciphers for use in low-power Internet of Things (IoT) devices.

Jassim and Farhan [3] do a thorough review of stream ciphers that perform well with constrained resources. Understanding the present lightweight encryption options is aided by their study.

Iqbal et al. [4] present LPsec, an optically optimized cryptography technology. LPsec is a lightning-fast encryption protocol designed to keep data safe while traveling through optical networks.

To strengthen cloud safety, Vignesh et al. [5] propose a new set of cryptographic principles. Their efforts to provide secure data access and management are particularly relevant in the modern era of cloud computing.

Windarta et al. [6] provide lightweight cryptographic hash algorithms for Internet of Things application scenarios. These hash functions were developed specifically for use in low-power IoT devices, where they may provide trustworthy checks of data integrity.

ACiS is a lightweight, safe homomorphic block cipher invented by Hariss and Noura [7]. This study contributes to the development of more robust cryptographic systems suitable for a variety of applications.

The FPGA implementation of the Present cipher is studied in depth by Datta and Sreehari [8]. The importance of hardware-based encryption methods in real-time software has made them the primary focus of this research.

Baksi et al. [9] use side-channel analysis to investigate stream ciphers and related structures. Understanding the vulnerabilities of cryptographic systems is essential for improving and expanding their use.

To evaluate the robustness of integrated circuits against side channels, Liu et al. [10] employ state-of-the-art transistor technology. The physical security of cryptography implementations is improved because to this research.

Hussain and Mohideen [11] probe how state-of-the-art machine learning techniques can be utilized to identify bogus Enigma encoded signals. The results of this work are useful in cryptanalysis and anomaly detection.

Irodah and Adriansyah [12] investigate the Vernam cipher cryptographic method to build a trustworthy customer-facing portal for a regional water company. This study demonstrates the various practical applications of cryptographic techniques beyond data encryption.

Effective cryptography approaches are presented by Ruiz et al. [13] to guarantee the privacy of optical communications. This research tackles the critical issue of providing secure data transfer within optical networks, which is becoming increasingly important.

Parida and Bhanja [14] investigate the smart meter cybersecurity issues. Their findings provide insight into securing smart grids' foundations, which is critical for ensuring the reliability and security of energy distribution networks.

Upadhyaya, Gay, and Polian [15] presented LEDA, a method for locking-enabled differential analysis of cryptographic circuits. This research makes a significant addition to hardware-oriented security by focusing on how to prevent attacks against cryptographic systems.

RESEARCH ARTICLE

Iqbal et al. [16] suggest integrating the ARIA Cipher 256 algorithm and mbedTLS with MQTT to better fortify security in IoT applications. This study discusses the importance of taking precautions to ensure the safety of the IoT ecosystem.

Garcia and Liu [17] study post-quantum key exchange systems. This research looks on possible alternatives to classical cryptography methods in light of the possibility that they could be attacked by quantum computers.

Salman, Farhan, and Shakir [18] analyze and compare lightweight variants of the Advanced Encryption Standard (AES) developed for IoT applications. Their findings show the tensions that can arise when prioritizing IoT security and resource conservation.

Segala [19], a practical guide for JavaScript developers, offers in-depth coverage of common cryptographic operations in Node.js and web browsers. Web developers who care about the safety of their users' personal information and communications can benefit from this guide.

Rajashree et al. [20] describe a homomorphic encryption technique for string concatenation. In order to perform computations on encrypted data without compromising the security of the data itself, homomorphic encryption is an essential tool.

Erbes et al. [21] offer evaluation metrics for GIFT block cipher side-channel leakage based on a case study. The research findings can be applied to evaluating and bolstering the side-channel attack resistance of cryptographic algorithms.

The use of a hybrid pseudorandom number generator for cryptography in Internet of Things applications is explored by Radie et al. [22]. An integral part of cryptographic systems, random number generation is investigated here for its role in protecting Internet of Things devices.

Smart cities rely heavily on cloud storage and mobile devices, and Bhat et al. [23] suggest employing blockchain technology to provide secure data transfer between the two. This research contributes to the expanding body of literature on the use of blockchain technology for safe data sharing in urban environments.

Mentens et al. [24] primarily emphasize on the importance of application-specific FPGAs in enabling cryptographic agility via specialized reconfigurable architectures. In this investigation, we explore the possibilities for adapting cryptographic algorithms implemented in hardware to specific applications.

Lavanya et al. [25] investigate the potential for safe data transfer using a hybrid crypto processor based on AES and HMAC. This inquiry is primarily concerned with the safety and confidentiality of sent information.

These articles showcase the breadth and depth of research into cryptography by discussing a wide range of cryptographic approaches, applications, and emerging challenges in the dynamic field of information security. Table 2 shows the comparison table of the related works. The contributions of each approach to secure communication over the Internet of Things are summarized in the table above. Each methodology has its own set of benefits and drawbacks, which are detailed in the corresponding research paper and presented in the strength and weakness columns, respectively.

Table 2 Comparison Table of the Related Works

Reference	Cipher Technique	IoT Relevance	Limitations
Windarta et al. [6]	Lightweight Cryptographic Hash Functions	Relevant for IoT, particularly in terms of efficiency	Utilizes hash functions with specific block sizes
Baksi et al. [9]	Side Channel Analysis on Stream Ciphers	Relevant for analyzing the security of IoT stream ciphers	Focuses on side-channel analysis rather than proposing a new cipher
Liu et al. [10]	Integrated Circuits Side-Channel Resilience	Relevant for IoT device security and resilience	Specific to silicon nanowire field-effect transistors, may not apply to all IoT devices
Hussain and Mohideen [11]	Machine Learning for Enigma Cipher Detection	Relevant for IoT security and anomaly detection	Focuses on machine learning and detection, not cryptographic techniques
Irodah and Adriansyah [12]	Vernam Cipher Cryptography Algorithm	Relevant for practical applications in securing self-service systems	Application-specific to a local water company, not a general-purpose cipher
Parida and Bhanja [14]	Cyber Security for Smart Meters	Highly relevant for IoT, particularly in smart grid security	Specific to smart meters, may not address broader IoT security concerns

RESEARCH ARTICLE

Upadhyaya et al. [15]	Locking Enabled Differential Analysis of Cryptographic Circuits	Relevant for hardware-oriented security, protecting cryptographic implementations	Focuses on hardware security and protection rather than proposing new ciphers
Iqbal et al. [16]	Integrating MQTT with ARIA Cipher 256 Algorithm	Relevant for enhancing security in IoT applications with MQTT	Specific to integrating MQTT with ARIA Cipher
Garcia and Liu [17]	Post Quantum Cipher Suites for Key Exchange	Addresses the post-quantum cryptography challenge in IoT key exchange	Focuses on post-quantum cryptography, may not be immediately applicable to all IoT devices
Salman et al. [18]	Lightweight AES Modifications for IoT	Relevant for resource-efficient encryption in IoT applications	Focuses on modifying AES for lightweight IoT use, potential trade-offs in security
Segala [19]	Essential Cryptography for JavaScript Developers	Relevant for web-based IoT applications with JavaScript	Focuses on educating JavaScript developers in cryptography, not a research paper
Rajashree et al. [20]	Homomorphic Encryption for String Concatenation	Relevant for secure computations on encrypted data in IoT	Specific to homomorphic encryption for string concatenation, not a general-purpose cipher
Unger et al. [21]	Side-channel Leakage Assessment Metrics for GIFT Ciphers	Relevant for evaluating GIFT block ciphers in IoT contexts	Focuses on evaluating side-channel leakage, may not propose new ciphers
Jassim et al. [22]	Pseudorandom Number Generator for IoT Cryptography	Relevant for secure random number generation in IoT	Focuses on pseudorandom number generation, not proposing new ciphers
Dar et al. [23]	Blockchain-based Secure Data Exchange in Smart Cities	Relevant for secure data exchange between IoT devices and cloud networks	Specific to blockchain-based solutions, not cryptographic algorithms
Biesmans et al. [24]	Application-specific FPGAs for Cryptographic Agility	Relevant for custom cryptographic solutions in IoT applications	Focuses on hardware customization, not proposing new cryptographic ciphers
Umamaheswari et al. [25]	Hybrid Crypto Processor based on AES and HMAC Algorithms	Relevant for secure data transmission in IoT applications	Specific to a hybrid processor, not proposing new cryptographic algorithms

3. METHODOLOGY

3.1. Hybrid Model

Combining RSA with symmetric ciphers such as AES and Blowfish, this paradigm provides robust encryption with the best of both worlds. For private key exchange, RSA is employed, while AES and Blowfish are used for strong encryption. Figure 3 shows the hybrid model.

3.1.1. Key Encryption and Storage

- An RSA key pair, consisting of a public key (used for encryption) and a private key (used for decryption), is created by the user during the key exchange phase. The private key remains private while the public key is distributed.
- An arbitrary symmetric key (AES or Blowfish) is produced whenever a user requests encrypted data transmission. To encrypt and decrypt the data, we will use this symmetric key.

3.1.2. Encryption of Symmetric Key

- Sender RSA Encryption: The sender uses the recipient's RSA public key to encrypt the symmetric key. This assures that the symmetric key may be decrypted only by the intended receiver, who owns the associated private key.

RESEARCH ARTICLE

- In the case of RSA encryption, the recipient uses their own private RSA key to decrypt the symmetric key.

3.1.3. Steganography for Secure Key Storage

- Container Data: The symmetric key is cloaked in what appears to be an unrelated set of information, or a "container." Digital information of any kind can serve as this container.
- Key Embedding: Using steganography techniques, the symmetric key is hidden inside the container's data. These methods maintain the original appearance and sound of the container data while embedding the encrypted symmetric key.
- Secure Storage: The container with the embedded key can be safely kept or sent. An adversary who manages to intercept the message would still need to be familiar with the steganography method utilized in order to extract the key.

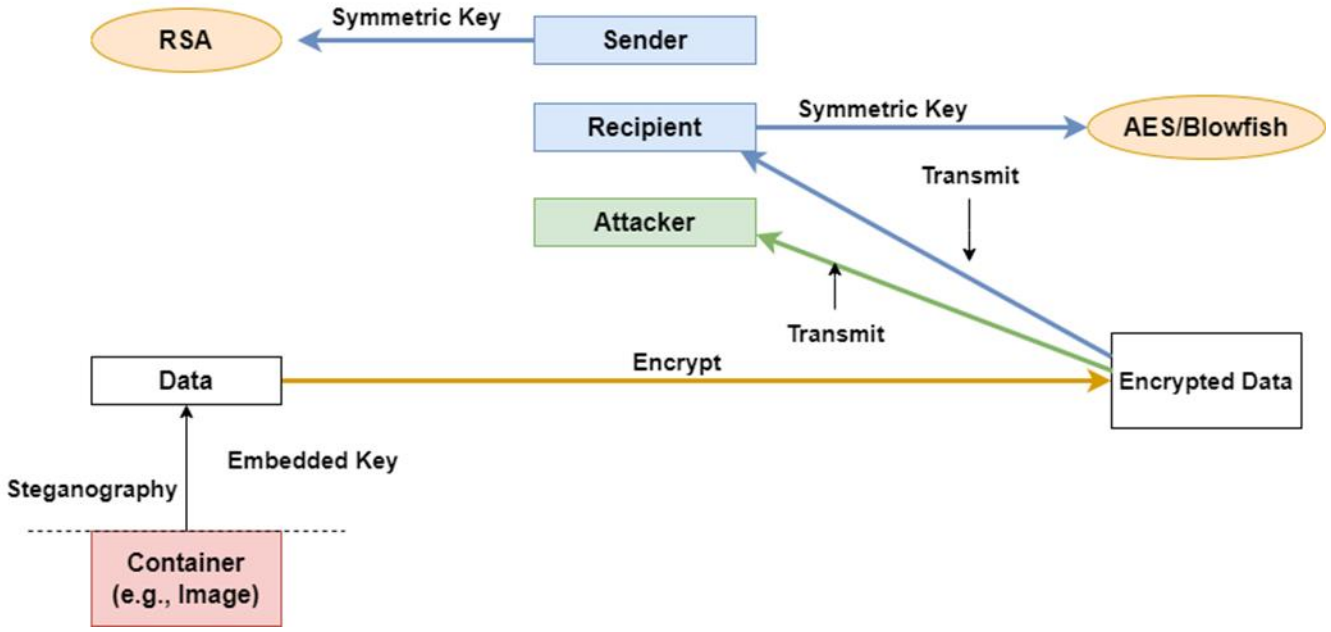

Figure 3 Hybrid Model

3.2. AES (Advanced Encryption Standard)

AES is a widely used symmetric encryption algorithm. It operates on fixed-size blocks of data, typically 128 bits, and supports key lengths of 128, 192, or 256 bits. AES is known for its efficiency and security.

The encryption process in AES consists of several steps:

Key Expansion: The key is expanded into a set of round keys, one for each round of encryption.

Initial Round: The plaintext is XORed with the initial round key.

Rounds (9/11/13 rounds for 128/192/256-bit keys): Each round consists of the following operations:

a. SubBytes: Substitutes each byte of the state with a corresponding byte from the S-box (a predefined substitution table).

b. ShiftRows: Shifts rows of the state matrix by different offsets.

c. MixColumns: Mixes the columns of the state matrix to introduce diffusion.

d. AddRoundKey: XORs the state with the round key.

Final Round: The final round omits the MixColumns operation.

The decryption process in AES is essentially the reverse of encryption, where the inverse operations of $SubBytes$, $ShiftRows$, and $MixColumns$ are applied.

The AES encryption equation for a single round can be summarized as follows in Eq. (1):

$$State = AddRoundKey\left(\begin{matrix}Sub_{Bytes}\left(Shift_{Rows}\left(MixColumns_{(State)}\right)\right), \\ Round_{Key}\end{matrix}\right) \quad (1)$$

Where:

RESEARCH ARTICLE

State is the current state of the data (a 4x4 matrix).

RoundKey is the round key for the current round. Figure 4 shows the AES architecture.

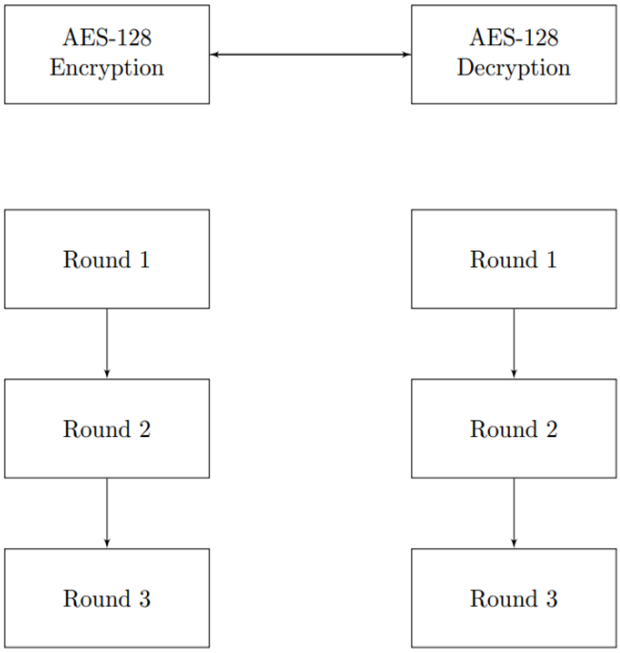

Figure 4 AES Algorithm [7]

3.3. Blowfish

Blowfish is a symmetric block cipher that is frequently used for secure communications and data storage due to its speed and efficiency.

Blowfish uses the following steps to encrypt data:

Key Expansion: A key-scheduling mechanism is used to convert the original, variable-length key into an array of subkeys.

The plaintext is first XORed with the first two subkeys, each of which is 32 bits long.

Number of Rounds: 16 The following steps make up one cycle:

a. Subkey Mixing: Each cycle of XORing the data blocks uses a new set of subkeys.

S-boxes (b) S-boxes are used to make substitutions where each 32-bit block is broken down into eight 4-bit sub-blocks based on the subkeys.

At the conclusion of each round, we exchange the two data blocks. c.

Subkeys are XORed with the data blocks when each round is complete.

Each round's Blowfish encryption equation is (2):

$$Left = Left\ XOR\ SubKey[i]$$

$$Right = F(Left) XOR\ Right$$

$$Swap(Left, Right) \ldots . (2)$$

Where:

Two 32-bit data blocks, labeled Left and Right.

The ith subkey for that round is represented by SubKey[i].

F(Left) is the output of an XOR and an S-box substitution.

Decryption in Blowfish is similar to encryption but works in reverse.

These equations summarize, in a nutshell, how AES and Blowfish encrypt data. Additional information, such as bitwise operations, fixed S-boxes, and constants, is required for practical implementations. The Blowfish architecture is depicted in Figure 5.

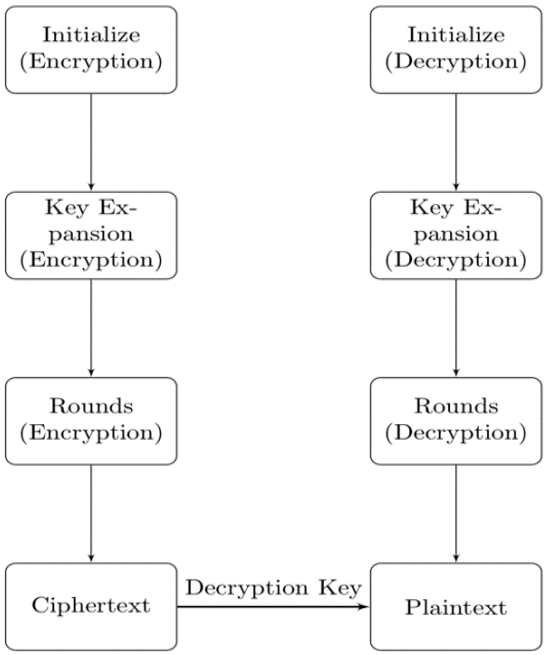

Figure 5 Blowfish Algorithm [4]

3.4. Rivest–Shamir–Adleman (RSA) Algorithm

RSA, or Rivest–Shamir–Adleman Algorithm, is a public-key/asymmetric-key cryptography algorithm that uses a Public Key, available to everyone on the network, to encrypt IoT Based Data Security and a Private-Key, accessible to only the Sender and Receiver, for decryption. The keys are large prime numbers of lengths 1024 / 2048 / 3072 / 4096 Bits. Figure 6 shows the RSA architecture.

RESEARCH ARTICLE

Two large prime numbers p and q, are selected.

The modulus n is calculated as,

n = p, q

Euler's Totient Function of (3)

$$n,'(n) = (p-1)(q-1) \ldots (3)$$

The Public-key, e is selected, such that e and the Euler's Totient Function of n are co-primes, i.e., $gcd(e,'(n)) = 1$

The Private key, d is calculated such that $(d\ e)\ mod\ '(n) = 1$

Hence the Public-Key pair is (e, n), and Private-Key Pair is(d, n).

The Plaintext, M is encrypted by the use of the Public Key, e as in Eq (4):

$$Ciphertext;\ C = Memod\ n \ldots (4)$$

The Cipher text, C is decrypted by use of the Private-Key, d as in Eq (5):

$$P\ plaintext;\ M = Cdmod\ n \ldots (5)$$

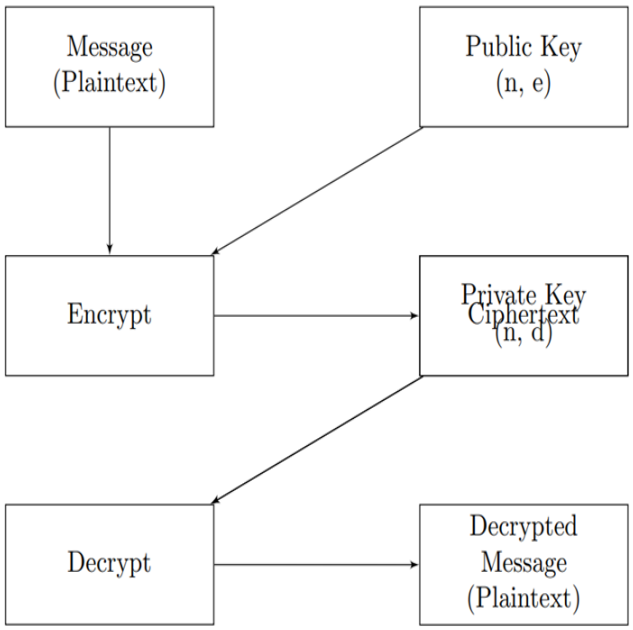

Figure 4 RSA [10]

3.5. LSB Image Steganography

IoT-based data security can be concealed in digital media using a method called least significant bit steganography. Each pixel in an image has a numerical value, and that number often corresponds to a specific color. These pixel values in a grayscale photograph can be anything from 0 to 255. In LSB Image Steganography, only the least significant bit of a pixel is altered, with little effects on the final image. The security for the data generated by IoT devices is concealed under a cover image. The message is encoded in binary, and the cover artwork is transformed to greyscale. Iterate through the image's pixels and create a new temporary variable, temp, for each one. Set temp to 0 if the least significant bit (LSB) of the Pixel Value and the message bit are the same, and 1 otherwise. Modify the output image pixel by adding the value of the temporary variable temp to the value of the image pixel. This process is repeated until the message is well ingrained.

3.6. SHA-1 Hashing Function

The message is digested into a hash using a one-way function in a safe hashing technique. The message digest will be updated to reflect any changes made to the message. Authentic message authentication codes, digital signatures, and random numbers can all be generated using this feature of SHA-1.

3.7. Symmetric Cryptography in WSN

A symmetric encryption technique can transform messages from their plaintext (M) form into their encrypted (C) form while still making use of the same secret key. Although the short-key advantage of this invertible encryption algorithm is great, the necessity of so many keys and so much IoT Based Data Security flow is a major downside. The two most popular examples of symmetric algorithms are block ciphers and stream ciphers. Block cipher algorithms, in contrast to cipher stream algorithms, which process IoT Based Data Security bits by bits, operate with IoT Based Data Security blocks of a specified size, such as 64-bit or 128-bit keys. We refer to some of the best-known cryptographic methods for WSNs, such as:

3.8. AES (Advanced Encryption Standard)

The IoT-based data security technique uses blocks for encryption. Similar algorithms, known as AES-128, AES-192, and AES-256, encrypt IoT-based data security with secret keys of lengths 128, 192, and 256 bits. Encryption and decryption techniques rely on the non-linear BYTE SUB function (Byte Substitution), which swaps out one byte at a time using a lookup table. Using the SHIFT ROW function, which shifts by a certain number of steps to the left, and the MIX COL function, which mixes bytes across columns, we may perform linear combinations of bytes on each byte to get new values.

3.9. RC4

An algorithm is a form of stream cipher. A series of bytes (or bits) K, generated from the key, are ANDed with a series of bytes (or bits) M, the plaintext, to generate the cipher text C, using the XOR operator ().

RESEARCH ARTICLE

3.10. Asymmetric Cryptography

Asymmetric or public-key cryptography employs a pair of keys: a publicly available "public" key for encrypting data, and a privately held "private" key for decrypting it. Some asymmetric cryptosystems that can be implemented in WSNs are given below:

3.11. RSA

It is currently the most widely used algorithm for asymmetric cryptography. The ability to factor large prime numbers is fundamental to the security provided by RSA. It consists of three parts: key generation, actual encryption, and decryption. A single modular exponentiation is used in both the encipherment and decipherment processes. Each node stores both an encryption public key (n,e) and a decryption private key (n,d).

3.12. Elliptic Curve Diffie-Hellman (ECDH)

Elliptic curves are incorporated into the Diffie-Hellman key exchange in this method. The standard ECDH goes as follows: When two nodes in a connected graph both land on the elliptic curve E (a,b) and the same point P, we have arrived at the initial state. A public key, P, is calculated by multiplying a secret pair of private keys, KA and KB. Once the public keys have been exchanged, the parties can generate a shared secret key by multiplying their private keys by the other's public key.

3.13. Hybrid Cryptography in WSNs

In hybrid cryptography systems, the benefits of both symmetric and asymmetric cryptosystems are integrated. Since asymmetric techniques are inefficient, they will only be used to trade a secret key that will be used in symmetric cryptography.

An Effective Algorithm for Hybrid Cryptography and Steganography-Based Privacy, Integrity, and Authentication

Biswas et al. (2019) present a system that uses AES-RSA IoT-based data security and key security in conjunction with LSB Steganography to protect an encrypted key from intrusion. So this technology guarantees privacy, security, and authenticity all at once.

4. RESULTS AND DISCUSSIONS

A wide range of encryption techniques are used by apps and services to safeguard user information. The development of revolutionary new technologies, however, is rendering these antiquated methods obsolete. Hardware improvements have greatly reduced the time required to crack a cryptographic system, while multiple attacks have rendered the existing systems more vulnerable.

As a result of crypto-analysis and other specialized mathematical attacks, cryptographers have found these systems to be extremely weak. Another weakness of current systems is the inability to properly secure crucial keys during storage and transmission. Maintaining optimal functionality is also crucial when it comes to protecting IoT-based data. Higher-security encryption methods typically employ longer key lengths, which reduces the efficiency of the system.

Key security and IoT-based data security leaks can both be compromised by a single-layered, stand-alone crypto-system. Data security is often compromised by a single system's flaws. Sometimes the speed and efficiency are compromised by the many problems inherent in stand-alone systems. As a result, there is an increasing need for a system that can compensate for the performance-security trade-offs inherent in the employment of individual cryptographic algorithms.

To resolve these problems, a combined strategy is more important than ever. The suggested system combines three of the most powerful and widely used algorithms for data encryption. Algorithms for Asymmetric Cryptography There are a number of symmetric algorithms, such as RSA, AES, and Blowfish. Over the Internet, and more specifically at the Transport Layer Security (TLS) level, RSA is one of the most used asymmetric encryption algorithms used for a broad variety of purposes in addition to data encryption. However, Blowfish and AES are examples of Symmetric Ciphers, which encrypt and decrypt data with the same key.

This study's proposed solution employs a layered encryption architecture to encrypt IoT-based data security three times over with the aforementioned three techniques. Steganography is employed to encrypt the keys and store them in an image for further safety. The password hash is then used as the key for AES encryption, with the hash itself encrypted using SHA-1. Experimental results show that the suggested Python-based system is an effective cryptosystem for protecting IoT-based data.

4.1. Experimental Results

All of the code for this study was written in Python using the following architecture.

The hybrid system takes the plaintext or IoT-based data security and encrypts it three times with the Blowfish, RSA, and AES algorithms in a cascade fashion. The utilized keys are kept in an AES-encrypted list, with the key being produced through SHA1 hashing a user-supplied password. There are two parts to the system:

4.2. Data Encryption

The IoT-Based Data Encryption Scheme is depicted in Figure 7. The system is composed of a key generator, a set of keys, and three distinct layers of encryption. A random n-bit key is generated by the Key Generator in accordance with the encryption technique, and this key is recorded in the List of Keys. The plaintext P is encrypted with a key size of either 32

RESEARCH ARTICLE

bits, 64 bits, or 128 bits using the Blowfish Algorithm. The key $KBlowfis$ is generated by the key generator for Blowfish Encryption. Then, it's added to L's List of Keys. In order to create Cipher text C1, the Plaintext, P, must first be encrypted as in Eq (6) and (7).

$$c_1 = Blowfish(P\ plaintext = P;\ Key = K_{Blowfish}) \ldots . (6)$$

$$L = [\,]\ KBlowfish \ldots (7)$$

The RSA Encryption algorithm is applied to the cipher text C1 using the public key K RSA Public, which has a bit length of 1024 or 2048 and was generated by the key generator. A separate private key, $K_{RSA_{Private}}$) , is produced for use in decryption. Even if the public key used for encryption is not stored, the resulting private key is added to the List of Keys. Encrypting C1 is necessary for generating Cipher Text C2 as shown in Eq (8)-(12).

$$c_2 = RSA(plaintext = c_1;\ Key = K_{RSA\ Public}) \ldots (8)$$

$$L = [\,KBlowfish\,]\ KRSA\ \ Private \ldots . (9)$$

The Cipher text, C2 is then encrypted using AES-128 Encryption with the 128 Bit, KAES generated by the Key Generator. The Key, KAES generated is appended to the List of Keys, L. This Step gives the final encrypted cipher text C.

$$Ciphertext;\ C = AES(Plaintext = c_2;\ Key = K_{AES}) \ldots (10)$$

$$L = [\,KBlowfish\,;\ KRSA\ \ P\ private\,]KAES \ldots (11)$$

The system's output is the Cipher text, C, and the list of keys L with all the keys.

$$List\ of\ Keys;\ L = [\,KBlowfish\,;\ KRSA\ \ P\ rivate\,;\ KAES\,] \ldots (12)$$

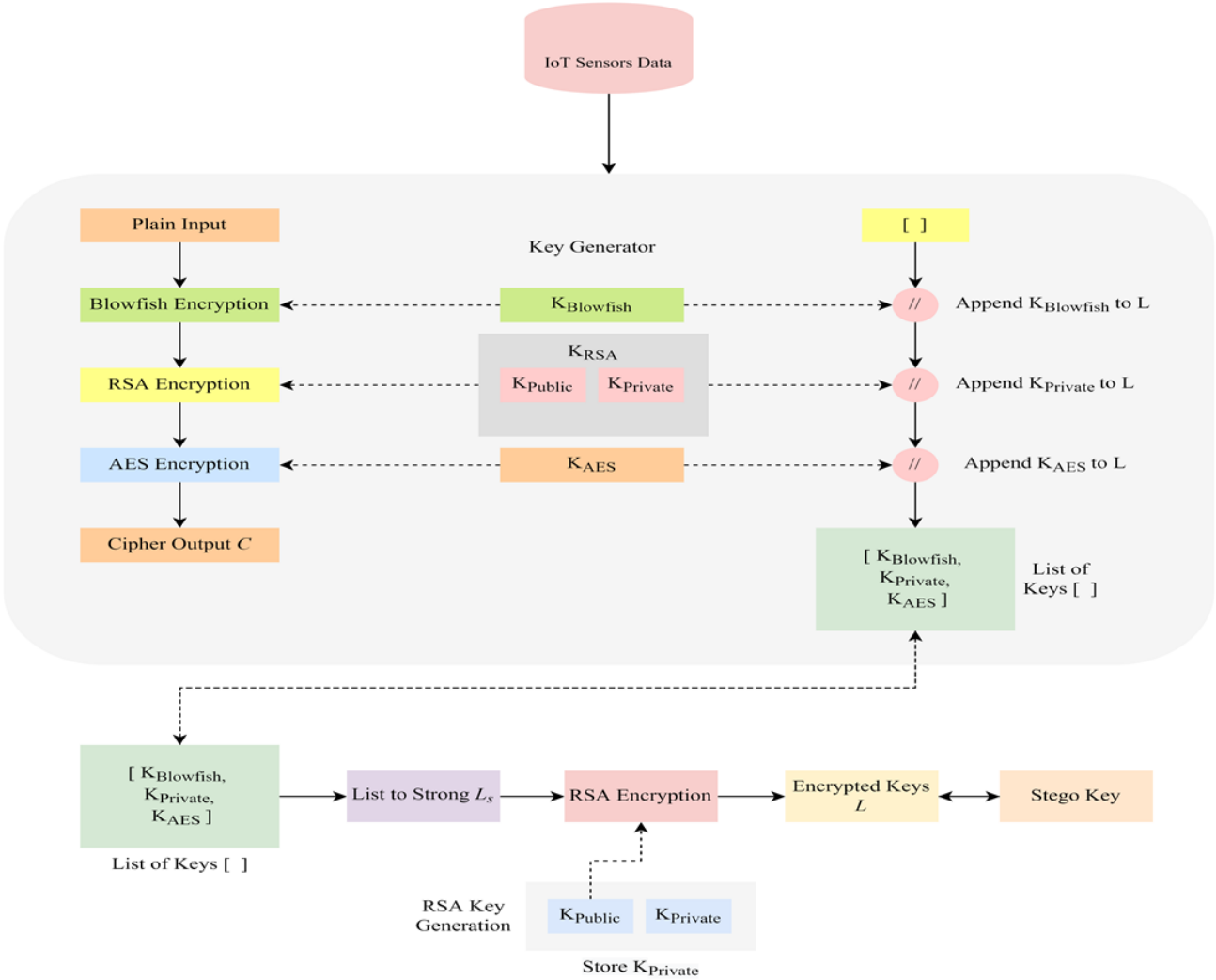

Figure 5 Proposed IoT-Based Data Encryption Scheme

RESEARCH ARTICLE

4.3.

Key Encryption

The suggested approach allows for the safe storage of the Keys required for encryption at each stage. All of the keys produced throughout the IoT-based data encryption process are stored in the List of Keys, L. The List of Keys, L, is updated each time a key for a specific encryption layer is generated.

The system uses the encryption layers Blowfish, RSA, and AES, therefore the keys are kept in the same order as shown in Eq (13)-(18):

$$List\ of\ Keys;\ L = [\ KBlowfish\ ;\ KRSA\ \ P\ private\ ;\ KAES\ ]\ \ldots(13)$$

This List, L is then passed into a function that converts the list into a single string of keys separated by separators (x ,$*$,/)

$$LS\ =\ Stringify(\ L;\ separator\ =\ 0{,}0\ )\ \ldots(14)$$

Then, using a Key created from the user-provided password, the String, LS is encrypted using the AES Encryption Algorithm. The user enters a password, PW, which is hashed using SHA1, and the key, KPassword, is created using the first 16 bits of the hash. The encrypted string LS Encrypted is created using the encryption key, KPassword.

$$HashedPassword;\ H_p\ =\ SHA(P_W\ )\ \ldots(15)$$

$$Key;\ K_{Password}\ =\ H_P\ [0:\ 16]\ \ldots(16)$$

$$LS\ \ Encrypted\ \ =\ \ AES(LS;\ KP\ password)\ \ldots(17)$$

This Encrypted string is then embedded into a Cover Image using the Least Significant Bit of Steganography, giving the embedded Stego-Image.

$$Stego\ Image = LSBSteganography\left(L_{S\ ,Encrypted}\ ;\ CoverImage\right)\ \ldots(18)$$

The Stego-Image is transferred to the receiver along with the Encrypted Data.

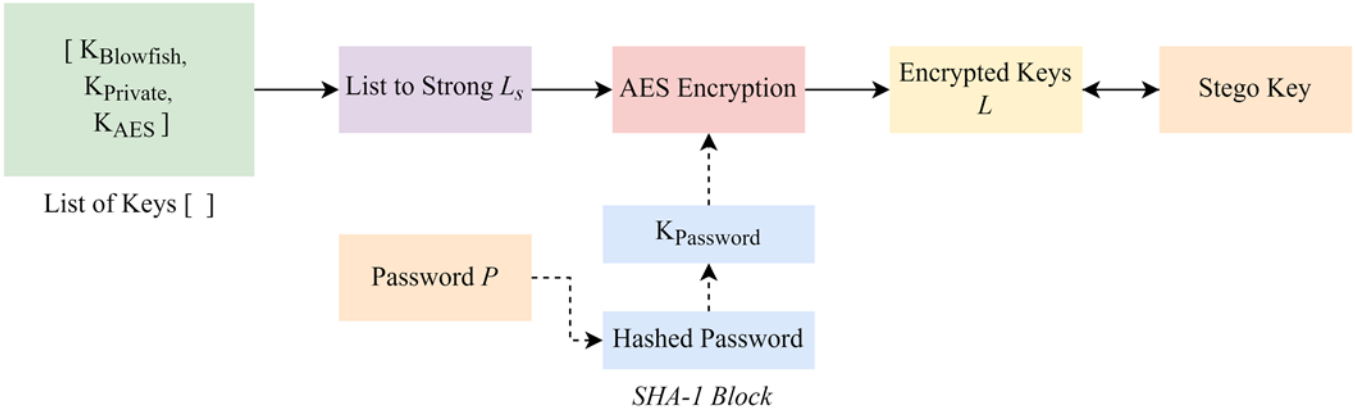

Figure 6 Proposed Key Encryption Scheme

```
Enter Plaintext: Hello, World! This is 2021!
Enter Password: enc2021
Ciphertext:
8afd0bbae83aff941a6c850d49a49bad5082f02bb6d9985c07efbab14c90dba02bc41f644553fa3b83e01b08f3b000d36ff82a32f9cc110bd2d30e710e20cd0
aafa18f562a3cd58b6e8c39e14a88ec00c90949d43b07918f2d6519bad3894ca3c68adf8a79384922b353f8ebd1653cfa7eb894136a7066562f49624929fcea
6cc65c809e7547a9cbe2f9c15444b9c4276798ace196932e73ade9abfc5ffa9e68646238de55d703d0694135353907c4e2beed80c9fdc0094a03ca0934db298
44ea90c6f65006ed053a0d612c9b921c6f3c2a06fca7ad261e7e89fe6f3bb0f42519e6397f8c025284d6ad79c21351768b3f44c1792ca8848a8c67695d126d3
5aea2a142cc7d73ec7e2297e96fe17fdec9d
Encryption Complete!
Encryption Metrics:
Length of Plaintext   :  54
Length of Ciphertext  :  544
Length of Password    :  7
Encryption Time (sec) :  0:00:00.959421
```

Figure 7 Output

RESEARCH ARTICLE

Figure 8 shows the Proposed Key Encryption Scheme. Python was used in the development of the Proposed Hybrid Crypto-system, and testing was conducted on a Windows machine equipped with an Intel i3 processor and 4 GB of RAM. The following plaintext has been encrypted to show how data encryption works. 'enc2021' was used as the password to encrypt the keys. The following findings emerged (Figure 9). Figure 9 shows the Output. The length of the cipher text is found to be ten times that of the plaintext. We can deduce that the system can encrypt the IoT-based data security to a form considerably different from the original plaintext because the plaintext and cipher text differ by a large margin and are completely random.

4.4. Performance Analysis of the Proposed System

For performance analysis, Files of different sizes and types were encrypted using the proposed system with the same password for every file. The password used was "enc2021". Figure 10 shows the Graph of Performance Analysis on Various Systems. Table 3 shows the Performance Analysis on Various Files.

Table 4 shows the Aggregate Performance Analysis. We find that the Encryption Rate is typically greater than the Decryption Rate. Research has demonstrated that cryptosystems with a longer decryption time are more secure since they are harder to crack. Therefore, the suggested cryptosystem is both secure and efficient in encrypting data due to its fast decryption time and rate.

Table 3 Performance Analysis on Various Files

File Type	File Size	Encryption Time	Decryption Time
Markdown File (.md)	7	17.24	1.81
MS-Excel Spreadsheet (.xlsx)	13	19.19	2.14
Portable Network Graphics (.png)	92	18.69	6.28
MS Word Document (.docx)	118	20.75	7.31
Image File (.jpg)	247	19.06	14.26
MS-PowerPoint Presentation (.pptx)	331	20.09	18.25
RAR Archive (.rar)	658	30.37	34.73
Photoshop Editable (.psd)	745	23.96	41.92
Archive File (.zip)	920	26.65	48.1
Font File (.otf)	4134	88	301
Video File (.mp4)	4980	69	254
Photoshop Editable (.psd)	9544	270	779
Audio File (.mp3)	15475	249	1162

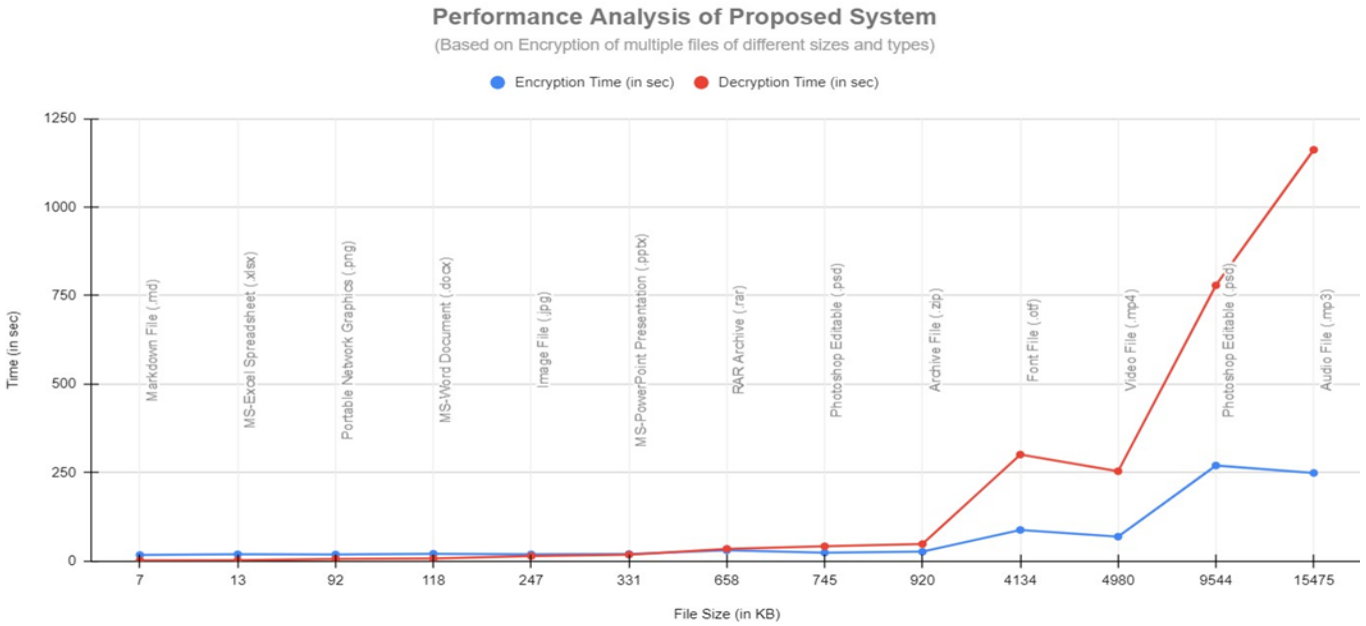

Figure 8 Graph of Performance Analysis on Various Systems

RESEARCH ARTICLE

Table 4 Aggregate Performance Analysis

Total Size of Files	Average Encryption Time	Average Decryption Time	Average Encryption Rate	Average Decryption Rate
(in KB)	(in sec):	(in sec):	(in KB/sec):	(in KB/sec):
37264	67.08	205.45	26.54	14.69

4.5. Comparison with Existing Systems

As a result of comparison tests, the Proposed Cryptosystem is found to be superior to both the Hybrid system (AES-RSA) and the Standalone system (Blowfish). The following are the summative findings from our comparative tests. Table 5 shows the Performance Comparison – Summary.

Table 5 Performance Comparison – Summary

Proposed Hybrid	Average Encryption Time	Average Decryption Time	Average Encryption Rate	Average Decryption Rate
Crypto-system (Blowfish-RSA-AES)	140.53	508.8	50.206	15.6
Standalone				
Crypto-system (Blowfish)	122.6	231.2	53.74450004	34.0260998
Hybrid				
Crypto-system (RSA-AES)	121.4	367	55.10561916	23.45880341

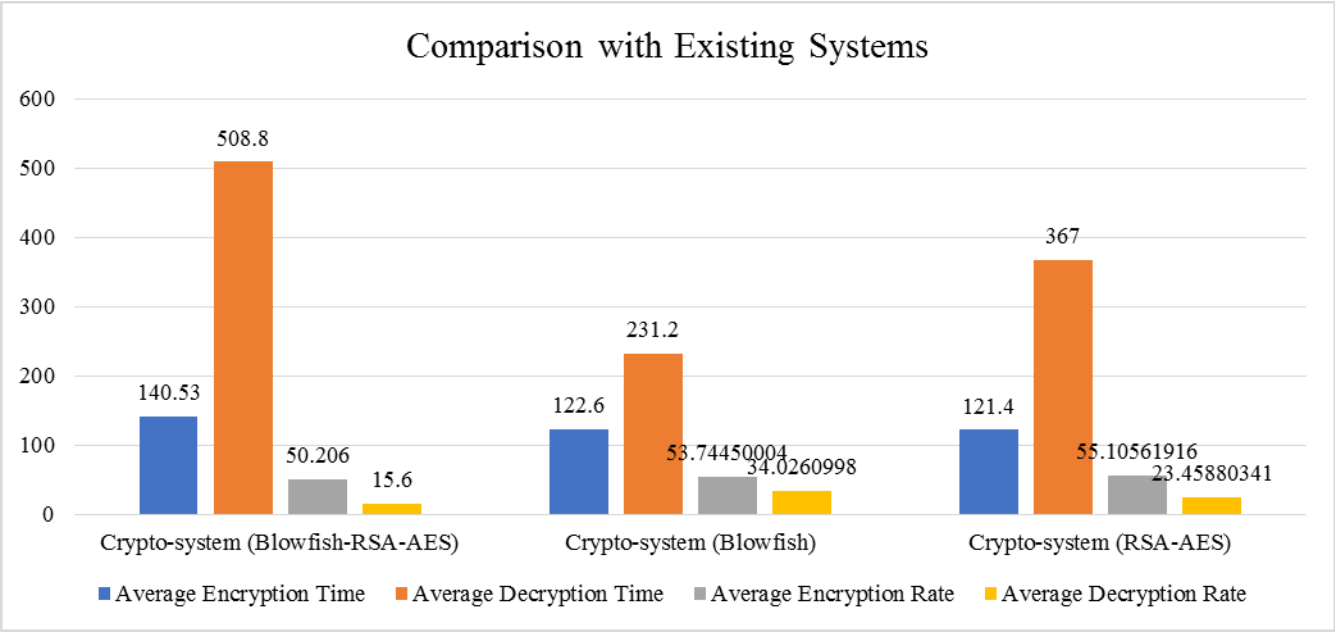

Figure 9 Performance

4.6. Complexity Analysis

Figure 11 shows the Comparison with Existing Systems. Symmetric Encryption: The key size and block size affect the complexity of symmetric encryption methods like AES. For instance, the key and block sizes of AES-128 are both 128 bits, while those of AES-256 are both 256 bits and 128 bits, respectively. The number of rounds needed for encryption and decryption, which varies with the size of the key, is a major factor in AES's complexity. It is generally agreed that AES

RESEARCH ARTICLE

has an O(n) time complexity, where n is the number of rounds.

Asymmetric Encryption: The difficulty of asymmetric encryption methods, like RSA, is proportional to the size of the key and the number of bits in the modulus. RSA encryption and decryption have an O(n2) time complexity, where n is the number of bits in the modulus. As a result, RSA's complexity can grow noticeably as key sizes get larger.

Message length and the number of iterations through the hash function determine the difficulty of hashing algorithms like SHA-1. In most cases, SHA-1's time complexity is estimated to be O(n), where n is the number of rounds.

Hybrid Encryption: The complexity of hybrid encryption techniques is a function of the underlying symmetric and asymmetric encryption algorithms. The suggested solution employs a hybrid Encryption based on the temporal complexity of both AES and RSA.

Key generation, key exchange, and data transmission all add to the difficulty of the suggested system, making it more than just the complexity of cryptographic methods. Therefore, these elements should be included in a full complexity study of the proposed approach.

5. CONCLUSIONS

The Decryption Time for the Proposed System is much longer than that of the other two systems, showing that it is difficult to break and will require more processing power and time than the other two systems, making it highly secure. In order to keep information safe, the suggested cryptosystem employs both symmetric and asymmetric cryptography. Encryption keys are encrypted via a new subprocess introduced by the system and then embedded in an image. Combining Blowfish, RSA, and AES has vastly strengthened security and eliminated weaknesses seen in individual systems. The technique also aids in boosting security without requiring longer keys. The lengthy decryption time demonstrated by the tests also indicates that the system is resistant against brute-force attacks. A high level of security is also ensured by the exponential growth of cipher text from plaintext. Although the system accomplishes its goal, it needs tweaks before it can be widely used. The proposed system has excellent safety and reliability. It's been shown to successfully encrypt IoT-based data security and safeguard keys. However, it was also discovered that the encrypted files are typically two to three times the size of the original file, despite the fact that they are both efficient and safe. That's why storing an encrypted file requires a lot of extra room. This shortcoming can be fixed by conducting additional research and making adjustments to the proposed system. Reducing the time required for encryption and decryption is another area for enhancement. Analyzing the effects of rearranging the order of the three algorithms is another avenue for exploring the suggested system. Alternately, one can increase performance by studying a slightly different variant by swapping out one of the algorithms.

REFERENCES

[1] Kundu, A., Das, J. C., De, D., & Debnath, B. (2022). Reversible Vigenere Cryptographic Cipher in Quantum-Dot Cellular Automata. In 2022 IEEE International Conference of Electron Devices Society Kolkata Chapter (EDKCON).

[2] Dorobantu, O. G., Apostol, A. G., & Datcu, O. (2022). The poly-alphabetic substitution ciphers - a viable solution for IoT applications? In 2022 International Symposium on Electronics and Telecommunications (ISETC).

[3] Jassim, S. A., & Farhan, A. K. (2021). A Survey on Stream Ciphers for Constrained Environments. In 2021 1st Babylon International Conference on Information Technology and Science (BICITS).

[4] Iqbal, M., Velasco, L., Costa, N., Napoli, A., Pedro, J., & Ruiz, M. (2022). LPsec: a fast and secure cryptographic system for optical connections. Journal of Optical Communications and Networking, 14(4).

[5] Vignesh, R. S., Chinnammal, V., Gururaj.D, Kishore Kumar, A. T. A., & Karthikeyan, K. V. (2023). Secured Data Access and Control Abilities Management over Cloud Environment using Novel Cryptographic Principles. In 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI).

[6] Windarta, S., Suryadi, S., Ramli, K., Lestari, A. A., Wildan, W., Pranggono, B., & Wardhani, R. W. (2023). Two New Lightweight Cryptographic Hash Functions Based on Saturnin and Beetle for the Internet of Things. IEEE Access, 11.

[7] Hariss, K., & Noura, H. (2022). ACiS: Lightweight and Robust Homomorphic Block Cipher Additive Scheme. In 2022 International Wireless Communications and Mobile Computing (IWCMC).

[8] Datta, B. P. R. V., & Sreehari, K. N. (2023). FPGA Implementation of Different Layers of Present Cipher. In 2023 3rd International Conference on Intelligent Technologies (CONIT).

[9] Baksi, A., Kumar, S., & Sarkar, S. (2022). A New Approach for Side Channel Analysis on Stream Ciphers and Related Constructions. IEEE Transactions on Computers, 71(10).

[10] Liu, Y., He, J., Ma, H., Qu, T., & Dai, Z. (2022). A Comprehensive Evaluation of Integrated Circuits Side-Channel Resilience Utilizing Three-Independent-Gate Silicon Nanowire Field Effect Transistors-Based Current Mode Logic. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(10).

[11] Hussain, S., & Mohideen, P. M. S. (2023). Advanced Machine Learning Approach for Suspicious Coded Message Detection using Enigma Cipher. In 2023 Second International Conference on Electronics and Renewable Systems (ICEARS).

[12] Irodah, R. M., & Adriansyah, A. (2022). Analysis and Design of Self-service Local Water Company (LWC) using Vernam Cipher Cryptography Algorithm. In 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS).

[13] Velasco, L., Iqbal, M., & Ruiz, M. (2023). Secure Optical Communications Based on Fast Cryptography. In 2023 23rd International Conference on Transparent Optical Networks (ICTON).

[14] Parida, D., & Bhanja, U. (2023). Smart Meters: Cyber Security Issues and Their Solutions. In 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN).

[15] Upadhyaya, D., Gay, M., & Polian, I. (2023). LEDA: Locking Enabled Differential Analysis of Cryptographic Circuits. In 2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST).

[16] Iqbal, M., Ari Laksmono, A. M., Prihatno, A. T., Pratama, D., Jeong, B., & Kim, H. (2023). Enhancing IoT Security: Integrating MQTT with ARIA Cipher 256 Algorithm Cryptography and mbedTLS. In 2023

RESEARCH ARTICLE

International Conference on Platform Technology and Service (PlatCon).
[17] Garcia, D., & Liu, H. (2021). A Study of Post Quantum Cipher Suites for Key Exchange. In 2021 IEEE International Symposium on Technologies for Homeland Security (HST).
[18] Salman, R. S., Farhan, A. K., & Shakir, A. (2022). Lightweight Modifications in the Advanced Encryption Standard (AES) for IoT Applications: A Comparative Survey. In 2022 International Conference on Computer Science and Software Engineering (CSASE).
[19] Segala, A. (2022). Essential Cryptography for JavaScript Developers: A practical guide to leveraging common cryptographic operations in Node.js and the browser. Packt Publishing.
[20] Rajashree, S., Vineetha, B., Mehta, A. B., & Honnavalli, P. B. (2022). Homomorphic Encryption Approach for String Concatenation. In 2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA).
[21] Unger, W., Babinkostova, L., Borowczak, M., & Erbes, R. (2021). Side-channel Leakage Assessment Metrics: A Case Study of GIFT Block Ciphers. In 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI).
[22] Jassim, S. A., Farhan, A. K., & Radie, A. H. (2021). Using a Hybrid Pseudorandom Number Generator for Cryptography in the Internet of Things. In 2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA).
[23] Dar, M. A., Askar, A., & Bhat, S. A. (2022). Blockchain based Secure Data Exchange between Cloud Networks and Smart Hand-held Devices for use in Smart Cities. In 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC).
[24] Biesmans, J., Regazzoni, F., & Mentens, N. (2023). Application-specific FPGAs: cryptographic agility through customized reconfigurable architectures. In 2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW).
[25] Umamaheswari, S., Vishal, N. R., Pragadesh, N. R., & Lavanya, S. (2023). Secure Data Transmission using Hybrid Crypto Processor based on AES and HMAC Algorithms. In 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication and Automation (ICAECA).

Author

Mohammed Naif Alatawi is currently working as an Assistant Professor in University of Tabuk, Saudi Arabia. He has completed his Ph.D. in Computer Information Systems from Nova Southeastern University USA in 2019 and Masters from Florida Institute of Technology USA in 2015. His research interests includes Computer Security, Software Engineering, Databases, and IoT.

How to cite this article:

Mohammed Naif Alatawi, "A Hybrid Cryptographic Cipher Solution for Secure Communication in Smart Cities", International Journal of Computer Networks and Applications (IJCNA), 10(5), PP: 776-791, 2023, DOI: 10.22247/ijcna/2023/223423.