International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 05 | May 2026 www.irjet.net p-ISSN: 2395-0072

A Layered for Protecting Data Using SM4 Ciphering and Image-based Concealment with Independent Key Delivery

Madhushree K S[1], Buvaneshwari S D[2], Sumathi A[3]

[1]*Department of Electronics and Communication Engineering, BNMIT, Bengaluru, India.*
[2]*Department of Electronics and Communication Engineering, BNMIT, Bengaluru, India.*
[3]*Department of Electronics and Communication Engineering, BNMIT, Bengaluru, India.*

Abstract - *In today's digital world, a large amount of sensitive information is shared through communication systems. However, even with existing protection methods, certain patterns in data transmission can still be identified and misused by attackers. To address this issue, this work proposes a two-stage security approach where encryption and data hiding are combined rather than treated as separate processes. In this system, the data is first encrypted using the SM4 block cipher and then embedded into a carrier image, making the information less noticeable during transmission. An additional processing layer is introduced to further enhance security, resulting in multiple levels of protection against unauthorized access. For secure key management, Zigbee communication is used along with an ESP32 microcontroller to enable local key transfer. At the same time, IoT integration allows keys to be delivered in real time through Telegram, ensuring both flexibility and security. The overall system is implemented using Python for encryption and steganography, supported by hardware-based key transmission. The results show that the proposed method improves confidentiality and data integrity, while also providing better resistance to threats such as eavesdropping and steganalysis. At the same time, it maintains low computational complexity. A key advantage of this approach is not only the strength of encryption, but also the ability to conceal the very presence of the data during transmission.*

Key Words: Dual-Layer Security, SM4 Encryption, Image Steganography, IoT-Based Key Exchange, Zigbee Communication, ESP32 Microcontroller, Secure Data Transmission.

1. INTRODUCTION

Data is continuously transmitted across modern communication networks, often passing through multiple intermediate points where complete control over its security cannot be guaranteed. In such environments, the challenge is not limited to preventing unauthorized access; it also involves reducing the likelihood of drawing attention to the data itself. Conventional cryptographic techniques are effective in protecting the content of information, but they do not hide the existence of encrypted data. This visibility can make the data a target for interception or further analysis. On the other hand, steganography focuses on concealing information within digital media, yet it may become vulnerable if the hidden data is discovered and extracted. When applied independently, both approaches reveal certain limitations in practical scenarios.

To overcome these limitations, a dual-layer secure data transmission framework is introduced in this work. The approach combines symmetric encryption with steganographic techniques to strengthen overall security. Initially, the data is encrypted using the SM4 block cipher, which operates on a 128-bit key and is designed to provide efficient and reliable encryption. The encrypted output is then embedded across multiple images using a cross-image steganographic method, thereby distributing and concealing the data within different carriers. This layered process ensures that even if the presence of hidden data is detected, the underlying information remains protected in its encrypted form.

In addition to safeguarding the data itself, the system addresses the critical aspect of key management. A separate communication mechanism based on IoT technology is used to transmit encryption keys securely. Zigbee protocol, along with an ESP32 microcontroller, facilitates localized wireless key transfer through an independent channel. This separation between data transmission and key distribution reduces the chances of both being intercepted simultaneously. Furthermore, IoT connectivity enables timely delivery of keys through secure messaging platforms, improving both accessibility and reliability.

The overall framework is designed to be lightweight and adaptable, making it suitable for environments where computational resources may be limited. Its structure supports scalability while maintaining cost efficiency, which makes it applicable in areas such as defense communication, healthcare data exchange, and financial transactions. By integrating encryption, steganographic data hiding, and independent key transmission, the proposed system enhances resistance to common security threats while ensuring efficient operation in real-world conditions.

2. LITERATURE SURVEY

Cryptography plays a central role in securing modern data communication by ensuring confidentiality, integrity, and authentication. Among the various techniques, symmetric key encryption is widely preferred because of its speed and efficiency, making it suitable for real-time applications. This is especially important in environments such as embedded systems and IoT devices, where memory and processing capabilities are limited. As a result, lightweight block ciphers have become increasingly relevant in recent research [2].

Many studies have explored ways to design and implement symmetric encryption algorithms that can handle large volumes of data while maintaining low latency. One such algorithm is SM4, a modern block cipher known for its strong resistance to cryptanalytic attacks. At the same time, it remains computationally efficient, which makes it well-suited for secure communication in resource-constrained systems. Its fixed block size and key length also simplify implementation in embedded platforms [1], [3].

To further improve security, different modes of operation are commonly used along with block ciphers. Techniques such as Electronic Codebook (ECB) and Cipher Block Chaining (CBC) help strengthen encryption. In practical scenarios, chaining methods like CBC are often preferred because they prevent the repetition of patterns in encrypted data, which could otherwise reveal useful information to attackers [2].

However, despite these advantages, relying solely on cryptography presents certain challenges. One of the major concerns is secure key exchange. If the encryption key is intercepted during transmission, the entire system becomes vulnerable. In many real-world applications, the effectiveness of encryption depends heavily on how securely the key is managed [10]. Another limitation is that encryption does not hide the existence of the data itself. Encrypted data can still attract attention and may be targeted for analysis or attack.

These issues highlight the need for additional layers of security beyond traditional cryptographic methods, particularly approaches that can both protect the content and conceal its presence.

Steganography plays an important role in improving data security by hiding confidential information in such a way that its presence is not obvious. Among different approaches, image-based steganography is widely preferred because digital images contain a large amount of redundant data, allowing extra information to be embedded without noticeable changes.

One of the most commonly studied techniques is Least Significant Bit (LSB) substitution. It is popular mainly because it is simple to implement and requires very little computational effort [4]. However, as research has progressed, more advanced approaches have been explored. Methods based on transform domains, such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), offer better resistance to common image processing operations like compression and filtering [5].

Recent developments have also introduced adaptive and multi-layer embedding techniques. In these methods, the secret data is spread across different parts of the image, making detection more difficult. Another approach, known as cross-image steganography, involves distributing encrypted data across multiple images or embedding it in layers within a single image. This increases the difficulty for attackers trying to retrieve the hidden information [6].

Despite these improvements, steganography still has several limitations. Techniques like LSB substitution are particularly vulnerable to statistical and visual analysis [4]. Although transform-domain methods provide better security, they come at the cost of increased computational complexity [5]. Multi-layer and cross-image techniques can also introduce synchronization problems, especially during data retrieval. In addition, many existing systems do not effectively combine steganography with strong encryption methods, which reduces the overall level of security.

To address the limitations of using cryptography or steganography alone, researchers have increasingly explored hybrid approaches that combine both techniques. In such systems, the data is first encrypted and then embedded into a cover medium, ensuring not only confidentiality but also concealment of its existence [7].

This dual-layer approach significantly improves protection against unauthorized access. Even if the hidden data is detected, interpreting it remains difficult without the correct decryption key. Some advanced methods go further by applying multi-level embedding, where an already encrypted image is hidden within another image, adding an extra layer of security [8], [9].

Comparative studies indicate that these hybrid techniques perform better than standalone methods, particularly in resisting various types of attacks. As a result, they are well-suited for sensitive applications such as defense systems and financial data transmission, where strong security is essential [7].

However, these advantages come with certain challenges. Hybrid systems often involve higher computational complexity and increased processing time. Proper coordination between encryption and embedding stages is critical, yet this aspect is not always carefully handled in existing models. In addition, many approaches do not

focus on optimizing resource usage, which limits their practicality in real-time environments and IoT-based systems. There is also a noticeable lack of standardized evaluation methods, leading to inconsistencies in how performance is measured and compared across different studies.

3. METHODOLOGY

The proposed system is designed as a dual-layer secure data transmission framework that combines encryption, data hiding, and IoT-based key exchange into a unified workflow. Instead of handling these processes as a single unit, the system is organized into separate functional stages, each responsible for a specific task such as data protection, communication, and key management.

At the sender side, sensitive data is not transmitted directly. It is first encrypted using the SM4 symmetric encryption algorithm, transforming the original information into an unreadable form. This encrypted data is then embedded into a digital image using a cross-image steganography technique, allowing the data to remain hidden without raising suspicion.

To further strengthen security, the process does not end here. The resulting stego-image is encrypted once again. This second layer of encryption is added deliberately to increase the difficulty of unauthorized access, even if one layer is somehow exposed.

The system uses two separate communication paths for transmission. The encrypted stego-image is sent through a standard internet-based channel, such as email or file-sharing platforms. Meanwhile, the encryption key is transmitted independently using Zigbee wireless communication. Separating these two components reduces the likelihood of both being intercepted at the same time.

On the receiver side, the Zigbee module interfaces with an ESP32 microcontroller, which facilitates secure handling of the received key. Instead of displaying the key directly, it is forwarded to the intended user through a Telegram notification. After obtaining both the encrypted image and the corresponding key, the receiver carries out the reverse operations to recover the original data.

Although this dual-path approach does not completely eliminate potential risks, it significantly increases the complexity involved in unauthorized data reconstruction.

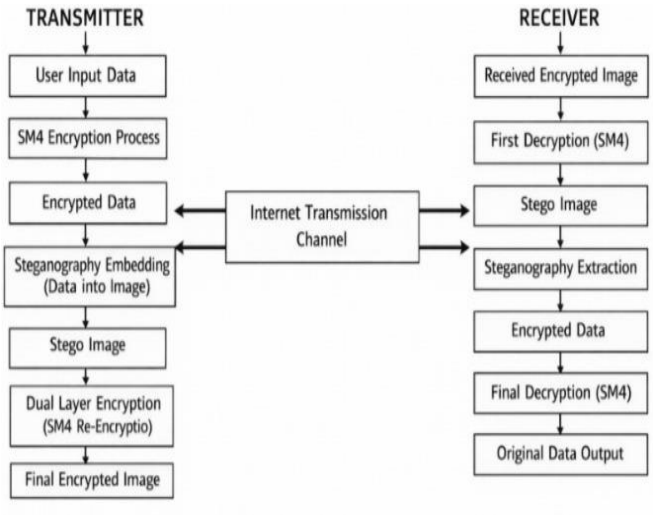

Fig-1: Overall System Architecture of Dual-Layer Secure Data Transmission Framework

As illustrated in Fig-1, the process starts with user-provided input, which may consist of sensitive information. Rather than transmitting this data directly, it is first encrypted using the SM4 algorithm, ensuring an initial level of security.

Once encrypted, the data is embedded into a cover image through a steganographic technique. This step does more than protect the data it conceals its very existence by hiding it within another medium, reducing the chances of detection during transmission.

To strengthen security further, the resulting stego-image undergoes an additional layer of encryption. This creates a dual-layer protection mechanism, where both the visibility of the data and its readability are safeguarded.

The fully encrypted image is then transmitted over a public network. Meanwhile, the encryption key is sent separately using Zigbee modules. On the receiving side, the Zigbee unit transfers the key to an ESP32 controller, which then delivers it to the user via a Telegram interface.

After receiving both the encrypted image and the key, the receiver follows a sequential recovery process. The outer encryption layer is removed first, allowing access to the stego-image. The hidden data is then extracted, and finally, SM4 decryption is applied to retrieve the original information.

Although this multi-step recovery process adds some complexity, it significantly improves security by reducing the risk associated with partial data interception.

The proposed system operates through a sequence of well-defined steps, although some of these processes remain transparent to the end user.

To begin with, the input data is converted into binary format and aligned with the block size required for SM4 encryption. A 128-bit secret key is generated using a randomization method, ensuring unpredictability. The data is then processed through multiple rounds of transformation as specified by the SM4 algorithm.

Once encryption is complete, the resulting ciphertext is embedded into a cover image using a pixel-level steganographic technique. This step is handled carefully to avoid noticeable changes in the image, thereby maintaining its visual quality and preventing suspicion.

Rather than sending this stego-image directly, an additional layer of security is applied by encrypting it again. Depending on the system design, this second layer may use either the same key or a modified version derived from it.

For secure key transmission, the generated key is encoded and sent through Zigbee modules operating in transparent mode. On the receiving side, the key is retrieved and passed to an ESP32 module, which functions as the IoT interface. The ESP32 then delivers the key to the intended user via a Telegram API, restricting access to authorized recipients only.

During the recovery phase, the outer encryption layer is first removed from the received image. The embedded encrypted data is then extracted using the reverse steganography process. Finally, SM4 decryption is applied with the received key to obtain the original data.

To ensure reliability, the recovered data is validated using format checks or simple checksum techniques, confirming that no errors were introduced during transmission or processing.

The system presents a layered approach to secure data transmission by integrating encryption, data hiding, and IoT-based key exchange. Rather than depending on a single method, security is applied in multiple stages to strengthen overall protection.

Initially, the data is encrypted and then embedded within an image using steganography. To further enhance security, an additional encryption layer is applied. The encryption key is transmitted separately through a Zigbee-based channel, ensuring it is not exposed alongside the data.

On the receiver side, the ESP32 acts as a communication bridge by delivering the key through a Telegram interface. The original data can only be reconstructed when both the hidden encrypted data and the key are successfully received.

This multi-step process reduces both the visibility of the data and the risk of unauthorized access. While it does not guarantee complete security, it creates multiple levels of defense that make intrusion more difficult.

Overall, the system is suitable for scenarios where layered security is more effective than relying on a single protection technique.

4. RESULTS AND DISCUSSION

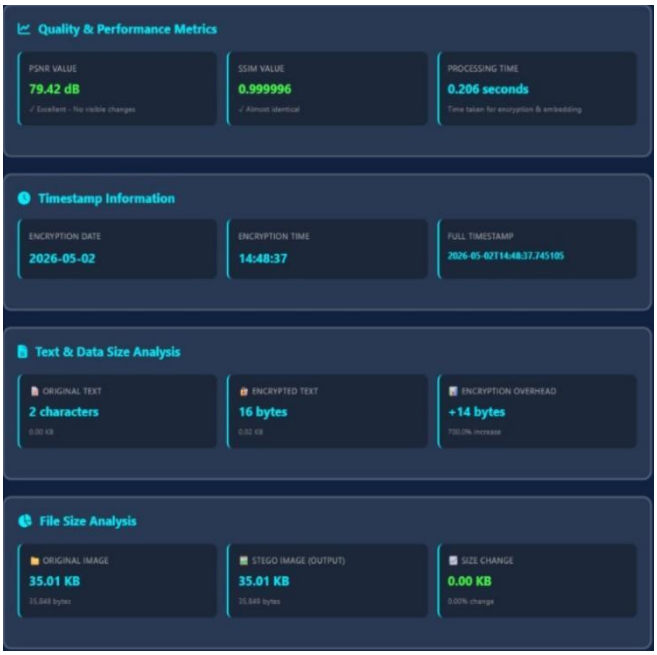

Fig -2: Comprehensive performance evaluation of the proposed system

To evaluate the performance of the proposed system in Fig-2, several experiments were carried out using text and image files of different sizes. The objective was to observe how consistently the system performed when handling varying amounts of data.

For text-based testing, file sizes ranged from 1 KB to 100 KB. For image-based testing, both PNG and JPEG formats were used, with resolutions starting from 256×256 pixels and extending up to 1024×1024 pixels. These variations helped in examining the system's behavior under different data loads and image capacities.

The software implementation was developed in Python, which was used to perform encryption, decryption, and steganographic embedding operations. On the hardware side, an ESP32 microcontroller was interfaced with Zigbee transceiver modules to enable secure transmission of encryption keys between sender and receiver units.

To provide immediate key delivery notifications, the Telegram API was integrated into the system. This allowed users to receive real-time updates whenever a key exchange was successfully completed.

All tests were conducted on a standard system equipped with an Intel i5 processor and 8 GB RAM. Since Telegram-based communication depends on network availability, a stable internet connection was maintained throughout the testing process. Zigbee communication trials were performed in a controlled indoor environment to ensure reliable signal transmission and to minimize interference during observation.

This setup provided a practical environment for evaluating the efficiency, reliability, and overall functionality of the proposed dual-layer secure transmission framework.

The performance of the system was assessed by considering execution time, accuracy of data recovery, and communication delay.

The time required for encryption was observed to increase gradually with the size of the input data. For smaller inputs of around 1 KB, the process took nearly 15 ms, while larger inputs of about 100 KB required up to 120 ms. A similar trend was seen during decryption, although it consistently took slightly less time, ranging from approximately 10 ms to 100 ms.

In terms of reliability, the system was able to recover the transmitted data without any loss. This was confirmed using checksum validation, ensuring that the received data exactly matched the original input.

Communication delay was also measured by considering the complete transmission process, including Zigbee-based key transfer and Telegram notification. The total latency varied between 1.5 and 3 seconds, depending on network conditions at the time of transmission.

Overall, the system demonstrates stable and efficient performance, making it suitable for real-time secure communication when handling moderate amounts of data.

The proposed framework strengthens data protection by applying multiple layers of security rather than relying on a single technique. This layered approach improves resistance to various types of attacks.

The use of the SM4 symmetric encryption algorithm provides a strong foundation for security. With a 128-bit key and 32 rounds of processing, it offers reliable protection against brute-force attempts and unauthorized access.

In addition to encryption, steganography is used to hide the encrypted data within image files. This makes the presence of sensitive information less obvious during transmission, thereby reducing the likelihood of detection.

The system also separates key transmission from the data channel. The encryption key is sent through a Zigbee-based communication path, supported by Telegram notifications. This separation minimizes the risk of both the data and key being intercepted together.

By combining encryption, data hiding, and independent key exchange, the framework enhances confidentiality, maintains data integrity, and improves overall system security.

The results show that combining cryptography with steganography strengthens overall data security without significantly increasing computational complexity. The SM4 algorithm provides efficient and lightweight encryption, making it suitable for systems with limited resources. At the same time, steganography introduces an additional layer by concealing the encrypted data, which helps reduce the chances of detection during transmission.

Another important aspect is the use of Zigbee for key exchange. By sending the encryption key through a separate communication channel, the system minimizes the risk of key interception.

In practical terms, this approach is well-suited for environments where both security and efficiency are critical. This includes applications in IoT systems, defense communication, and healthcare data transfer, where protecting sensitive information without overloading system resources is essential.

Table - 4.1: Comparison with Existing Methods

Feature	Existing Systems	Proposed System
Encryption	Single-layer (AES/RSA)	SM4 (Dual-layer)
Steganography	Not used	Used
Key Exchange	Internet-based	Zigbee + IoT
Data Concealment	No	Yes
Security Level	Moderate	High
Data Recovery Accuracy	High	100%

Most existing approaches focus only on a single layer of encryption. While this helps in protecting the content, it does not prevent others from noticing that sensitive data is being transmitted. The proposed system addresses this limitation by combining SM4 encryption with steganography, ensuring that the data is not only secured but also hidden from plain view.

In addition, the encryption key is sent separately using Zigbee and IoT-based communication, which further lowers the chances of interception. By distributing security across multiple steps, the system enhances

overall protection while maintaining reliable and accurate data recovery.

Even though the system performs well overall, a few practical limitations were noticed during testing:

Zigbee Range Constraint: Communication through Zigbee works reliably only within a limited distance, roughly up to 500 meters in open environments.

Internet Dependency: Since the key delivery relies on Telegram, a stable internet connection is necessary for proper operation.

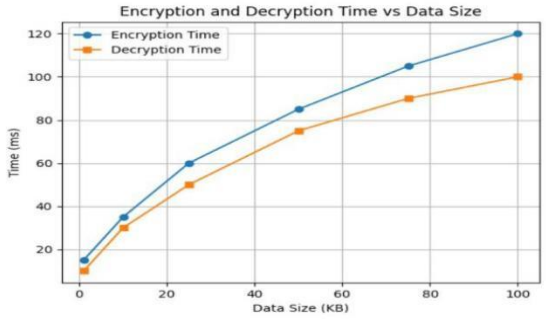

Chart -1: Graphical Representation of Time vs Data Size

The graph illustrates how execution time changes as the size of the input data increases for both encryption and decryption processes. Data size, measured in kilobytes, is shown along the horizontal axis, while the time taken for processing, in milliseconds, is represented on the vertical axis.

As the input size becomes larger, the processing time rises gradually. This pattern is expected since larger data requires more rounds of computation. It is also noticeable that encryption takes slightly more time than decryption, which can be attributed to the additional forward operations involved in the SM4 process.

The plotted lines remain smooth without major fluctuations, showing that the system behaves consistently across different data sizes. Such uniformity indicates that performance remains stable and predictable even as the workload increases.

In general, the outcome highlights that the system manages to provide stronger security while keeping the computational cost within a reasonable range, without introducing any major performance issues.

5. CONCLUSIONS

This work focuses on a practical concern in modern communication protecting data while also making it less visible during transmission. The proposed dual-layer framework brings together SM4 symmetric encryption, cross-image steganography, and an IoT-supported key exchange mechanism to address growing security risks in open networks. Instead of relying on a single method, the system combines encryption with data hiding so that sensitive information is not only secured but also concealed from potential attackers.

An additional encryption layer applied to the stego-image further strengthens the system, making it more resistant to both cryptographic attacks and steganalysis. This layered design improves overall robustness and ensures that even if one level is compromised, the data remains protected through other safeguards.

A key strength of the approach lies in separating data and key transmission. The use of Zigbee for local key transfer, along with ESP32-based IoT support and Telegram alerts, reduces the likelihood of key interception and adds reliability to the process. From the experimental results, the system achieves accurate data recovery with only a slight increase in processing time and acceptable latency, making it suitable for real-time use with moderate data sizes.

When compared to traditional single-layer methods, the proposed system offers better confidentiality, integrity, and concealment. That said, some practical challenges remain, such as the limited range of Zigbee communication and dependence on internet connectivity for certain operations.

Future improvements could focus on extending communication range, handling larger datasets more efficiently, and introducing stronger authentication techniques. Adapting the system to changing conditions in dynamic environments would also enhance its effectiveness. Overall, this framework provides a balanced and practical solution for secure data transmission, especially in distributed and resource-limited settings.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to B. N. M. Institute of Technology (BNMIT) for providing us with the opportunity, resources, and supportive environment required to carry out this research work successfully.

We extend our deepest thanks to our respected guide for their continuous guidance, valuable suggestions, and encouragement throughout the development of this work. Their insights and expertise have played a crucial role in shaping the direction and quality of our research.

REFERENCES

[1] X. Wang, Y. Zhang, and L. Chen, “Efficient Implementation of SM4 Block Cipher for Resource-Constrained IoT Devices,” IEEE Internet of Things Journal, vol. 9, no. 6, Mar. 2022, pp. 4512–4523.

[2] J. Liu, H. Wu, and Z. Qin, "Lightweight Symmetric Encryption Algorithms for Secure IoT Communication: A Comparative Study," IEEE Access, vol. 10, 2022, pp. 33721–33735.

[3] Y. Li and K. Chen, "Hardware Optimization of SM4 Encryption Algorithm for Embedded Systems," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 4, Apr. 2022, pp. 1802–1806.

[4] S. Roy and A. K. Das, "A Robust Image Steganography Technique Using LSB and Random Pixel Selection," IEEE Access, vol. 9, 2021, pp. 102345–102356.

[5] M. Sharma and P. Singh, "Secure Image Steganography Based on DWT and Encryption Techniques," in Proc. IEEE Int. Conf. Signal Processing and Communication (ICSPC), 2021, pp. 1–5.

[6] H. Kaur and R. Singh, "Cross-Image Data Hiding Using Multi-Layer Steganography for Secure Communication," IEEE Access, vol. 11, 2023, pp. 55678–55689.

[7] T. Nguyen, D. Nguyen, and M. Vo, "Hybrid Cryptography and Steganography Framework for Secure Data Transmission," IEEE Access, vol. 10, 2022, pp. 88901–88912.

[8] A. Verma and S. Kumar, "Dual-Layer Security Model Using AES and Image Steganography," in Proc. IEEE Int. Conf. Computing, Communication and Automation (ICCCA), 2022, pp. 1123–1128.

[9] P. R. Reddy and K. S. Rao, "Multi-Level Encryption and Adaptive Steganography for Secure Data Communication," IEEE Access, vol. 11, 2023, pp. 22345–22358.

[10] M. Alqahtani and F. Alharbi, "IoT-Based Secure Key Management Using Zigbee Communication Protocol," IEEE Internet of Things Journal, vol. 8, no. 14, Jul. 2021, pp. 11521–11530.