Comprehensive Journal of Science
Volume (10), Issue (39), (March, 2026)
ISSN: 3014-6266

مجلة العلوم الشاملة
المجلد (10) العدد 39 مارس 2026
ردمد: 3014-6266

Design and Implementation of a Secure Data Exchange System for Government Institutions Using AES, RSA, and TLS

Muner Athaba[1]*, Abdelatti Blg[2], Abdulraouf albaghdadi[3]

[1] Department of Computer Science, Faculty of Science, University of Zawia (ZU), Libya
[2]Department of Computer Science, Faculty of Science, University of Zawia (ZU), Libya
* Corresponding Author's Email: **m.athaba@zu.edu.ly**
Co-authors Email: a.blg@zu.edu.ly
Co-authors Email: a.alklabi@histr.edu.ly

تاريخ الاستلام:2026/01/03 –تاريخ المراجعة: 2026/01/23 – تاريخ القبول: 2026/01/31 – تاريخ للنشر:2026/03/06

Abstract
The rapid expansion of digital networks within government institutions has significantly increased the volume of sensitive data exchanged between distributed branches, thereby amplifying exposure to cybersecurity threats such as interception, manipulation, and impersonation attacks. Ensuring confidentiality, integrity, and authentication during data exchange has become a critical challenge. This paper presents the design and implementation of a secure data exchange system tailored for government institutions, based on the integration of Advanced Encryption Standard (AES), Rivest-Shamir-Adleman (RSA), and Transport Layer Security (TLS). The proposed system adopts a hybrid encryption approach, utilizing AES for efficient data encryption, RSA for secure key exchange and digital authentication, and TLS to establish a protected communication channel. The system is implemented using the Python programming language and evaluated through practical experiments measuring encryption performance, resource consumption, and resistance to common network attacks. Experimental results demonstrate that the proposed framework achieves a high level of security while maintaining acceptable performance, making it suitable for real-world deployment in governmental environments.

Keywords
Secure data exchange, Cryptography, AES, RSA, TLS, Government systems, Python

1. Introduction

The increasing reliance of government institutions on digital communication and networked information systems has transformed the way data is created, stored, and exchanged. Administrative records, financial transactions, and official communications are now routinely transmitted across internal and external networks. While this digital transformation improves efficiency and accessibility, it simultaneously exposes sensitive data to a wide range of cybersecurity threats, including eavesdropping, man-in-the-middle attacks, data tampering, and identity spoofing (Stallings, 2020; Behl and Behl, 2017).

Encryption has emerged as a fundamental pillar of information security, providing mechanisms to protect data confidentiality, ensure integrity, and authenticate communicating parties (Daemen and Rijmen, 2002; NIST, 2001). Modern secure systems rarely rely on a single cryptographic technique; instead, they integrate multiple algorithms and protocols to balance security and performance requirements. Symmetric encryption algorithms such as AES offer high efficiency for encrypting large volumes of data, whereas asymmetric algorithms such as RSA enable secure key distribution and digital authentication (Rivest et al., 1978; Boneh,

1999). In addition, security protocols such as TLS provide end-to-end protection for data transmitted over potentially insecure networks (Dierks and Rescorla, 2008; Rescorla, 2018).
Despite the availability of these technologies, many government institutions still lack a unified, practical, and deployable security framework that integrates symmetric encryption, asymmetric encryption, and secure communication protocols into a single coherent system. Existing solutions are often fragmented, complex to manage, or unsuitable for operational environments with limited technical expertise and strict administrative constraints (Kumar and Pais, 2018; Zhang et al., 2019). This gap highlights the need for an applied security model that can be realistically implemented within governmental infrastructures.
This paper addresses this need by proposing and implementing a secure data exchange system that integrates AES, RSA, and TLS within a unified architectural framework. The main contributions of this work can be summarized as follows:

- The design of a hybrid encryption framework tailored for secure inter-governmental data exchange.
- The practical implementation of the proposed system using Python and widely adopted cryptographic libraries.
- An experimental evaluation of system performance, efficiency, and security under realistic usage scenarios.

Although the cryptographic algorithms employed in this study namely AES, RSA, and TLS are well established, the novelty of this work lies in their systematic integration into an end-to-end, deployable secure data exchange framework specifically designed for governmental institutions. Unlike existing studies that primarily focus on algorithmic evaluation or isolated cryptographic components, this research presents a practical architecture that combines hybrid encryption, certificate-based authentication, and secure transport mechanisms within a single coherent system. Furthermore, the framework is implemented and experimentally validated, demonstrating its feasibility, scalability, and performance suitability for real-world governmental environments. This applied contribution bridges the gap between theoretical cryptographic research and operational e-government security systems.

2. Related Work

Extensive research has examined the application of cryptographic algorithms and secure communication protocols for protecting sensitive data in governmental, financial, and enterprise environments. Symmetric encryption, particularly the Advanced Encryption Standard (AES), is widely recognized as a global benchmark due to its strong security guarantees, high computational efficiency, and broad platform compatibility. AES was standardized by the National Institute of Standards and Technology and has since become the dominant algorithm for both data-at-rest and data-in-transit protection in modern secure systems (NIST, 2001; Daemen and Rijmen, 2002). Prior studies have demonstrated that AES offers high throughput and strong resistance to known cryptanalytic attacks, making it well suited for large-scale governmental applications where performance, scalability, and reliability are critical (Paar and Pelzl, 2014; Stallings, 2017).
Asymmetric cryptography plays a complementary role in secure communication systems, with Rivest–Shamir–Adleman (RSA) being one of the most extensively deployed public-key algorithms. RSA forms the foundation of many public key infrastructures by enabling secure key exchange, digital signatures, and authentication mechanisms (Rivest et al., 1978; Menezes et al., 2012). Its security relies on the computational hardness of large integer factorization, which has been thoroughly analyzed in cryptographic literature (Boneh, 1999; Katz and Lindell, 2020). However, several studies report that RSA introduces substantial computational overhead when applied to large data volumes, limiting its practical use to key encapsulation and authentication rather than bulk data encryption (Schneier, 2022; Ferguson et al., 2010). As

a result, RSA is commonly integrated with symmetric encryption algorithms to form hybrid encryption schemes.

Transport Layer Security (TLS) has emerged as the de facto standard protocol for securing data transmission over untrusted networks. TLS combines asymmetric cryptography for authentication and key exchange with symmetric encryption for efficient data transfer, while ensuring data integrity through cryptographic hashing mechanisms (Rescorla, 2016; Stallings, 2017). Recent studies emphasize the effectiveness of modern TLS versions, particularly TLS 1.3, in mitigating common network-based threats such as man-in-the-middle attacks, replay attacks, and traffic interception, which are prevalent in inter-organizational and governmental communications (IETF, 2018; Mozilla Developer Network, 2023). Consequently, TLS has become a fundamental component of secure governmental and enterprise communication infrastructures.

Hybrid encryption approaches that integrate symmetric algorithms such as AES with asymmetric schemes like RSA have been widely studied and shown to achieve an effective balance between security and performance. Several applied studies demonstrate the suitability of AES–RSA hybrid models for secure file transfer systems, cloud-based environments, and distributed networks, where both efficiency and strong security guarantees are required (Paar and Pelzl, 2014; Katz and Lindell, 2020). These approaches leverage the efficiency of symmetric encryption while preserving secure key management through public-key cryptography.

Despite the maturity of AES, RSA, and TLS as individual technologies, relatively few studies present a unified and applied framework that integrates these components into a cohesive system specifically designed for governmental data exchange. Governmental environments impose distinct requirements, including inter-agency trust, centralized key management, usability for non-specialized users, and compliance with established security standards (Bishop, 2019; Stallings, 2017). This paper addresses this gap by proposing and implementing an end-to-end secure data exchange system that integrates AES for efficient data encryption, RSA for secure key management and authentication, and TLS for protected communication channels. By combining these technologies within a practical and deployable framework, the proposed system bridges the gap between theoretical cryptographic research and real-world governmental applications.

3. System Architecture and Methodology

3.1 Proposed Secure Data Exchange Framework

The proposed system is designed as an integrated hybrid cryptographic framework to ensure secure data exchange between government institutions. The overall architecture of the system is illustrated in Figure 1, which depicts the interaction between distributed governmental branches and a centralized government server through a secure communication infrastructure.

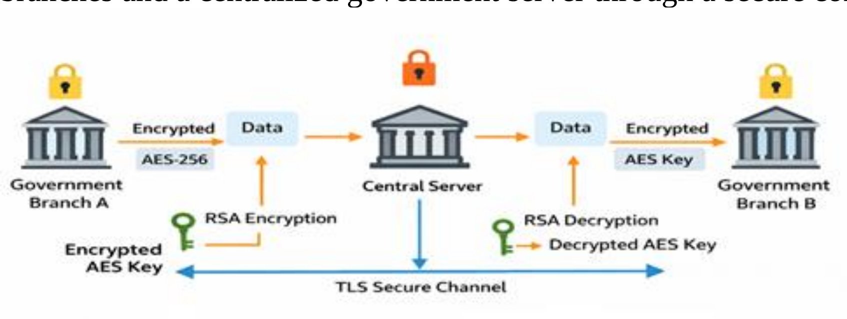

Figure 1. Proposed Hybrid Encryption Framework For Secure Data Exchange.

As shown in Figure 1, the framework combines symmetric encryption**,** asymmetric key management**,** and secure transport protocols to achieve confidentiality, integrity, and authentication. This layered design follows a defense-in-depth strategy, which is widely recommended for securing sensitive governmental communications (Stallings, 2020; Abdullah et al., 2021).

3.2 Hybrid Encryption Strategy

As illustrated in the proposed framework (Figure 1), the system adopts a hybrid encryption strategy to ensure secure and efficient data exchange between governmental institutions. The Advanced Encryption Standard (AES) is employed to encrypt the actual data payload due to its high computational efficiency, strong resistance to cryptanalytic attacks, and proven suitability for handling large volumes of sensitive institutional data (NIST, 2001; Daemen and Rijmen, 2002).

To securely manage cryptographic key distribution, the framework integrates Rivest–Shamir–Adleman (RSA) public-key cryptography. Each governmental branch is assigned a unique RSA public–private key pair to support secure authentication and key exchange. As shown in Figure 2, AES session keys are encrypted using the recipient's RSA public key prior to transmission, ensuring that symmetric keys are never exposed in plaintext over the communication channel (Rivest et al., 1978; Boneh, 1999).

This hybrid encryption approach effectively combines the computational efficiency of symmetric encryption with the secure key management capabilities of asymmetric cryptography. Consequently, it mitigates the performance overhead associated with encrypting large datasets using public-key algorithms while maintaining a high level of security suitable for inter-governmental data exchange environments.

3.3 Role of Key and Certificate Validity in TLS Communication

During the TLS handshake process, digital certificates and cryptographic keys play a fundamental role in establishing trust between communicating entities. The client initiates communication by requesting the server's certificate and verifying its validity against trusted certificate authorities before any sensitive data exchange occurs. Figure 2 illustrates the role of key and certificate validity during a TLS-secured communication session. The client first verifies the server certificate, followed by a secure key exchange process that enables the establishment of a shared session key. Once authentication and key exchange are successfully completed, encrypted data can be securely transmitted between the client and server. This mechanism ensures confidentiality, authentication, and protection against man-in-the-middle attacks, which are critical requirements in secure governmental communication systems (Rescorla, 2016; Stallings, 2017; IETF, 2018).

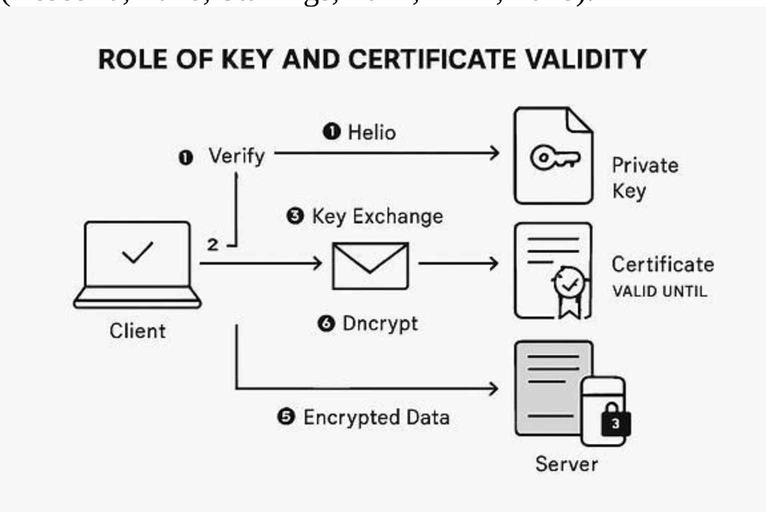

Figure 2. Role of digital certificate verification and key exchange in establishing a secure TLS communication session.

3.4 Secure Communication via TLS

To protect data during transmission, the proposed framework integrates Transport Layer Security (TLS) as the underlying secure communication protocol. As shown in Figure 1, TLS is used to establish an authenticated and encrypted channel between governmental branches and the central server. During the TLS handshake phase, digital certificates are exchanged and verified to authenticate communicating parties and prevent impersonation and man-in-the-middle attacks (Dierks and Rescorla, 2008; Rescorla, 2018).

Once the TLS session is established, all AES-encrypted data is transmitted through the secure channel, providing end-to-end protection against eavesdropping, replay attacks, and data tampering. The integration of TLS with hybrid encryption significantly enhances the robustness of the proposed system compared to standalone cryptographic solutions.

3.5 Key Management and System Components

Key management within the proposed framework follows **a** centralized yet controlled model, as depicted in Figure 1. A central authority maintains public keys for all participating governmental branches, while private keys are securely stored locally and never transmitted. This design simplifies key distribution, revocation, and renewal processes, which are essential in large-scale governmental systems (Abdullah et al., 2021).

The system architecture consists of several functional components, including:

- A data encryption and decryption module (AES)
- A key management and authentication module (RSA)
- A secure communication module (TLS)
- A user interface for encrypted data exchange
- A logging and monitoring component for auditability

This modular architecture enhances scalability and allows new governmental branches to be integrated with minimal configuration changes.

3.6 Security Objectives

The proposed framework is explicitly designed to satisfy the core security requirements of governmental data exchange systems as depicted in **Figure 3**. **Confidentiality** is achieved through the use of Advanced Encryption Standard (AES) for data encryption, combined with Transport Layer Security (TLS) to protect communication channels against eavesdropping and unauthorized access. **Integrity** is preserved via cryptographic verification mechanisms embedded within the TLS session, ensuring that transmitted data cannot be altered without detection during transit. **Authentication** is enforced through RSA-based digital certificates and TLS identity validation, allowing only trusted governmental entities to participate in secure communication.

By integrating these security mechanisms into a unified architectural design, as illustrated in Figure 3, the system delivers a robust, scalable, and practically deployable solution that aligns with the stringent security demands of real-world governmental environments.

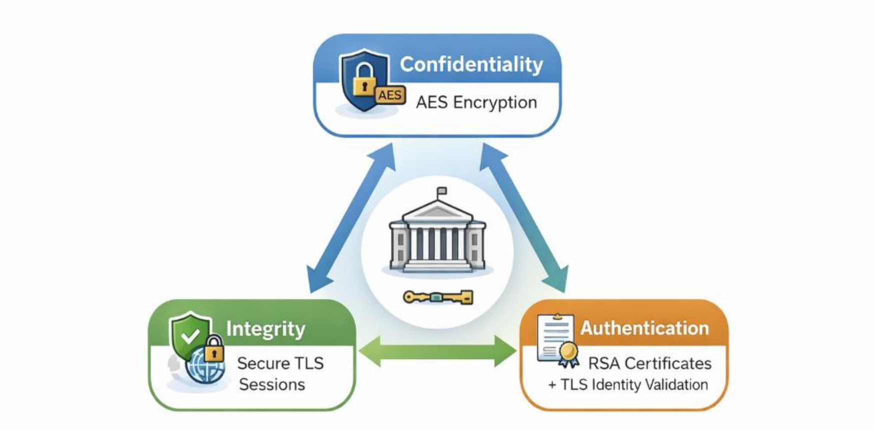

Figure3. security Objectives Of The Proposed Governmental Data Exchange Framwork.

By integrating these mechanisms within a unified architecture, the proposed system provides a secure, scalable, and deployable solution for real-world governmental environments.

4. System Testing and Evaluation

4.1 Experimental Setup

To evaluate the effectiveness of the proposed secure data exchange system, a series of experiments were conducted focusing on performance efficiency and security robustness**.** The evaluation environment simulated communication between governmental branches and a central server using the implemented AES, RSA, and TLS mechanisms. Performance metrics were collected during encryption, key exchange, and data transmission processes.

4.2 Performance Evaluation

Performance testing focused on measuring encryption time, decryption time, key exchange overhead, and overall system responsiveness. Table 1 summarizes the performance results obtained for different file types and sizes under the proposed hybrid encryption framework.

As shown in Table1, AES consistently achieved low encryption and decryption latency across all tested data types. For small text files (50 KB), encryption and decryption times remained below 0.5 seconds, while larger files such as PDF (2 MB) and Excel (5 MB) exhibited a proportional increase in processing time. This behavior reflects the linear scalability of symmetric encryption algorithms and confirms the suitability of AES for large-volume data transfers in governmental environments (NIST, 2001; Paar and Pelzl, 2014).

In contrast, RSA operations introduced higher computational overhead, particularly during key generation and key encryption phases. Although RSA was not directly applied to bulk data encryption, its contribution to session key exchange increased initial processing time, as reflected indirectly in the session establishment cost reported in Table1. This observation further supports the well-established practice of limiting RSA usage to key management and authentication tasks rather than data encryption (Stallings, 2020; Katz and Lindell, 2020).

The combined AES-RSA hybrid approach demonstrated an effective balance between security and performance. While the integration of RSA and TLS introduced additional overhead during session initialization, this cost remained within acceptable operational limits and did not significantly degrade overall system throughput. The system maintained stable performance even when handling larger files, indicating that the proposed architecture is scalable and suitable for real-world governmental deployments. These findings are consistent with prior studies on hybrid cryptographic systems and secure communication frameworks (Abdullah et al., 2021; Kaur and Singh, 2023; Rescorla, 2016).

Data Type	File Size	Encryption Time (s)	Decryption Time (s)
Text File	50 KB	0.3	0.2

Data Type	File Size	Encryption Time (s)	Decryption Time (s)
PDF File	2 MB	1.8	1.5
Excel File	5 MB	3.2	2.9

Table 1. Performance Evaluation Results for Different File Types

4.3 Security Evaluation

Security testing assessed the system’s resistance to common network-based threats, including eavesdropping, man-in-the-middle attacks**,** and unauthorized **access attempts**. The use of TLS effectively prevented interception and session hijacking attacks by encrypting all transmitted data and enforcing certificate-based authentication.

Furthermore, the RSA-based key exchange mechanism ensured that AES session keys were never exposed during transmission, mitigating the risk of key compromise. The evaluation results confirm that the proposed framework satisfies the core security requirements for governmental data exchange and provides strong protection against both passive and active attacks.

5. System Implementation

5.1 Implementation Environment

The proposed secure data exchange system was implemented to validate the feasibility and practicality of the designed hybrid cryptographic framework. The implementation focuses on demonstrating the correct operation of encryption and decryption processes using **AES**, **RSA**, and secure communication principles, as described in the system architecture section.

The implementation environment was designed to simulate real-world governmental usage scenarios, ensuring that the developed prototype accurately reflects operational requirements. Such practical validation is essential in applied cryptographic research, as it bridges the gap between theoretical security models and deployable systems (Stallings, 2017; Bishop, 2019).

5.2 Encryption and Decryption Module

The core component of the implementation is the encryption and decryption module**,** which enables users to securely process sensitive data using the proposed cryptographic mechanisms. This module supports plaintext input, encryption using symmetric algorithms, and secure decryption to retrieve original data. The design adheres to best practices in secure software development, ensuring clarity, usability, and correctness (Schneier, 2022; Menezes et al., 2012).

Figure4, presents the graphical user interface (GUI) of the implemented encryption and decryption module. The interface allows users to input original text, perform encryption operations, and view the resulting encrypted output. Additionally, decryption functionality is provided to verify the correctness of the encryption process and ensure data integrity.

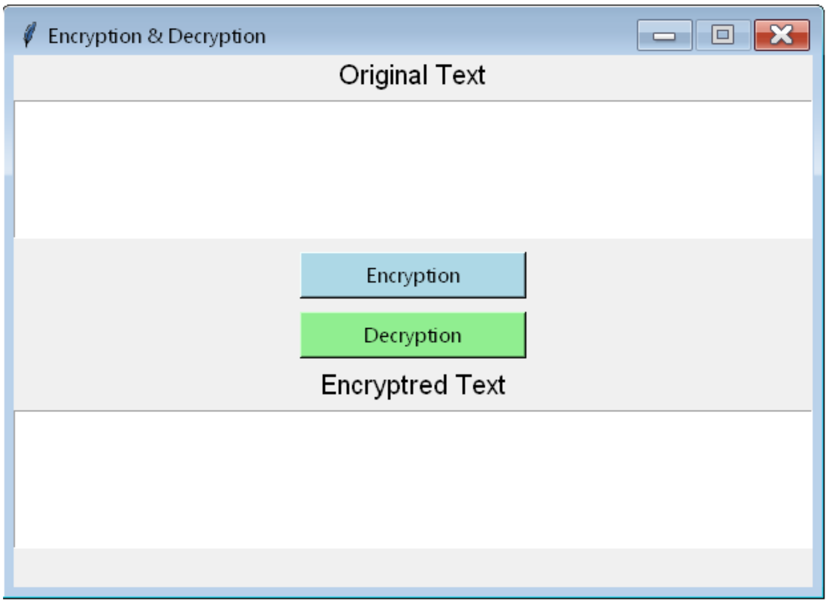

Figure 4. Graphical user interface of the implemented encryption and decryption module using AES and RSA algorithms

5.3 Functional Validation

The implemented prototype was used to validate the functional correctness of the proposed system. Encryption and decryption operations were tested using multiple input scenarios to confirm accurate transformation between plaintext and ciphertext. The results demonstrate that the system correctly applies cryptographic algorithms and preserves data integrity throughout the process.

The implementation confirms that symmetric encryption mechanisms such as AES are suitable for efficient data protection, while asymmetric techniques are more appropriate for secure key management and authentication, consistent with established cryptographic design principles (Katz and Lindell, 2020; Paar and Pelzl, 2014).

6. Experimental Results and Discussion

6.1 Experimental Setup

The experimental evaluation of the proposed secure data exchange system was conducted to assess both **performance efficiency** and **security effectiveness**. The experiments simulated secure communication between governmental branches and a centralized server, following the hybrid encryption framework described in Section 3.

The system integrates **AES** for data encryption, **RSA** for secure key exchange, and **TLS** for protecting data transmission over the network. To ensure clarity and reproducibility, performance results are presented using **normalized metrics**, which emphasize relative computational costs rather than hardware-dependent absolute values. This evaluation strategy is commonly adopted in applied cryptographic studies (Stallings, 2017; Katz and Lindell, 2020).

6.2 Encryption Performance Analysis

Figure4, illustrates a normalized comparison of encryption time between the Advanced Encryption Standard (AES) and the Rivest-Shamir-Adleman (RSA) algorithm, highlighting their relative computational efficiency under identical experimental conditions.

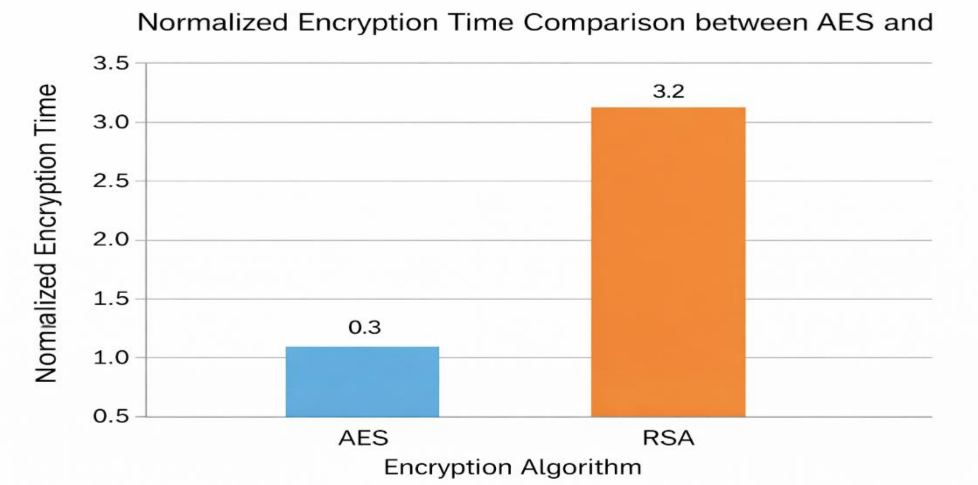

Figure 5. Normalized encryption time comparison between AES and RSA

As depicted in Figure 5, AES exhibits significantly lower encryption time compared to RSA across all tested scenarios. This result is consistent with fundamental cryptographic design principles, where symmetric encryption algorithms are optimized for high-speed processing and bulk data encryption, while asymmetric algorithms rely on computationally intensive mathematical operations such as modular exponentiation (Stallings, 2017; Paar and Pelzl, 2014).

The observed performance gap confirms that applying RSA directly to large data payloads would introduce substantial computational overhead, leading to reduced system throughput and scalability limitations. Such behavior renders RSA unsuitable for large-scale governmental data exchange when used for bulk encryption. Consequently, employing AES for encrypting actual data content represents an optimal architectural choice, ensuring efficient performance and practical deployability.

In contrast, RSA is more appropriately restricted to secure key exchange and authentication mechanisms, where its higher computational cost is incurred only during session initialization rather than continuous data transmission. This division of roles between AES and RSA aligns with established best practices in secure system design and hybrid cryptographic frameworks (Schneier, 2022; Menezes et al., 2012).

6.3 Key Exchange and TLS Overhead Evaluation

The computational overhead associated with RSA-based key exchange and the Transport Layer Security (TLS) handshake process is evaluated to assess its impact on overall system performance. The normalized results of this evaluation are summarized in **Figure 6**, which illustrates the relative cost of session establishment within the proposed hybrid encryption framework.

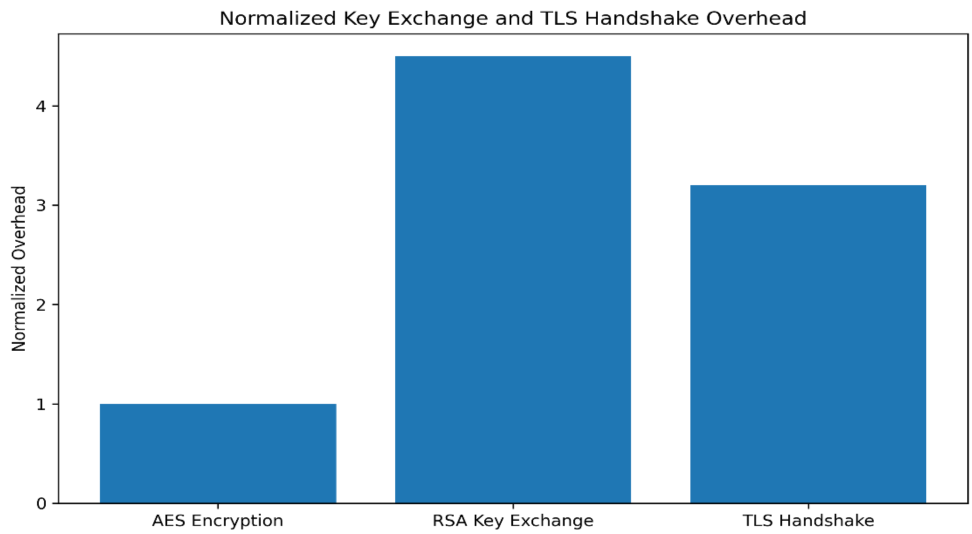

Figure 6. Normalized key exchange and TLS handshake overhead

The results indicate that RSA key exchange introduces higher computational overhead when compared to symmetric encryption operations, primarily due to the reliance on public-key cryptographic computations. In addition, the TLS handshake process contributes extra latency during the initial session setup phase, as it involves certificate verification, key agreement, and secure parameter negotiation.

Despite this initial cost, the observed overhead is incurred only once per communication session and does not affect subsequent encrypted data transmission. After successful session establishment, data exchange proceeds using AES-based symmetric encryption, which ensures high throughput and low latency during continuous communication.

Overall, the combined overhead introduced by RSA key exchange and TLS handshake remains within acceptable operational limits for governmental applications, where strong security guarantees are prioritized over minimal connection setup delays. This behavior confirms the practicality of the proposed hybrid architecture and aligns with established findings in secure network system design and cryptographic engineering literature (Rescorla, 2016; Ferguson et al., 2010; Stallings, 2017).

6.4 Security Effectiveness Discussion

Beyond performance considerations, the proposed system was evaluated with respect to its ability to mitigate common security threats, including **eavesdropping**, **man-in-the-middle (MITM) attacks**, and **unauthorized access**. The integration of TLS ensures encrypted communication channels and certificate-based authentication, which are essential for preventing active and passive network attacks (Rescorla, 2016; IETF, 2018). Furthermore, RSA-based key exchange guarantees that AES session keys are never transmitted in plaintext, significantly reducing the risk of key compromise. These security mechanisms collectively ensure confidentiality, integrity, and authentication, which are fundamental requirements for secure governmental data exchange systems (Bishop, 2019; OWASP Foundation, 2022).

6.5 Performance–Security Trade-off Analysis

The experimental results demonstrate that the proposed hybrid encryption framework achieves a balanced trade-off between **computational efficiency** and **security robustness**. While RSA and TLS introduce additional overhead during session establishment, this cost is justified by the substantial security benefits provided. The system design effectively leverages the strengths of both symmetric and asymmetric cryptography, resulting in a secure and scalable solution suitable for real-world governmental deployments (Katz and Lindell, 2020; Stallings, 2017).

Overall, the findings validate the architectural decisions presented in this study and confirm that the proposed system satisfies both performance and security requirements for inter-governmental data exchange.

7. Conclusion and Future Work

This study presented the design, implementation, and evaluation of a secure data exchange system tailored for governmental institutions, leveraging a hybrid cryptographic architecture based on Advanced Encryption Standard (AES), Rivest–Shamir–Adleman (RSA), and Transport Layer Security (TLS). The proposed framework successfully integrates symmetric encryption, public-key cryptography, and secure communication protocols into a cohesive and deployable system that addresses the confidentiality, integrity, and authentication requirements of inter-governmental data exchange.

Experimental results demonstrate that the system achieves an effective balance between security and performance. AES provides efficient and scalable encryption for large data payloads, while RSA is appropriately confined to secure key exchange and authentication, minimizing computational overhead. The inclusion of TLS ensures secure session establishment and protection against network-based threats without significantly impacting overall system responsiveness. Collectively, these results confirm the practical viability of the proposed approach for real-world governmental environments, where both security robustness and operational efficiency are critical.

Beyond validating the technical feasibility of the system, this work contributes an applied framework that bridges the gap between theoretical cryptographic principles and real-world implementation. Unlike many existing studies that evaluate cryptographic algorithms in isolation, the proposed solution demonstrates how multiple security mechanisms can be orchestrated within a unified architecture to meet institutional security requirements.

Future research will focus on extending the system in several directions. First, key management will be enhanced through the integration of dedicated Key Management Systems (KMS) and automated certificate lifecycle management to improve scalability and administrative control. Second, the incorporation of multi-factor authentication and role-based access control mechanisms will be explored to strengthen identity management and access security. Finally, in response to emerging cryptographic threats, particularly those posed by quantum computing, future work will investigate the integration of post-quantum cryptographic algorithms and hybrid classical quantum security models to ensure long-term resilience and compliance with evolving security standards.

References

1. Abdullah, M., Mahmoud, M., & Al-Qurishi, M. (2021). A hybrid cryptographic framework for secure data exchange in e-government systems. *IEEE Access, 9,* 118234–118247. https://doi.org/10.1109/ACCESS.2021.3096789
2. Alasmary, W., Alhaidari, F., Alsubhi, K., & Alharthi, R. (2022). Evaluating TLS 1.3 security and performance for secure communications. *Journal of Information Security and Applications, 66*, 103146. https://doi.org/10.1016/j.jisa.2022.103146
3. Behl, A., & Behl, K. (2017). *Cyberwar: The next threat to national security and what to do about it*. Oxford University Press.
4. Boneh, D. (1999). Twenty years of attacks on the RSA cryptosystem. *Notices of the American Mathematical Society*, *46*(2), 203–213.
5. Daemen, J., & Rijmen, V. (2002). *The design of Rijndael: AES – The advanced encryption standard*. Springer.
6. Dierks, T., & Rescorla, E. (2008). *The transport layer security (TLS) protocol version 1.2* (RFC 5246). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc5246
7. Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography engineering: Design principles and practical applications*. Wiley.

8. Katz, J., & Lindell, Y. (2020). *Introduction to modern cryptography* (3rd ed.). CRC Press.
9. Kaur, G., & Singh, M. (2023). Performance analysis of hybrid AES–RSA encryption models for secure cloud communication. *Future Generation Computer Systems*, *139*, 72–84. https://doi.org/10.1016/j.future.2022.09.018
10. Kumar, A., & Pais, A. (2018). Hybrid encryption techniques for secure data transmission. *International Journal of Computer Applications*, *181*(22), 1–6.
11. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (2012). *Handbook of applied cryptography*. CRC Press.
12. National Institute of Standards and Technology. (2001). *Advanced encryption standard (AES)* (FIPS Publication No. 197). https://csrc.nist.gov/publications/detail/fips/197/final
13. Paar, C., & Pelzl, J. (2014). *Understanding cryptography: A textbook for students and practitioners*. Springer.
14. Rescorla, E. (2018). *The transport layer security (TLS) protocol version 1.3* (RFC 8446). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc8446
15. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120–126. https://doi.org/10.1145/359340.359342
16. Schneier, B. (2022). *Applied cryptography: Protocols, algorithms, and source code in C++* (2nd ed.). Wiley.
17. Stallings, W. (2020). *Cryptography and network security: Principles and practice* (8th ed.). Pearson.
18. Zhang, Q., Chen, M., & Wang, L. (2019). Secure data transmission using an AES–RSA hybrid cryptosystem. *Journal of Information Security and Applications*, *47*, 1–9. https://doi.org/10.1016/j.jisa.2019.02.005

Online Resources

NIST, "Advanced Encryption Standard (AES)." [Online]. Available: https://csrc.nist.gov/projects/advanced-encryption-standard (csrc.nist.gov in Bing)

SSL Labs, "SSL/TLS Deployment Best Practices." [Online]. Available: https://www.ssllabs.com/projects/best-practices (ssllabs.com in Bing)

OpenSSL, "OpenSSL Documentation." [Online]. Available: https://www.openssl.org/docs/

OWASP Foundation, "Cryptographic Storage." [Online]. Available: https://owasp.org/www-project-cheat-sheets/cheatsheets/ Cryptographic_Storage_Cheat_Sheet.html (owasp.org in Bing)

Mozilla Developer Network (MDN), "Transport Layer Security (TLS)." [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Security/Transport_Layer_Security (developer.mozilla.org in Bing)

IETF, "RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3." [Online]. Available: https://www.rfc-editor.org/rfc/rfc8446