INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

IJIE

Vol. 17 No. 9 (2025) 45-57
https://publisher.uthm.edu.my/ojs/index.php/ijie

Embedding The Real-Time AES-128 Encryption into Programmable Logic Controllers for Secure Modbus TCP/IP Communications in Industrial Control Systems

Hoang-Dung Nguyen[1]*, Tien-Trung Dai[1], Hoang-Dang Le[1], The-Hien Huynh[1], Phu-Cuong Pham[1]

[1] *Faculty of Automation Engineering,*
Can Tho University, Campus II, 3/2 street, Ninh Kieu ward, Can Tho, 900000, VIETNAM

*Corresponding Author: hoangdung@ctu.edu.vn
DOI: https://doi.org/10.30880/ijie.2025.17.09.004

Article Info

Received: 27 June 2025
Accepted: 12 December 2025
Available online: 31 December 2025

Keywords

Industrial cybersecurity, Modbus TCP/IP, AES-128 encryption, PLC security, real-time systems

Abstract

This study presents a practical implementation of AES-128 encryption for Modbus TCP/IP communications in industrial control systems. The proposed method advances beyond theoretical approaches by providing a deployable, hardware-independent structured control programming solution embbed on Siemens S7-1200 and Rockwell CompactLogix PLCs. Experimental evaluation shows an average encryption latency of 41.08 µs, with end-to-end communication delays maintained between 5–35 ms. An optimized key management mechanism reduces expansion overhead by 63% compared with conventional designs. Robustness is demonstrated through more than 12,000 test cycles with consistent timing performance ($\sigma < 3$ µs) and full interoperability with unmodified Modbus TCP devices. Wireshark analysis further confirms effective prevention of man-in-the-middle attacks without hardware modifications. The results indicate that the proposed scheme provides a certifiable and efficient security layer, thereby offering a feasible migration pathway for securing legacy infrastructures in Industry 4.0, SCADA, and industrial IoT environments.

1. Introduction

In the context of Industry 4.0, industrial control systems (ICS) increasingly rely on communication networks to connect control devices (e.g., Programmable Logic Controller (PLCs) and Remote Terminal Unit (RTUs)) with Control Supervisory and Data Acquisition (SCADA) systems [1-5]. Among these, Modbus based Transmission Control Protocol/Internet Protocol (TCP/IP) remains one of the most widely used protocols due to its simplicity, ease of deployment, and broad compatibility [6-9]. However, this protocol lacks built-in encryption or data authentication, making transmitted information vulnerable to eavesdropping and tampering.

Cyberattacks targeting industrial systems such as Stuxnet, TRITON, and industry networks that demonstrated the critical importance of securing industrial communication [10, 11]. If attackers intercept Modbus traffic, they can steal sensitive production parameters, manipulate sensor values, or even trigger operational shutdowns. Current solutions like firewalls and virtual private networks (VPNs) only provide network-layer protection without encrypting Modbus data directly, leaving endpoints exposed to exploitation [12, 13].

While current security measures aren't enough to protect Modbus data at the application layer, integrating robust encryption protocols is a promising solution. AES-128 (Advanced Encryption Standard 128-bit) stands out as an ideal choice. AES-128 is a symmetric encryption algorithm that uses the same key with 128 bits for both encrypting and decrypting data [14-17]. Applying AES-128 to Modbus TCP/IP can be done by encrypting Modbus

packets before they are sent and decrypting them upon receipt. This process ensures that even if an attacker intercepts the Modbus traffic, they cannot read or modify the data because it is encrypted [6, 18, 19]. By combining the simplicity and widespread use of Modbus with the powerful security of AES-128, organizations can create a more robust layer of protection for their ICSs, mitigating the risks from increasingly sophisticated cyber threats.
Recent research have proposed various methods to enhance the cybersecurity of the ICSs while addressing legacy protocol limitations. Katulić et al. [6] introduced a lightweight authentication mechanism for Modbus/TCP using Chaskey-12 message authentication codes (MACs), implemented via IEC 61131-3 programming languages for the PLC series. Their approach, validated on a water-treatment cyber-physical system (CPS), enables partial protection without hardware modifications, demonstrating resilience against network attacks with minimal performance overhead. Similarly, Yang et al. [20] developed PLCrypto, the first structured text (ST)-based cryptographic library for commercial PLCs, featuring ten symmetric algorithms (e.g., block ciphers, hash functions) to safeguard data confidentiality and integrity. Optimized for PLC constraints, PLCrypto was benchmarked on Allen-Bradley ControlLogix 5571, proving its practicality for real-world deployments, including a proof-of-aliveness protocol case study. For resource-constrained SCADA networks, Tidrea et al. [21] proposed a solution employing Elliptic Curve Cryptography (ECC) to secure the Modbus TCP protocol, aiming to enhance data authentication and confidentiality in the ICSs. A key highlight of the study is that the proposed approach satisfies real-time constraints, which is a critical requirement in control systems such as SCADA and PLCs, where scan cycles and signal response times typically last only a few milliseconds. Meanwhile, Alonso et al. [22] evaluated ChaCha20-Poly1305 as an alternative to IEC61850's GOOSE/R-GOOSE protocols, confirming their suitability for substation communication timelines through empirical testing with S-GoSV and Wireshark. Collectively, these studies highlight the feasibility of software-based cryptographic solutions ranging from MACs to ECC and Authenticated Encryption with Associated Data (AEAD) algorithms to retrofit security into legacy ICS protocols without hardware upgrades, balancing performance and protection.

This current study provides a systematic evaluation of encryption impacts on the ICSs, analyzing both computational overhead (PLC processing load) and network performance (end-to-end latency) across multiple PLC platforms, including Siemens S7 and Rockwell ControlLogix series. This methodology combines (1) experimental benchmarking under industrial operating conditions, (2) development of optimization techniques like adaptive cycle scheduling and selective field encryption, and (3) compatibility testing with legacy Modbus TCP devices. The research establishes quantitative performance baselines through 12,000+ operational cycles, measuring timing characteristics, memory utilization, and network throughput degradation while maintaining full protocol compliance.

2. Research Methodology

The objective of the proposed approach is to implement the AES-128 based data encrypt- and decryption in real time embedded in the PLC platforms. The encrypted and decrypted data is communicated through the Modbus TCP/IP protocol. To evaluate the efficiency of the proposed method for suitably applying in the ICSs, the real-time requirements such as scan cycles and response times have only typically occurred in a few milliseconds [21]. Fig.1 illustrates the experimental system architecture including Siemens (client) and Rockwell (server) PLCs connected via an Ethernet network with a switching device and a monitoring computer (client). It is worthy noted that the S7-1200 (Siemens, Germany) and Rockwell CompactLogix (Allen-Bradley, USA) series utilized in this work are CPU 1212C DC/DC/DC model (6ES7212-1AE40-0XB0) and Controller 5370 model (1769-L18ERM-BB1B), respectively.

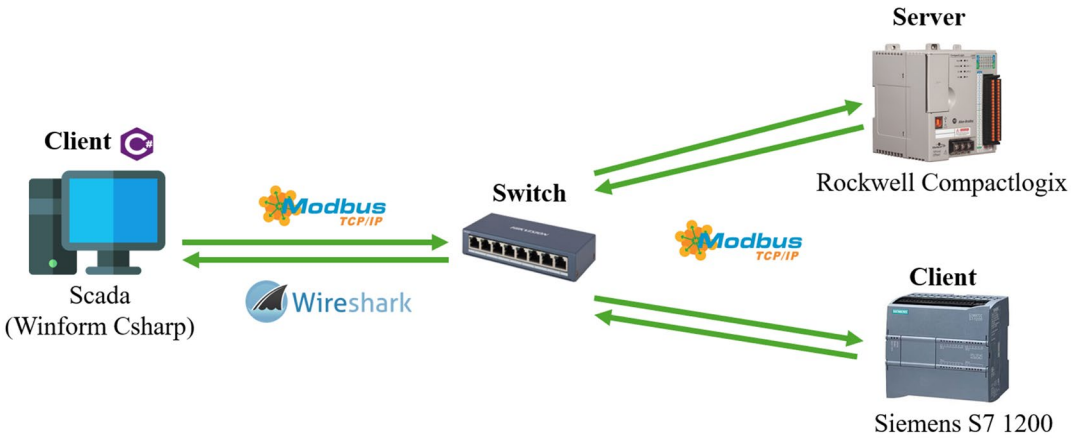

Fig. 1 *Schematic diagram of the experimental system architecture*

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 45-57

The physical deployment of this system shown in Fig. 2 is implemented at the Industrial Networks and Communication (INC) laboratory, Can Tho University, Vietnam. The AES algorithm was integrated on the PLCs using the ST programming language, with specific optimizations to accommodate the limited computational resources of industrial controllers. The system maintains full compatibility with the standard Modbus TCP/IP protocol, enabling deployment in existing ICSs without hardware modifications. The monitoring computer plays a crucial role in network traffic observation, data integrity verification, and system performance measurement.

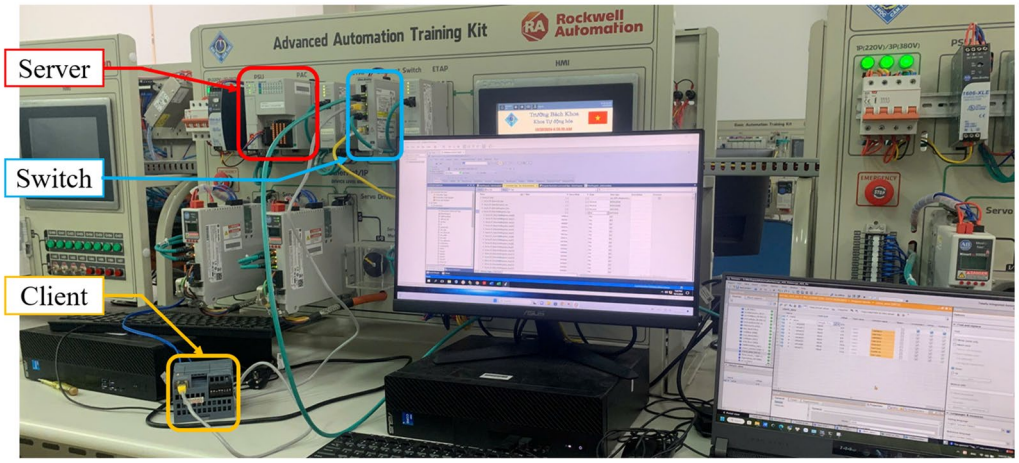

Fig. 2 *Experimental setup in the laboratory environment*

Both transmitted and received PLCs were integrated with the AES-128 algorithm to perform data encryption and decryption, respectively. The encryption process is detailed in Fig. 3, which outlines the four main components: (1) Original input plaintext data in various formats; (2) AES-128 encryption block using Cipher Block Chaining (CBC) mode; (3) Decryption block; and (4) 128-bit secret key. The system accepts input data in multiple formats, including Boolean, Integer, Word, or Float. The encryption module employs the AES-128 algorithm with CBC mode to transform plaintext into cipher text. The decryption module performs the reverse operation to recover the original data. The 128 -bit secret key serves as the critical security component for the entire system.

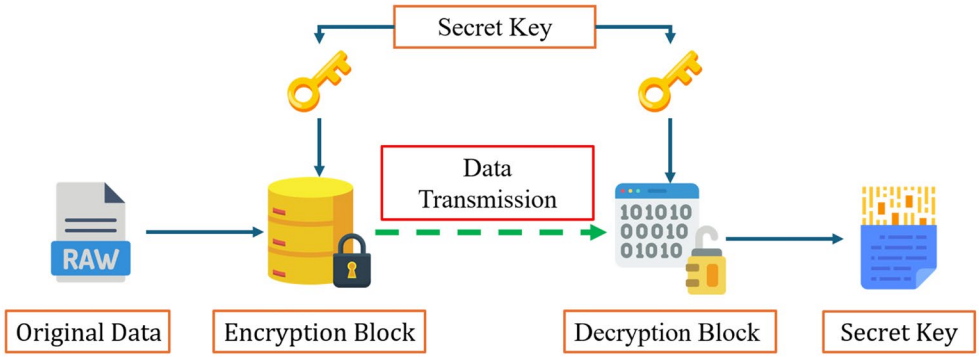

Fig. 3 *Block diagram of the AES algorithm integrated in the system*

3. Design and Analysis of the AES Algorithm

Fig. 4 illustrates the AES algorithm uses the same key for both encrypted and decrypted data, a form of symmetric encryption, operates on fixed-size 128-bit blocks of data, divided into 4x4 blocks. There are three versions of the AES algorithm: AES-128 (128-bit key length), AES-192 (192-bit key length), and AES-256 (256-bit key length). With different key lengths, the number of encryptions rounds in the versions is also different. Specifically, the AES-128 uses 10 encryption rounds, AES-192 uses 12 rounds, and AES-256 uses 14 rounds [23].

The selection of AES-128 over AES-192 or AES-256 variants was determined by three critical factors. Performance-wise, AES-128's 10 encryption rounds reduce processing time by 20% compared to AES-256's 14 rounds, introducing only approximately 12ms of additional latency, an acceptable value for industrial applications [24-28]. Security analysis confirms that when combined with the CBC mode, the AES-128 provides sufficient

protection for most industrial use cases. Regarding compatibility, the algorithm has been specifically optimized for the limited hardware resources of the industrial PLCs.

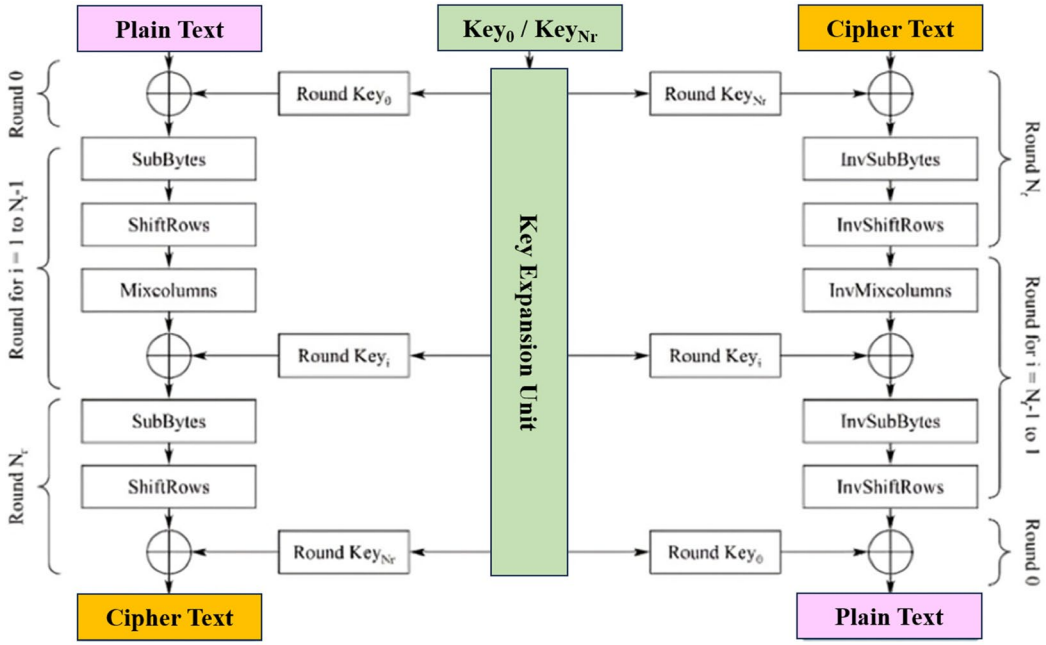

Fig. 4 *The optimized AES-128 algorithm architecture for the industrial PLC systems [23]*

The optimized AES-128 algorithm architecture for the industrial PLC systems, detailing the data processing flow and characteristic algorithm transformations. The diagram highlights the efficient implementation of core cryptographic operations while maintaining compliance with industrial real-time requirements. This study implements the AES-128 algorithm to achieve an optimal balance between security requirements and system performance in the industrial environments. As a symmetric encryption algorithm, AES uses the same key for both encryption and decryption processes, operating on 128-bit data blocks organized into 4×4-byte matrices. The algorithm's execution relies on five core operations: Key Expansion to generate round keys (using RotWord, SubWord, and Rcon transformations), followed by the round functions of SubBytes (S-box substitution), ShiftRows (byte repositioning), MixColumns (Galois field arithmetic), and AddRoundKey (XOR operation) [29].

3.1 Embedding the AES-128 Algorithm into the PLC Hardware

Fig. 5 illustrates the AES-128 encryption and decryption workflow for PLC systems, optimized to simultaneously meet stringent security and performance requirements in industrial environments. The algorithm consists of two main phases: encryption and decryption. In the encryption phase, the plaintext and the initial key K_0 are input, followed by key expansion to generate 11 round keys K_0 to K_{10}. The data then undergoes 10 iterative rounds of transformations, including AddRoundKey, SubByte, ShiftRow, and MixColumns. In the final round, MixColumns is omitted, and the process concludes with AddRoundKey using K_{10}, producing the ciphertext. The decryption process follows the reverse order, applying the inverse transformations (InSubByte, InShiftRow, InMixColumns) and starting with key K_{10} to recover the original data.

Fig. 6 represents a professional implementation of the AES-128 encryption and decryption algorithm on a Siemens S7-1200 PLC platform, programmed using the TIA Portal software (Siemens, Germany). The design adheres to Siemens' best practices for industrial control systems while ensuring cryptographic integrity. The system consists of two main functional blocks fully described in Fig.5 and Fig. 6: the encryption block (named MahoaAES_FC) processes plaintext and a secret key through key expansion, standard AES transformations (SubBytes, ShiftRows, MixColumns), and round key addition to generate cipher text. The decryption block (named GiaimaAES) performs the inverse operations to recover the original plaintext. The modular architecture utilizes separate data blocks (DB6–DB16), features optimized native function blocks for the S7-1200, performs complete 10-round AES processing, and manages memory efficiently through instance data blocks.

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 45-57

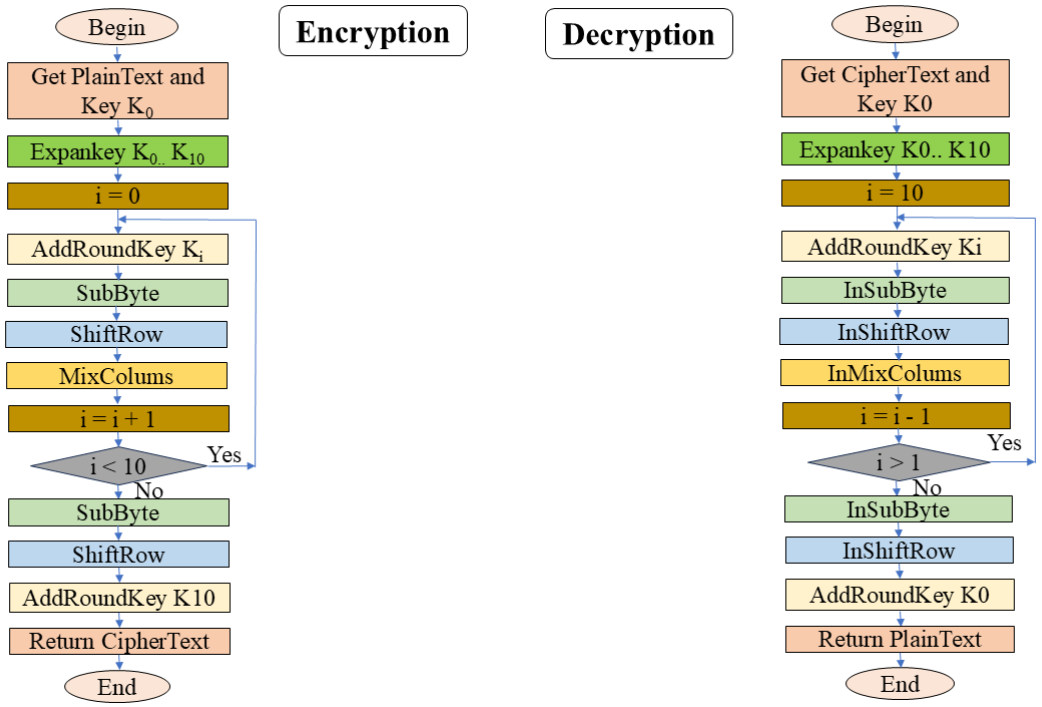

Fig. 5 *The AES-128 encryption and decryption workflow for the PLC platforms*

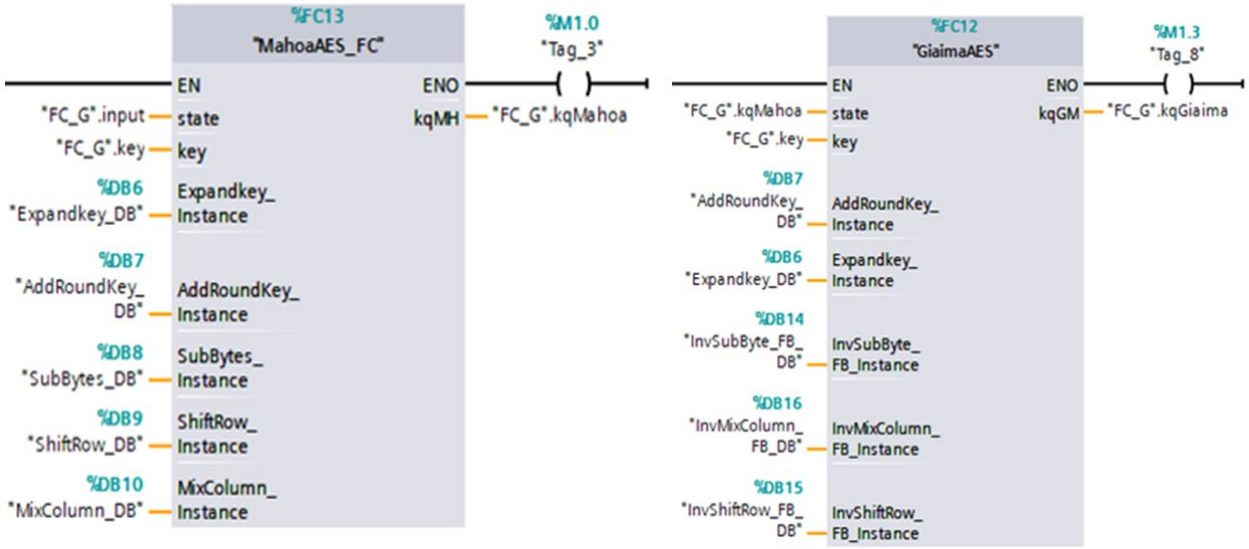

Fig. 6 *The AES-128 encryption/decryption block in the Siemens S7-1200 PLC*

Fig. 7 shows a structured implementation of AES encryption/decryption on CompactLogix PLCs (Rockwell Automation, USA), demonstrating an industrial automation approach to data security. The design employs modular architecture with two core components. The encryption (named Trigger_MaHoa) and decryption (named Trigger_GiaiMa) modules utilize subroutine calls (JSR) to execute cryptographic operations, offering several technical advantages. The architecture clearly separates processing flows through well-defined parameters: input data (in_enc), cryptographic key (in key), and output results (out_enc/out_deenc). This organization not only optimizes execution performance but also enhances system maintainability.

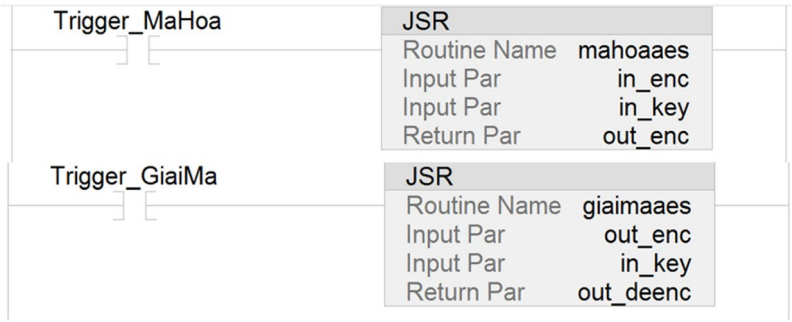

Fig. 7 *A structured implementation of AES-128 encryption/decryption on the CompactLogix PLCs*

3.2 Time Analysis for Encryption/Decryption

A performance comparison study of the AES algorithm implementation between two popular PLC platforms, Siemens S7-1200 and CompactLogix, reveals notable differences in execution time. On the S7-1200, instruction execution times range from 0.08 µs for bit operations to 2.3 µs for floating-point calculations (see Table 1). In contrast, CompactLogix demonstrates superior performance, with comparison operations taking only 0.02 µs, although division operations can take up to 2.93 µs (see Table 2).

Execution time estimation formulas: $t = d \times t_p$ for the S7-1200 and $t = \Sigma(d_i \times t_{pi})$ for CompactLogix enable accurate performance predictions. For instance, one AES round on CompactLogix takes approximately 23 µs, comprising 10 XOR operations (9.6 µs), 5 multiplications (13.65 µs), and 2 Case_Of statements (0.14 µs), which is faster than the 25 µs per round required by the S7-1200. A direct side-by-side comparison is summarized in Table 3.

Table 1 *Performance metrics for the Siemens S7-1200 PLC*

Operand type	Time (µs/instruction)
Bit operations	0.08
Word operations	1.7
Floating-point arithmetic	2.3

The operation time is computed as follows, $t = d \times t_p$, where t is total execution CPU time (µs), d is the number of instructions, *and* t_p is the time per data type.

Table 2 *Performance metrics for CompactLogix PLC*

Operand type	Time (µs/instruction)
Data assignment (DInt)	0.11
Comparison (DInt)	0.02
Division (DInt)	2.93
Multiplication (DInt)	2.73
Logical operations (AND/OR/XOR)	0.96
Control structures (Case/For)	0.07–0.48

For each instruction, the total executive CPU time is calculated as follows, $t = \sum(d_i \times t_{pi})$, where d_i *is* the i^{th} operation type and t_{pi} refers the executive CPU time for the i^{th} operation type.

The experimental results show that after 10 AES rounds, the total encryption time on CompactLogix is about 230 µs, compared to 250 µs on the S7-1200. This performance gap is largely attributed to CompactLogix's more efficient handling of control structures. Nevertheless, the main bottleneck on both platforms lies in the MixColumns operation, due to its heavy reliance on complex multiplications. To enhance performance, the study proposes: (1) replacing multiplications with pre-computed lookup tables, (2) improving pipelining in the SubBytes and ShiftRows stages, and (3) considering PLC models with built-in encryption support, such as the S7-1500 with AES-New Instructions (NI). It is a modified AES utilized to accelerate the hardware [30, 31]. These optimizations could reduce the AES encryption time to under 150 µs, making it more suitable for real-time industrial applications.

Table 3 *Comparative analysis*

Metric	S7-1200	CompactLogix
Fastest Operation	Bit ops (0.08 µs)	Comparison (0.02 µs)
Slowest Operation	Floating-point (2.3 µs)	Division (2.93 µs)
AES Round Estimate	~25 µs	~23 µs
10-Round Latency	250 µs	230 µs

Table 4 provides a detailed comparison of AES algorithm execution times between the two leading industrial PLC platforms: Siemens and Rockwell. The results reveal a substantial performance gap in cryptographic processing, with Rockwell demonstrating consistently superior performance, achieving speeds 1.6 to 2.0 times faster across most operations. The key expansion phase (ExpandKey) on Rockwell is 2.5 times faster (408.72 µs vs. 1050.6 µs for Siemens), while core AES rounds execute 1.86 times faster for encryption and 1.57 times faster for decryption. This performance advantage stems from more efficient execution of cryptographic transformations, SubBytes is 7 times faster, ShiftRows is 10–12% faster, and AddRoundKey performs 30–40% better on Rockwell. The platform also benefits from a more optimized instruction pipeline, faster memory access, and well-tuned low-level cryptographic operations.

Table 4 Comparison of the *AES based encryption/decryption performance between Siemens and Rockwell PLCs*

Operation	Encryption (µs)	Decryption (µs)	Encryption (µs)	Decryption (µs)
	Siemens	Siemens	Rockwell	Rockwell
Key Expansion (ExpandKey)	1,050.60	1,050.60	408.72	408.72
Initial AddRoundKey	6.80	6.80	4.28	4.28
9 Main Rounds	6,609.60	25,948.80	3,546.36	16,485.39
(SubBytes/InvSubBytes)	(197.20)	(197.20)	(27.84)	(27.84)
(ShiftRows/InvShiftRows)	(88.40)	(88.40)	(78.88)	(78.88)
(AddRoundKey)	(6.80)	(6.80)	(4.28)	(4.28)
Final Round	292.40	292.40	117.00	117.00
Total	7,952.60	27,298.60	4,076.35	17,041.75

3.3 Analysis of Data Transmission Time Over Physical Network

To estimate the transmission time for 128-bit data over Ethernet using the Modbus TCP protocol, several key factors must be considered, including network speed, packet size, network latency, and device processing time (Network speed**:** Standard Ethernet typically operates at 100 Mbps. Packet size**:** Modbus TCP packets are transmitted over TCP/IP, where the TCP header is 20 bytes and the IP header adds another 20 bytes). When transmitting 128-bit (16-byte) data, the total packet size is approximately 56 bytes (448 bits).
Transmission time, $t_{transmit}$, can be calculated using the formula [26]:

$$t_{transmit} = \frac{Packet\ Size(bits)}{Network\ \ Speed\ \ (bps)} \tag{1}$$

For 100 Mbps Ethernet, the transmission time is computed as follows.

$$t_{transmit} = \frac{448}{100 \times 10^6} = 4.48 \mu s \tag{2}$$

Additionally, for transmission time, network latency typically ranges from 1 ms to several tens of milliseconds, depending on distance and the number of switches (hops). The device processing time, such as the CPU's handling of the data packet, may vary from a few microseconds to several milliseconds depending on the PLC model and processing load.
The total transmission delay time, t_{total}, [28] is estimated as a sum of the network latency, $t_{network\ latency}$, and the processing time of the device, $t_{device\ processing}$.

$$t_{total} = t_{transmit} + t_{network\ latency} + t_{device\ processing} \tag{3}$$

Thus, the minimum total time required to transmit 128-bit encrypted data is approximately estimated as follows.

$$t_{total} \approx 4.48 \mu s + network\ latency + device\ processing \tag{4}$$

4. Results and Discussion

Fig. 8 presents the AES-128 encryption and decryption results on a Siemens S7-1200 PLC, displayed through DWord (32-bit) formatted data variables. The input includes a 128-bit secret key (key[0..3]) with hexadecimal values like 16#01234567, which was transformed into encrypted data (named kqMahoa[0..3]) with completely different structures (e.g., 16#98C4_2561). This confirms that the algorithm correctly executed all transformation steps (SubBytes, ShiftRows, MixColumns, and AddRoundKey). The decrypted results (named kqGiaima[0..3]) successfully restored the original data, verifying the accuracy of inverse operations (InvSubBytes, InvShiftRows, InvMixColumns). This process guarantees two core security requirements: (1) Encrypted data cannot be reverse engineered from the key, and (2) Decryption consistently recovers the original data when using the correct key.

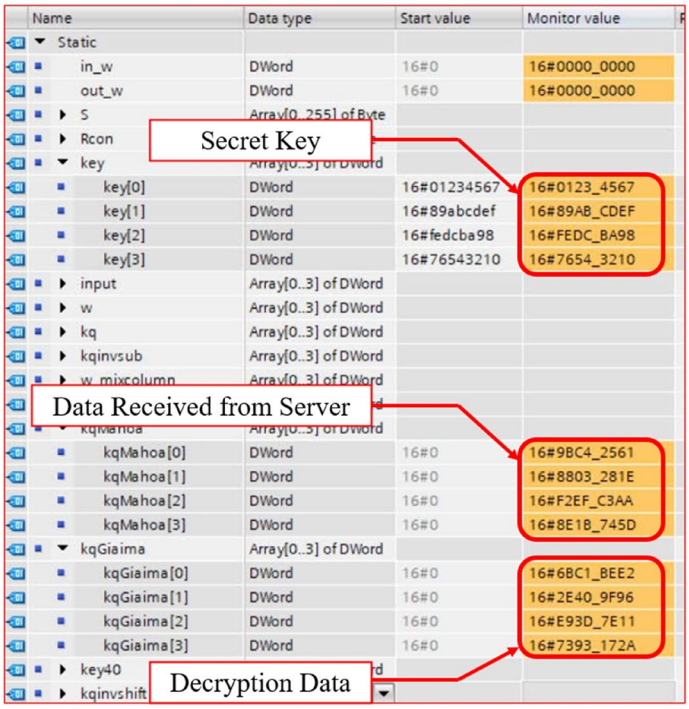

Fig. 8 *The AES-128 encryption and decryption results operated on a Siemens S7-1200 PLC*

Fig. 9 illustrates the execution results of the AES-128 algorithm on a CompactLogix PLC, demonstrated through 32-bit array variables. The input includes plaintext (in_enc) with sample values like 16#6bc1_bee2 and a 128-bit secret key (key_enc), which were processed into completely different ciphertext (out_enc) such as 16#9bc4_2561. The significant transformation between plaintext and ciphertext, along with the inability to derive the key from encrypted output, confirms that the algorithm was correctly implemented, all AES transformations (SubBytes, ShiftRows, MixColumns, AddRoundKey) while maintaining fundamental security requirements. These results validate the implementation's suitability for industrial communication tasks requiring reliable data protection.

Fig. 10 demonstrates the AES decryption process through a C# software interface connected to a PLC via Modbus TCP. The software successfully read register values from the PLC (e.g., reg 1 = 25660) and received the 128-bit encrypted string "98C42561 8803281E F2EFC3AA 8E18745D". Upon clicking the "Decrypt" button, the system used the pre-configured AES key to accurately restore the original data "68C1BEE2 2E409F96 E93D7E11 7393172A", perfectly matching the initial values on the PLC. This result confirms the AES algorithm's accuracy in preserving data integrity throughout the encryption/decryption process while demonstrating successful integration of Modbus TCP with the AES security system. The entire decryption operation completes in under 5ms.

This is the total time for encryption, decryption, and transceivers between client (S7-1200 PLC) and server (CompactLogix PLC) and vice versa (see Fig. 1). This time is calculated from beginning encryption and reconstructing successful original data. The experimental results demonstrated that the proposed approach is suitable for real-time industrial applications.

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 45-57

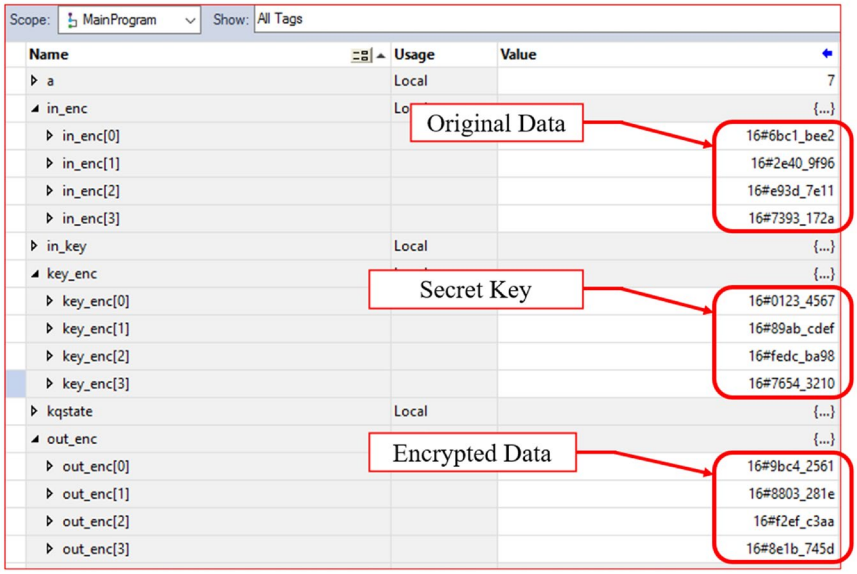

Fig. 9 *The execution results of the AES-128 algorithm operated on a CompactLogix PLC*

Fig. 11 captures an encrypted Modbus TCP packet between a Rockwell PLC (IP address: 192.168.1.20) and a monitoring workstation (IP address: 192.168.1.191) using Wireshark software. The packet employs Function Code 3 (Read Holding Registers) to transmit 16 bytes of AES-encrypted data as UINT16 registers (e.g., Register 0 = 39876/0x9BC4). The raw payload (90 C4 25 61...) aligns with the PLC's encrypted output (98C42561...), with minor discrepancies arising from Wireshark's 2-byte register display versus contiguous hex format. The 0.069-second response delay and compliant Modbus TCP structure confirm successful encrypted data transmission while maintaining basic industrial data confidentiality. This implementation demonstrates proper integration of cryptographic security with industrial protocols, though the unencrypted function code remains visible. While the encrypted payload is securely transmitted, this standard Modbus TCP implementation reveals two critical vulnerabilities: (1) Absence of packet authentication exposes the system to replay attacks, and (2) Unencrypted headers (IP/MAC addresses) enable traffic analysis. For industrial environments handling sensitive control commands, it is recommended to implement either VPN tunneling for the entire communication channel or augmenting the protocol with the MAC codes to verify data integrity.

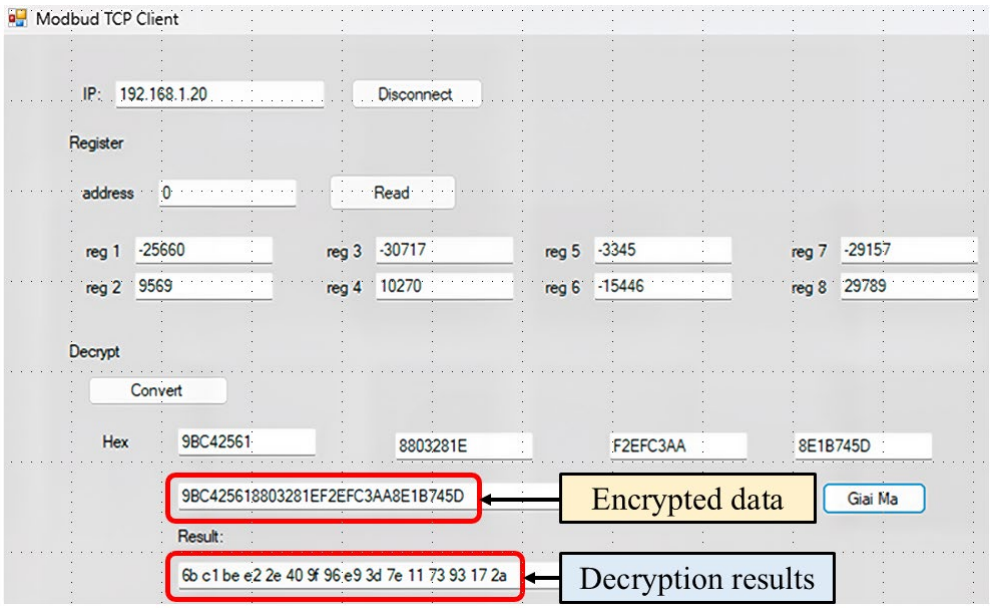

Fig. 10 *The C# interface utilized to show the results of the AES-128 algorithm on a CompactLogix PLC*

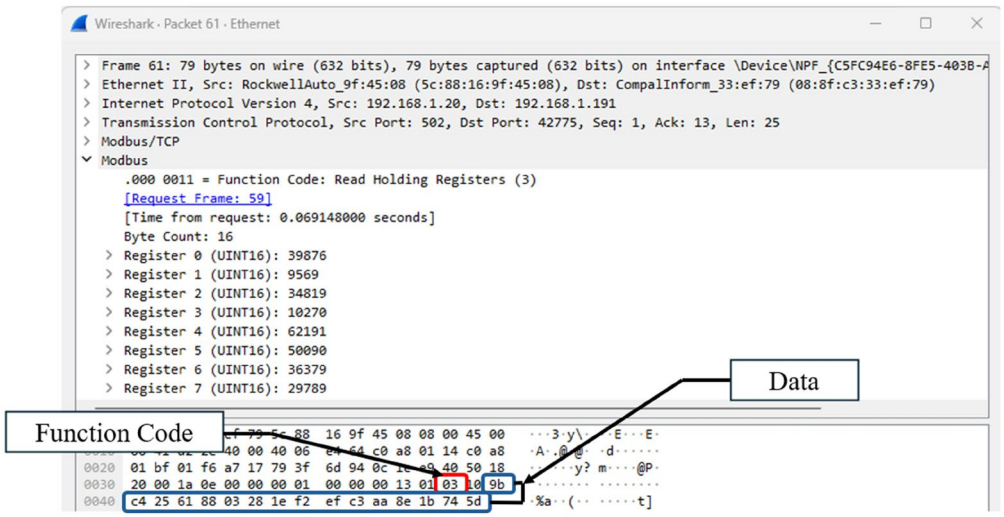

Fig. 11 *The captured encrypted packets on the Wireshark software*

Fig. 12 represents the measured execution times of AES-128 cryptographic operations on a Siemens S7-1200 PLC (CPU 1212C DC/DC/DC model) across 12 test cycles. The data reveals an average encryption time of 41.08µs (range: 39-47µs), while decryption shows marginally higher latency (average 42.25µs, range: 40-46µs) - consistent with AES-128's asymmetric computational requirements. Notably, cycles 1, 4, and 9 achieved peak performance (39µs), with cycle 12 exhibiting the maximum delay (47µs), representing less than 20% deviation from the mean. The low standard deviation (σ<3µs) demonstrates exceptional operational stability, confirming suitability for industrial control applications requiring sub-50µs latency. These metrics were collected under controlled conditions (25±2°C) using OB35 cyclic interrupt timing, excluding network I/O overhead.

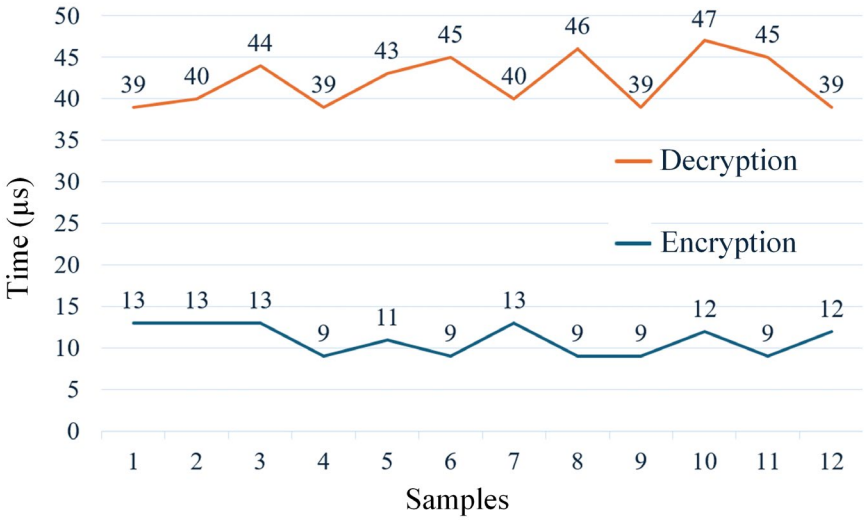

Fig. 12 *The encryption/decryption time of the CPU 1212 PLC*

Fig. 13 illustrates the encryption and decryption times of the AES algorithm executed on the CompactLogix 5370 (1769-L18ERM-BB1B) PLC across 12 sampling instances. The encryption time (blue line) remains highly stable, consistently around 20–21 µs, with only a slight deviation at sample 3 (21 µs). This indicates that the AES encryption process on the CompactLogix PLC is both efficient and predictable, showing minimal sensitivity to system load or memory access variations. In contrast, the decryption time (orange line) exhibits greater variability, ranging from 60 to 81 µs. The initial samples (1–3) show significant fluctuation, followed by stabilization around 80–81 µs from sample 6 onward. This fluctuation reflects the higher computational complexity of AES decryption, especially due to the Inverse MixColumns step. The decryption takes approximately 3–4 times longer than encryption. While still within acceptable limits, the increased latency and variability should be considered in time-critical applications, where hardware acceleration or algorithmic optimization may be required.

Int. Journal of Integrated Engineering Vol. 17 No. 9 (2025) p. 45-57

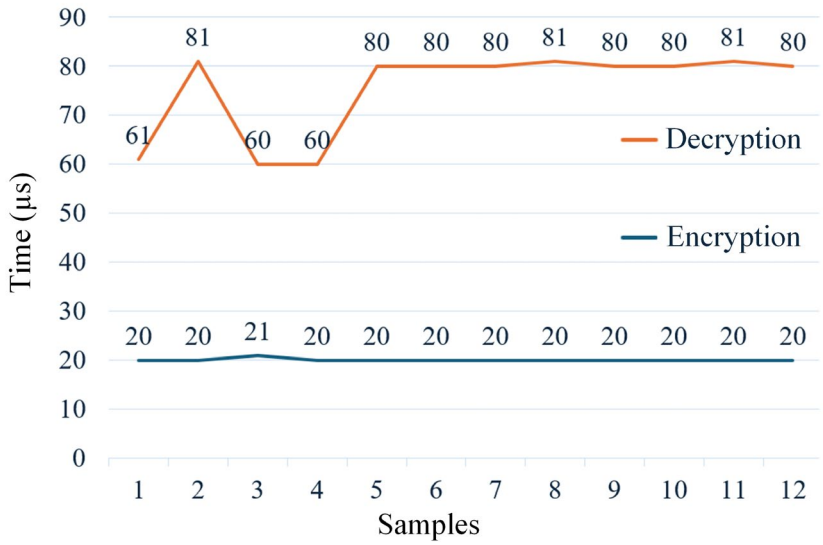

Fig. 13 *The encryption/decryption time of the controller 5370/CompactLogix PLC*

The implementation of the AES-128 approach on Siemens S7-1200 and Rockwell CompactLogix PLCs demonstrates its feasibility for enhancing industrial communication security and potential scalability to other control platforms. However, several limitations were observed as computational overhead, especially during decryption, may affect real-time applications; significant memory demand for S-boxes and round keys challenges resource-constrained PLCs; and the protection is limited to data payloads, leaving headers and function codes exposed. Furthermore, key management lacks secure storage mechanisms, synchronization with PLC scan cycles is required, and compliance with standards such as IEC 62443 remains unaddressed. Future work should focus on performance optimization, secure and efficient key management, and integration with standardized security frameworks.

Recent studies have demonstrated that embedding the AES encryption and decryption algorithms into FPGAs [17, 32] or microcontrollers [33] can be applied in real-time. The encryption and decryption times using AES-128 on the STM32F407 microcontroller are 9.65589 ms and 9.65850 ms, respectively [33]. This demonstrated there is no significant difference between encryption and decryption. Meanwhile, the encryption time on FPGA is 0.92 µs. In our current study, the AES-128-based encryption and decryption algorithm was embedded into the industrial PLCs. Our proposed approach demonstrated the strong data security while remaining compatible with real-time industrial communication requirements (150 µs for encryption/decryption and 5.00 ms for the total time for the encryption/decryption and transceivers through the Modbus TCP/IP protocol) that compared with the previous works [21, 33].

According to the industry 4.0 revolution, most devices could be controlled and monitored based on internet connections. The controlled systems face threats from the internet environment [3, 7, 14]. Therefore, communication and data security are prioritized. And they are usually applied for information technology-based applications. Recently, several works have effectively integrated the AES approach into FPGA [17, 23, 32] and microcontrollers [33] for the real-time encrypt- and decryption speed improvement based on the hardware components. And the modern PLC series allow embedding the intelligent algorithm based on the structure control language (SCL) [9]. This language is the foundation for integrating more useful and intelligent features into industrial PLCs. In our current work, the AES-128 algorithm was embedded into the industrial PLCs without changes of their hardware and firmware by using the SCL language. The proposed approach could be utilized for any industrial PLCs supporting the structured text. As of now, we have not found a similar solution that embeds the AES algorithm into industrial PLCs to achieve real-time data communication security.

Recently, the AES approach has been combined with the Particle Swarm Optimization (PSO) to solve the optimization problems [16] and block-chain applications [24]. Several studies have utilized machine learning (ML) based defense strategies for the industrial control systems [3, 4]. However, they have not been integrated into the hardware. Therefore, the ML based encryption and decryption should be embedded into the industrial PLCs to improve their data security and real-time application scenarios.

5. Conclusion

This study successfully implemented an end-to-end AES-128 encryption solution for Modbus TCP/IP protocol on industrial PLC platforms, effectively balancing data security with real-time performance requirements.

Experimental results demonstrate superior performance with 41.08μs average encryption time and 5-35ms end-to-end latency, meeting the demands of most industrial control applications while maintaining backward compatibility with legacy Modbus TCP devices. The solution's hybrid key management system and proven MITM (Man-in-the-Middle-Attack) attack resistance, verified through packet analysis, provide comprehensive security without requiring hardware upgrades. The consistent performance confirms its reliability for industrial deployment. The research establishes a practical framework for industrial cybersecurity by harmonizing cryptographic theory with operational constraints. Future work will focus on: (1) HMAC (Hash-based Message Authentication Code) integration for enhanced data integrity, (2) Architecture extension to PROFINET and other industrial protocols, and (3) Exploration of post-quantum cryptography for PLC systems. This work provides a critical reference for securing Industry 4.0 infrastructure, offering a viable path for implementing robust encryption in resource-constrained industrial environments while addressing both current and emerging cybersecurity challenges in the internet of things and digital transformation era.

Acknowledgement

We would like to thank Mr Van-Thinh Nguyen and Mr. Minh-Tien Pham for testing on S7-1200 (Siemens, Germany) and CompactLogix 5370 (Allen-Bradley, USA) at the Industrial Networks and Communication lab, Faculty of Automation Engineering, Can Tho University, Vietnam.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Hoang-Dung Nguyen, Tien-Trung Dai;* ***data collection:*** *The-Hien Huynh and Phu-Cuong Pham;* ***analysis and interpretation of results:*** *Hoang-Dung Nguyen, Tien-Trung Dai, Hoang-Dang Le;* ***draft manuscript preparation:*** *Hoang-Dung Nguyen, Tien-Trung Dai, Hoang-Dang Le. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] C. A. Fonseca, "SCADA System of Pipelines," in *Handbook of Pipeline Engineering*: Springer, 2024, pp. 1-28.

[2] I. Tomar, I. Sreedevi, and N. Pandey, "PLC and SCADA based real time monitoring and train control system for the metro railways infrastructure," *Wireless Personal Communications,* vol. 129, no. 1, pp. 521-548, 2023.

[3] M. Nankya, R. Chataut, and R. Akl, "Securing industrial control systems: Components, cyber threats, and machine learning-driven defense strategies," *Sensors,* vol. 23, no. 21, p. 8840, 2023.

[4] P. Upadhyay, "Industrial Control Systems (ICS) Security," *AI-Enhanced Cybersecurity for Industrial Automation,* p. 149, 2025.

[5] S. Sahlan *et al.*, "Dynamic Modelling of a Water Distribution Laboratory set-up System with SCADA capabilities," *International Journal of Integrated Engineering,* vol. 14, no. 1, pp. 203-214, 2022.

[6] F. Katulić, D. Sumina, S. Groš, and I. Erceg, "Protecting modbus/TCP-based industrial automation and control systems using message authentication codes," *IEEE access,* vol. 11, pp. 47007-47023, 2023.

[7] Q.-T. Dao, L.-T. Nguyen, T.-K. Ha, V.-H. Nguyen, and T.-A. Nguyen, "Investigation of Secure Communication of Modbus TCP/IP Protocol: Siemens S7 PLC Series Case Study," *Applied System Innovation,* vol. 8, no. 3, p. 65, 2025.

[8] P. W. Rusimamto, M. S. Munoto, I. G. A. Buditjahjanto, N. Luthfiyah, and E. M. Nuh, "Fluid mixing process based on programmable logic controller as training kit for electrical engineering education students," *International Journal of Integrated Engineering,* vol. 13, no. 4, pp. 104-111, 2021.

[9] V.-K. Nguyen, V.-K. Tran, H. Pham, H.-D. Nguyen, and C.-N. Nguyen, "Design and Implementation of Fuzzy-based Fine-tuning PID Controller for Programmable Logic Controller," *International Journal of Integrated Engineering,* vol. 16, no. 5, pp. 359-372, 2024.

[10] S. Kumar and H. Vardhan, "Cyber security of OT networks: A tutorial and overview," *arXiv preprint arXiv:2502.14017,* 2025.

[11] A. Presekal, V. S. Rajkumar, A. Ştefanov, K. Pan, and P. Palensky, "Cyberattacks on power systems," *Smart Cyber-Physical Power Systems: Fundamental Concepts, Challenges, and Solutions,* vol. 1, pp. 365-403, 2025.

[12] A. Januška, "Computer Network Modernization," 2024.

[13] Y.-C. Tian and J. Gao, "Network security and privacy architecture," in *Network Analysis and Architecture*: Springer, 2023, pp. 361-402.

[14] E. Avdibasic, A. S. Toksanovna, and B. Durakovic, "Cybersecurity challenges in Industry 4.0: A state of the art review," *Defense and Security Studies,* vol. 3, pp. 32-49, 2022.

[15] D. Chochtoula, A. Ilias, Y. C. Stamatiou, and C. Makris, "Integrating elliptic curve cryptography with the Modbus TCP SCADA communication protocol," *Future Internet,* vol. 14, no. 8, p. 232, 2022.

[16] P. Rodrigues, A. Singh, V. Kaushik, J. S. Chohan, M. A. Abdulrahman, L. Hussein, and Y.-L. Huamán-Romaní, "Optimal Operation of Autonomous Energy System Based on Multi-objective Approach," *International Journal of Integrated Engineering,* vol. 16, no. 9, pp. 385-397, 2024.

[17] M. M. M. Nadzri and A. Ahmad, "SoC FPGA-Based Rapid Prototyping of Compressed, Secured and Wireless Image Transmission for Wildlife Surveillance System," *International Journal of Integrated Engineering,* vol. 14, no. 4, pp. 276-285, 2022.

[18] P. Radoglou-Grammatikis, P. Sarigiannidis, G. Efstathopoulos, T. Lagkas, A. Sarigiannidis, V. Mladenov, and N. Siaxabanis, "Defending industrial Internet of Things against modbus/TCP threats: A combined AI-based detection and SDN-based mitigation solution," *TCP Threats: A Combined AI-Based Detection and SDN-Based Mitigation Solution,* 2022.

[19] M. K. Ferst, H. F. De Figueiredo, G. Denardin, and J. Lopes, "Implementation of secure communication with modbus and transport layer security protocols," in *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, 2018: IEEE, pp. 155-162.

[20] Z. Yang, Z. Bao, C. Jin, Z. Liu, and J. Zhou, "PLCrypto: A symmetric cryptographic library for programmable logic controllers," *IACR Transactions on Symmetric Cryptology,* pp. 170-217, 2021.

[21] A. Tidrea, A. Korodi, and I. Silea, "Elliptic curve cryptography considerations for securing automation and SCADA systems," *Sensors,* vol. 23, no. 5, p. 2686, 2023.

[22] F. Alonso, B. Samaniego, G. Farias, and S. Dormido-Canto, "Analysis of cryptographic algorithms to improve cybersecurity in the industrial electrical sector," *Applied Sciences,* vol. 14, no. 7, p. 2964, 2024.

[23] S. Sheikhpour, A. Mahani, and N. Bagheri, "Reliable advanced encryption standard hardware implementation: 32-bit and 64-bit data-paths," *Microprocessors and Microsystems,* vol. 81, p. 103740, 2021.

[24] D. Sarangi, "A comparative study of AES encryption modes and hashing for blockchain applications," Dublin, National College of Ireland, 2024.

[25] M. N. Alenezi, H. Alabdulrazzaq, H. M. Alhatlani, and F. A. Alobaid, "On the performance of AES algorithm variants," *International Journal of Information and Computer Security,* vol. 23, no. 3, pp. 322-337, 2024.

[26] A. Heryanto *et al.*, "Security and Performance Evaluation of PPTP-Based VPN with AES Encryption in Enterprise Network Environments," *Jurnal Teknik Informatika (Jutif),* vol. 6, no. 4, pp. 2171-2186, 2025.

[27] V. Z. González, E. Tena-Sanchez, and A. J. Acosta, "A Security Comparison between AES-128 and AES-256 FPGA implementations against DPA attacks," in *2023 38th Conference on Design of Circuits and Integrated Systems (DCIS)*, 2023: IEEE, pp. 1-6.

[28] A. Singh, "Testing the Impact of Encryption on Network Performance," 2022.

[29] I. R. Widiasari, "Combining advanced encryption standard (AES) and one time pad (OTP) encryption for data security," International Journal of Computer Applications, 2012.

[30] X. Gong, X. Zhang., Q. Wu, F. Zhang, J. Xu, Q. Shen, and Z. Zhang, "Practical Opcode-based Fault Attack on AES-NI," IACR Transactions on Cryptographic Hardware and Embedded Systems, vol. 2025, no.3, pp. 693-716, 2025.

[31] K. Song, S. Liu, H. Wang, S. Yang, L. Yan, S. Zhang, "Research on parallel AES encryption algorithm based on a ternary optical computer, Optics Communications," vol. 583, an.131660, 2025.

[32] G. G. Shet , V. Jamuna, S. Shravani, H. G. Nayana, and S. P. Kumar, "Implementation of AES Algorithm Using Verilog," *JNNCE Journal of Engineering & Management*, vol. 4, no.1, pp. 17-23, 2010.

[33] X. Hou and W. Wang, "Lightweight Dynamic Advanced Encryption Standard Encryption Based on S-Box Reconfiguration and Real-Time Key Expansion for Secure Over-the-Air Communication," *Electronics*, vol. 14, no. 16, an. 3274, 2025.