frontiers | Frontiers in Mechanical Engineering

TYPE Original Research
PUBLISHED 31 July 2024
DOI 10.3389/fmech.2024.1407665

Check for updates

OPEN ACCESS

EDITED BY
Hamid Reza Karimi,
Polytechnic University of Milan, Italy

REVIEWED BY
Jie Fan,
Beijing Institute of Technology, China
Saša Milojević,
University of Kragujevac Faculty of Engineering,
Serbia

*CORRESPONDENCE
Chenzhe Mu,
✉ muchenzhe2023@163.com

RECEIVED 27 March 2024
ACCEPTED 10 July 2024
PUBLISHED 31 July 2024

CITATION
Mu C (2024), Application of optimizing advanced encryption standard encryption algorithm in secure communication of vehicle controller area network bus.
Front. Mech. Eng 10:1407665.
doi: 10.3389/fmech.2024.1407665

Application of optimizing advanced encryption standard encryption algorithm in secure communication of vehicle controller area network bus

Chenzhe Mu*

School of Mechanical Engineering, Jiangxi Technical College of Manufacturing, Nanchang, China

Introduction: As the main means of information exchange within vehicles, the safety of the controller area network bus directly affects the safe operation of the vehicle and the safety of passengers' lives and property.

Methods: To enhance its secure communication function, this study utilizes advanced encryption standard algorithms and improves the S-box of the algorithm to solve problems such as extended processing time. A secure communication system for the local area network bus of the vehicle controller is designed based on optimized advanced encryption standard algorithms.

Results and Discussion: The results showed that when the file size was 200MB, the encryption and decryption time spent by the research method was 469.8 s and 528.5 s, respectively, which are significantly lower than traditional methods. In the simulation results, under both non-encrypted and encrypted transmission, the information remained intact throughout the entire transmission process. This indicated that the optimization algorithm effectively reduced encryption processing time and system resource consumption while ensuring data confidentiality and integrity. The new system meets the security requirements of the local area network bus of vehicle-mounted controllers.

Conclusion: This study not only enhances the security of in-vehicle networks but also promotes the application and development of related encryption technologies in the field of vehicle networking. It provides strong technical support for the further development of vehicle networking and the safe operation of intelligent vehicles.

KEYWORDS

AES encryption algorithm, CAN bus, S-box, encryption and decryption, secure communication

1 Introduction

The rapid development of Internet of Vehicles (IoV) technology has made the security of Controller Area Network (CAN) in vehicles an important issue. CAN is a serial bus network used for communication between various electronic control units within a vehicle. The design goal of the bus is to provide high-speed and reliable communication to ensure high flexibility and scalability in complex vehicle systems (Balaska et al., 2020; Bottarelli et al., 2021). Self-driving cars need to process large amounts of sensor data, actuator control signals, and command and status information between other electronic control units in real time via CAN. Any communication security vulnerability may lead to serious safety accidents. Therefore, improving the communication security of the on-board CAN bus is of great significance for the safe operation of self-driving cars (Bentoutou et al., 2020; Cai et al., 2020). Researchers have proposed a variety of cryptographic algorithms and security protocols to deal with potential CAN network threats. For example, public key infrastructure-based authentication mechanisms and symmetric key-based encryption methods have been widely used in in-vehicle networks to improve communication security. Furthermore, machine learning and artificial intelligence techniques have been integrated into the field of in-vehicle security with the objective of enhancing the system's protection by enabling the intelligent detection and prevention of attacks (Khan et al., 2023; Panic et al., 2023). Nevertheless, these methods remain inadequate in terms of processing delay and system resource consumption, which is not conducive to the demand for efficient and real-time communication in self-driving cars. Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm used to protect the security of electronic data (Gong et al., 2021). Due to its high efficiency and security, AES is widely used in various data encryption scenarios, including government, finance, and private sectors. However, traditional AES algorithms have problems such as long processing delays and high resource consumption in vehicle environments, which are not conducive to fast and efficient vehicle communication. Therefore, this study adopts a new affine transformation method to improve the S-box and optimize the key extension algorithm, aiming to optimize AES to meet the dual requirements of secure and efficient communication for the vehicle CAN bus. This is also the innovation of this study. The main contribution of the research is that the proposed novel in-vehicle CAN encryption method can reduce the encryption and decryption processing time and system resource consumption, improve the efficiency and security of the algorithm, and guarantee the data confidentiality and integrity of the in-vehicle network. Compared with the existing research, the difference of this research is the introduction of a new affine transformation method and an optimized key extension algorithm, which improve the applicability of the AES algorithm in the in-vehicle environment. The results not only enhance the security of in-vehicle networks, but also promote the application and development of related cryptographic techniques in the field of Telematics. This provides a robust technical foundation for the continued evolution of Telematics and the secure operation of intelligent vehicles. The research content is mainly divided into four parts. Part 1 is a literature review that summarizes the current research achievements in communication security and encryption algorithms. Part 2 is the research method, which mainly optimizes the AES algorithm and designs the SC-VCANb network based on it. Part 3 mainly analyzes the simulation results of the research methods. Part 4 is the conclusion, summarizing the research findings and shortcomings.

2 Related work and significance

The advancement of information technology and network communication has made communication security an important challenge facing the world, especially in the fields of the Internet of Things (IoT) and high-speed physical layers. Researchers are constantly exploring and developing new algorithms to address increasingly complex security threats (Ametepe et al., 2022). Kumari et al. (2022) proposed a signature-based hash multiplication to address the security communication issues exacerbated by the use of 6G technology in IoT networks. The signature algorithm was enhanced through the Bernoulli Karatsuba multiplication algorithm, achieving data protection, and the secure communication performance was significantly improved in the model results (Kumari et al., 2022). To address the global technical challenges of providing physical layer security in fiber optic communication systems, Gao et al. (2022) proposed a pure hardware optical encryption scheme based on time extension and self-feedback phase encryption. This scheme achieved high bit rate distance product recording of 6400 Gb/s km by securely transmitting 32 Gb/s confidential signals over a 200 km fiber optic link. It demonstrated full compatibility with traditional optical transmission systems and enormous potential for future ultra high speed physical layer secure optical communication (Gao et al., 2022). To address the security issues of IoT devices during communication, Kannan et al. (2021) proposed a technique based on deep artificial structure learning. Through multi-layer training and knife cut regression Schmidt Samoa encryption, this technology ensured high data confidentiality while achieving faster processing speed and low memory usage. Compared with existing technologies, this method had significant advantages in confidentiality and processing efficiency (Kannan et al., 2021). Gao et al. (2020) proposed a symbol by symbol optical phase encryption technology based on ultra long, reconfigurable optical phase modes and commercial dispersion elements to address the challenges of high-speed physical layer security. This technology has successfully achieved secure optical communication of 40 Gb/s differential phase shift keying modulation signals in experiments, demonstrating its high security and robustness against eavesdropping attacks under different optical codes (Gao et al., 2020). Li and Wu proposed a node oriented secure data transmission algorithm to address the security issues of mobile node data transmission in IoT social networks. This algorithm could identify the authenticity of nodes, analyze and prevent malicious behavior, and ensure the secure transmission of data. This algorithm has been proven to effectively improve the security and success rate of data transmission (Li and Wu, 2020).

Encryption algorithm is one of the core technologies for protecting communication security, with a wide range of applications, covering direct data transmission from the physical layer to data processing and storage at the application layer. The research of different scholars has demonstrated the diversity and applicability of encryption technology. Lin et al. (2022) proposed a new method combining chaotic synchronization and recursive fuzzy brain emotion learning cerebellar model joint controller to address the synchronization problem in chaos-based secure communication. This method synchronized chaotic signals between the transmitter and receiver, and encrypted and decrypted the information. Stability analysis and audio and image simulation data indicated that this method was effective and superior in secure communication (Lin

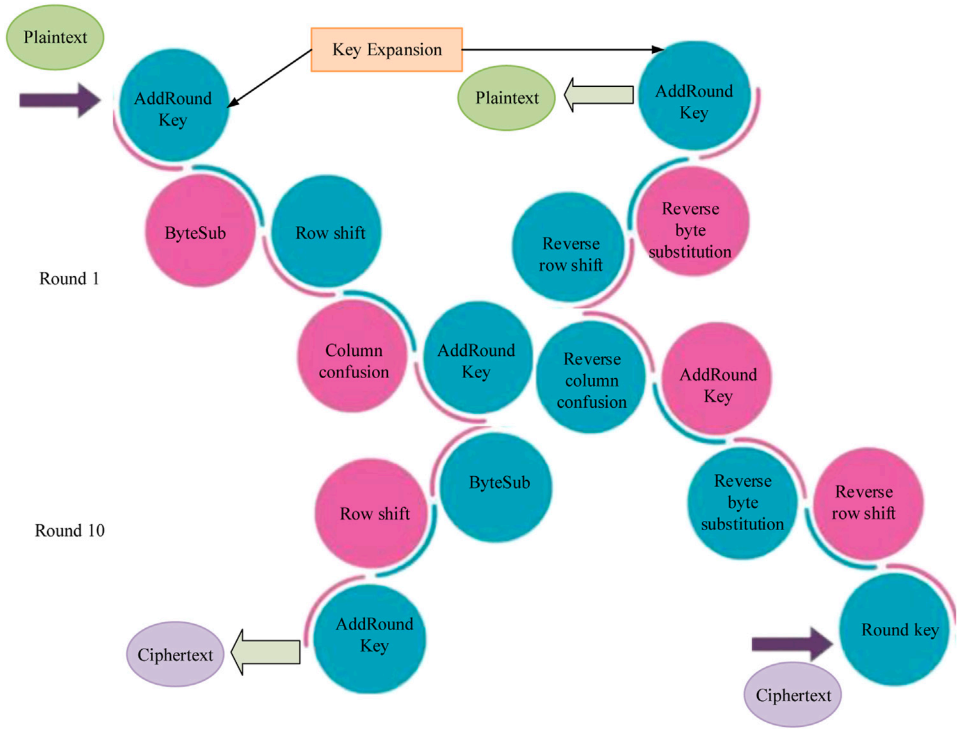

FIGURE 1
AES algorithm workflow.

et al., 2022). To improve the difficulty of anti-counterfeiting, Luo et al. (2021) proposed an encryption method that utilizes short and long fluorescence lifetimes and their weighted average lifetime parameters. This method obtained four fluorescence lifetime parameters through double exponential fitting, and could achieve asymmetric, multi-level, switchable, and reversible encryption of these parameters, effectively improving the encryption complexity (Luo et al., 2021). To enhance the security of image encryption, Jia integrated chaotic systems and linear functions and proposed a cross color field obfuscation method. This method scrambled pixels in a tricolor matrix, which can effectively resist various attacks and has excellent image encryption performance (Jia, 2020). To ensure the security of electronic medical record data in smart healthcare systems, Anand et al. (2020) developed a dual watermarking technique based on compression and encryption. This technology utilized multiple transformations to embed dual watermarks, and through hierarchical tree set segmentation compression and watermark containing image encryption, it outperformed existing technologies in terms of robustness and security (Anand et al., 2020).

The above research indicates that ensuring communication security is an urgent need in different fields such as 6G, optical communication systems, and IoT devices, and encryption algorithms play a core role in modern communication security systems. Given the low processing efficiency and high resource consumption of traditional AES algorithms in vehicle environments, this study utilizes affine transformation to improve the S-box, thereby optimizing the AES algorithm and ensuring high security of the vehicle CAN bus during efficient communication.

3 Application of improved-AES encryption algorithm in SC-VCANb

This study first applies the new affine transformation (“A7”, “6F”) to obtain a new S-box and optimizes the traditional AES encryption algorithm. Then, a SC-VCANb scheme based on improved-AES encryption algorithm is designed, and data encryption and decryption are carried out using this optimization algorithm.

3.1 Optimization of AES encryption algorithm

AES is a popular symmetric encryption algorithm used to ensure the security of electronic data. AES supports three key lengths: 128 bit, 192 bit, and 256 bit, all using a block length of 128 bits. It performs the encryption process through a series of encryption steps, such as byte replacement, row shifting, column mixing, etc (Hao et al., 2020; Hong et al., 2020). Taking AES-128 as an example, its encryption process consists of 10 iterations, with the first 9 rounds having the same structure, while the last round omits the column mixing step. The decryption process is the inverse of encryption, and each round of encryption requires the use of sub keys generated through key expansion. The workflow of the AES algorithm is Figure 1 (Yumin et al., 2023).

In Figure 1, the encryption process of the AES encryption method is divided into 10 iteration rounds, each round including four key steps: byte replacement, row displacement, column mixing, and round key addition. This process begins with a 16 byte original key, which is extended to generate a series of sub-keys. To perform multiple iterations of the AES encryption algorithm, the initial 16 byte key is divided into a 4*4 matrix, with each column containing 4 bytes, forming a matrix array called W (Liu et al., 2022). Due to the fact that each round of encryption process requires the use of different sub-keys for round key addition, and a round key addition is also required before the round function is executed, a total of 11 matrix sized keys need to be expanded to generate 44 column W matrices. The calculation formula is Eq. 1.

$$W[n] = \begin{cases} W[n-4] \oplus W[n-1], if\, n\%4 = 0 \\ W[n-4] \oplus Mix\left(W[n-1] \oplus rcon\left[\frac{n}{4} - 1\right]\right), if\, n\%4! = 0 \end{cases} \tag{1}$$

Eq. 1 is cited in the literature (Alexandrov and Tikhonov, 2022). n represents the key. In AES key extension, there is a function T that calculates the matrix (Alexandrov and Tikhonov, 2022). Its expression is Eq. 2.

$$Min(x) = SubWord(RotWord(x)) \tag{2}$$

Eq. 2 is cited in the literature (Cecchinato et al., 2023). $RotWord()$ represents a left shift in a loop, and $SubWord$ represents byte replacement (Cecchinato et al., 2023). Each byte is processed through a specific non-linear transformation (S-box transformation) during replacement. The S-box itself is a preset 16*16 matrix used to achieve byte level nonlinear transformations. This transformation process includes initial settings, mapping to the Galois domain, and subsequent affine transformations. The AES algorithm also includes two important steps, row shifting and column mixing, to enhance the complexity and security of encrypted data. Row shift is a linear transformation that improves the level of data confusion by rearranging rows in a matrix. Column mixing is the process of multiplying a matrix that has undergone row shift transformation with a predefined matrix to ensure sufficient mixing of data between columns. Finally, the round key addition adds the encrypted data from each round to the round key in the Galois domain, which is equivalent to performing an Exclusive OR (XOR) operation to ensure the diversity of encryption results for different rounds.

The key extension module plays a crucial role in encryption, as it quickly generates the required keys for each round from the initial key, ensuring the efficiency of encryption. At present, the improvement of AES algorithm focuses on two aspects. Firstly, due to the short cycle period of the existing S-box, it is easy to become a weak link in the algorithm being cracked. Therefore, it can enhance the iteration cycle period and nonlinear characteristics of the S-box to improve security. By adopting a new affine transformation method, it is possible to develop an advanced S-box with an iteration cycle that satisfies the full capacity of GF (2^8) space. The second is to optimize the key extension algorithm to reduce the correlation between seed keys and round keys, and use one-way generation strategies and random functions or cyclic shift techniques to increase the difficulty of cracking. Considering that the second method may sacrifice the immediacy of the algorithm, this study aims to address the fragility of the S-box and key extension in the AES algorithm to design a more secure encryption scheme.

The S-box operation in the AES algorithm is a reversible nonlinear transformation performed in the GF (2^8) domain, expressed as Eq. 3.

$$BS\left(a_{i,j}\right) = \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} a_{i,j}^{-1} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{3}$$

Eq. 3 cited in literature (JosephNg et al., 2025). $a_{i,j}^{-1}$ represents the multiplication inverse of $(a_{i,j})$ in the GF (2^8) domain (JosephNg et al., 2025). The expression derived from Eq. 3 is shown in Eq. 4.

$$a_{i,j}^{-1} = \begin{cases} \left(a_{i,j}\right)^{254}, a_{i,j} \neq 0 \\ 0, a_{i,j} = 0 \end{cases} \tag{4}$$

Due to the operation is performed on the GF (2^8) field, the result of the operation is also on the GF (2^8) field, and the final S-box obtained through this transformation is a matrix composed of 16*16 bytes. Non-linearity is generated by the inverse of multiplication. However, the short iteration period of traditional S-boxes in the GF (2^8) field limits their algebraic properties and the overall anti-attack performance of the algorithm. Changing the parameters of affine transformation and adjusting the computational process of S-box can effectively prolong the cycle period and enhance its algebraic structure. Therefore, this study will use new affine transformations ("A7," "6F") to obtain new S-boxes. The first is to perform an affine transformation on ("A7," "6F"), then to calculate the inverse of the multiplication and perform the same affine transformation again to produce the final output. Compared with the traditional AES algorithm's S-box, the improved S-box increases the algebraic expression of the number of terms to 9–255 through additional affine transformations, enhancing complexity and overcoming the problem of simple structure in the original S-box. Choosing different affine transformation pairs can prolong the affine transformation and iteration period of the S-box, and improve the security of the algorithm.

In Table 1, the affine transformation period of the improved S-box has increased to 16, and the iteration period has reached 256, which is a significant improvement compared to the original S-box's 4 and periods less than 88. The nonlinear characteristics and anti-cryptanalysis ability of the new S-box have been optimized, while maintaining the original query efficiency.

3.2 Design of SC-VCANb based on improved-AES encryption algorithm

The confidentiality of network data is a key indicator for evaluating security, especially in CAN systems where data protection is particularly important. The CAN bus transmits un-encrypted messages, which are easily monitored and stolen. Attackers may use this data for malicious operations, threatening the safety of vehicles (Hu et al., 2021). Introducing improved-AES encryption algorithm is an effective means to ensure the confidentiality of data in the CAN bus. In this scheme, the

TABLE 1 Improved S-box.

Hexadecimal	High 4 bits of S-box																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Lower 4 bits of S-box	0	6F	20	C0	A5	B0	D6	0A	A0	88	2A	B3	1B	D5	BD	80	F7
	1	9C	E8	C5	19	09	2D	55	DE	32	E4	0E	5C	90	BC	2B	4B
	2	96	8C	AC	49	3A	B4	5B	2E	54	8B	46	E9	72	81	BF	85
	3	C1	CE	A2	12	D7	47	F6	22	98	00	86	C3	4D	7E	7D	AE
	4	93	DF	9E	B9	8E	3D	74	60	CD	68	8A	4A	75	F4	C7	5A
	5	FA	0F	1D	FF	FB	BB	24	43	E1	AB	18	06	07	B5	1A	77
	6	38	91	B7	3E	89	82	D1	6D	3B	50	73	79	A3	ED	C9	D9
	7	94	39	D0	BA	9B	A7	31	35	76	A1	EF	CC	6E	E2	87	56
	8	19	78	37	C8	9F	70	0C	EB	97	E5	4E	DA	EA	F1	E2	1C
	9	36	B1	EC	5D	95	2C	F5	3F	62	A6	AA	B2	33	A8	FD	D8
	A	AD	E6	5F	CA	5E	AF	27	C6	25	61	05	58	C2	1E	71	CB
	B	28	F2	0D	64	D4	D3	DB	EE	53	11	02	7A	DD	92	6B	A4
	C	C4	FC	10	F0	0B	D2	CF	F9	14	6A	99	52	30	41	66	51
	D	45	93	F8	84	69	44	6C	16	01	63	26	9D	34	7C	3C	FE
	E	9A	67	4C	59	B8	B6	8D	29	15	65	03	23	40	48	42	04
	F	E3	57	08	4F	2F	1F	BE	8F	E7	7B	A9	21	13	7F	F3	DC

sending node first encrypts the data before sending it out. After receiving encrypted data, the receiving node decrypts it to obtain the original information. This processing ensures that the data transmitted on the bus is encrypted, thereby avoiding the risk of data being easily read or stolen by unauthorized nodes. The process of designing secure communication is Figure 2.

In the secure communication framework of CAN bus, the sending node encrypts plaintext data P into ciphertext C by improved-AES encryption algorithm E and key k before transmission, ensuring the security of the transmitted data. After receiving the ciphertext, the receiving node must use the same key k to decrypt C, restore the original plaintext data P, and maintain the effective operation of the network. The encrypted expression for the sending node data after derivation from the AES formula is shown in Eq. 5.

$$E_k(P) = C \tag{5}$$

The receiver node data decryption expression is derived as shown in Eq. 6.

$$D_{k^{-1}}(C) = P \tag{6}$$

The data that is encrypted and then decrypted should meet the derivation Eq. 7.

$$D_{k^{-1}}(E_k(P)) = P \tag{7}$$

Based on the optimization of the AES encryption algorithm, which is a symmetric encryption algorithm, the key k used for encryption and the key used for decryption are the same. Therefore, the encryption process of the sending node and the decryption process of the receiving node are consistent and symmetrical. The SC-VCANb system mainly consists of three parts: the data encryption module of the sending node, the data transmission module of the CAN bus, and the data decryption module of the receiving node. The entire system relies on improved-AES encryption algorithm for data encryption and decryption. The sending and receiving nodes share a user-defined 16 byte AES key and use a 16 byte plaintext plain consisting of an 8-byte counter and 8 bytes of additional information. In the sending node, the data is first AES encrypted using key and plain to generate a 16 byte ciphertext ciphertext. Then, 8 bytes are extracted from the low bit of the cipher to generate the encrypted session key k. When the sending node is preparing to send the message, it performs XOR operation on the key k and 8-byte plaintext data to obtain the ciphertext data, which is then sent through the CAN bus. The encryption part of the sending node is to convert the plaintext data of the sending node into ciphertext data, as shown in the flowchart in Figure 3.

In Figure 3, after the system is started, the sending node generates a 16 byte algorithm ciphertext using the improved-AES encryption algorithm through the algorithm key and algorithm plaintext plain. Combining the key generation rules, the encrypted session key k is obtained by extracting the low bit from the algorithm ciphertext cipher. The main task of the receiving node is to restore the received ciphertext to plaintext, which is the opposite of encryption. The decryption process of the receiving node is Figure 4.

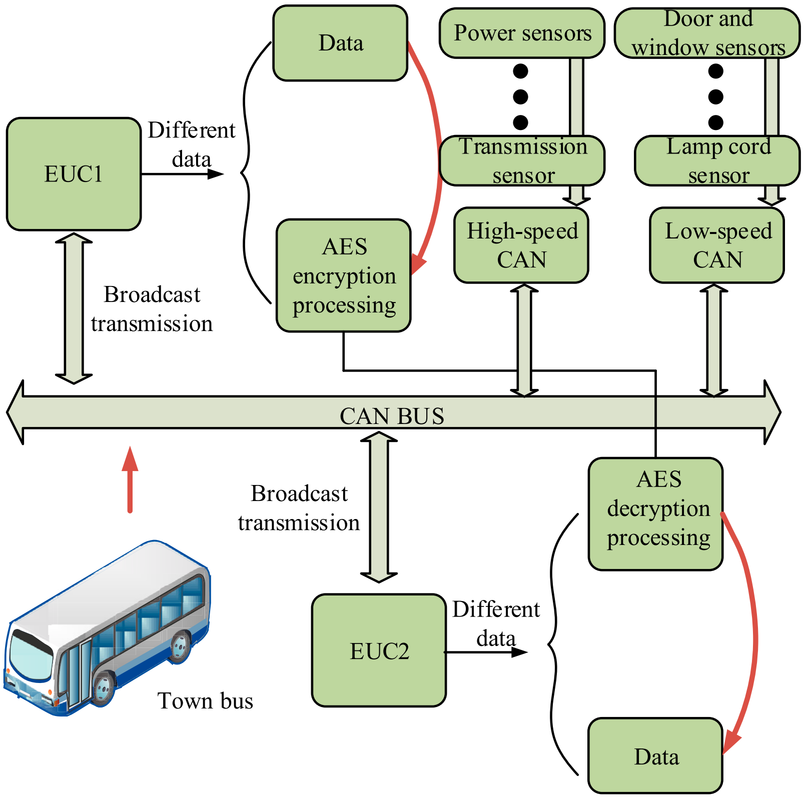

FIGURE 2
Security communication design process.

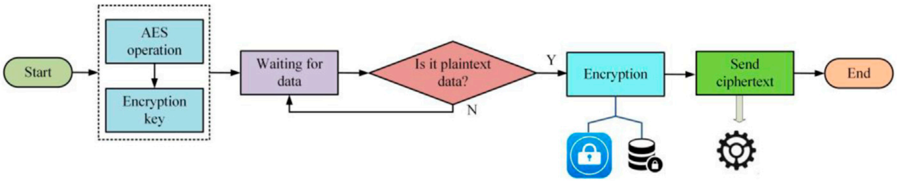

FIGURE 3
Process of sending node encryption.

In Figure 4, in a symmetric encryption system, the session key used for data encryption is also used for data decryption, and the decryption process uses the same key G as the sending node. The receiving node must decrypt all received packets, which requires the node to be in a waiting state before receiving the ciphertext. Once the ciphertext is received, the key G is utilized and converted into plaintext for subsequent operations. After decryption is completed, the node waits for the new ciphertext to arrive to repeat the decryption process.

During system initialization, the sending and receiving nodes on the CAN bus use the same improved-AES encryption algorithm key and plaintext plain for initialization. By optimizing the AES encryption algorithm for 10 iterations, a 16 byte ciphertext cipher can be generated. The derived expression is shown in Eq. 8.

$$cipher = AES(key, plain) \tag{8}$$

The session key G, with a length of 8 bytes, is extracted from the low bit of the 16 byte ciphertext cipher generated by the algorithm. The derived expression is shown in Eq. 9.

$$G = cut\left(cipher\right)_{8byte} \tag{9}$$

Sending node ECU_{tx} uses session key G to perform XOR operation on plaintext data D, generating ciphertext data D, and achieving data encryption. The derived expression is shown in Eq. 10.

$$D' = G \oplus D \tag{10}$$

Receiving node ECU_{tx} successfully restores plaintext data D by performing XOR on the received ciphertext data D' using the same session key G as the sending node, completing the decryption process. The derived expression is shown in Eq. 11.

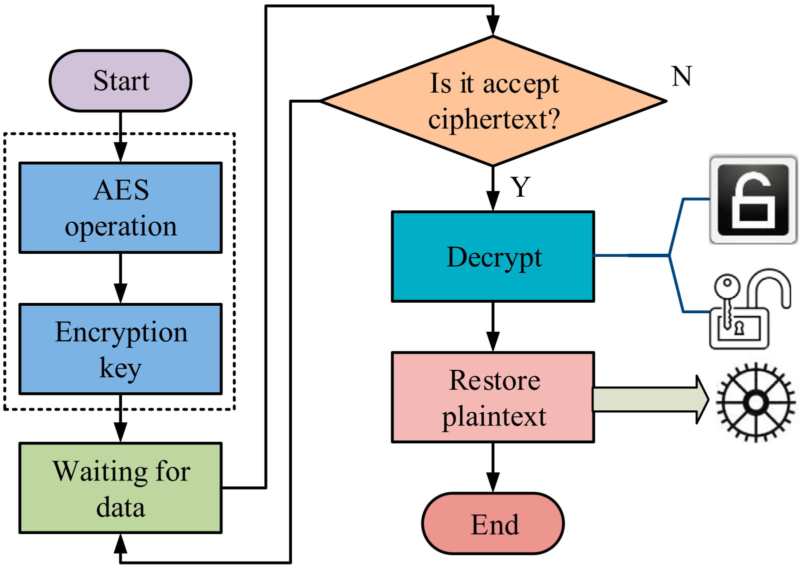

FIGURE 4
Receiving node decryption process.

$$D = D' \oplus G \tag{11}$$

Once decryption is complete, the receiving node executes the requisite functions in accordance with the instructions set forth in the plaintext data. This ensures the security and confidentiality of the CAN bus secure communication system, effectively preventing unauthorized access to sensitive information.

4 Application analysis of improved-AES encryption algorithm in SC-VCANb

Firstly, the optimization performance of the AES encryption algorithm was evaluated, followed by an in-depth exploration of the SC-VCANb scheme based on this optimization algorithm. The effectiveness and practicality of the encryption scheme were verified by comparing the data transmission effects before and after encryption.

4.1 Performance analysis of improved-AES encryption algorithm

To analyze the performance of the improved-AES encryption algorithm, plaintext data files with sizes of 10MB, 50MB, 100MB, 150MB, and 200 MB were selected as the dataset, and their encryption time and effectiveness were compared. To increase the richness of the experiment, a comparison was made between the improved-AES encryption algorithm and the traditional AES encryption algorithm, as well as the Advanced Encryption Standard in IoT (AES-I) data encryption method. Table 2 compares the software and hardware environments of the experiment.

The operating system used to set the version number of the software used in the on-board test-bed is Windows 10 Professional 64-bit, with the development environment of Visual Studio Code. The programming language is Python 3.8, the CAN bus network simulator is CANoe, and the data analysis tool is Matlab. Firstly, the time spent on encrypting and decrypting plaintext data of different file sizes using different algorithms is analyzed, and the results are shown in Figure 5.

Figure 5A shows the encryption time of different algorithms for different file sizes. The results show that the encryption time of the optimized AES encryption algorithm is significantly lower than that of the traditional AES and AES-I encryption algorithms. For example, for a 200 MB file, the encryption time of the optimized AES encryption algorithm is 469.8 s, while that of the traditional AES and AES-I encryption algorithms is 524.6 s and 502.1 s, respectively. This indicates that the optimized AES encryption algorithm further reduces the time required for encryption while ensuring high efficiency. Figure 5B shows the decryption time of different algorithms for different file sizes. The optimized AES encryption algorithm also shows a significant advantage in decryption time. For a 200 MB file, the decryption time of the optimized AES encryption algorithm is 528.5 s compared to 632.4 s and 592.3 s for the traditional AES and AES-I encryption algorithms, respectively. This indicates that the efficiency of the optimized algorithm in the decryption process has also been significantly improved. Next, the acceleration ratio effect is analyzed, and the acceleration ratio results of the three algorithms are shown in Figure 6.

As can be seen from Figure 6, the optimized AES encryption algorithm is slightly lower than the traditional AES encryption algorithm and AES-I encryption algorithm in terms of overall performance. This is mainly due to the fact that the optimized AES encryption algorithm requires more computation time in the S-box generation and keystream sequence generation phases. Although the speedup ratios of the optimized AES encryption algorithm are slightly lower than the original algorithm, they are all very close to the theoretically desirable values4. This indicates that the experimental results of improved-AES are consistent with theoretical analysis, and even with additional computational steps,

TABLE 2 The software and hardware environment of the experiment.

Software and hardware environment	Name	Project	Unit	Accoutrements	Producers
Hardware	Processor	Intel Core i7-9700K	/	Santa Clara, California, United States	Intel Corporation
	Memory	16 (DDR4 RAM)	GB		
	Hard disk	1 (SSD)	TB		
	Network	1000 (Ethernet)	Mbps		
Software environment	Operating system	Windows 10 Professional 64	/	Redmond, Washington, United States	Microsoft Corporation
	Development environment	Visual Studio Code	/	Redmond, Washington, United States	Microsoft Corporation
	Programming Language	Python 3.8	/	Beaverton, Oregon, United States	Python Software Foundation
	Encryption library	PyCrypto 2.6.1	/	Beaverton, Oregon, United States	Python Software Foundation

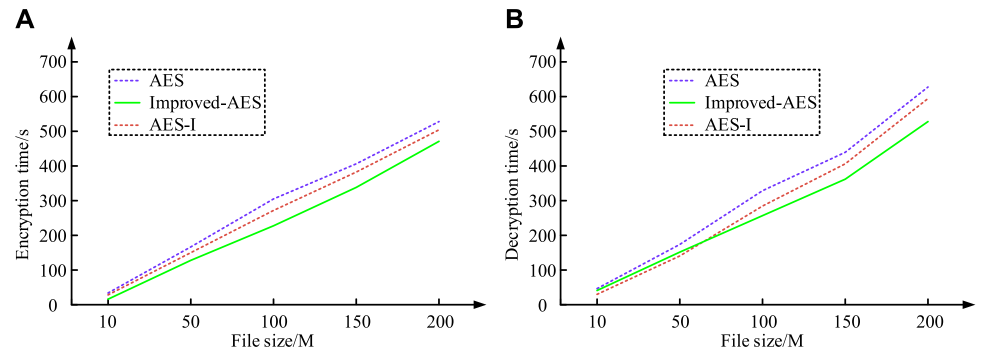

FIGURE 5
Spending time of different algorithms to encrypt and decrypt plaintext data of different file sizes. **(A)** Encryption time **(B)** Decryption time.

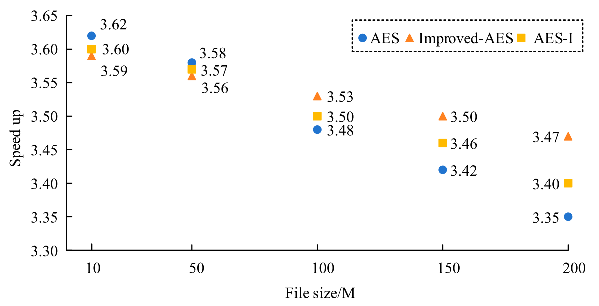

FIGURE 6
Acceleration ratio of three algorithms.

the improved-AES algorithm still achieves significant performance improvement in the same environment. Finally, the CPU usage of three algorithms during the encryption and decryption process is analyzed, as shown in Figure 7.

Figures 7A, B respectively represent the CPU usage of three algorithms for encrypting and decrypting plaintext data of different file sizes. For files of different sizes, the CPU usage trend for encryption and decryption is the same for each algorithm, and the CPU usage for decryption is higher than that for encryption. Overall, the three algorithms occupy CPU in descending order: traditional AES, AES-I, and improved-AES. Therefore, the performance of the research method is relatively better, which proves that the method is progressiveness and can be applied to SC-VCANb.

4.2 SC-VCANb simulation analysis

To verify the application of improved-AES in SC-VCANb, a dedicated experimental environment is first set up, as shown in Table 3.

Data transmission in non-encrypted state in the set CANoe simulation network was verified. To ensure the reproducibility of the results, the experiment was repeated several times under the same simulation conditions. The encryption and decryption process was repeated 10 times for each file size and the encryption and decryption times were recorded each time. The results of these repeated experiments were averaged through. This method of repeating the experiments ensured the reliability and

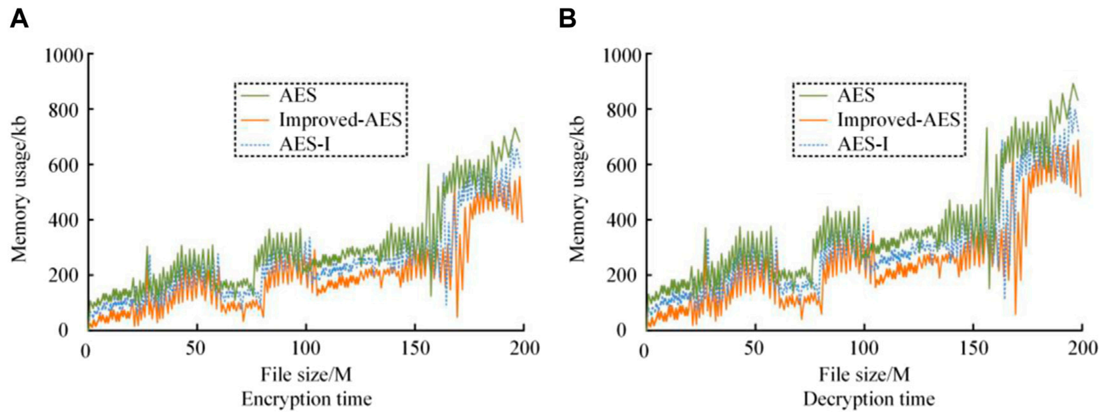

FIGURE 7
CPU usage during encryption and decryption using three algorithms. **(A)** Encryption time **(B)** Decryption time.

TABLE 3 Vehicle CAN bus safety communication simulation experimental environment.

Software and hardware environment	Name	Project	Accoutrements	Producers
Hardware	CAN bus network simulator	CANoe	Stuttgart, Baden-Württemberg, Germany	Vector Informatik GmbH
	ECU simulator	Sending and receiving nodes	Stuttgart, Baden-Württemberg, Germany	Vector Informatik GmbH
	Data collection and analysis equipment	Logic analyzer pr PC	Natick, Massachusetts, United States	MathWorks
Software environment	Operating system	Windows	Redmond, Washington, United States	Microsoft Corporation
	CAN network simulation and analysis software	CANoe	Stuttgart, Baden-Württemberg, Germany	Vector Informatik GmbH
	Encryption algorithm	Improved-AES	Belgium	Joan Daemen and Vincent Rijmen
	Data analysis tools	Matlab	Natick, Massachusetts, United States	MathWorks

reproducibility of the results. The Engine node acted as the data sender, sending the Engine Speed signal to the network, while the Display node acted as the receiver. The non-encrypted results of data transmission on the network, Engine node sending data, and Display node receiving data are shown in Figure 8.

Figures 8A–C represent the non-encrypted results of data transmission on the network, data transmission by the Engine node, and data reception by the Display node, respectively. It can be seen from Figures 8A–C that the data curves of the sending node, the network transmission, and the receiving node completely overlap, indicating that the data is not encrypted in any way in the case of non-encrypted transmission, and cannot provide a security feature to prevent data theft. When conducting encrypted transmission tests in the CAN bus network simulated by CANoe, Figure 9 shows the un-encrypted Engine Speed signals of the Engine node, the encrypted data in network transmission, and the data received and decrypted by the Display node.

Figures A–C represent the encryption results of data transmission on the network, Engine node sending data, and Display node receiving data, respectively. The un-encrypted data of the sending node coincides with the decrypted data curve of the receiving node, indicating that the information has remained intact throughout the entire transmission process, and the encryption operation has not affected the normal operation of the node. In addition, as the data transmitted through the network is different from the original and decrypted data of nodes, it proves that the network has anti-theft capabilities. In addition, since the data transmitted by the network is different from the original and decrypted data of the nodes, it is proved that the encrypted transmission possesses an anti-stealing function. Overall, the CAN bus network can achieve effective data encryption transmission and ensure that the receiving node can correctly decrypt and the network can still operate normally. This result verifies that the CAN bus security communication design based on improved-AES effectively improves the network's anti-theft ability and meets the original intention of security design. To verify whether the above results can be described by a mathematical model, the study uses relevant mathematical methods to analyze the experimental data. Through the regression analysis of the relationship between encryption and decryption time and file size,

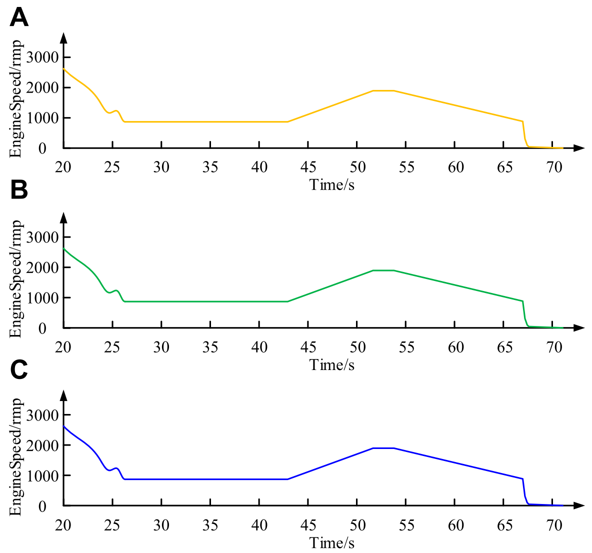

FIGURE 8
Un-encrypted network data. **(A)** Transferring data over the network **(B)** Engine node sends data **(C)** Display node receives data.

the corresponding mathematical model is established. By building the mathematical model, the study predicts the quantities and parameters that may be obtained under other simulation conditions, i.e., decryption time and CPU usage. The prediction results of the model are consistent with the actual experimental data, proving the validity and predictability of the model.

5 Conclusion

In response to the security threats such as information theft and tampering faced by the CAN bus, this study first optimized the AES algorithm by improving the S-box to make it more suitable for the security communication needs of the vehicle CAN bus, and then developed a security communication design. Experimental data showed that when the file size was 200MB, improved-AES saved 54.8 s and 103.9 s in encryption and decryption time compared to traditional AES and AES-I algorithms, respectively, while also occupying the least CPU resources. These results validate the expectations and assumptions of the study that the improved-AES encryption algorithm is able to significantly reduce the computation time and system resource consumption while maintaining high efficiency. The experimental results are consistent with the expectations, and there are no obvious deviations or completely opposite results. Based on the analysis and discussion, the application of the improved-AES encryption algorithm in the secure communication of in-vehicle CAN bus shows significant performance enhancement. The study shows that the algorithm can not only enhance the security of the in-vehicle network, but also promote the application and development of related encryption technologies in the field of Telematics. However, the study still has some shortcomings, such as not effectively reducing the encryption and decryption execution time of the AES algorithm. Future research can further explore the performance of the optimization algorithm under other simulation conditions and use mathematical models for error prediction. Especially when dealing with larger datasets or more complex network environments, such as IoT devices, smart homes

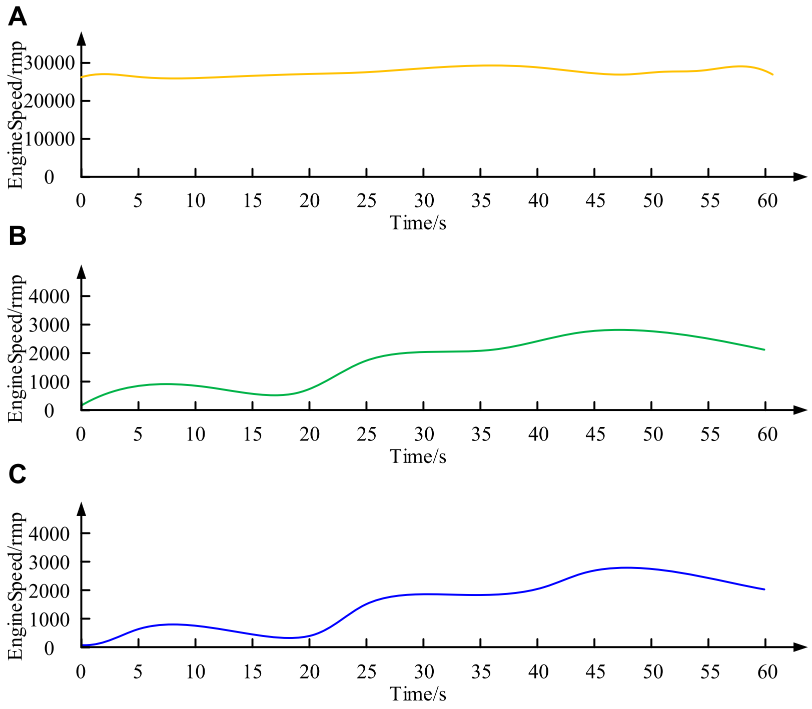

FIGURE 9
Encrypt network data. **(A)** Transferring data over the network **(B)** Engine node sends data **(C)** Display node receives data.

and industrial control systems, the above algorithms can be applied to optimize and adapt them to meet the requirements of practical applications. This research makes a significant contribution to the field by proposing an enhanced AES encryption algorithm based on an improved S-box. Furthermore, it develops a secure communication scheme for in-vehicle CAN bus networks, which is founded upon the aforementioned algorithm. It not only provides technical support for the secure communication of in-vehicle network, but also lays the foundation for the promotion and development of other application fields in the future.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

CM: Conceptualization, Data curation, Formal Analysis, Methodology, Project administration, Resources, Validation, Writing–original draft, Writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. The research is supported by Science and technology research project of Jiangxi Provincial Department of Education, Design of Multi-ECU Body Control System based on CAN Bus, PROJECT NO: GJJ204709.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Alexandrov, A. Y., and Tikhonov, A. A. (2022). Electrodynamic control with distributed delay for AES stabilization in an equatorial orbit. *Cosmic Res.* 60 (5), 366–374. doi:10.1134/s0010952522040013

Ametepe, A. F. X., Ahouandjinou, A. S. R. M., and Ezin, E. C. (2022). Robust encryption method based on AES-CBC using elliptic curves Diffie–Hellman to secure data in wireless sensor networks. *Wirel. Netw.* 28 (3), 991–1001. doi:10.1007/s11276-022-02903-3

Anand, A., Singh, A. K., Lv, Z., and Bhatnagar, G. (2020). Compression-then-Encryption based secure watermarking technique for smart healthcare system. *IEEE Multimed.* 27 (4), 133–143. doi:10.1109/mmul.2020.2993269

Balaska, N., Ahmida, Z., Belmeguenai, A., and Boumerdassi, S. (2020). Image encryption using a combination of Grain-128a algorithm and Zaslavsky chaotic map. *IET Image Process.* 14 (6), 1120–1131. doi:10.1049/iet-ipr.2019.0671

Bentoutou, Y., Bensikaddour, E., Taleb, N., and Bounoua, N. (2020). An improved image encryption algorithm for satellite applications. *Adv. Space Res.* 66 (1), 176–192. doi:10.1016/j.asr.2019.09.027

Bottarelli, M., Karadimas, P., Epiphaniou, G., Ismail, D. K. B., and Maple, C. (2021). Adaptive and optimum secret key establishment for secure vehicular communications. *IEEE Trans. Veh. Technol.* 70 (3), 2310–2321. doi:10.1109/tvt.2021.3056638

Cai, Y., Chen, X., Tian, L., Wang, Y., and Yang, H. (2020). Enabling secure NVM-based in-memory neural network computing by sparse fast gradient encryption. *IEEE Trans. Comput.* 69 (11), 1596–1610. doi:10.1109/tc.2020.3017870

Cecchinato, N., Toma, A., Drioli, C., Oliva, G., Sechi, G., and Foresti, G. L. (2023). Secure real-time multimedia data transmission from low-cost UAVs with a lightweight AES encryption. *IEEE Commun. Mag.* 61 (5), 160–165. doi:10.1109/mcom.001.2200611

Gao, Z., An, Y., Wang, A., Li, P., Qin, Y., Wang, X., et al. (2020). 40Gb/s secure optical communication based on symbol-by-symbol optical phase encryption. *IEEE Photonics Technol. Lett.* 32 (14), 851–854. doi:10.1109/lpt.2020.3000215

Gao, Z., Li, Q., Zhang, L., Tang, B., Luo, Y., Gao, X., et al. (2022). 32 Gb/s physical-layer secure optical communication over 200km based on temporal dispersion and self-feedback phase encryption. *Opt. Lett.* 47 (4), 913–916. doi:10.1364/ol.451314

Gong, L. H., Zeng, J., and Li, X. Z. (2021). Image encryption algorithm based on the fractional Hermite transform. *J. Mod. Opt.* 68 (19/21), 1026–1040. doi:10.1080/09500340.2021.1968054

Hao, W., Sun, G., Zhang, J., Xiao, P., and Hanzo, L. (2020). Secure millimeter wave cloud radio access networks relying on microwave multicast fronthaul. *IEEE Trans. Commun.* 68 (5), 3079–3095. doi:10.1109/tcomm.2020.2974743

Hong, S., Pan, C., Ren, H., Wang, K., and Nallanathan, A. (2020). Artificial-noise-aided secure MIMO wireless communications via intelligent reflecting surface. *IEEE Trans. Commun.* 68 (12), 7851–7866. doi:10.1109/tcomm.2020.3024621

Hu, Z., Song, P., and Chan, C. K. (2021). Chaotic non-orthogonal matrix-based encryption for secure OFDM-PONs. *IEEE Photonics Technol. Lett.* 33 (20), 1127–1130. doi:10.1109/lpt.2021.3109029

Jia, M. (2020). Image encryption with cross colour field algorithm and improved cascade chaos systems. *IET Image Process.* 14 (5), 973–981. doi:10.1049/iet-ipr.2019.0310

JosephNg, P. S., EricMok, Z. C., and Phan, K. Y. (2025). Mitigating social media cybercrime: revolutionising with AES encryption and generative AI. *J. Adv. Res. Appl. Sci. Eng. Technol.* 46 (2), 124–154. doi:10.37934/araset.46.2.124154

Kannan, C., Dakshinamoorthy, M., Ramachandran, M., Patan, R., and Kumar, A. (2021). Cryptography-based deep artificial structure for secure communication using IoT-enabled cyber-physical system. *IET Commun.* 15 (6), 771–779. doi:10.1049/cmu2.12119

Khan, J., Lim, D. W., and Kim, Y. S. (2023). Intrusion detection system can-bus in-vehicle networks based on the statistical characteristics of attacks. *Sensors* 23 (7), 3554. doi:10.3390/s23073554

Kumari, S., Singh, M., Singh, R., and Tewari, H. (2022). Signature based Merkle Hash Multiplication algorithm to secure the communication in IoT devices. *Knowledge-Based Syst.* 253 (Oct.11), 109543–109543.12. doi:10.1016/j.knosys.2022.109543

Li, X., and Wu, J. (2020). Node-oriented secure data transmission algorithm based on IoT system in social networks. *IEEE Commun. Lett.* 24 (12), 2898–2902. doi:10.1109/lcomm.2020.3017889

Lin, C. M., Pham, D. H., and Huynh, T. T. (2022). Encryption and decryption of audio signal and image secure communications using chaotic system synchronization control by TSK fuzzy brain emotional learning controllers. *IEEE Trans. Cybern.* 52 (12 Pt.2), 13684–13698. doi:10.1109/tcyb.2021.3134245

Liu, J., Zhang, M., Tong, X., and Wang, Z. (2022). Image compression and encryption algorithm based on 2D compressive sensing and hyperchaotic system. *Multimed. Syst.* 28, 595–610. doi:10.1007/s00530-021-00859-6

Luo, T., Zhou, T., and Qu, J. (2021). Lifetime division multiplexing by multilevel encryption algorithm. *ACS Nano* 15 (4), 6257–6265. doi:10.1021/acsnano.0c09177

Panic, S., Petrović, V., Drašković, S., Kontrec, N., and Milojevic, S. (2023). Performance analysis of hybrid fso/rf communication system with receive diversity in the presence of chi-square/gamma turbulence and rician fading. *Bull. D. Serikbayev EKTU*, 304–312. doi:10.51885/1561-4212_2023_4_304

Yumin, D., Hengrui, L., Yanying, F., and Che, X. (2023). Improving the success rate of quantum algorithm attacking RSA encryption system. *J. Appl. Phys.* 134 (2), 024401.1–024401.10. doi:10.1063/5.0153709