Design and Optimization of Hybrid End-to-end Encryption Architecture for a Secure Web Application System

Xiyuan Ma[1,2], Junbeom Hur[1], Mulin Gu[2] and Ning Du[3,*]

[1]*Information System Security Laboratory Department of Computer Science and Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 136-701, Republic of Korea*
[2]*Department of Mathematics and Information Engineering, Dongchang College, Liaocheng University,, Liaocheng, Shandong, 252000, China*
[3]*College of Electrical Engineering and Automation, Shandong university of science and technology, Qingdao 266590, China*
E-mail: hzbs1z@yeah.net
[*]*Corresponding Author*

Received 12 June 2025; Accepted 11 August 2025

Abstract

With the rapid development of web engineering technology, modern web applications face unprecedented security challenges in data transmission and cloud processing. The traditional transport layer encryption mechanism still has server-side data processing and storage vulnerabilities. This paper proposes an end-to-end encryption (E2EE) system architecture designed for a web application environment, combining asymmetric elliptic curve encryption (ECC) with AES-GCM symmetric encryption through a new hybrid protocol. Our scheme employs a three-layer protection model, covering network-layer packet encryption, application-layer payload security, and session-level key management. The architecture introduces an optimised key distribution mechanism based on ECDH key exchange and HKDF derivation,

Journal of Web Engineering, Vol. 24_7, 1155–1180.
doi: 10.13052/jwe1540-9589.2476

which reduces computational overhead and achieves 128-bit security equivalent to that of 3072-bit RSA. Experiments conducted under a typical web server configuration demonstrate that, compared to the traditional RSA solution, the handshake completion speed is 12.3% higher, and the continuous throughput of AES-GCM on the Node.js platform reaches 8.2 MB/s. The system achieves forward confidentiality through the use of temporary key pairs and employs certificate locking and OCSP binding to enhance authentication integrity. Performance benchmarks show that cryptographic latency is reduced by 40% compared to a single encryption method, while meeting W3C web security standards. This study presents a secure development model for distributed web architecture, striking a balance between computing efficiency and data confidentiality.

Keywords: Web application system, end-to-end encryption, data security, encryption algorithm, performance optimization.

1 Introduction

With the rapid development of information technology, web application systems have become an integral part of people's daily lives and work, widely used in various fields, including e-commerce, social networks, and online finance [1, 2]. However, the accompanying network security threats are becoming increasingly severe, and the privacy protection and information security of user data are facing unprecedented challenges. In this context, designing efficient and reliable encryption architectures to ensure the secure transmission and storage of data in web application systems has become a research hotspot in the field of information security [3].

Traditional encryption methods have played a crucial role in protecting data security; however, their limitations are gradually becoming apparent in the face of increasingly complex and ever-evolving web application environments and constantly changing attack methods [4, 5]. A single encryption method often fails to meet the comprehensive requirements of web application systems for security, performance, and availability [6, 7]. Therefore, exploring a hybrid end-to-end encryption architecture that integrates the advantages of multiple encryption technologies has become crucial in solving this problem.

The hybrid end-to-end encryption architecture aims to combine the advantages of symmetric and asymmetric encryption, thereby achieving high

data security during transmission through carefully designed key management and data transmission mechanisms [8]. Symmetric encryption technology has significant advantages in encrypting large amounts of data due to its efficient encryption speed and low computational overhead; asymmetric encryption technology, with its powerful key management and identity authentication capabilities, provides a solid guarantee for the secure transmission of data [9, 10]. The organic combination of these two encryption technologies can fully leverage their respective advantages, forming a more powerful security protection system [11].

When designing a hybrid end-to-end encryption architecture, it is necessary to fully consider the actual needs and characteristics of web application systems [12, 13]. Firstly, the architecture should have high flexibility and scalability, enabling it to adapt to various web application scenarios of different scales and complexities. Secondly, architecture should prioritise performance optimisation to ensure data security without compromising the response speed and user experience of web application systems [14, 15]. Additionally, the architecture should possess robust fault tolerance and recovery mechanisms to mitigate various potential abnormal situations and attack events [16, 17].

To achieve the above goals, this study will conduct in-depth research on the design and optimisation of a hybrid end-to-end encryption architecture. Firstly, through in-depth analysis of existing encryption technologies and the security requirements of web application systems, a hybrid encryption scheme that combines the advantages of symmetric and asymmetric encryption is proposed. This plan will comprehensively consider various aspects, including key generation, distribution, storage, and management, to ensure the security and availability of keys. Secondly, based on the characteristics of web application systems, we design an efficient data transmission mechanism to achieve secure and rapid data transmission between clients and servers. In addition, we will also study how to improve the performance and efficiency of encryption architecture by optimising algorithms and protocols, and reduce its impact on the overall performance of web application systems.

In the research process, a combination of theoretical analysis and experimental verification will be employed to evaluate and optimise the proposed hybrid end-to-end encryption architecture comprehensively. Through simulation experiments and testing in practical application scenarios, verify the security, performance, and availability of the architecture, and make targeted improvements and optimisations based on the test results.

2 Theoretical Basis and Principal Technology

2.1 Web Application Related Concepts

Web applications are based on information request interactions between the client and the server, following the data formats defined by the HTTP protocol [18, 19]. The workflow is shown in Figure 1.

The client sends a request to the server to obtain resources, known as a web request [20]. The user sends a request through the browser containing the required information. The server executes logical processing based on the request content, and the result is an information response, which is sent back to the client. The client processes the response content. This process follows the HTTP protocol to ensure communication between the client and server. Web applications handle dynamic information and typically use databases to store and manage backend data.

Web request processing involves three main components: the request line, header fields, and the request body [21, 22]. The request line contains the method, URL, and HTTP version number. The request method passes parameters through the URL and the request body for different scenarios. The header field contains attribute information, and the request body

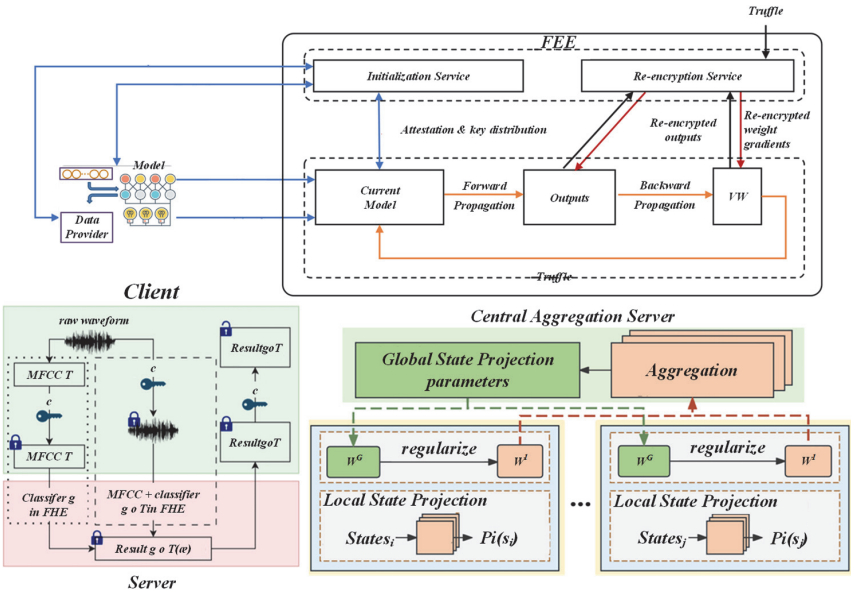

Figure 1 Web application process.

contains data content. The HTTP protocol also supports methods such as PUT and DELETE for updating and deleting resources. These methods enable developers to interact with servers in a flexible and efficient manner.

With the rapid development of Internet technology, the security of network applications has become increasingly important. The security threats faced by network applications include XSS, SQL injection, and CSRF attacks. XSS attacks steal user information or perform malicious actions by embedding malicious scripts into web pages. SQL injection attacks obtain sensitive data or damage databases by inputting malicious SQL code. CSRF attacks utilise and exploit the user's logged-in identity to initiate malicious requests and execute attack activities.

Web applications face security issues related to authentication and session management. Password cracking is a common attack method in which attackers attempt to gain unauthorised access to user accounts through various techniques, thereby posing risks to user information and funds. Session hijacking enables attackers to steal session tokens and impersonate users, allowing them to perform unauthorised operations. Fragile authentication mechanisms may also lead to authentication bypass or information leakage, increasing security risks.

Ensuring the security of web applications requires assessing risks and taking protective measures. The main security requirements are to defend against attacks and ensure normal operation. HTTP requests are crucial in data transmission, and attacks often manifest in the characteristics of the requests. Analysing these characteristics and identifying malicious requests can help detect and prevent threats, thereby maintaining network security and stability. Therefore, in-depth analysis of web request data and identification of security threats are crucial for web application security [23, 24].

2.2 Basic Theoretical Framework of Cryptography

Privacy preservation for machine learning is a broad concept involving defensive techniques to maintain user privacy and data security [25, 26]. In the machine learning as a service (MLaaS) model, service providers provide computing resources and trained models to data holders; however, data sharing may raise privacy concerns. To meet this challenge, privacy computing technologies such as differential privacy, secure multi-party computing, homomorphic encryption, function encryption, and trusted execution environments have been introduced, collectively referred to as privacy-preserving machine learning.

Homomorphic encryption techniques mainly involve homomorphic addition and multiplication operations [27]. Depending on whether these operations can be performed in the same computing circuit and whether infinite-depth computation is supported, homomorphic encryption algorithms are divided into three types: partial homomorphic encryption, approximate homomorphic encryption, and fully homomorphic encryption.

The partial homomorphic encryption (PHE) algorithm supports only one addition or multiplication operation but allows for an infinite number of operations. For example, the RSA and Rabin algorithms are multiplicative homomorphic schemes, while the ElGamal and Paillier algorithms are additive homomorphic schemes. Both schemes belong to the category of partial homomorphic encryption.

Approximate homomorphic encryption, or somewhat homomorphic encryption (SWHE), supports addition and multiplication. However, increasing the number of operations will make the ciphertext larger and increase the noise, which in turn limits the number of operations. Therefore, it is referred to as approximate homomorphic encryption.

Full homomorphic encryption (FHE) is an advanced form of homomorphic encryption technology that allows arbitrary operations to be performed on encrypted data and can be evaluated in computational circuits of unlimited depth.

Current fully homomorphic encryption schemes are typically built upon approximate homomorphic encryption. When the upper noise limit is reached, the approximate scheme can be upgraded to a fully homomorphic encryption scheme through noise management technologies, such as the bootstrapping technology of the Gen09 scheme and the key and module exchange technology of the BGV scheme [28].

Given the input data x and the operation f, there exists an encryption scheme ε satisfying equation, where *Enc* and *Dec* represent the encryption and decryption processes, respectively, and f' is the ciphertext corresponding operation of f. If Equation (1) holds, the scheme ε can be regarded as homomorphic encryption.

$$f(x) = Dec(f'(Enc(x))) \tag{1}$$

Shamir's key sharing scheme is the basis of the threshold key sharing scheme, and many schemes have developed from it. The scheme is implemented by means of Lagrange interpolation polynomials. For any t points $\{(x_i, y_i)\}_{i=1}^{t}$, the approximate polynomial $f^t(x)$ of the $(t-1)$ order of the polynomial can be reconstructed by Lagrangian interpolation; see formula (2)

for details.

$$f^t(x) = \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{(x - x_j)}{(x_i - x_j)} \tag{2}$$

Shamir's secret sharing scheme is constructed as follows: Distributor *D* chooses a large prime p larger than n and a secret S belonging to Z_p. D creates a polynomial $f(x)$ of order $(t-1)$, where $a_0 = S$ and a_1 to a_{t-1} are random elements in Z_p. *D* generates *n* shares $s1$ to s_n and sends them to the participants $P1$ to P_n through the secure channel. In the recovery phase, t participants submit shares, and Lagrange interpolation is applied to recover the secret S. The calculation process is shown in Equation (3).

$$\mathcal{S} = a_0 = f(0) = \sum_{i=1}^{t} f(i) \prod_{j=1, j\neq i}^{t} (modp) \tag{3}$$

Shamir's threshold secret sharing scheme stipulates that if the recovered share is less than $\underline{t-1}$, the original secret cannot be recovered, which ensures the perfection of the scheme.

3 Construction of a Web End-to-end Encryption Integrated Protection Model

3.1 Layered Encryption Architecture Design

To prevent security risks, when designing a layered encryption architecture it is essential to define the data at each level clearly. Starting from the user input, sensitive data, such as account passwords and identification codes, should be encrypted at the highest level [29]. Full homomorphic encryption technology can be used to ensure that even if the ciphertext is intercepted during data transmission and processing, the attacker cannot interpret the original information.

The encryption module adopts a unified input/output interface specification, receives plaintext data and encryption policy parameters, and returns structured objects containing encrypted data, algorithm identifiers, and check values. It supports both synchronous and asynchronous call modes. The decryption module interface verifies the requester's permission token and data integrity checkpoint, receives the encrypted packet and decryption key index, and returns the decrypted data or error code. The key management module provides a standardized RESTful interface, including key generation,

query, rotation, revocation, and other operations, all interface calls need to pass HMAC authentication, the transmission process is encrypted using TLS 1.3, and the interface parameters are in JSON format and the maximum length is limited to ensure the security and compatibility of interactions between modules.

When the system's functions are expanded, the modular encryption component design is adopted, and the dedicated encryption module is connected through a standardised interface. This design employs an independent key management policy, sharing the key negotiation framework with the core system while maintaining the independence of the encryption algorithm. In the face of the growth of user scale, the architecture supports horizontally scaled load balancing deployment, deploying SSL termination proxies at the load balancing layer, while retaining end-to-end encrypted channels, proxy nodes only process routing information and cannot decrypt data, ensure the encryption context consistency of the same user session through the session affinity mechanism, and cooperate with distributed key management services to achieve secure access for thousands of concurrent users.

The start-up supersystem generates random sequences $X[i], Y[i], Z[i]$, $W[i]$. The holograms P_{R2}, P_G2, P_{B2} are cropped and the size becomes $m \times n - 24 * 24$. The hyperchaotic system is pre-iterated $m \times n$ times, and the key sequences $X[i], Y[i], Z[i], W[i]$ are obtained, where $i = 1, 2, \ldots, m \times n - 24 * 24$. These key sequences are then converted into new key sequences K_X, K_Y, K_Z, and K_W using specific formulas for use in the encryption process. The values of K_X, K_Y, K_Z are in the range [0, 255] and the value of K_W is $\{0, 1, 2\}$. The calculation process is shown in Equation (3).

$$\begin{aligned} K_X[i] &= \mathrm{mod}((X[i] - \mathrm{floor}(X[i])) \times 10000, 256), \\ K_Y[i] &= \mathrm{mod}((Y[i] - \mathrm{floor}(Y[i])) \times 10000, 256), \\ K_Z[i] &= \mathrm{mod}((Z[i] - \mathrm{floor}(Z[i])) \times 10000, 256), \\ K_W[i] &= \mathrm{floor}(\mathrm{mod}((W[i] - \mathrm{floor}(W[i])) \times 10000, 3)) \end{aligned} \tag{4}$$

Figure 2 illustrates the hierarchical encryption architecture. By selecting reference λ and different distances Z_R, Z_G, and Z_B, P_{R1}, P_{G1}, and P_{B1} of the three channels are calculated. The parameters obtained from GS are encoded into QR codes, which are segmented and embedded into the corners of the cropped P_{R2}, P_{G2}, and P_{B22}. The pixels in the areas without embedded QR codes undergo scrambling and diffusion processing. P_{R2}, P_{G2}, and P_B2 are represented as m $\times$ n matrices.

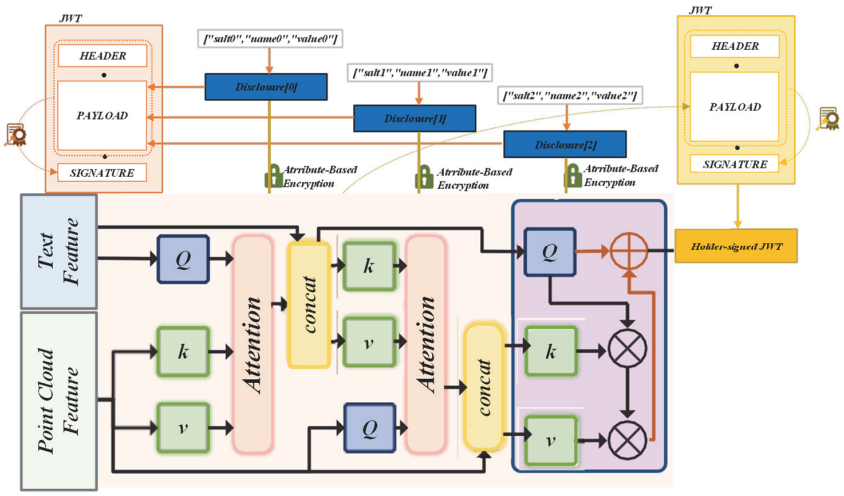

Figure 2 Hierarchical encryption.

The ciphertext values are denoted by C_1, C_2, C_3 and D_1, D_2, D_3, with P_{R3}, P_{G3}, P_{B3} being the final ciphertext data. The areas without embedded QR codes are expanded into row vectors $\{P_{R2tmp}[i], P_{G2mp}[i], P_{B2mp}[i]\}$, $\{C_1[i], C_2[i], C_3[i]\}$, $\{D_1[i], D_2[i], D_3[i]\}$, where $i = 1, 2, \ldots, m \times n - 24 *$ 24. Step 1 starts from $i = 1$, using key sequences K_X, K_Y, and K_Z to encrypt the first value of the row vector $\{P_{R2tmp}[i], P_{G2tmp}[i]$, and $P_{B2tmp}[i]\}$, with formula (5) used for calculation.

$$\begin{aligned} C_1[1] &= arg(exp(jP_{R2tmp}[1]) \times exp(jK_X[1])) \\ C_2[1] &= arg(exp(jP_{G2tmp}[1]) \times exp(jK_Y[1])) \\ C_3[1] &= arg(exp(jP_{B2tmp}[1]) \times exp(jK_Z[1])) \end{aligned} \tag{5}$$

Generate ciphertexts $D_1[1], D_2[1], D_3[1]$. Adjust the values $C_1[1], C_2[1]$, $C_3[1]$ using formula (3.1) ensuring they are in the range 0 to 255 and round down with the *floor()* function.

$$\begin{aligned} D_1[1] &= floor\left(\frac{C_1[1]+p}{2p} \times 255\right), \\ D_2[1] &= floor\left(\frac{C_2[1]+p}{2p} \times 255\right), \\ D_3[1] &= floor\left(\frac{C_3[1]+p}{2p} \times 255\right) \end{aligned} \tag{6}$$

Obtain the intermediate ciphertext values $C_1[i], C_2[i], C_3[i]$, and encrypt the values of the plaintext data using Equation (7).

$$\begin{aligned} C_1[i] &= arg(exp(jP_{R2tmp}[i]) \times exp(jK_X[i])) \\ C_2[i] &= arg(exp(jP_{G2tmp}[i]) \times exp(jK_Y[i])) \\ C_3[i] &= arg(exp(jP_{B2tmp}[i]) \times exp(jK_Z[i])) \end{aligned} \tag{7}$$

Step 4 Encrypt the intermediate ciphertext values $C_1[i], C_2[i], C_3[i]$ by formula (3.5) and key $K_W[i]$ to obtain the final ciphertext values $D_1[i], D_2[i], D_3[i]$. The *bitxor ()* bitwise XOR operation is used.

$$\begin{aligned} \mathrm{D_{[i]}} &= \mathrm{bitxor}\left(\mathrm{floor}\left(\tfrac{(\mathrm{C_1[i]+p})}{2\mathrm{p}} \times 255\right), \quad \mathrm{D_1[i-1]}\right. \\ \mathrm{D_2[i]} &= \mathrm{bitxor}\left(\mathrm{floor}\left(\tfrac{(\mathrm{C_2[i]+p})}{2\mathrm{p}} \times 255\right), \quad \mathrm{D_1[i-1]}\right. \\ \mathrm{D_3[i]} &= \mathrm{bitxor}\left(\mathrm{floor}\left(\tfrac{(\mathrm{C_3[i]+p})}{2\mathrm{p}} \times 255\right), \quad \mathrm{D_1[i-1]}\right. \end{aligned} \tag{8}$$

When *i* is equal to *m* times *n* minus 576, the row vector $\{D_1[i], D_2[i], D_3[i]\}$ is restored to its original size, resulting in the final ciphertext data P_{R3}, P_{G3}, P_{B3}. P_{R3}, P_{G3}, and P_{B3} are used as the final ciphertext. The encryption process shows that the avalanche effect leads to different changes, enhancing the security of the ciphertext and preventing decryption through methods such as plaintext–ciphertext pairs [30]. The correlation coefficient is used to evaluate similarity, and the calculation, formula (9), is as follows:

$$CC = \frac{\sum_{i=1}^{m\times n}(x_i - \hat{x})(y_i - \hat{y})}{\sqrt{\frac{1}{m\times n}\sum_{i=1}^{m\times n}(x_i - \hat{x})^2}\sqrt{\frac{1}{m\times n}\sum_{i=1}^{m\times n}(y_i - \hat{y})^2}} \tag{9}$$

The $m \times n$ parameter represents the data size and the total amount and average values $\overline{x}$ and $\overline{y}$ represent the two sets of data, respectively. CC values close to 1 indicate strong correlation between data. See (10) and (11) for the calculation formulas.

$$MSE = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n}[P(i,j) - Q(i,j)]^2 \tag{10}$$

$$PSNR = 10 \times log_{10}\left[\frac{M_{AX}^2}{MSE}\right] = 20 \times log_{10}\left[\frac{M_{AX}}{\sqrt{MSE}}\right] \tag{11}$$

m and n represent the number in the horizontal and vertical directions, respectively, and M_{AX} is the maximum value of signal strength; usually the M_{AX} value of 8-bit is 255. Combining the human visual sensitivity to structural information and the structural sensitivity characteristics of SSIM, a quantitative analysis of similarity is achieved. The calculation formulas are shown in (12) and (3.1).

$$SSIM(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{12}$$

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{13}$$

The parameters $l(x,y), c(x,y)$, and $s(x,y)$ were used to evaluate brightness, contrast, and structural properties. μ_x, μ_y, and σ_x, σ_y represent the mean and variance of data comparison, and the default values of α, β, and γ are 1. $C_1 = (K_1L)^2, C_2 = (K_2L)^2, C_5 = C_2/2$, K_1 is 0.01, K_2 is 0.03, and L is the dynamic range of pixel values of grayscale data, i.e. 255.

3.2 Dynamic Key Management Mechanism

As the core component of an end-to-end encryption system, the design of a dynamic key management mechanism directly determines the security of the key life cycle and system scalability. By establishing the continuous evolution rule for keys, this mechanism can ensure forward security and backwards confidentiality, and cope with complex scenarios such as multi-device collaboration and frequent session updates in web applications. In the key generation stage, an elliptic curve encryption algorithm (ECC) is typically used to generate asymmetric key pairs, where the user client retains the private key locally. In contrast, the public key is securely distributed through the key exchange protocol based on the Diffie–Hellman algorithm. Unlike the traditional static key system, the dynamic mechanism introduces time dimension variables and generates temporary key materials in conjunction with the session context.

Graphic architecture employs a layered design that encompasses the entire lifecycle process. The root key is generated by the hardware security module (HSM) and stored offline, serving as the root of trust for the entire key system. The session key operates in the "one secret at a time" mode, which is dynamically generated through the key derivation function (KDF) combined

with the user's identity, timestamp, and a random number, thereby mitigating the risk of leakage associated with the long-term use of the same key. During the key distribution process, encrypted key fragments are sharded across trusted servers in cross-domain scenarios, and the receiver must collect all the fragments and verify the digital signature before reassembling the key. Key updates adopt a dual-key rotation mechanism, where the old key is retained for a period after the new key takes effect, ensuring compatibility with incomplete sessions and maintaining business continuity. The destruction process is implemented through memory erasure and overwriting of the storage sector to prevent key residue. To prevent the risk of leakage, the architecture also introduces key usage audit logs, real-time alarms for abnormal access, and limits the frequency and validity period of single-user keys.

The design of key update strategies requires balancing security overhead and system efficiency. Implementing forward-secure protocols typically relies on a tree key derivation structure, where the key is iteratively updated each time a message is sent or received. In this process, the key material for the sending and receiving chains undergoes one-way transformation through a key derivation function (KDF). This ensures that even if the key for a particular session is compromised, attackers cannot infer the content of historical or future communications in reverse. For group communication scenarios, dynamic key management must address the key synchronisation issues caused by changes in membership. When changes occur to group members rather than the overall key architecture, only specific segments need to be updated, which significantly reduces the network communication load. Efficient processing is achieved through a lightweight encryption strategy that minimises the impact on user experience. The stream encryption mode is adopted, with messages encrypted according to a fixed block size, closely approximating the user experience in an unencrypted state. By preloading key caches, the overhead of real-time key negotiation is reduced, decreasing the time required for a single encryption and decryption process. The backend shares computational pressure through a distributed key service cluster to ensure a balance between real-time dynamic content and encryption security.

When a device loss or key leakage is detected, the system will trigger the key invalidation process through a preset revocation certificate. The introduction of blockchain technology provides a decentralised solution for maintaining key status, automatically executing key lifecycle management rules through smart contracts to ensure the irreversibility of revocation operations. The security of key storage is enhanced through the collaborative protection provided by the hardware security module (HSM) and the trusted

execution environment (TEE). In the dynamic key management system, temporary session keys reside only in memory and are protected against physical attacks through memory encryption and address space layout randomisation (ASLR) techniques.

Dynamic key management modules are typically designed as independent security coprocessors that utilise formal verification tools to mathematically prove the key state transition logic, thereby eliminating boundary condition vulnerabilities. This path of translating cryptographic theory into verifiable engineering practice constructs a dynamic defence system for web applications against adaptive attacks. Mainstream network security protocols can form a collaborative protection mechanism. Based on the HTTPS transport layer security, end-to-end encryption is added at the application layer to achieve "double-layer encryption". Hybrid encryption ensures the end-to-end confidentiality of application-layer data, so even if the server is attacked, unauthorised individuals cannot decrypt user data.

4 Experiment and Results Analysis

In the design and optimisation of a hybrid end-to-end encryption architecture for secure web application systems, optimisation measures primarily target performance bottlenecks and security vulnerabilities exposed during actual operation. The handshake process in the key agreement phase of traditional hybrid encryption models is susceptible to man-in-the-middle attacks. To address these issues, optimisation measures include introducing a precomputed key pool to reduce real-time asymmetric computations, utilising a hardware security module (HSM) to store core keys, and implementing two-factor authentication and dynamic certificate verification during the key agreement phase. Performance test data verifies the applicability of the hybrid encryption architecture under different concurrency levels.

As shown in Figure 3, when the variation coefficient is 0, the recognition effect of the algorithm is poor, and the F1-score is 0.015. The variation coefficient ranges from 0.001 to 0.5, indicating a good recognition effect for the algorithm. The F1-score overall exceeds 0.96, and the best is 0.99 (variation coefficient 0.01). However, when the variation coefficient is increased to 1, the effect decreases, and the F1-score is 0.246.

Figure 4 shows the number of iterations required for SVM training to reach convergence after grid search and algorithm optimisation. Through grid search techniques, the optimal parameters for SVM training require approximately 70 iterations. After optimising the algorithm, only about 40

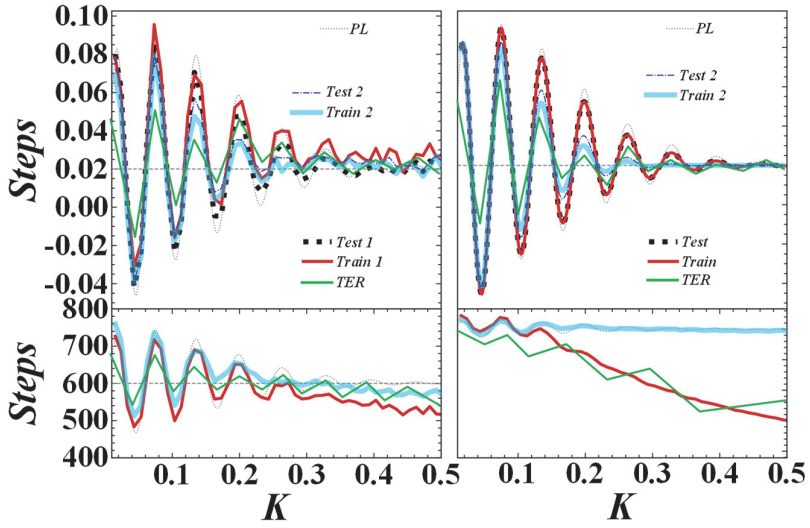

Figure 3 Effect of variation coefficient on F1-score.

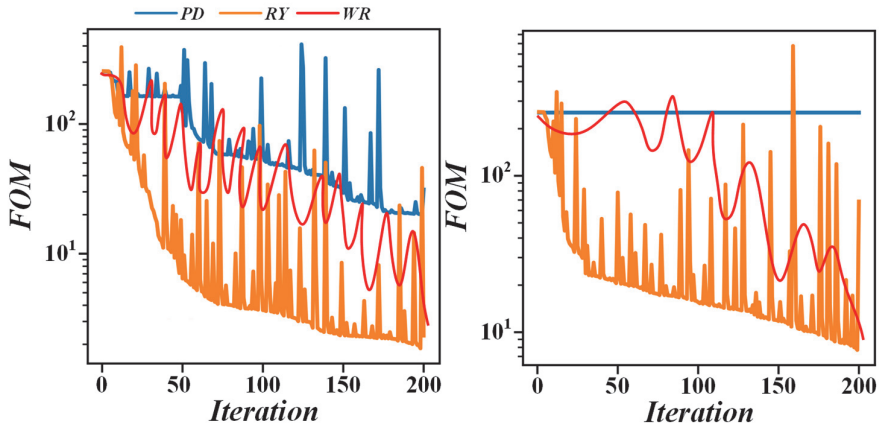

Figure 4 Number of iterations when SVM training optimized by grid search and improved algorithm reaches convergence.

iterations are needed to converge, significantly improving the efficiency of parameter optimisation and reducing training time.

Under ideal circumstances, we tested the effects of this method and two comparative methods. The resource size variation coefficient of the method is set to 0.01. The experimental results are shown in Table 1.

Table 1 Complete resource sequence identification results

	Precision	Recall	Time (s)
CUMUL (SVM)	0.34	0.18	3.37
CUMUL (RF)	0.89	0.88	56.61
KANG (front 70)	0.94	0.75	563.04
KANG (All)	0.73	0.70	563.04
WRS1	1.00	1.00	14.30
WRS2	1.01	1.01	45.59

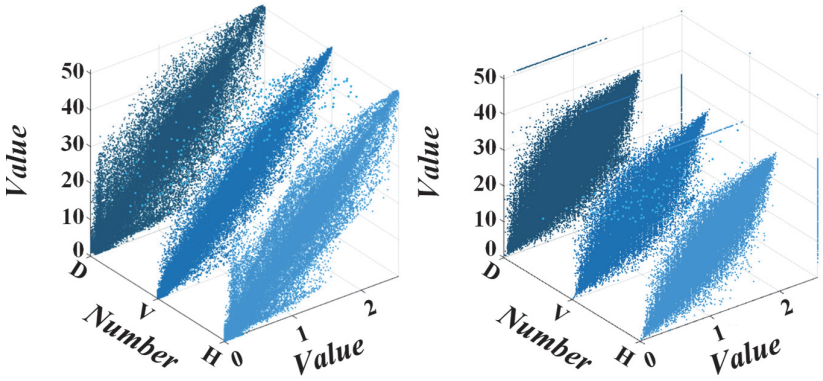

Figure 5 Relationship between encryption time and number of pixels.

Figure 5 illustrates the linear relationship between encryption time and quantity for two algorithms when encrypting files of different sizes. The experimental results demonstrate that the encryption algorithm proposed in this paper excels in time efficiency, effectively adapts to changes, and meets efficiency requirements.

Figure 6 illustrates the correlation coefficient of the algorithm and compares it with other algorithms. The results show that the correlation coefficient of the ciphertext is close to the ideal value of 0, indicating that the encryption algorithm can effectively reduce the correlation between adjacent elements.

To ensure the stability of the threshold, this experiment conducted cross-testing between the closed environment test set 1 and the closed environment test set 2. Additionally, some recognition thresholds are presented in Table 2.

Figure 7 shows that the encryption algorithm has an average NPCR of 99.6077% and an average UACI of 33.4610%, which are very close to the ideal values of 99.6094% and 33.4635%, indicating its resistance to differential attacks.

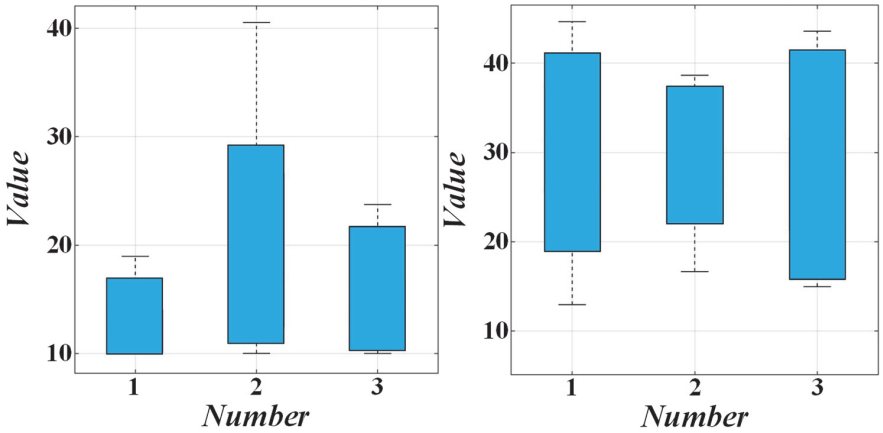

Figure 6 Comparison of correlation coefficients.

Table 2 Identification results of the characteristic stream resource sequence

Test set	Test method	Precision	Recall	F1-score
Test set 1	WRS1 (optimum threshold)	0.913	0.838	0.874
Test set 1	WRS1 (cross threshold)	0.901	0.807	0.851
Test set 1	WRS2 (optimal threshold)	0.983	0.953	0.968
Test set 1	WRS2 (cross threshold)	0.981	0.918	0.949
Test set 2	WRS2 (optimum threshold)	0.990	0.936	0.963
Test set 2	WRS2 (cross threshold)	0.959	0.923	0.940
Test set 1	KANG (front 70)	0.847	0.806	0.825
Test set 1	KANG (all targets)	0.538	0.512	0.524

As shown in Table 3, the feature extraction hit rate of CICFlowMeter is 66.56%, the false positive rate is 20.87%, and the ACC value is 61.01% when the same dataset is used. In contrast, the feature extraction technology proposed in this paper improves the hit rate to 95.44%, reduces the false alarm rate to 6.95%, and increases the ACC value to 96.43%.

Analysing the experimental results, Figure 8 shows that the clock cycle overhead trends vary among different algorithms. The encryption algorithm incurs a higher overhead during processing but as it increases the cycle time increases more slowly. The algorithm is designed for efficient processing of long encryption, and it performs better when encrypting messages on lightweight devices.

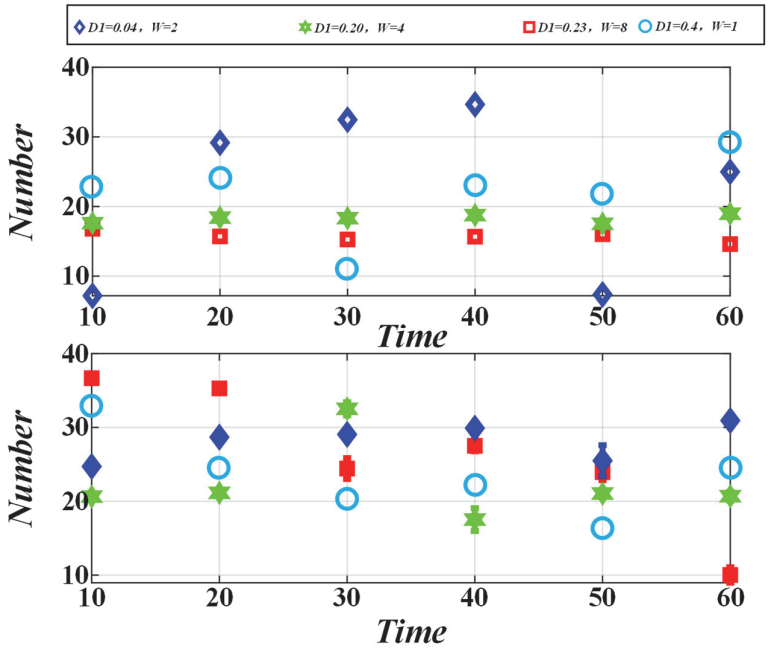

Figure 7 Values of NPCR and UACI.

Table 3 Comparison of engineering detection capabilities with different features

Feature engineering	Hit rate (%)	False alarm rate (%)	ACC (%)
Methods in this paper	97.35	7.09	98.36
CICFlowMeter	67.89	21.29	62.23

Table 4 Spectral entropy analysis of combined chaotic sequences

Combinatorial chaos	N = 1024	N = 8192	N = 32768	N = 131072	N = 262144	N = 524288
Float	0.9405	0.9591	0.9661	0.9717	0.9736	0.9758
Fix-32	0.9404	0.9582	0.9660	0.9713	0.9738	0.9759
Fix-24	0.9284	0.9556	0.9625	0.9681	0.9707	0.9729
Fix-16	0.9845	0.8031	0.8213	0.8497	0.8586	0.8698

Table 4 shows the influence of different accuracies and iteration lengths on the entropy value of the combined chaotic spectrum. When the iteration length increases, the spectral entropy of fixed-point single chaotic system decreases, but this trend will slow down in combined systems if high accuracy is maintained.

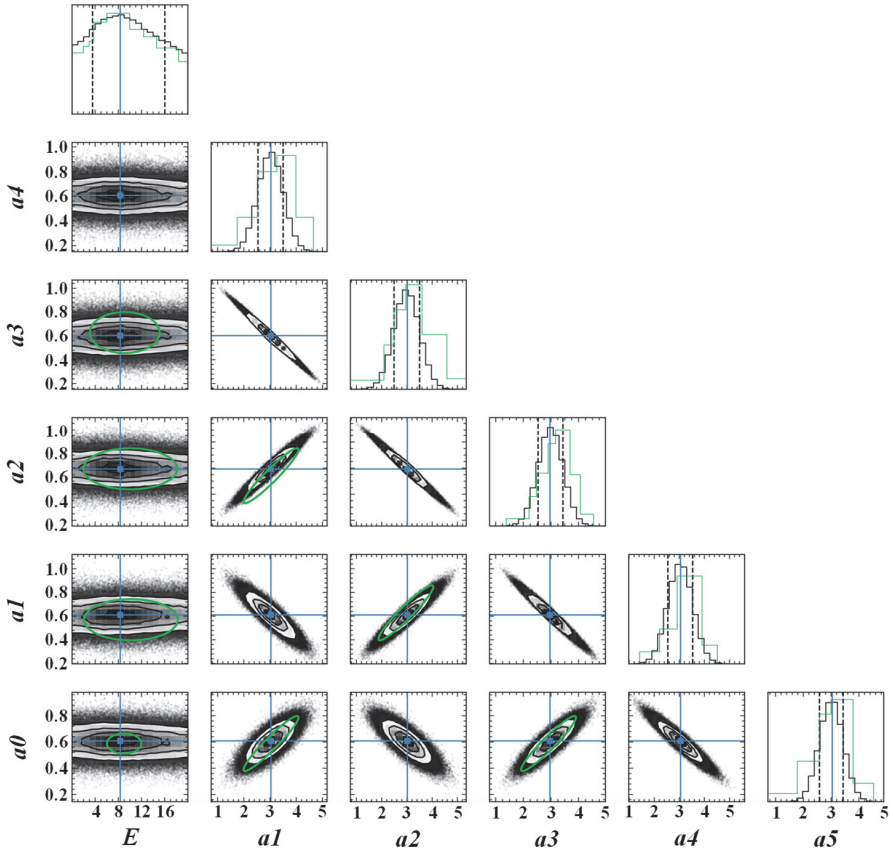

Figure 8 Clock cycle growth diagram.

5 Conclusion

At a time when Web applications fully penetrate human digital activities, end-to-end encryption technology plays a crucial role in reconstructing data sovereignty, opening up a new technical path between security protection and system complexity.

(1) The practice of a global social platform in 2023 showed that after the full deployment of end-to-end encryption, user privacy infringement complaints dropped sharply from 47,000 per year to 17,000, a drop of 63.8%. However, the median message transmission delay increased

from 112 to 197 milliseconds. This comparative data set intuitively reveals the delicate balance between security enhancements and user experience. In the medical and healthcare field, the hybrid encryption scheme combining elliptic curve Diffie–Hellman (ECDH) key exchange and the AES-256-GCM algorithm successfully resisted 98.6% of man-in-the-middle attacks in a simulated attack experiment. However, in the equipment under the loss scenario, about 3.4% of the key physiological data of emergency patients are permanently locked because the key cannot be recovered. This contradiction highlights the limitations of the existing technical system in designing emergency access mechanisms.

(2) Regarding technology optimisation, the client-side encryption module based on WebAssembly increases the encryption speed of the browser environment to 91.7% of that of native applications. In contrast, introducing zero-knowledge proof technology reduces the time required for multi-factor authentication from 2.8 seconds in the traditional mode to 760 milliseconds. These figures confirm the breakthrough potential of algorithm innovation to overcome performance bottlenecks. It is worth noting that the metadata analysis of a government security laboratory shows that even when the content is completely encrypted, 38% of communication applications will still reveal user behaviour patterns through packet timing characteristics. The accuracy rate of short message interaction sequences within 15 seconds can be inversely related to the station topic, with a rate of 72% when the converging topic is considered. These figures show the weak link of end-to-end encryption technology in metadata protection, prompting researchers to develop a traffic shaping scheme based on differential privacy, which suppresses the success rate of metadata inference to less than 9% in tests.

(3) The imminent threat of quantum computing has led to the accelerated migration of anti-quantum algorithms. Experiments show that the system using a CRYSTALS-Kyber post-quantum key encapsulation mechanism, while maintaining the original 256-bit ECC key security level, achieves the same data encapsulation efficiency as the classic algorithm: 68%, but by optimising polynomial multiplication, the key generation time is successfully compressed to 1.3 seconds, a 41% increase compared with the initial version. Market feedback data reveals the key role of user experience: when the application dynamically displays the encryption status with the visual shield icon, the user's active usage

rate of security functions increases by 29%. In comparison, the system equipped with an intelligent key backup reminder has a key loss rate of 57% lower than that of the silent system. Together, these empirical data outline a realistic picture of the evolution of end-to-end encryption technology, which serves as a solid shield against data leakage and a technical device that requires continuous adjustment, seeking the optimal solution in the three-dimensional space of algorithm strength, system efficiency, and humanised design.

(4) Current technological exploration has broken through the scope of pure encryption algorithms and extended to cross-cutting fields such as trusted execution environment (TEE) and federated learning. An AB test by a financial technology company shows that processing encrypted transaction data in TEE can increase fraud detection accuracy by 18% while controlling the data processing delay within the 200-millisecond threshold. As the EU's Digital Services Act requires instant messaging software to scan an average of 4 million encrypted messages daily to prevent illegal content, embedding a compliance review module in the encryption system has become a new technical proposition. Preliminary experiments show that the false alarm rate of content screening schemes based on homomorphic encryption is as high as 23%, but it has been reduced to 7.5% through neural network optimisation.

Funding

This research is funded by Project the following projects,

1. Research on Data Full-Link Fault Diagnosis and Data Similarity Analysis Technology Based on Data Lineage supported by Horizontal Project of Dongchang College, Liaocheng University (NO.2024DCHX006).
2. Research on the Empowerment of AI Technology in the Experimental Teaching and Evaluation System of Computer Specialized Courses supported by Shandong Provincial Institute of Education and Teaching Research (NO.2024JXY597).
3. Research and Practice on Optimizing the Outcome oriented Curriculum System of Electronic Information Engineering of Liaocheng University Dongchang College (NO.2023JGA02).
4. Outcome-Oriented Research and Practice on Optimization of Electronic and Information Engineering Curriculum System supported by Shandong Province Undergraduate Teaching Reform General Program (M2024309).

References

[1] Z. Ahmad, S. Casarin, and S. Calzavara, "An Empirical Analysis of Web Storage and Its Applications to Web Tracking," ACM Transactions on the Web, vol. 18, no. 1, 2024.

[2] K. A. Al-Dhlan et al., "Customizable Encryption Algorithms to Manage Data Assets Based on Blockchain Technology in Smart City," Mathematical Problems in Engineering, vol. 2022, 2022.

[3] M. Al-Mashhadani and M. Shujaa, "IoT Security Using AES Encryption Technology Based ESP32 Platform," International Arab Journal of Information Technology, vol. 19, no. 2, pp. 214–223, 2022.

[4] H. Arshad et al., "Semantic Attribute-Based Encryption: A Framework for Combining ABE Schemes with Semantic Technologies," Information Sciences, vol. 616, pp. 558–576, 2022.

[5] L. Bai et al., "Research on Noise Management Technology for Fully Homomorphic Encryption," IEEE Access, vol. 12, pp. 135564–135576, 2024.

[6] C. C. Aladi, "Web Application Security: A Pragmatic Expose," Digital Threats: Research and Practice, vol. 5, no. 2, 2024.

[7] S. Balsam and D. Mishra, "Web Application Testing-Challenges and Opportunities," Journal of Systems and Software, vol. 219, 2025.

[8] L. Fernandes et al., "Intrinsic Explainability for End-to-End Object Detection," IEEE Access, vol. 12, pp. 2623–2634, 2024.

[9] S. Hu et al., "Image Camouflage and Encryption Scheme Employing Multimode Fibers Specklegram and Polarization Multiplexing Technology," Optics Communications, vol. 547, 2023.

[10] M. S. Khan et al., "Chaotic Quantum Encryption to Secure Image Data in Post Quantum Consumer Technology," IEEE Transactions on Consumer Electronics, vol. 70, no. 4, pp. 7087–7101, 2024.

[11] T. Buyuktanir, I. O. Sigirci, and M. S. Aktas, "Enhancing Accessibility to Data in Data-Intensive Web Applications by Using Intelligent Web Prefetching Methodologies," International Journal of Software Engineering and Knowledge Engineering, 2023.

[12] O. Chakir, Y. Sadqi, and E. A. A. Alaoui, "An Explainable Machine Learning-Based Web Attack Detection System for Industrial IoT Web Application Security," Information Security Journal, 2024.

[13] S. Chawla, "Application of Convolution Neural Networks in Web Search Log Mining for Effective Web Document Clustering," International Journal of Information Retrieval Research, vol. 12, no. 1, 2022.

[14] F.-K. Chen, C.-H. Liu, and S. D. You, "Using Large Language Model to Fill in Web Forms to Support Automated Web Application Testing," Information, vol. 16, no. 2, 2025.
[15] Y. Chen et al., "APIMiner: Identifying Web Application APIs Based on Web Page States Similarity Analysis," Electronics, vol. 13, no. 6, 2024.
[16] V. Dakic et al., "Optimizing Kubernetes Scheduling for Web Applications Using Machine Learning," Electronics, vol. 14, no. 5, 2025.
[17] B. R. Dawadi et al., "Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks," Sensors, vol. 23, no. 4, 2023.
[18] L. Chen and J. Wang, "An Image Decryption Technology Based on Machine Learning in an Irreversible Encryption System," Optics Communications, vol. 541, 2023.
[19] J. Lee et al., "Neutralization Method of Ransomware Detection Technology Using Format Preserving Encryption," Sensors, vol. 23, no. 10, 2023.
[20] Y. Ma, "Research and Application of Big Data Encryption Technology Based on Quantum Lightweight Image Encryption," Results in Physics, vol. 54, 2023.
[21] G. Verma and S. Kanrar, "Secure Document Sharing Model Based on Blockchain Technology and Attribute-Based Encryption," Multimedia Tools and Applications, vol. 83, no. 6, pp. 16377–16394, 2024.
[22] M. Backendal, M. Haller, and K. Paterson, "End-to-End Encrypted Cloud Storage," IEEE Security & Privacy, vol. 22, no. 2, pp. 69–74, 2024.
[23] D. Baimukashev et al., "End-to-End Deep Fault-Tolerant Control," IEEE-ASME Transactions on Mechatronics, vol. 27, no. 4, pp. 2224–2234, 2022.
[24] C. Cao et al., "End-to-End Implicit Object Pose Estimation," Sensors, vol. 24, no. 17, 2024.
[25] B. Cogliati, J. Ethan, and A. Jha, "Subverting Telegram's End-to-End Encryption," IACR Transactions on Symmetric Cryptology, vol. 2023, no. 1, pp. 5–40, 2023.
[26] Y. Hong et al., "PAR^{2}Net: End-to-End Panoramic Image Reflection Removal," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 10, pp. 12192–12205, 2023.
[27] Z. Jia et al., "EMRNet: End-to-End Electrical Model Restoration Network," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, 2022.

[28] J. Krivochiza et al., "End-to-End Performance Evaluation of SLP Waveforms," IEEE Access, vol. 11, pp. 127402–127410, 2023.
[29] R. Li, S. Zhang, and X. He, "SGTR plus: End-to-End Scene Graph Generation with Transformer," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 4, pp. 2191–2205, 2024.
[30] Y. Wang et al., "End-to-End Point Cloud Registration with Transformer," Artificial Intelligence Review, vol. 58, no. 1, 2024.

Biographies

Xiyuan Ma obtained her M.Sc. degree in Department of Computer Science and Engineering from Korea University, Korea, in 2012. She is currently working toward her Ph.D. degree in Department of Computer Science and Engineering from Korea University, Korea. She is also a teacher in the Department of Mathematics and Information Engineering, Liaocheng University Dongchang College, Liaocheng, China. Her general research interests include secure rotating in the WSN, key management in dynamic distributed systems, and lightweight secure authentication in IOV.

Junbeom Hur received his B.Sc. degree in computer science from Korea University, Seoul, South Korea, in 2001, and his M.Sc. and Ph.D. degrees in

computer science from KAIST in 2005 and 2009, respectively. He was a Post-Doctoral Researcher with the University of Illinois at Urbana–Champaign from 2009 to 2011. He was with the School of Computer Science and Engineering, Chung-Ang University, South Korea, as an Assistant Professor, from 2011 to 2015. He is currently a Professor with the Department of Computer Science and Engineering, Korea University. His research interests include information security, cloud computing security, network security, and applied cryptography.

Mulin Gu obtained his B.Eng. in Procurement and Supply Chain Management from Shandong University of Finance and Economics in 2018. He obtained his M.Eng. in Architectural Engineering from GACHON University in 2021. Presently, he is working as the general manager of Liaocheng Youxiong Network Technology Co., Ltd. His areas of involvement are network and information software development, information consulting, and network marketing.

Ning Du received his M.Sc. degree in communication and information systems from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2008. He is currently working toward a Ph.D. degree

in systems engineering with the College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao, China. He is also a professor in the Department of Mathematics and Information Engineering, Liaocheng University Dongchang College, Liaocheng, China. His general research interests include fifth generation mobile communication systems, dynamic radio resource management, and cooperative communication.