Hybrid RSA–SHA-256 Scheme for Enhancing Physical-Layer Security in MIMO-OFDM Systems Using SDR Implementation

M Wisnu Gunawan, I Gede Puja Astawa*, Amang Sudarsono

Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya (PENS), East Java, Indonesia

Received 08 September 2025; received in revised form 05 January 2026; accepted 06 January 2026

DOI: https://doi.org/10.46604/aiti.2026.15667

Abstract

This research aims to enhance the physical-layer security (PLS) of multi-input multi-output orthogonal frequency division multiplexing (MIMO-OFDM) wireless communication systems. The proposed method integrates Rivest–Shamir–Adleman (RSA)-based subcarrier index scrambling with secure hash algorithm (SHA)-256 hashing to ensure both confidentiality and integrity without altering the transmitted waveform. The scheme is implemented on a 2×2 Universal Software Radio Peripheral (USRP)-based MIMO-OFDM software-defined radio (SDR) testbed and evaluated using symbol distribution analysis, bit-error-rate (BER) performance, and security assessment. Experimental results indicate that RSA-based scrambling substantially increases symbol randomness while preserving Shannon-consistent BER performance across multiple modulation orders, with minimal penalty in signal-to-noise ratio (SNR). Additionally, security analysis demonstrates that the combination of RSA-2048 and SHA-256 is resistant to Wiener's attack and the common modulus attack, thereby supporting efficient and secure real-time MIMO-OFDM communication.

Keywords: MIMO-OFDM, RSA, SHA-256, subcarrier scrambling, physical layer security

1. Introduction

The rapidly growing need for secure wireless communication has made data security a pressing challenge in modern systems [1]. Multi-input multi-output orthogonal frequency division multiplexing (MIMO-OFDM) uses multiple antennas to simultaneously transmit and receive data over many closely spaced subcarriers. This approach is widely adopted in current wireless standards due to its high spectral efficiency, resilience to multipath fading, and large data capacity [1-2]. However, the openness of wireless channels exposes MIMO-OFDM to threats such as eavesdropping (unauthorized interception) and signal manipulation [3-4]. Unlike wired systems, which restrict physical access, wireless signals are broadcast, increasing vulnerability to unauthorized access and cyberattacks. Thus, developing advanced security mechanisms for MIMO-OFDM is crucial to ensure both confidentiality and data integrity (accuracy and trustworthiness).

Traditionally, wireless communication security has relied on upper-layer cryptographic approaches, which encrypt data at the network or application layers [3, 5]. While these methods can secure user data through encryption, they have limitations. For example, if upper-layer encryption is compromised, transmitted data remains susceptible to exposure or manipulation. Additionally, conventional cryptographic algorithms often introduce considerable computational complexity and latency, decreasing their suitability for low-power or real-time wireless environments [6]. These recognized limitations have generated interest in physical-layer security (PLS) techniques. PLS incorporates security directly into signal design [5-8], thereby facilitating confidentiality at the waveform level and enhancing resilience against interception targeting system parameters, such as subcarriers in OFDM systems [4, 7-8].

* Corresponding author. E-mail address: puja@pens.ac.id

Numerous PLS methods have been proposed in the literature. Weng et al. [9] introduced a carrier scrambling technique using a three-dimensional Lorenz chaotic system to conceal key information and enhance confidentiality. Guo et al. [2] proposed a dual three-dimensional ultra-chaotic memory system that achieves a key space of 10^{343} and demonstrates resistance to brute-force attacks, albeit with increased system complexity and degraded bit-error-rate (BER) performance. Li et al. [10] presented a chaos-based encryption scheme combined with the secure hash algorithm (SHA)-256 for key generation, thereby improving both key space and confidentiality.

Other investigated methods include OFDM subcarrier reordering [3-4], chaotic modulation [6], and discrete spectral encryption [7], all targeting enhanced PLS. While these approaches improve randomness and confidentiality, they often result in trade-offs such as increased computational load, degraded BER, or vulnerability to advanced cryptanalytic attacks [11-12]. These factors highlight the ongoing need for low-complexity, robust mechanisms that ensure both confidentiality and integrity in response to contemporary threats.

Recent research has shown that physical-layer encryption can enhance wireless confidentiality and data integrity in OFDM-based systems. Mohammed and Saha [13] introduced Encrypted-OFDM, a secure wireless waveform using a two-stage time-domain encryption scheme. The method involves scrambling and phase distortion, both of which are controlled by a secret key shared between the transmitter (Alice) and the receiver (Bob). This process eliminates the OFDM signal's orthogonality, making it appear as random noise to eavesdroppers (Eve). Tested on a Universal Software Radio Peripheral (USRP) X310 software-defined radio (SDR) platform, Encrypted-OFDM showed only minor BER degradation (1–4 dB) compared to standard OFDM, while improving resilience to eavesdropping. Since the encryption is applied directly in the time domain, the method maintains power efficiency and does not increase peak-to-average power ratio (PAPR), making it suitable for real-time SDR use.

Hajomer et al. [14] proposed a different approach to secure OFDM transmission using Chaotic-Discrete Hartley Transform (DHT) precoding. DHT is a mathematical method similar to the Discrete Fourier Transform (DFT), but it operates with real numbers. In this scheme, the DHT matrix is permuted based on a four-dimensional hyper-chaotic system. This dynamic system, known for its complex and unpredictable behavior, generates a vast key space of approximately 10^{394} possible combinations. The chaotic transformation randomizes both subcarrier mapping (the assignment of frequency bands to data) and training sequence generation (the creation of reference signals), providing high resistance to brute-force and signal reconstruction attacks. Experimental results on an 8.9 Gb/s optical OFDM system over 20 km of standard single-mode fiber showed a PAPR reduction of 1.8 dB and an improvement in receiver sensitivity of 1.4 dB at a BER of 10^{-3}. These outcomes indicate that chaotic DHT enhances both security and power efficiency without compromising spectral performance.

Recent studies have identified complementary strategies for securing OFDM systems. Encrypted-OFDM employs time-domain cryptographic obfuscation by modifying the transmitted waveform, while Chaotic-DHT utilizes chaos-based frequency randomization for spectral-domain transformation. Building on these approaches, the present work combines the mathematical rigor of Rivest–Shamir–Adleman (RSA) with stochastic subcarrier index scrambling to establish a hybrid RSA–SHA-256 framework for PLS in MIMO-OFDM systems. In this framework, RSA-based scrambling improves confidentiality, and SHA-256 hashing ensures message integrity. The proposed scheme enhances resistance to common modulus and Wiener's attacks, which frequently target RSA-based systems. Implementation and validation on a USRP-based SDR testbed confirm the practicality of real-time secure waveform generation and reception in wireless communication environments.

To address this gap, recent studies have integrated cryptographic algorithms with physical-layer schemes. RSA is a well-established public-key algorithm that uses large integer factorization for encryption and decryption. RSA has been used for secure subcarrier scrambling due to its mathematical strength and resistance to factorization-based attacks [15-16]. Meanwhile, SHA-256 is a widely used hash function that generates fixed-size digests to ensure data integrity. It is adopted for integrity

verification in diverse applications, from image encryption to cloud and healthcare systems [17-19]. Combining RSA scrambling with SHA-256 hashing is a promising hybrid method to ensure confidentiality and integrity [20-22]. However, practical implementation and evaluation in real-time MIMO-OFDM testbeds remain limited in existing literature.

This study proposes a hybrid PLS framework that combines RSA-based subcarrier scrambling with SHA-256 hashing for secure MIMO-OFDM transmission. RSA randomizes subcarrier allocation in the frequency domain, making it difficult for eavesdroppers to reconstruct the modulation pattern without the decryption key. Meanwhile, SHA-256 hashing verifies signal integrity and detects unauthorized modifications during transmission. This dual-layer security resists cryptanalytic techniques such as Wiener's attack and the common modulus attack, which can compromise RSA when small key sizes or shared moduli are used.

This study makes three primary contributions: (1) design and implementation of an RSA–SHA-256 hybrid cryptographic scheme for MIMO-OFDM systems; (2) real-time evaluation using a 2×2 USRP-based SDR testbed; and (3) detailed system analysis, including symbol distribution, deviation metrics, and resistance to advanced RSA attacks. Results show RSA scrambling reduces symbol deviation. Integrating SHA-256 enhances uniformity and enables robust integrity verification. Cryptographic evaluation confirms that RSA with 2048-bit keys and SHA-256 is resistant to Wiener's and common modulus attacks, whereas smaller key configurations are not.

2. Methodology

This study implements a MIMO-OFDM system incorporating a subcarrier scrambling algorithm to rearrange data across frequency channels for enhanced communication security. The system utilizes SDR technology with National Instruments USRP-2920 devices, enabling radio function configuration through software control. A 2×2 MIMO scheme is adopted, in which both the transmitter and receiver are equipped with two antennas. Text data are transmitted, as shown in Fig. 1, and the proposed scrambling mechanism restricts data recovery to authorized receivers.

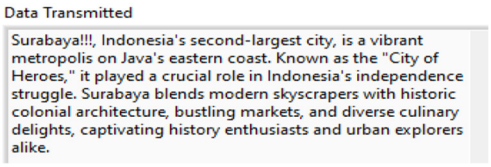
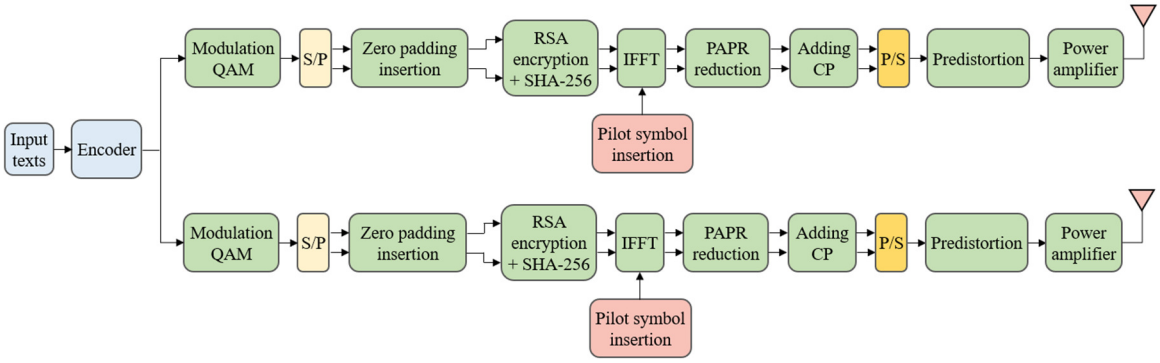


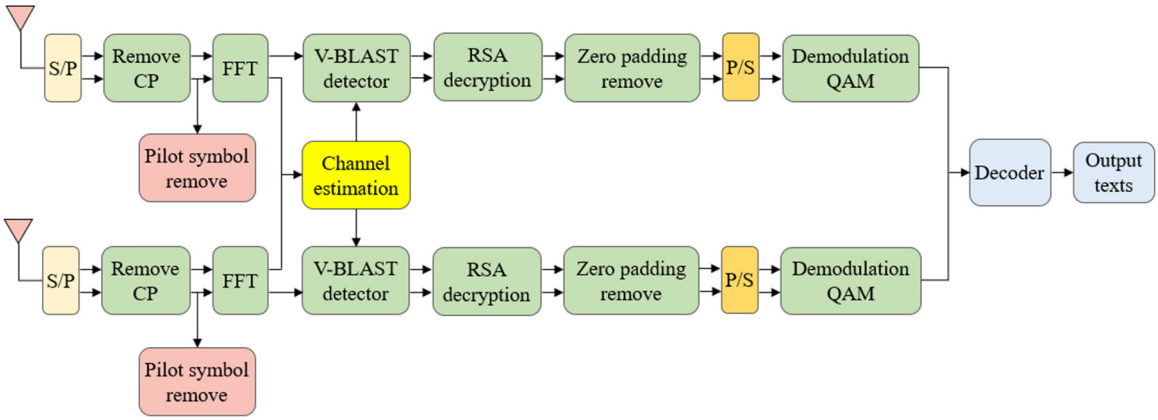
Fig. 1 Data transmitted

Fig. 2 illustrates the implementation process of the subcarrier scrambling algorithm. Applied at both the transmitter and receiver, the algorithm enables encryption at the transmitter (Fig. 2(a)) and decryption at the receiver (Fig. 2(b)). Secure implementation of the algorithm depends on consistent scrambling and descrambling operations at both ends of the communication link. In Fig. 2, the identical processing blocks represent parallel transmit chains corresponding to each antenna in the MIMO configuration.



(a) Block Diagram Transmitter

Fig. 2 Implementation of the subcarrier scrambling process in a transceiver system



(b) Block Diagram Receiver

Fig. 2 Implementation of the subcarrier scrambling process in a transceiver system (continued)

2.1. *Rivest–Shamir–Adleman* (*RSA*)

In the RSA algorithm, both plaintext and ciphertext are represented as integers ranging from 0 to n–1. The value of n can be up to 2048 bits [23]. The plaintext is usually divided into blocks for encryption, and these blocks are transmitted to the receiver. The block size must not exceed $\log_2(n)$; more precisely, the block size is k, where $2^k < n < 2^{k+1}$ [24-25]. Both the sender and the receiver know the value of n. The sender uses the public key (e, n) to encrypt the plaintext block M, while the receiver uses the private key (d, n) to decrypt the corresponding ciphertext block C. The RSA algorithm is illustrated in Fig. 3.

Bob	**Alice**
Key creation	
• Choose secret primes p and q. • Choose encryption exponent e with $\gcd\{e, (p-1)(q-1)\} = 1$. • Publish $N = pq$ and e.	
Encryption	
	• Choose plaintext m. • Use Bob's public key (N, e) to compute $c \equiv m^e \pmod N$. • Send ciphertext c to Bob.
Decryption	
• Compute d satisfying, $ed \equiv 1 \{\mod (p-1)(q-1)\}$. • Compute $m' = c^d \pmod N$. • Then m' equals the plaintext m.	

Fig. 3 RSA algorithm

The steps involved in the RSA algorithm are given below [23]:

Step 1 Choose two different random prime numbers, p and q.

Step 2 Compute the modulus n as:

$$n = p \times q \tag{1}$$

Step 3 Compute Euler's totient function $\varphi(n)$ using:

$$\varphi(n) = (p-1)(q-1) \tag{2}$$

This value is kept private.

Step 4 Select a public exponent e such that:

$$\gcd\{j(n), e\} = 1 \text{ and } 1 < e < \varphi(n) \tag{3}$$

Step 5 Compute the private key d, which is the modular multiplicative inverse of e modulo $\varphi(n)$:

$$d \equiv e^{-1} \{\bmod \varphi(n)\} \tag{4}$$

Step 6 Encrypt the plaintext message M using the public key (e, n) to generate the ciphertext C:

$$C = M^e \bmod n \tag{5}$$

Step 7 Transmit the ciphertext C to the receiver.

Step 8 The receiver decrypts the ciphertext using the private key (d, n) to recover the plaintext M:

$$M = C^d \bmod n \tag{6}$$

2.2. Secure hash algorithm (SHA)

Recently, the SHA family has remained among the most widely used cryptographic hash functions, due to their security properties and standardization status. Earlier functions, such as message digest 5 (MD5) and SHA-1, are now considered insecure following successful cryptanalytic attacks [26]. In contrast, SHA-2 and SHA-3 provide improved resistance to known vulnerabilities [27]. Based on Table 1, larger block sizes and word sizes are associated with higher processing efficiency on 64-bit architectures and increased resistance to cryptographic attacks. However, these advantages are accompanied by increased memory usage and longer computational time, requiring a careful balance between performance and security. The SHA-2 variants (such as SHA-256 and SHA-512) provide a higher level of security than SHA-1, which is now considered unsuitable for modern cryptographic applications.

Table 1 Comparison of SHA parameters

Algorithm	Message size	Block size	Word size	Message digest size
SHA-1	$< 2^{64}$	512	32	160
SHA-244	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

The basic structure of the SHA-256 algorithm can be shown as:

Step 1 Add a single '1' bit to the original message. Then, add enough '0' bits so that the total length is congruent to 448 modulo 512. This leaves space for a 64-bit message length field:

$$\text{length}(M') = 448 \bmod 512 \tag{7}$$

Step 2 Next, append a 64-bit big-endian representation of the original message length l (in bits), where l is the length of the original message in bits, to the padded message, resulting in:

$$M'' = M' \| \text{ 64-bit representation of } l \tag{8}$$

Step 3 Initialize eight 32-bit variables. Use predefined constants from the fractional parts of the square roots of the first eight prime numbers:

$$a_0, b_0, \ldots, h_0 = \text{Initial hash values} \tag{9}$$

Step 4 Divide the padded message into 512-bit blocks:

$$M'' = M_1, M_2, \ldots, M_N \tag{10}$$

Each block prepared in the previous step will be processed independently in the subsequent steps.

Step 5 For each block, prepare a message schedule consisting of 64 words:

$$W_t = \begin{cases} M_i[t], \text{ for } 0 \le t \le 15 \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}, \text{ for } 16 \le t \le 63 \end{cases} \tag{11}$$

W_t represents the t-th element of the message schedule array of 64 words (32-bit values) per 512-bit message block. For t = 0 to 15, W_t, it is derived by dividing the original block into sixteen 32-bit segments. For t = 16 to 63, each W_t value, previous elements with bitwise operations produce diffusion and non-linearity. The functions σ_0 and σ_1 use right rotations (ROTR) and right shifts (SHR) to maximize mixing and resist cryptanalytic attacks. This expansion ensures thorough influence of the original message across compression rounds.

Step 6 For each round Σ_1, compute two temporary variables T_1 and T_2, use logical function choose (*Ch*), majority (*Maj*), bitwise rotations and shifts (Σ_0 and Σ_1):

$$T_1 = h + \sum_1(e) + Ch(e, f, g) + K_t + W_t \tag{12}$$

$$T_2 = \sum_0(a) + Maj(a, b, c) \tag{13}$$

For completeness, all symbols appearing in Eqs. (12) and (13) are defined as follows. The variable K_t denotes the t-th 32-bit round constant in SHA-256, where t = 0, 1, 2, …, 63. These constants are predefined and derived from the fractional parts of the cube roots of the first 64 prime numbers. The function *Ch*(*e*, *f*, *g*) is a nonlinear bitwise operation in SHA-256 where the variable *e* acts as *a* selector, choosing bits from *f* when its corresponding bit equals 1 and from *g* otherwise. The *Maj*(*a*, *b*, *c*) function outputs the majority bit among the corresponding bits of *a*, *b*, and *c*, enhancing nonlinearity and diffusion in the SHA-256 compression stage. The variables *a*, *b*, *c*, *e*, *f*, *g*, and *h* denote the eight 32-bit working registers in the SHA-256 compression process. Σ_0 and Σ_1 represent nonlinear bitwise rotation-based transformation functions applied to the 32-bit working variables *e* and *a* in the SHA-256 compression process.

Step 7 After processing all 64 rounds for block M_i update, the hash values:

$$H_i = H_{i-1} + (a, b, c, d, e, f, g, h) \tag{14}$$

This addition is performed modulo2^{32} for each 32-bit word. After processing all *N* message blocks, concatenate the final values *a*, *b*, *c*, *d*, *e*, *f*, *g*, *h* to form the final 256-bit hash output:

$$\text{Hash}(M) = H = a \,||\, b \,||\, c \,||\, d \,||\, e \,||\, f \,||\, g \,||\, h \tag{15}$$

2.3. Subcarrier randomization with RSA-2048 and SHA-256

Building on the importance of robust cryptographic primitives, subcarrier randomization is a data security technique employed in MIMO-OFDM systems, in which the sequence of subcarriers is scrambled to prevent eavesdroppers from accessing the transmitted information. In this implementation, the scrambling process utilizes the RSA algorithm, which is based on a public–private key cryptographic framework. As shown in Fig. 4, the process begins on the transmitter side (Alice), where OFDM symbols from quadrature amplitude modulation (QAM) modulation are first converted from base-4 to base-10 to improve computational efficiency. After conversion, these symbols are encrypted using Bob's public key, resulting in a ciphertext. Prior to transmission, the ciphertext is processed with SHA-256 to generate a hash code, which is then prepended to ensure data integrity. The placement of this hash code is illustrated in Fig. 5.

Bob	Alice
Key creation	
• Choose secret primes p and q. • Calculating $N - pq$ as modulus $\varphi N = (p - 1)(q - 1)$. • Choose the encryption exponent e such that $\gcd\{e, \varphi N\} = 1$. • Publish (e, N) as a public key.	
Encryption	
	• Encrypting the subcarrier sequence using the public key (e, N) with equation $c \equiv m^e \pmod N$. • Converting ciphertext to hash form using SHA-256. • Sending the ciphertext c with the added hash to Bob.
Decryption	
	• Separating hash from ciphertext. • Converting ciphertext into a hash using SHA-256 as a comparison hash. • Comparing the hash with the reference hash. If they match, the decryption process continues; if they don't match, it does not continue. • Calculate d such that $e.d \equiv 1 \pmod{\varphi N}$. • Decryption of c with the equation $m' = c^d \pmod N$ that provides the actual subcarrier sequence.

Fig. 4 Subcarrier randomization algorithm

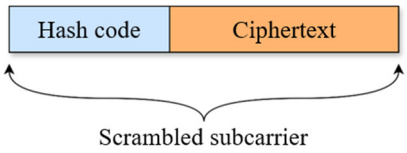


Fig. 5 Illustration of hash code placement within the ciphertext

On the receiver side (Bob), the process begins with the encrypted data, which combines the ciphertext and the hash code. This combined data is converted back to base-4 to match the original OFDM symbol format. Upon reception, Bob converts the OFDM symbols to base-10 for computational efficiency, then separates the ciphertext from the hash and verifies its integrity by applying SHA-256 to the ciphertext and comparing it with the attached hash. If the hashes match, RSA decryption proceeds using the private key and modulus N to recover the original subcarrier order. If the hashes do not match, decryption halts, and a notification indicates possible data tampering.

The proposed scrambling process is performed in the frequency domain before the inverse fast Fourier transform (IFFT) stage, thereby preserving each subcarrier's QAM constellation and avoiding waveform distortion. In contrast, time-domain scrambling modifies the composite OFDM signal after the IFFT stage. As a result, frequency-domain scrambling preserves modulation integrity, enables direct subcarrier and spectrum control, and simplifies SDR implementation. Additionally, time-domain scrambling tends to increase the PAPR and may disrupt receiver synchronization, whereas the proposed method maintains low complexity and stable BER performance. The subcarrier scrambling process occurs in the frequency domain prior to the IFFT block. In this stage, subcarrier index mapping is randomized using the RSA encryption key, thereby hindering unauthorized receivers from reconstructing the original data without the corresponding decryption key. The modulation symbols remain unchanged, ensuring that system performance with respect to constellation distortion is maintained.

Fig. 6 illustrates the subcarrier allocation in the frequency domain before and after the implementation of the proposed RSA-based scrambling process. In the pre-scrambling scenario, QAM symbols are sequentially mapped to fixed subcarrier indices, resulting in a deterministic and predictable frequency allocation. After scrambling, the subcarrier indices are permuted according to the RSA-generated scrambling pattern, while the QAM constellation points remain unchanged. This demonstrates

that the scrambling operation modifies only the subcarrier index mapping without introducing constellation distortion. Consequently, modulation integrity and BER performance are preserved, and the randomized subcarrier positions enhance resistance to eavesdropping and subcarrier-reconstruction attacks.

The figures depict the subcarrier mapping process in an MIMO-OFDM system before and after scrambling. In the 'Before scrambling' scenario, each symbol produced by QAM modulation (S_0–S_7) is sequentially mapped to subcarrier indices (0–7). In the 'After scrambling' scenario, the subcarrier indices are reordered using the RSA encryption key (for example, [3, 7, 0, 5, 2, 6, 1, 4]), while the QAM symbols remain unchanged. Consequently, the modulation constellation is preserved, but the subcarrier index mapping is randomized, which hinders unauthorized receivers from reconstructing the data sequence without the appropriate decryption key.

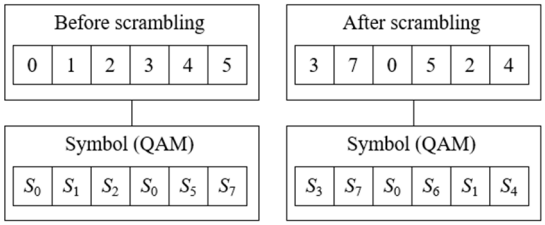


Fig. 6 Illustrates the subcarrier allocation before and after scrambling

2.4. Wiener's attack

Wiener's attack is a cryptanalytic technique targeting RSA systems by exploiting continued fractions to recover the private exponent d, particularly when d is small, specifically when $d < 1/3\sqrt[4]{n}$, where n is the RSA modulus [28]. This vulnerability arises when the public exponent e is chosen as a small number, and its modular inverse d is calculated such that it is expressed as in Eq. (4), where $\varphi(n)$ is Euler's totient function, n is the RSA modulus, e is the public exponent, and d is the private exponent. Under these conditions, an attacker can leverage the relationship between e, d, and n to perform a continued fraction attack and efficiently recover the private key. The attack proceeds through the following key steps:

Step 1 The public exponent e and modulus n are used to form the fraction e/n, which is then expressed as a continued fraction expansion, denoted as:

$$\frac{e}{n} = \left[a_0, a_1, a_2, \ldots, a_k\right] \tag{16}$$

where a_0, a_1, a_2,⋯, a_k are the terms of the continued fraction. These terms are later used to generate convergents k/d, which serve as candidates for the private exponent d in the RSA cryptosystem.

Step 2 Compute each convergent from the continued fraction expansion of the public exponent fraction e/n using Eq. (16). Recursively evaluate each convergent k/d as shown in:

$$\frac{k}{d} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}} \tag{17}$$

These convergents are considered as candidates for the private key parameter k/d. Here, k and d are integers, and d is the possible RSA private key exponent. The algorithm iteratively checks each convergent. It determines whether the key congruence condition in Step 3 is satisfied.

Step 3 For each candidate k/d, the numerator is assigned to k and the denominator to d. The algorithm first verifies whether d is odd. If it is, then it checks the congruence condition as:

$$e.d \equiv 1(\text{mod } k) \tag{18}$$

If this congruence is satisfied, the algorithm computes Euler's totient function using the following:

$$\varphi(n) = \frac{ed-1}{k} \tag{19}$$

Once $\varphi(n)$ is known, the algorithm constructs the following quadratic polynomial, which is used to recover the prime factors p and q.

$$x^2 - \left[n - \varphi(n) + 1\right]x + n = 0 \tag{20}$$

If this equation yields integer roots, then the correct private key d has been recovered.

Step 4 If none of the tested convergents satisfy the conditions in Steps 3 and 4, then it can be concluded that the given RSA parameters (e, n) are not vulnerable to Wiener's attack.

2.5. Common modulus attack

A common modulus attack involves generating different public and private key pairs for multiple users, all sharing the same modulus. The steps for this attack are outlined below [26]:

Step 1 The common modulus attack begins by preparing several key elements. These include the modulus N, two public exponents e_1 and e_2, and two ciphertexts c_1 and c_2. Both ciphertexts are created from the same original message m, but each is encrypted with a different public exponent. In other words, $c_1 = m^{e1} \bmod N$ and $c_2 = m^{e2} \bmod N$, where N is the shared modulus.

Step 2 Before proceeding, it is necessary to verify that the public exponents e_1 and e_2 are relatively prime:

$$\gcd(e_1, e_2) = 1 \tag{21}$$

This condition is crucial for applying the extended Euclidean algorithm in the subsequent step.

Step 3 Use the Extended Euclidean Algorithm to find integers a and b such that:

$$ae_1 + be_2 = 1 \tag{22}$$

These coefficients a and b are crucial for reconstructing the original message without needing to factor N.

Step 4 Once a and b has been determined, the next step is to compute the original message m. This is done by combining the ciphertexts using the computed exponents:

$$m = c_1^a c_2^b \bmod N \tag{23}$$

If either a or b is negative, the corresponding modular inverse is used to ensure the operation remains valid.

Step 5 Finally, the result of this calculation yields the original message m, without requiring factorization N. Thus, the common modulus attack enables an attacker to exploit two different ciphertexts encrypting the same message under the same modulus, effectively recovering sensitive information without breaking RSA through traditional factoring methods.

3. Result and Discussions

The RSA–SHA-256 algorithm was evaluated experimentally within a real-time MIMO-OFDM system implemented on the USRP N2920 platform. The testbed supports over-the-air transmission and reception, enabling assessment of the proposed

security scheme under realistic channel and hardware conditions. Encryption and scrambling were performed at the transmitter before OFDM modulation, while descrambling and decryption were performed at the receiver. Key OFDM system parameters used in the experiments, such as fast Fourier transform (FFT) size, subcarrier spacing, modulation order, and cyclic prefix length, are summarized in Table 2. These parameters directly influence spectral efficiency, resilience to multipath fading, and overall system performance.

Table 2 OFDM parameters

Parameter	Value
FFT size	128
Number of subcarriers	128
Number of data subcarriers	96
Number of pilot subcarriers	8
Number of null subcarriers	24
Modulation	4-QAM
Cyclic prefix	32
Center frequency	915 Hz
Channel estimation	Least square
Channel equalizer	Zero forcing

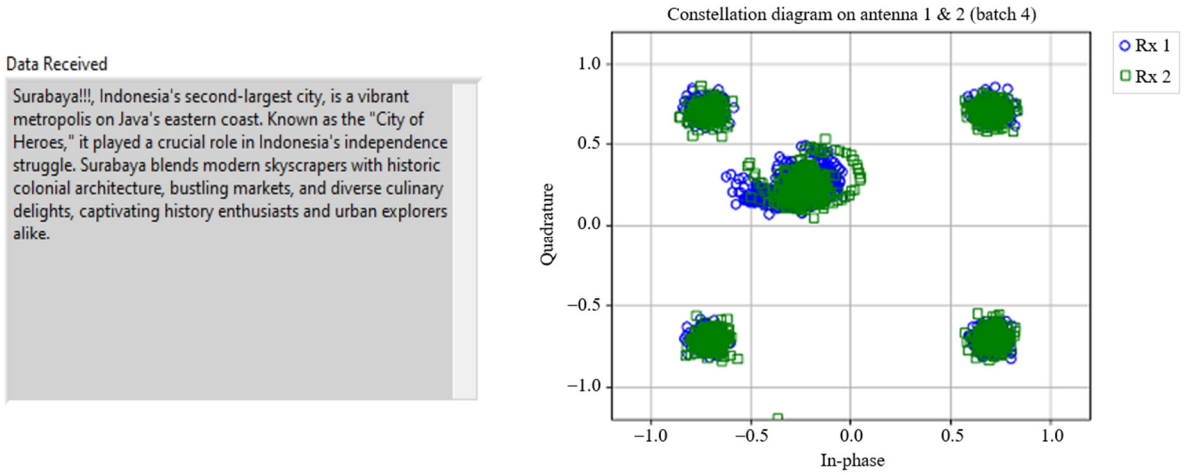
The chosen OFDM parameters align with those specified in contemporary wireless communication standards, particularly IEEE 802.11ac. IEEE 802.11ac utilizes wide bandwidths, high-order modulation schemes, and efficient OFDM configurations to achieve high data rates and enhanced spectral efficiency. By adopting comparable parameter settings, the experimental setup replicates realistic deployment scenarios and allows the evaluation of the RSA–SHA-256–based scrambling scheme under conditions relevant to practical applications. The results thus indicate the feasibility of the proposed security approach and its potential applicability to modern Wi-Fi and broadband wireless systems.

The experimental setup utilized two USRP N2920 devices as transmitters and two USRP devices as receivers, as shown in Fig. 7. This configuration models a practical 2×2 MIMO-OFDM transmission scenario with multiple receiving nodes. Receivers were categorized as authorized or unauthorized to evaluate the effectiveness of the RSA–SHA-256–based scrambling scheme in enhancing PLS. Authorized receivers, equipped with the correct cryptographic keys, were able to descramble and decode the transmitted data. In contrast, unauthorized receivers attempted to demodulate the signal without access to the scrambling key, allowing for an assessment of security performance under eavesdropping conditions.

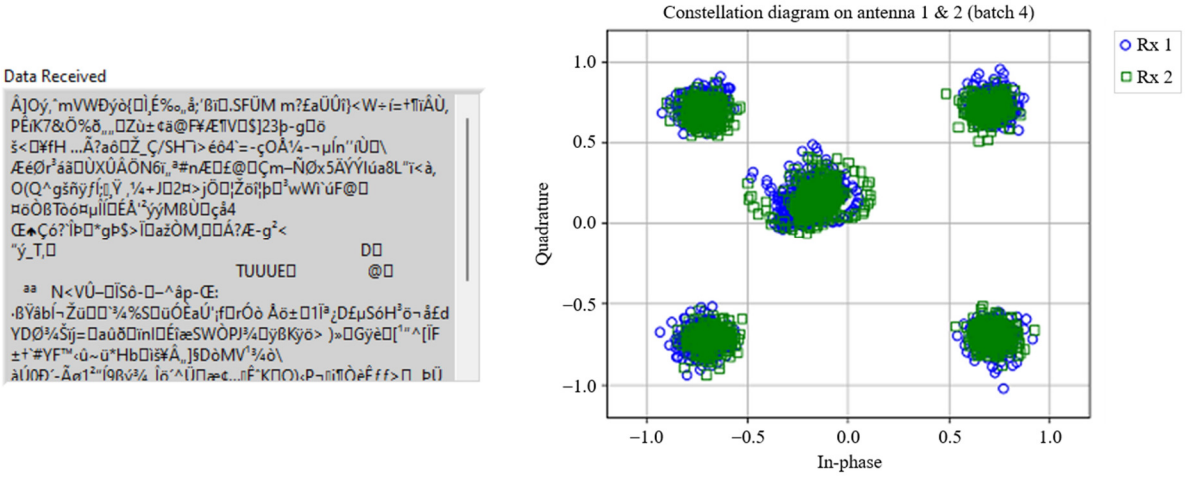
Fig. 7 System testing scenario

3.1. Evaluation with authorized and unauthorized receiver

In Fig. 8, the receiver side is categorized into authorized and unauthorized receivers. An authorized receiver is assumed to possess the RSA–SHA-256 algorithm to decrypt the transmitted message, whereas an unauthorized receiver does not have access to the algorithm. Fig. 8(a) shows that the authorized receiver received the transmitted text without character errors and with well-clustered constellation points. In contrast, Fig. 8(b) shows that the unauthorized receiver received text data containing multiple character errors, although the constellation diagram exhibited little dispersion.



(a) Authorized receiver



(b) Unauthorized receiver

Fig. 8 Received signal constellation at the receiver side

Performance evaluations and constellation visualizations were conducted at a fixed signal-to-noise ratio (SNR) of 20 dB. This SNR level represents a moderate channel condition in which noise effects are present but do not dominate the received signal. Under these conditions, the influence of the proposed RSA–SHA-256–based scrambling scheme can be distinguished from severe channel impairments. Employing a fixed SNR allows a fair comparison between authorized and unauthorized receivers, as both are subject to identical channel conditions. Consequently, any observed degradation in constellation clarity or decoding performance at the unauthorized receiver can be primarily attributed to the scrambling mechanism rather than channel noise.

To clarify, the base-4-to-base-10 conversion is introduced solely for RSA processing and does not affect modulation performance. The BER performance of legacy OFDM and the proposed RSA–SHA-256–based MIMO-OFDM system was evaluated over multiple SNR levels. Both systems exhibit nearly identical BER trends. The slight difference (<10%) is attributed to processing overhead rather than constellation distortion. After encryption and decryption, QAM modulation continues in base-4, so the physical-layer constellation remains unaffected by the base-10 conversion step. This paragraph

compares the BER performance of the legacy OFDM (4-QAM), RSA-OFDM with base-10 conversion, and the proposed hybrid RSA–SHA-256 MIMO-OFDM system, as shown in Fig. 9. The results indicate that the BER difference across the three systems remains below 10% over the 0–30 dB SNR range, demonstrating that the base-10 conversion in RSA encryption has a negligible impact on physical-layer performance. The minor BER variation is primarily attributable to computational and mapping overhead rather than constellation distortion.

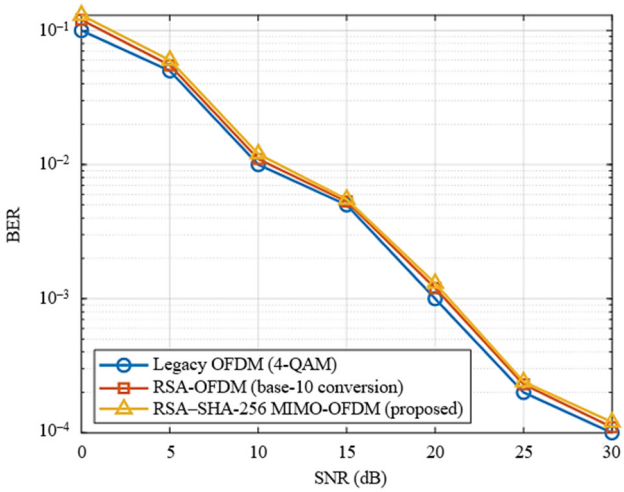
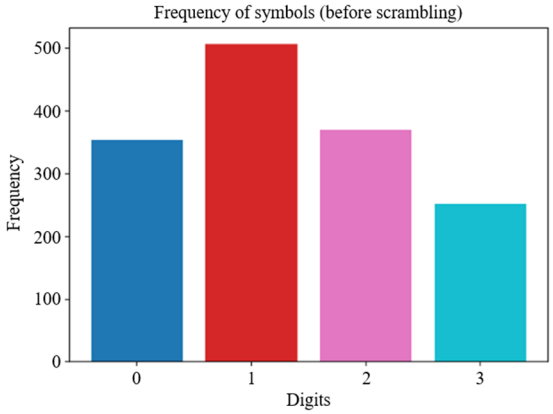


Fig. 9 BER performance of legacy and secure MIMO-OFDM systems

3.2. Evaluation symbol distribution

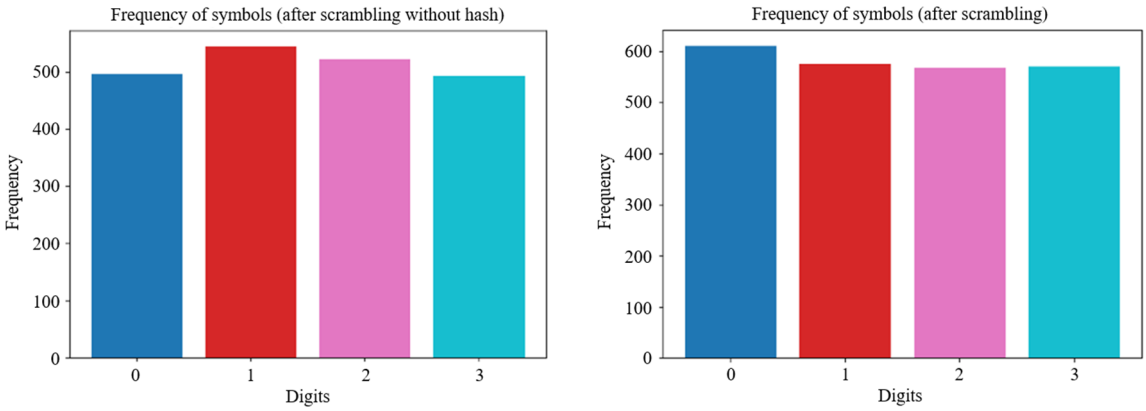
This step evaluated symbol distribution to assess the effectiveness of the RSA-based scrambling process and SHA-256 hash integration in the MIMO-OFDM system. The analysis compared base-4 digit frequencies at three stages: before scrambling, after scrambling without hashing, and after scrambling with hashing.

Fig. 10 presents an evaluation of base-4 symbol distribution (digits 0–3) under three different conditions in the MIMO-OFDM system. In Fig. 10(a), which represents the distribution before scrambling, the symbol frequencies are highly uneven; digit ‘1’ appears most frequently, while digit ‘3’ appears least frequently. This imbalance suggests the presence of patterns that could potentially be exploited by unauthorized receivers, thereby reducing system security. Fig. 10(b) shows the distribution after scrambling without hash integration. Although the distribution becomes more balanced, slight variations remain, indicating that scrambling alone may be insufficient. Fig. 10(c) illustrates the distribution after the combination of scrambling and SHA-256 hashing. In this case, the symbol frequencies are nearly identical, indicating that the combined approach achieves a more uniform symbol distribution. Such uniformity is important for enhancing security, as it reduces exploitable patterns and increases resistance to statistical analysis or prediction.



(a) Before scrambling

Fig. 10 Frequency of symbols



(b) After scrambling without a hash (c) After scrambling with SHA-256

Fig. 10 Frequency of symbols (continued)

Table 3 presents the symbol distribution and standard deviation results. Scrambling substantially balances the symbol distribution in the MIMO-OFDM system. Before scrambling, the distribution was highly imbalanced: symbol '01' occurred most frequently with a proportion of 0.34, whereas symbol '11' was the least frequent at 0.17. This imbalance resulted in a high standard deviation of 90.76, increasing susceptibility to unauthorized pattern analysis. After scrambling with the RSA algorithm, the symbol distribution became more uniform, with frequencies ranging from 0.23 to 0.29. The standard deviation decreased to 16.45, indicating a more randomized symbol distribution. This result indicates that the RSA algorithm effectively enhances data randomness and security.

In the third scenario, RSA is combined with the SHA-256 hash function. The symbol distribution becomes nearly uniform, with all four symbols exhibiting a probability of 0.25. The standard deviation in this scenario is 17.27, which is slightly higher than that obtained with RSA alone. Nevertheless, the overall symbol distribution shows improved uniformity. This indicates that integrating RSA with SHA-256 contributes to more effective symbol scrambling. The resulting distribution exhibits both enhanced statistical randomness and uniformity across symbols, which is important for maintaining data confidentiality and reducing the risk of pattern analysis by unauthorized parties.

Table 3 Comparison of standard deviation between MIMO-OFDM and MIMO-OFDM with subcarrier scrambling

Scenario	Mean symbol				Standard deviation
	00	01	10	11	
Before scrambling	0.24	0.34	0.25	0.17	90.75
Scrambling (RSA)	0.29	0.26	0.23	0.22	16.45
Scrambling (RSA + SHA-256)	0.26	0.25	0.25	0.25	17.26

The compatibility of the proposed scheme with higher-order modulation formats was evaluated by testing the system with 16-QAM, 64-QAM, and 256-QAM under identical channel and system conditions, as shown in Fig. 11. As the modulation order increases, the BER increases due to the reduced Euclidean distance between constellation points. Nevertheless, the scrambling behavior remains consistent across different modulation orders, as evidenced by the uniform symbol distribution and the preserved resistance to RSA-based attacks. These results indicate that the proposed RSA–SHA-256–based subcarrier scrambling approach exhibits robustness that is largely independent of modulation order, supporting its applicability to secure communications employing high-order modulation schemes such as 256-QAM.

Comparison of BER for the proposed RSA–SHA-256–based MIMO-OFDM system using QPSK (4-QAM), 16-QAM, 64-QAM, and 256-QAM under identical channel conditions. As the modulation order increases, the BER increases due to the reduced Euclidean distances between constellation points, which makes the system more sensitive to noise. Nevertheless, the scrambling uniformity and cryptographic effectiveness remain consistent across different modulation orders, indicating that the proposed approach is scalable and largely independent of the modulation format.

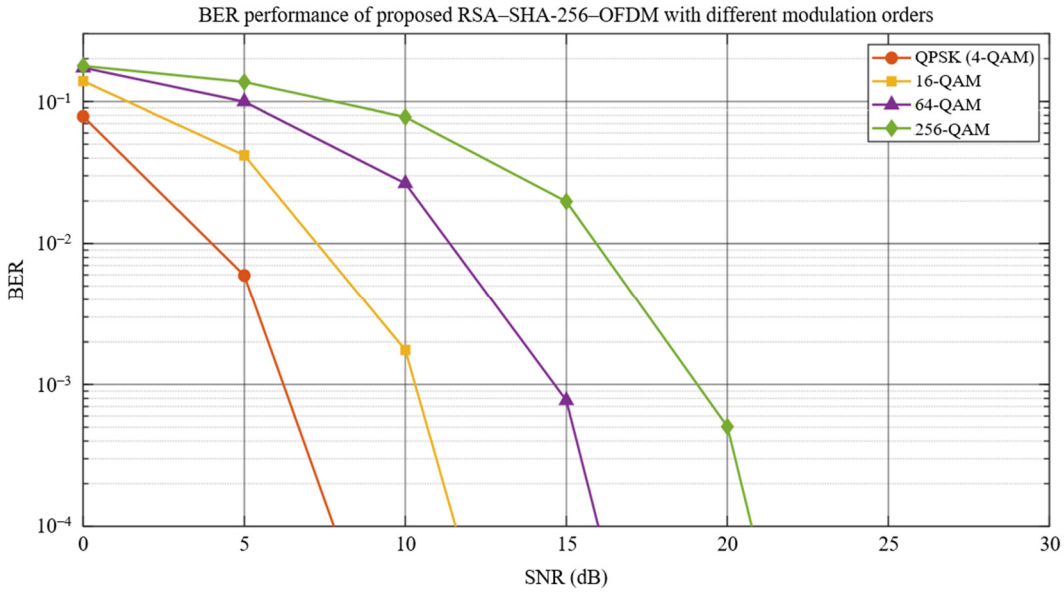


Fig. 11 BER vs SNR for different modulation orders

3.3. Evaluation of RSA–SHA-256 with Wiener's attack and common modulus attack

The RSA–SHA-256 algorithm implemented in the MIMO-OFDM system was evaluated against two well-known RSA attacks: the Wiener attack and the common modulus attack. These evaluations examined whether the proposed scheme can withstand typical vulnerabilities associated with RSA-based systems. For the Wiener attack, the evaluation focused on whether the private exponent d is sufficiently large to prevent efficient recovery through continued fraction analysis. The common modulus attack assessed the system security in scenarios where different users share the same modulus N but employ different public exponents e.

The Wiener attack analysis was conducted using different RSA parameter sets, as summarized in Table 4. Two parameter sets were examined, each consisting of a public exponent and a modulus. The modulus is assumed to be unfactored. Parameter Set 1 satisfies the known vulnerability conditions for a Wiener attack, whereas Parameter Set 2 employs a commonly used public exponent representative of typical RSA implementations.

Table 4 RSA parameter configuration for Wiener attack testing

No	Parameter	Value	Result	No	Parameter	Value	Result
1	e	e 364089297256633736371348118514445348660 422783054505185981131140314875399419168 869745566811711488096062244228850656054 683798599325361038510123817237961701698 569599435695930452286212666915986364801 113169473677089800150075530816113288544 307122777998519655760825745014614917605 935351438737551196806582464931627 25	Found	2	e	e 65537	Not Found
	N	N 926192069893001210798975756534224062403 963480547193697142686597531486177910356 433646186747311329598695159407111253817 824438502388919928629295526506635877219 804239984413327907018102577781372197313 990656342862475173761926440792095293363 801945998601604297103194588795707886706 474769545926238041786969583658671 23			N	N 109168092567607198158464717074958164286 969425169536252849709733568149113631065 135205433830830238057794421376741281917 667020966152951228489052586100822081200 619831753331388811966170362177084171535 413435221673286762729814859427008216259 516094764862176809206072669646354772741 753341482349170988423973074911062991147 881107207628332522568248523778 79	

To evaluate this vulnerability, Wiener's attack was implemented as pseudocode. The process begins by computing the continued fraction expansion of e/n. The corresponding convergents k_i/d_i are then generated. Each candidate d_i is examined by testing whether it correctly decrypts a ciphertext to recover the original plaintext. If a valid candidate is identified, the private exponent d is successfully recovered. The pseudocode for Wiener's attack is presented in Table 5:

Table 5 Pseudocode Wiener's attack

```
function WienerAttack(e, n)
    Choose a random test message m
    Compute ciphertext c ← m^e mod n
    Compute continued fraction expansion CF ← e/n
    Generate a convergent list Conv from CF
    for each fraction i in Conv do
    k ← numerator( )
    d ← denominator( )
      if k = 0 then
            continue
        end if
        try
           m1 ← c^d mod n
           if m1 = m then
              return d
           end if
        catch error
           continue
     end for
     return "Not Found"
end function
```

When RSA–SHA-256 uses Parameter Set 1, the result indicates that the private key is successfully recovered, with the status marked as "Found," revealing a vulnerability to Wiener's attack. This outcome occurs because Parameter Set 1 employs a small private exponent, which satisfies the mathematical condition for a successful Wiener attack, specifically when the private exponent is smaller than one-third of the fourth root of the modulus. In contrast, when Parameter Set 2 is used, the private key cannot be recovered. This is because Parameter Set 2 adopts a commonly used public exponent, which typically results in a sufficiently large private exponent, thereby making the RSA system resistant to Wiener attacks.

Table 6 Pseudocode common modulus attack

```
function CommonModulusAttack(m, e1, e2, p, q)
    n ← n × q
    φ ← (p − 1)(q − 1)
    if gcd (e1, φ) ≠ 1 or gcd (e2, φ) ≠ 1 then
        return "Invalid exponents"
    end if
    c1 ← m^e1 mod n
    c2 ← m^e2 mod n
    c1 ← SHA-256(Base4(c1)) → Decimal → Base4
    c2 ← SHA-256(Base4(c2)) → Decimal → Base4
    (g, a, b) ← ExtendedEuclidean(e1, e2)
    if a < 0 then
        c1 ← inverse(c1, n)^−a mod n
    else
        c1 ← c1^a mod n
    end if
    if b < 0 then
        c2 ← inverse(c2, n)^−a mod n
    else
        c2 ← c2^a mod n
    end if
    ct ← (c1 × c2) mod n
    m2 ← (ct)^(1/g)
    if m2 = m then
        return "Attack Successful", m2
    else
        return "Attack Failed"
    end if
end function
```

The common modulus attack algorithm is presented in Table 6. The process begins by computing the modulus $n = p \times q$ and Euler's totient $\varphi(n)$. The public exponents e_1 and e_2 are verified to be relatively prime to $\varphi(n)$. The plaintext m is then encrypted into two ciphertexts, each of which is transformed into base-4, hashed using SHA-256, and reconverted. Using the extended Euclidean algorithm, coefficients a and b are obtained such that $ae_1 + be_2 = g$. Depending on the signs of a and b, the ciphertexts are processed using either positive exponents or modular inverses. A combined ciphertext is then computed, and the g-th root is taken to recover the message. If the recovered message matches the original plaintext, the attack is considered successful; otherwise, it fails.

Furthermore, to ensure that the proposed security algorithm is resistant to attacks during channel transmission, a common modulus attack test was conducted. The common modulus attack targets RSA systems in which two or more parties use the same modulus N but different public exponents e to encrypt the same message. In this evaluation, the RSA key size was varied using 512-bit, 1024-bit, and 2048-bit configurations. The results of RSA system testing against both Wiener's attack and the common modulus attack, with and without hash-code integration, for key sizes of 512, 1024, and 2048 bits are summarized in Table 5.

Based on the results in Table 7, RSA without SHA-256 is shown to be vulnerable to the common modulus attack at key sizes of 512, 1024, and 2048 bits. In contrast, integrating SHA-256 significantly improves resistance to the common modulus attack, even for RSA configurations with 512- and 1024-bit keys. Wiener's attack is effective primarily against RSA systems with small key sizes and becomes ineffective at the 2048-bit level. Overall, the combination of a 2048-bit RSA key and SHA-256 demonstrates resistance to both Wiener's attack and the common modulus attack under the evaluated conditions.

Table 7 Test results of Wiener's and common modulus attacks for various RSA key sizes

Scenario	Common modulus attack	Wiener's attack
RSA-512	Attackable	Attackable
RSA-512 + SHA-256	Resistant	Attackable
RSA-1024	Attackable	Attackable
RSA-1024 + SHA-256	Resistant	Attackable
RSA-2048	Attackable	Resistant
RSA-2048 + SHA-256	Resistant	Resistant

3.4. Evaluation of comparative performance with prior works

This subsection compares the proposed RSA–SHA-256–based frequency-domain subcarrier scrambling scheme with two representative PLS approaches: (i) Encrypted-OFDM [13] and (ii) Chaotic -DHT precoding [14]. The comparison considers BER performance, SNR penalty, sensitivity improvement, and modulation-order scalability for 4-QAM, 16-QAM, 64-QAM, and 256-QAM. Encrypted-OFDM applies encryption in the time domain, which distorts the OFDM waveform and disrupts subcarrier orthogonality. Such distortion increases the burden on channel estimation and equalization, resulting in a reported SNR penalty of approximately +2–4 dB (with an average of about 2.5 dB at a BER of approximately 10^{-3}) [13]. Consequently, Encrypted-OFDM requires a higher SNR to achieve the same BER performance as conventional OFDM, particularly for low-to-medium modulation orders. Although time-domain encryption enhances confidentiality, the induced waveform distortion leads to noticeable degradation in physical-layer performance.

Fig. 12 shows that the baseline OFDM system provides a reference lower bound for BER performance, as it operates close to the Shannon capacity under a normalized and ideal channel assumption. Consequently, additional security schemes or signal-processing techniques are not expected to outperform the baseline system in terms of BER at the same SNR. Encrypted-OFDM exhibits an SNR penalty due to signal distortion introduced by time-domain encryption. In contrast, the Chaotic-DHT curves closely overlap with the baseline OFDM results, indicating that this scheme does not improve channel capacity under

ideal conditions. The BER improvement reported for Chaotic-DHT [14] arises primarily from reduced PAPR and mitigation of hardware nonlinearity in non-ideal optical communication systems, rather than from exceeding the Shannon capacity limit. The proposed RSA–SHA-256 scheme achieves BER performance nearly identical to that of the baseline OFDM, demonstrating that frequency-domain subcarrier index randomization does not alter modulation symbols or signal power characteristics.

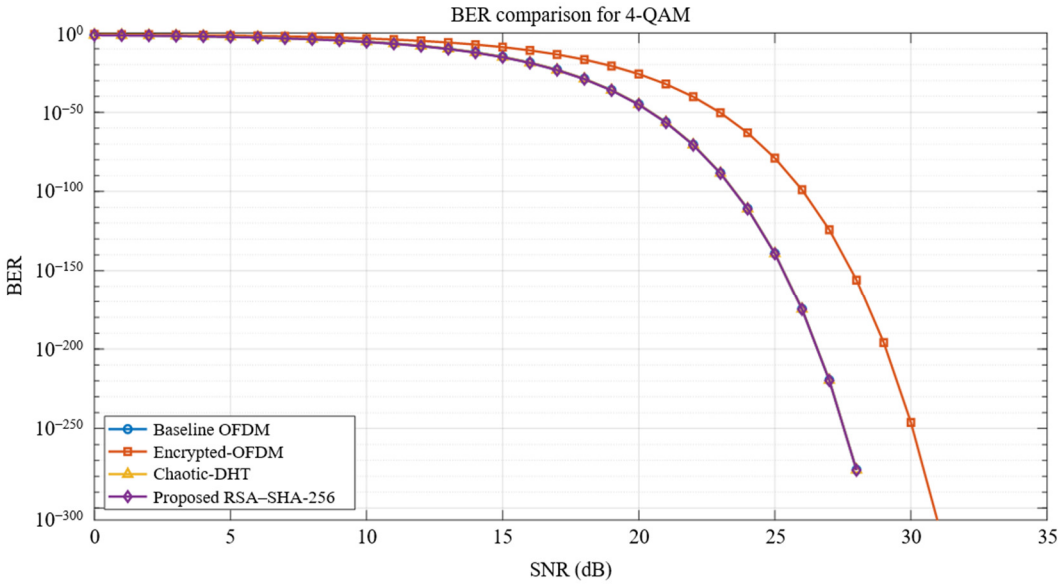


Fig. 12 BER performance comparison of OFDM-based systems

Compared to these approaches, the proposed RSA–SHA-256 scheme scrambles only subcarrier indices during frequency-domain mapping. The QAM symbols remain unchanged, preserving constellation integrity and avoiding waveform distortion. Consequently, the method incurs a negligible SNR penalty and achieves a BER performance comparable to baseline OFDM across all tested modulation orders. Fig. 12 illustrates this stable BER performance under QPSK (4-QAM) modulation. BER performance for higher-order modulations (16-QAM, 64-QAM, and 256-QAM) is presented in Fig. 13, confirming modulation-order independence. Integrating SHA-256 provides strong data integrity assurance, while RSA-2048 delivers high cryptographic strength, surpassing that of chaos-based schemes.

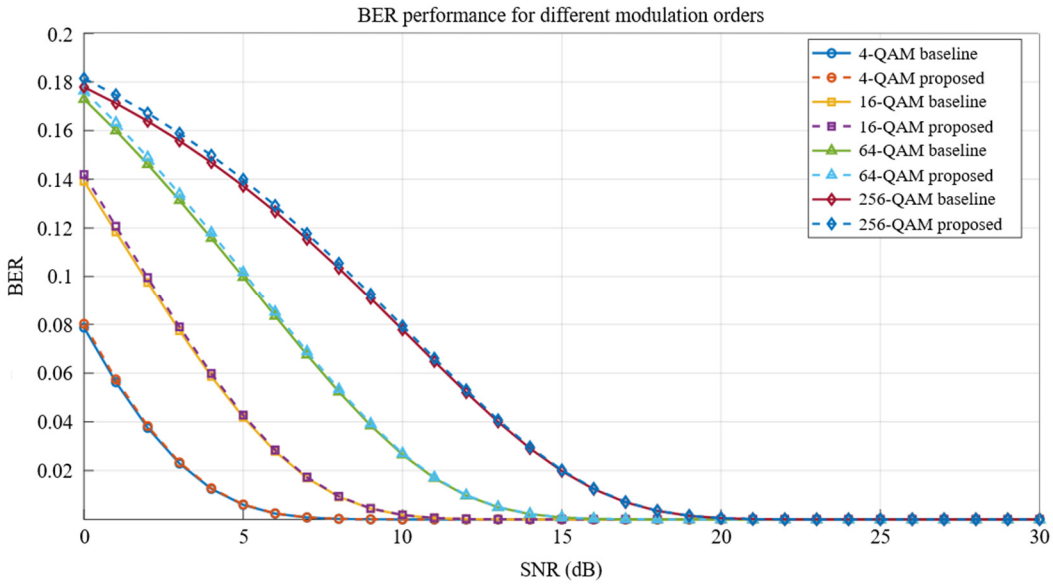


Fig. 13 The BER performance for higher-order modulations (16-QAM, 64-QAM, and 256-QAM)

Table 8 summarizes the SNR penalty, PAPR effect, constellation distortion, and security strength. Compared with existing PLS techniques, the proposed scheme provides stronger cryptographic security, reduced performance degradation, and greater compatibility with SDR-based MIMO-OFDM systems, making it a practical and efficient alternative. It should be noted that

the reported SNR gain for Chaotic-DHT is observed in non-ideal optical intensity-modulation/direct-detection (IM/DD) systems, primarily due to PAPR reduction and mitigation of hardware non-linearities. Under ideal AWGN or Rayleigh channels with normalized noise power, however, Chaotic-DHT does not surpass baseline OFDM and remains consistent with the Shannon capacity limit.

Table 8 Comparison of physical-layer security methods

Feature	Encrypted-OFDM [13]	Chaotic-DHT [14]	Proposed RSA–SHA-256
Encryption domain	Time-domain	Frequency-domain (chaotic precoding)	Frequency-domain (subcarrier scrambling)
SNR impact at BER 10^{-3}	+2–4 dB penalty (average ≈ 2.5 dB)	≈ 0 dB under ideal channels (AWGN/Rayleigh channels); ≈ −1.4 dB gain in optical IM/DD systems	≈ 0 dB (negligible SNR penalty)
Effect on constellation	Distorted (time-domain modification)	Preserved	Preserved
PAPR impact	Higher	Reduced (≈ −1.8 dB at CCDF = 10^{-2})	No change
Security strength	Weak cryptographic basis	Chaos-based (non-cryptographic)	Strong cryptographic security (RSA-2048 + SHA-256)
Implementation	High receiver complexity	Moderate	Simple SDR implementation, low overhead
Modulation scalability	Limited (sensitive to higher-order modulation)	Limited by chaotic processing complexity	High (4-QAM–256-QAM)

The hybrid RSA–SHA-256 scheme introduces additional cryptographic operations; however, the associated computational load remains manageable for real-time MIMO-OFDM transmission. RSA encryption is applied exclusively to subcarrier index mapping, which involves a substantially smaller data vector than the time-domain waveform, while SHA-256 operates on a fixed-length digest. Because scrambling is performed in the frequency domain before the IFFT, the proposed scheme does not introduce waveform distortion, iterative decoding, matrix inversion, or chaos-based transformations. Experimental validation on the USRP-2920 SDR platform demonstrates that the processing latency remains within a single OFDM symbol duration. These results indicate that the proposed scheme preserves Shannon-consistent BER performance while maintaining real-time operability and enhanced data confidentiality and integrity.

The analysis addresses common cryptanalytic and physical-layer attacks in multicarrier wireless systems, including subcarrier reconstruction and key-recovery attempts. Each scheme is evaluated based on its ability to preserve symbol confidentiality and prevent unauthorized signal interpretation. The proposed RSA–SHA-256–based approach is examined in detail to demonstrate its effectiveness in maintaining security while minimizing performance degradation. This comparative discussion elucidates the trade-offs between security strength and system complexity among the evaluated schemes.

This section clarifies the security robustness of the system by describing attack models relevant to the compared schemes and evaluating the robustness of the proposed RSA–SHA-256–based MIMO-OFDM scheme. At the physical layer, common threats include passive eavesdropping, brute-force key guessing, statistical pattern analysis, replay attacks, and cryptanalytic attacks that exploit weak key structures. To further evaluate the security robustness under the aforementioned attack models, the following schemes are analyzed in terms of their resistance to these threats.

(1) In Encrypted-OFDM, the security mechanism relies on time-domain waveform distortion. Although this approach provides a degree of obfuscation against passive eavesdroppers, the protection is not cryptographic and is accompanied by signal distortion, which degrades BER performance. Moreover, once synchronization and channel estimation are successfully achieved, attackers may exploit residual signal structures, thereby limiting resilience to advanced signal-processing attacks.

(2) The Chaotic-DHT scheme employs chaos-based frequency-domain precoding to increase signal randomness and provide resistance to basic eavesdropping. However, its security is non-cryptographic and depends heavily on the confidentiality of

the chaos parameters. The scheme remains vulnerable to parameter estimation, statistical reconstruction, and brute-force attacks, particularly when the chaos map or initial conditions are partially exposed. Moreover, Chaotic-DHT does not inherently provide mechanisms for data integrity protection.

(3) The proposed RSA–SHA-256 scheme achieves enhanced robustness by combining cryptographic key-based randomization with integrity verification. Subcarrier indices are randomized using RSA encryption, preventing unauthorized receivers from reconstructing the correct frequency mapping even when modulation and channel parameters are known. Because RSA encryption is applied exclusively to subcarrier indices rather than time-domain samples, brute-force attacks and statistical analysis become computationally infeasible for sufficiently large key sizes, such as RSA-2048. In addition, SHA-256 enables data integrity verification, allowing effective detection of replay attacks and data manipulation.

In summary, the proposed scheme provides multi-layered protection by addressing confidentiality, integrity, and resilience against both cryptanalytic and signal-processing-based attacks, while maintaining BER performance consistent with Shannon capacity limits and enabling real-time operability on SDR systems.

4. Conclusions

A hybrid PLS scheme for MIMO-OFDM systems is presented and experimentally validated, integrating RSA-based subcarrier index scrambling with SHA-256 hashing for data integrity verification. The proposed approach operates exclusively in the frequency domain, thereby preserving the inherent OFDM waveform characteristics while enhancing both confidentiality and integrity. Implementation and evaluation on a real-time 2×2 MIMO-OFDM SDR testbed using USRP devices demonstrate practical feasibility under realistic transmission conditions. The principal findings are summarized as follows:

(1) RSA-based subcarrier index scrambling significantly increases signal randomness without altering the OFDM constellation, as indicated by the reduction in symbol standard deviation from 90.75 to 16.45.
(2) The integration of SHA-256 hashing enables effective integrity verification and results in a near-uniform symbol distribution, with a measured standard deviation of 17.26.
(3) Experimental BER results confirm that the proposed security scheme maintains Shannon-consistent performance and does not introduce additional degradation compared with conventional OFDM transmission.
(4) Security analysis indicates that employing 2048-bit RSA keys effectively mitigates vulnerabilities associated with Wiener's attack and common modulus attacks.
(5) The overall scheme achieves a practical balance between enhanced security and real-time operability, making it suitable for implementation in modern MIMO-OFDM wireless communication systems.

Acknowledgments

This research was supported by the Ministry of Higher Education, Science, and Technology (Kemdiktisaintek) through the Directorate General of Research and Development (Ditjen Risbang). The research was funded under contract 019/C3/DT.05.00/PL/2025.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] H. A. Hussein Al-Delfi and F. S. Hasan, "Hybrid Ciphering and Frequency Domain Scrambling for Secure Speech Transmission through MIMO-OFDM System," International Journal of Intelligent Engineering and Systems, vol. 15, no. 6, pp. 563-575, 2022.

[2] Z. Guo, B. Liu, J. Ren, Y. Mao, X. Wu, S. Chen, et al., "High Security and Low Complexity SCMA-OFDM Transmission System Based on Dual Three-Dimensional Memory Hyperchaotic in Seven-Core Optical Fibers," Journal of Lightwave Technology, vol. 42, no. 16, pp. 5458-5465, 2024.

[3] M. M. Banat, J. Bas, and A. A. Dowhuszko, "Improved Physical-Layer Security for OFDM Using Data-Based Subcarrier Scrambling," IEEE Globecom Workshops, pp. 1-6, 2021.

[4] M. M. Hasan, M. Cheffena, and S. Petrovic, "Physical-Layer Security Improvement in MIMO OFDM Systems Using Multilevel Chaotic Encryption," IEEE Access, vol. 11, pp. 64468-64475, 2023.

[5] A. Mukherjee, S. A. A. Fakoorian, J. Huang, and A. L. Swindlehurst, "Principles of Physical Layer Security in Multiuser Wireless Networks: A Survey," IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1550-1573, 2014.

[6] S. Golstein, T. H. Nguyen, F. Horlin, P. D. Doncker, and J. Sarrazin, "Physical Layer Security in Frequency-Domain Time-Reversal SISO OFDM Communication," International Conference on Computing, Networking and Communications, pp. 222-227, 2020.

[7] M. L. F. Abbade, W. S. Souza, M. O. Santos, I. E. L. Rodrigues, I. Aldaya, L. H. Bonani, et al., "Discrete Spectral Encryption of Single-Carrier Signals with Pseudo Random Dynamic Keys," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 4914-4929, 2024.

[8] R. P. Hudhajanto, I. G. P. Astawa, and A. Sudarsono, "Covert Communication in MIMO-OFDM System Using Pseudo Random Location of Fake Subcarriers," EMITTER International Journal of Engineering Technology, vol. 4, no. 1, pp. 150-163, 2016.

[9] Z. Weng, J. Ren, B. Liu, Y. Mao, X. Wu, X. Song, et al., "Physical Layer Security Scheme for Key Concealment and Distribution Based on Carrier Scrambling," Optics Express, vol. 32, no. 9,pp. 15053-15064, 2024.

[10] M. Li, G. Zhang, G. Li, H. Li, and X. Zhang, "Secure Transmission Algorithm Based on Subcarrier Sorting and XOR Operation in OFDM Systems," IEEE International Conference on Communication Systems, pp. 147-151, 2018.

[11] D. Ramakrishna and M. A. Shaik, "A Comprehensive Analysis of Cryptographic Algorithms: Evaluating Security, Efficiency, and Future Challenges," IEEE Access, vol. 13, pp. 11576-11593, 2025.

[12] X. Liang, C. Zhang, Y. Luo, X. Wang, and K. Qiu, "Secure Encryption and Key Management for OFDM-PON Based on Chaotic Hilbert Motion," Journal of Lightwave Technology, vol. 41, no. 6, pp. 1619-1625, 2023.

[13] H. Mohammed and D. Saha, "Encrypted-OFDM: A Secured Wireless Waveform," Computer Networks, vol. 255, article no. 110871, 2024.

[14] A. A. E. Hajomer, X. Yang, and W. Hu, "Secure OFDM Transmission Precoded by Chaotic Discrete Hartley Transform," IEEE Photonics Journal, vol. 10, no. 2, pp. 1-9, 2018.

[15] J. Ayad, A. Sadiq, B. Makki, Z. A. Abbood, H. J. Abdulkareem, and Z. K. Hussein, "Secure Images Transmission through OFDM System," 2nd International Conference on Computing and Machine Intelligence, pp. 1-4, 2022.

[16] A. Dash, A. Sarkar, A. Chatterjee, S. Darshana, M. Pandey, and R. K. Barik, "Multi-Factor Analysis of RSA Based on Variations in Primes Used for Modulus Generation," 3rd International Conference on Innovative Sustainable Computational Technologies, pp. 1-6, 2023.

[17] Z. Wang, "Secure Image Transmission in Wireless OFDM Systems Using Secure Block Compression-Encryption and Symbol Scrambling," IEEE Access, vol. 7, pp. 126985-126997, 2019.

[18] N. Kaur, A. Mittal, U. K. Lilhore, S. Simaiya, S. Dalal, K. Saleem, et al., "Securing Fog Computing in Healthcare with a Zero-Trust Approach and Blockchain," EURASIP Journal on Wireless Communications and Networking, vol. 2025, no. 1, article no. 5, 2025.

[19] N. Karnik, A. Kumar, P. Mahajan, A. Srivastav, S. Das, and B. Singh, "An Efficient Technique for Securing a Multi-Cloud Storage Environment," International Journal of System Assurance Engineering and Management, in press. https://doi.org/10.1007/s13198-025-02751-2

[20] Z. M. Yahya and M. F. Al-Gailani, "Secure Data Delivery in a Software-Defined Wireless Body Area Network," Journal of Telecommunications and Information Technology, no. 2, pp. 41-47, 2024.

[21] R. Acheampong, D. M. Popovici, T. Balan, A. Rekeraho, and M. S. Ramos, "Enhancing Security and Authenticity in Immersive Environments," Information, vol. 16, no. 3, article no. 191, 2025.

[22] D. M. Acasamoso, A. M. Sison, and R. P. Medina, "Enhancing Data Security with RSA-SHA 256: Mitigating Timing Attacks," 15th International Conference on Information and Communication Technology Convergence, pp. 574-578, 2024.

[23] M. M. Ahmed and S. H. Alnajjar, "Cipher Text to Secure Li-Fi System Using Hybrid Encryption Algorithm," Iraqi Journal for Computer Science and Mathematics, vol. 6, no. 2, article no. 16, 2025.

[24] K. Baskar, K. Muthumanickam, P. Vijayalakshmi, and S. Kumarganesh, "A Strong Password Manager Using Multiple Encryption Techniques," Journal of The Institution of Engineers (India): Series B, vol. 106, no. 4, pp. 1207-1214, 2025.

[25] U. Gulen and S. Baktir, "Side-Channel Resistant 2048-Bit RSA Implementation for Wireless Sensor Networks and Internet of Things," IEEE Access, vol. 11, pp. 39531-39543, 2023.

[26] H. Dymova, "Study of Cryptographic Security of Computer Networks," Computer-Integrated Technologies: Education, Science, Production, no. 57, pp. 15-19, 2025. (In Ukrainian)

[27] C. Gilbert and M. Gilbert, "Exploring Secure Hashing Algorithms for Data Integrity Verification," SSRN Electronic Journal, in press. https://doi.org/10.2139/ssrn.5251606

[28] M. Zheng, "Revisiting Small Private Key Attacks on Common Prime RSA," IEEE Access, vol. 12, pp. 5203-5211, 2024.