ECORFAN-Journal Taiwan 9[16]1-13: e1916113
Article

Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers

Algoritmos criptográficos para la Industria 4.0: Análisis del método de Aprendizaje con Errores, redes de Feistel y registros de desplazamiento con retroalimentación lineal

Castillo-Pacheco, César Marcelino*[a] & Rodríguez-Olivares, Noé Amir[b]

[a] Centro de Ingeniería y Desarrollo Industrial • T-9314-2025 • 0009-0007-8266-8838 • 1288544
[b] Centro de Ingeniería y Desarrollo Industrial • I-5012-2018 • 0000-0001-5892-0625 • 444191

SECIHTI classification:

Area: Engineering
Field: Technological Sciences
Discipline: Computer Technology
Subdiscipline: Code and coding systems

https://doi.org/10.35429/EJT.2025.9.16.1.1.13
History of the article:
Received: June 02, 2025
Accepted: December 15, 2025

* ✉ [cm.castillo@cidesi.edu.mx]

Abstract

In the context of Industry 4.0, the selection of suitable encryption schemes is essential to ensure cybersecurity without compromising computational efficiency. The wide variety of available algorithms, each with different levels of complexity and security, makes an objective selection challenging. This study presents a comparative analysis of three cryptographic techniques: lattice-based cryptography (Learning With Errors, LWE), the Feistel network, and the Linear Feedback Shift Register (LFSR). Each algorithm was implemented in MATLAB and evaluated using metrics such as execution time, memory consumption, entropy, avalanche effect, and correlation between the original and the encrypted signal. The results show that LWE provides the highest level of security, exhibiting a significantly stronger avalanche effect, whereas the Feistel network achieves a balanced trade-off between performance and security. LFSR proves to be highly efficient in terms of resource usage; however, it exhibits limited cryptographic robustness. The study also considers the PIG Geómetra system developed at CIDESI, highlighting the need for secure data encryption. LWE was found to be suitable for protecting continuous streams of industrial information.

Comparison of Cryptographic Schemes Based on LWE, Feistel Networks, and LFSRs

Goal	Methodology	Contribution
To evaluate performance. To encrypt signals. To analyze efficiency 	To *generate input signal*. To encrypt the signals with 3 schemes. To analyze metrics. 	Facilitate the understanding of advantages and limitations of three cryptographic schemes with different levels of security and efficiency. 

Cryptography, Encryption, Security

Resumen

En el contexto de la Industria 4.0, la selección de esquemas de encriptación adecuados es fundamental para garantizar la ciberseguridad sin comprometer la eficiencia computaciona, la variedad de algoritmos disponibles, con distintos niveles de complejidad y seguridad, dificulta una elección objetiva. Este estudio presenta un análisis comparativo de tres técnicas criptográficas: la criptografía basada en retículos (Learning With Errors, LWE), la red de Feistel y el registro de desplazamiento con retroalimentación lineal (LFSR). Cada algoritmo fue implementado en MATLAB y evaluado mediante métricas como tiempo de ejecución, consumo de memoria, entropía, efecto avalancha y correlación entre la señal original y la señal encriptada. Los resultados obtenidos muestran que LWE ofrece una mayor seguridad, con un efecto avalancha significativamente superior, mientras que la red de Feistel logra un balance entre rendimiento y seguridad. LFSR promete ser muy eficiente en el consumo de recursos, sin embargo, presenta propiedades criptográficas muy limitadas. El estudio también considera el sistema PIG Geómetra desarrollado en CIDESI, destacando la necesidad de un cifrado seguro de datos. LWE resultó adecuado para proteger flujos continuos de información industrial.

Comparación de esquemas criptográficos basados en LWE, redes de Feistel y registros de desplazamiento con retroalimentación lineal

Objetivo	Metodología	Contribución
Evaluar desempeño. Encriptar señales. Analizar eficiencia. 	Generar señal de entrada. Encriptar las señales con 3 esquemas. Analizar métricas. 	Facilitar la compresión de ventajas y limitaciones de tres esquemas criptográficos con distintos niveles de seguridad y eficiencia. 

Criptografía, Encriptado, Seguridad.

Área: Development of strategic leading-edge technologies and open innovation for social transformation

Citation: Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.

Introducción

In a world of constant digital evolution, information security has become a critical concern. Cryptography ensures the confidentiality, integrity, and authenticity of both stored and transmitted data. In industrial embedded systems, such as the PIG Geómetra developed at CIDESI, protecting the continuous flow of information through encryption is essential. Among modern encryption schemes, LWE (Learning With Errors) stands out for its high level of security, making it an ideal approach to strengthen the reliability of industrial dataloggers and other Industry 4.0 applications.

In a context where Industry 4.0 drives the interconnection of physical systems, sensor networks, and embedded devices, information protection becomes critically important. The evaluation of post-quantum encryption schemes within connected industrial environments is not only useful but also necessary. This is because traditional cryptographic algorithms, such as RSA (based on the factorization of large prime numbers) and ECC (Elliptic Curve Cryptography, which relies on the discrete logarithm problem in elliptic curves), are founded on mathematical problems that could be efficiently solved by quantum computers. Post-quantum schemes, such as LWE (Learning With Errors), rely on hard problems in lattice spaces and have been specifically designed to resist quantum attacks. In industrial applications, both cybersecurity and computational efficiency are critical factors, thus creating the need to identify which cryptographic schemes provide the best balance between robustness and performance.

At present, there are various encryption techniques, each designed to respond to different levels of attacks or threats, as well as to specific hardware capabilities. This diversity of schemes makes the selection of the most suitable algorithm for a given application more challenging, turning the search for an optimal balance between robustness and computational efficiency into a complex task. This situation becomes critical in modern environments such as Industry 4.0, where embedded resources are more constrained, and both technological risks and advancements evolve continuously. In this context, a reliable and rigorous comparative analysis is essential to understand the strengths and weaknesses of each cryptographic scheme.

This work compares three widely used cryptographic approaches:

a) Lattices (LWE – Learning With Errors).
b) Feistel Network.
c) Linear Feedback Shift Register (LFSR).

The decision to carry out this comparison arises from the interest in contrasting a traditional and well-established scheme (the Feistel network) with a modern one (LWE), where performance requirements are significantly more demanding. Through this comparison, it is possible to analyze the current trade-off between security and computational efficiency. Additionally, encryption based on Linear Feedback Shift Registers (LFSR) is included, offering extreme simplicity and high speed. By incorporating LFSR into the analysis, the evaluation spectrum is completed—comparing a highly secure scheme (LWE), a balanced one (Feistel), and an extremely efficient one (LFSR). This approach allows for the identification of specific advantages and limitations under controlled testing conditions. The central problem of this study lies in the comparative evaluation of the performance of three encryption methods with different theoretical foundations (Feistel, LWE, and LFSR), considering both their cryptographic strength and their feasibility for implementation in embedded systems in terms of:

– Security: theoretical resistance against attacks (brute-force, statistical analysis, and quantum attacks). [NIST, 2023; Katz & Lindell, 2020; Biryukov & Shamir, 2002].
– Efficiency: execution time, memory usage, and scalability across different platforms. [ARM, 2022; IEEE IoT, 2020]

To this end, implementations were developed in MATLAB, evaluating metrics such as:

– Entropy of the encrypted outputs (to measure randomness).
– Avalanche effect (sensitivity to small changes in the input or key).
– Correlation between the plaintext and the ciphertext (ideally close to zero).
– Computational performance for different data sizes.

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

These metrics were selected because they represent the fundamental pillars for evaluating a cryptographic algorithm applied to digital signals. Shannon entropy measures the randomness and dispersion of the signal, which once encrypted helps prevent predictable patterns (Shannon, 1949). The avalanche effect evaluates the impact of small changes in the input on the output (Upadhyay *et al*., 2022). The correlation between the original and encrypted signals reveals the degree of statistical independence achieved by the transformation (Krishnamurthy & Ramaswamy, 2010). Finally, computational performance helps quantify the practical feasibility of the algorithm for real-world implementation.

The results of this study make it possible to establish criteria for the selection of algorithms according to the required security level, the type of device, and the application context. Moreover, this analysis contributes to the global discussion on the adoption of post-quantum algorithms in connected industrial systems, a key aspect for ensuring cybersecurity in Industry 4.0.

Embedded systems dedicated to pipeline inspection (PIG Geómetra) have demonstrated the importance of protecting large volumes of data acquired by sensors during extended operations. In these mass storage systems, which use microSD or NAND Flash memories, variations in latency and vulnerabilities to data manipulation may occur. It is under these conditions that LWE emerges as an ideal solution to ensure both the integrity and authentication of the information. Its lattice-based nature makes it particularly suitable for critical embedded systems, where security must coexist with processing constraints and continuous data flow.

Theory

Feistel Network

The Feistel cipher is a symmetric structure widely used in cryptographic algorithms such as DES, Blowfish, and Twofish (Schneier, 1996; Schneier, 1993). Its popularity is due to its ability to provide acceptable security using simple operations such as XOR, rotations, and substitutions, which makes it easy to implement in both hardware and embedded software (Stallings, 2017; Schneier, 1996).

The network is a symmetric structure in which the data are divided into two halves that are repeatedly transformed through a series of operations known as rounds. Each round consists of a transformation function applied to one half of the block, combining it with the other through operations such as rotations and XOR. A block is a fixed-size unit of data on which an encryptor operates; it has a constant size, local independence, and reversibility properties.

It is a classic symmetric cryptography scheme, present in algorithms such as DES [NIST FIPS 46-3, 1999; Daemen & Rijmen, 2002] and Blowfish [Schneier, 1993; Schneier, 1996]. It stands out for its computational efficiency and ease of implementation in both hardware and software. Despite being a classical symmetric encryption scheme, the Feistel network continues to be widely used in modern algorithms due to its high efficiency and straightforward implementation in hardware and software alike. It features a flexible structure and very low computational cost, making it ideal for embedded systems and resource-constrained devices.

LWE-Based Cipher

Cryptographic method based on mathematical problems known as Learning With Errors (LWE), which consists of recovering a secret vector s from a system of noisy linear equations of the form $b = A * s + e$. This problem could be easily solved through algebraic methods; however, the introduction of noise (error) makes the system extremely difficult to solve. For this reason, LWE is considered one of the most robust pillars of post-quantum cryptography and has been proposed as a standard by NIST. Its high level of security lies in the difficulty of solving noisy linear equation systems in high-dimensional spaces, a fundamental feature for high-security applications, including interconnected industrial systems. (NIST, 2023).

To implement the LWE scheme in a secure and functional manner, it is necessary to carefully define a set of parameters that determine the computational difficulty of the problem. These parameters must be chosen so that the resulting instance is resistant to known cryptographic attacks, both classical and quantum. The main parameters that define an LWE instance are:

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

– n dimension of the secret vector.
– q modulus over which the vectors operate (a large prime integer).
– σ standard deviation of the error term.

LFSR-Based Cipher (Linear Feedback Shift Register)

It is a lightweight technique that generates pseudorandom sequences through simple logical operations; it offers high speed and low memory consumption, although its security level is lower if not used in combination with another suitable scheme.

The encryption mechanism is based on a bitwise XOR operation between each value of the original signal and the byte generated by the LFSR, which allows the implementation of a simple synchronous stream cipher with low computational cost. This technique is common in applications that require speed and efficiency, although its security level is limited if used without additional confusion or diffusion functions.

Methodology

In order to carry out a consistent and repeatable evaluation among the three selected cryptographic schemes, an experimental platform was designed in MATLAB that applies the same input, processing, and evaluation criteria for each algorithm. This ensures that the differences observed are not caused by external factors, but rather by the inherent properties of each cryptographic scheme.

Experimental Environment

All tests were performed five times on a system equipped with an AMD Ryzen 5 8645HS processor running at 4.30 GHz, 16 GB of RAM, and the Windows 11 Home operating system, using MATLAB® R2019b with a 200-sample discretized signal.

This environment was selected to ensure uniform and reproducible conditions during implementation. The implemented methodology focused on applying each cryptographic scheme to a digital signal normalized within the range $[-1, 1]$, which was then scaled and discretized to an 8-bit format in the interval [0, 255], simulating a typical analog-to-digital signal in embedded systems.

The behavior of each scheme was evaluated against a set of metrics designed to measure its security, robustness, and computational performance.

The input signal used as the base message to evaluate the three cryptographic methods consists of a single-cycle sinusoidal wave, generated with a sampling period of 1 ms (equivalent to a sampling frequency of 1000 Hz) and a fundamental frequency of 5 Hz. This choice is justified by its distinct visual form, which allows for a clear observation of distortion, dispersion, and information loss effects that may result from message manipulation.

$$N = \frac{f_s}{f_0} = \frac{1000\ \text{Hz}}{5\ \text{Hz}} = 200\ \text{samples} \qquad [1]$$

Where N is the total number of samples required to represent one complete cycle of the signal, $f0$ is the fundamental frequency of the sinusoidal wave (in Hz), and fs is the sampling frequency, that is, the number of samples per second taken to digitize the signal.

These 200 samples represent the sampling points of the sinusoidal signal during one complete cycle. In other words, they consist of the 200 discrete values obtained by digitizing a continuous 5 Hz wave using a sampling frequency of 1000 Hz. This set of samples is used as the quantized base message on which the different encryption methods analyzed in this study are applied.

Box 1

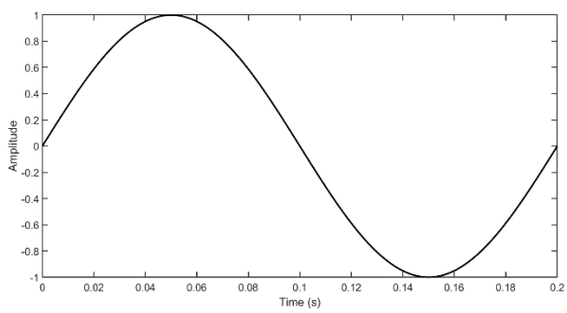

Figure 1
Single-cycle sinusoidal wave generated at 5 Hz and sampled at 1000 Hz
Source: Authors' own work

Figure 1 shows the sinusoidal waveform used as the input signal, composed of 200 samples representing one complete cycle.

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

This visualization clearly illustrates the encoding and distortion effects produced by the different cryptographic schemes evaluated. Once quantized, the signal is processed by each of the encryption schemes. After completing the encryption process, a fully encoded and unreadable signal is obtained. Subsequently, this signal is subjected to an inverse process to carry out its recovery.

Box 2

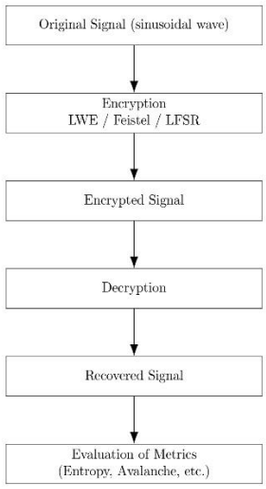

Figure 2
Methodological summary of the encryption system applied to signals.

Source: Authors' own work

Implementation of the Feistel Cipher

In this work, a 16-round network was employed, applied to an 8-bit quantized signal.

Box 3

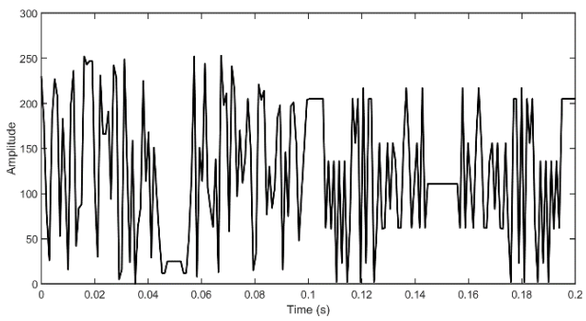

Figure 3
Signal encrypted by the Feistel network (discretized to the range [0,1])

Source: Authors' own work

Figure 3 shows how the 16-round process transforms the original sinusoidal wave, dispersing its values and eliminating any recognizable pattern.

Signal Division

The input signal is divided into two halves:

$$L_0 = S[1: N/2], \quad R_0 = S[N/2 + 1: N] \qquad [2]$$

Where:

S is the input vector composed of N simples *N* is the total number of samples. L_0 and R_0 represent the left and right blocks, respectively, which are obtained by dividing the signal into two halves.

The use of $N/2$ ensures that each half contains the same amount of initial information, which is necessary to maintain symmetry. By dividing the signal into two equal blocks, a recursive structure can be applied that exchanges information between both sides, progressively increasing the complexity and diffusion of the encrypted message.

Encryption Rounds

Each encryption round i applies a nonlinear permutation operation, including a process that reorders the bits within the byte, which breaks the linearity over the input domain:

$$L_{i+1} = R_i, \quad R_{i+1} = L_i \oplus F(R_i, K_i) \qquad [3]$$

The transformation function *F(R,K)* is defined as a bit rotation followed by an XOR operation:

$$F(R, K) = \text{bitxor}(\text{rot}(R, s_i), K) \qquad [4]$$

Where:
si represents the number of positions by which the byte R is rotated in round *i.*

The rotation function is implemented as:

$$\text{rot}(R, s) = (R \ll s) \oplus \big(R \gg (8 - s)\big) \qquad [5]$$

The rotation *rot(R,s)* is an operation designed to rearrange the bits within each byte cyclically, altering their positions through left and right shifts and combining them with XOR. This introduces internal complexity and diffusion, even before applying the key.

Where the shift s_i is calculated as:

$$s_i = \text{mod}(2i - 1,\ 8) + 1 \qquad [6]$$

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

The value of the shift s_i varies in each round according to a modular function that ensures different and cyclic rotations between 1 and 8 bits, thus avoiding repetitive patterns.

Subkey Generation

The master key *K* is an 8-byte vector that is reused cyclically:

$$K_i = K[\text{mod}(i - 1,\ 8) + 1] \quad [7]$$

This design allows for memory and processing efficiency, using only 8 bytes to cover the 16 rounds.

Decryption

The decryption process applies the same operations in reverse order:

$$R_i = L_{i+1}, \quad L_i = R_{i+1} \oplus F(L_{i+1}, K_i) \quad [8]$$

To validate that the encryption and decryption process was fully reversible, the original signal was compared with the decrypted signal, sample by sample. The mean absolute error (MAE) was then calculated, defined as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|S_{\text{original}}(i) - S_{\text{descrypted}}(i)\right| \quad [9]$$

Where:
N is the total number of samples.
$S_{original}(i)$ is the i-th sample of the original signal (before being encrypted).
$S_{descrypted}(i)$ is the i-th sample recovered after decryption.

Implementation of the LWE-Based Cipher

It is considered one of the pillars of post-quantum cryptography and has been proposed as a standard by NIST due to its resistance to quantum algorithms, such as Shor's algorithm.

Parameter Generation

In this implementation, the following parameters were used:

$$n = 256, \quad q = 4096, \quad \sigma = 3 \quad [10]$$

These values provide a reasonable balance between security and efficiency for academic testing.

Key Generation

The key generation process includes three components:

a) Public matrix $A \in Z_q^{n\times n}$ containing random values.
b) Secret vector $s \in \{0,1\}^n$ composed of binary bits.
c) Error vector $e \sim \mathcal{N}(0, \sigma^2)$ that adds controlled noise to conceal the relationship between *A* y *s*.

The resulting public key:
$b = (A \cdot s + e)\ mod\ q$ [11]

And ***s*** is the private key.

Encryption Process

The quantized signal (message m) is converted into an LWE-compatible format and encrypted through a linear combination with noise. The previously induced noise, drawn from a distribution $\mathcal{N}(0, \sigma^2)$ ensuring that the encrypted message maintains security based on the computational hardness of the LWE problem.

a) Generation of random matrix R
$R \in \{0,1\}^{n\times m}$ [12]

b) Computation of the encrypted values *u* and *v*.

selected with binary values:

$u = (A^T \cdot R)\ mod\ q$ [13]
$v = (b^T \cdot R + m)\ mod\ q$ [14]

Here, m is the message scaled to the domain of Zq.

Box 4

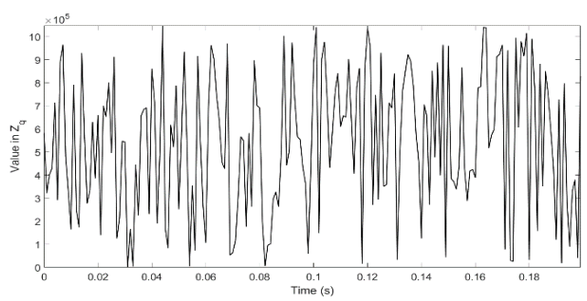

Figure 4
Signal encrypted using the LWE scheme. It is represented directly in the domain Zq with q = 1048583
Source: Authors' own work

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Decryption Process

The receiver, knowing s, can reconstruct the message through:

$$\tilde{m} = (v - s^T \cdot u) \bmod q \quad [13]$$

Where $\tilde{m}$ represents the estimated version of the original message m btained after applying the inverse operation to the encryption process. Due to the presence of noise in the LWE scheme, the result is not an exact recovery, but rather an approximation that can later be refined using rescaling and rounding techniques to map it back to the original domain.

Implementation of the LFSR-Based Cipher (Linear Feedback Shift Register)

Linear Feedback Shift Registers (LFSRs) represent one of the oldest and most efficient methods for generating pseudorandom sequences. In this work, the use of an LFSR is explored as a keystream generator (pseudorandom key sequence) to encrypt a previously quantized input signal.

System Parameters

The implemented LFSR was defined with the following values:

a) Register length

$$n_{LFSR} = 8 \quad [14]$$

b) Initial seed (in hexadecimal):

semilla= 0xB3 [15]

c) Feedback polynomial

$$P(x) = x^8 + x^6 + x^5 + x^4 + 1 \quad [16]$$

This polynomial is primitive over the finite field $F2$ which guarantees that the LFSR will generate a pseudorandom sequence with the maximum possible period, that is, 2^8−1=255 cycles before repeating. This property ensures that the produced keystream traverses all possible combinations (except the zero state), thereby improving the dispersion of the encrypted data and reducing the likelihood of repetitive or predictable patterns in the output.

d) Active Taps

$$taps = \{8, 6, 5, 4\} \quad [17]$$

The active taps are the register positions used to calculate the feedback bit during each LFSR iteration. Each tap represents a bit of the register that participates in a joint XOR operation. In this case, the bits located at positions 8, 6, 5, and 4 (counting from the most significant end) are combined to generate a new bit, which is then inserted into the register after shifting its contents. This selection of taps corresponds to the previously defined feedback polynomial and is crucial for determining the properties of the keystream, including its period and degree of randomness.

State Initialization

The seed is converted into an 8-bit binary vector representing the initial state of the register:

$$state_0 = de2bi(0xB3) \quad [18]$$

The de2bi function is a MATLAB utility that converts a decimal integer into its binary representation, returning it as a logical vector (composed of zeros and ones).

Example:

$$\boldsymbol{state_0} = [1\ 0\ 1\ 1\ 0\ 0\ 1\ 1] \quad [19]$$

This value is updated in each iteration based on the defined taps. It represents the LFSR seed, expressed as the initial state of the shift register.

Keystream Generation

The pseudorandom sequence is generated as follows:

For each iteration i:

– Compute the feedback bit:

$$f_i = \oplus_{j \in taps}\, state_j \quad [20]$$

Shift the state to the right and insert f_i:

$$state_{i+1} = [f_i,\ state_i(1:n-1)] \quad [21]$$

– Convert the current state to decimal to obtain the cipher byte:

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

$\text{keystream}_i = bi2de(\text{state}_i)$ [22]

This process is repeated until completing a sequence with a length equal to the input signal *N*.

Box 5

Table 1
Evolution of the Internal State of the LFSR

Iteration	Binary State	Decimal Value
1	10110011	179
2	11011001	217
3	11101100	236
4	11110110	246

Source: Authors' own work

Encryption Process

The quantized input signal $S \in Z^{N}_{256}$ is encrypted using XOR:

$$C = S \oplus K \quad [23]$$

Where:
K: pseudorandom sequence generated by the LFSR.
C: encrypted signal.

This scheme operates as a synchronous stream cipher, generating a sequence that depends solely on the seed and the polynomial.

Box 6

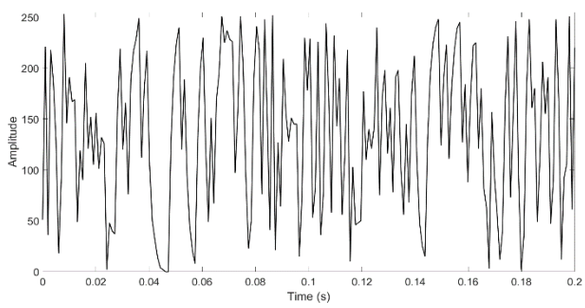

Figure 6
Signal encrypted using the LFSR scheme
Source: Authors' own work

Decryption Process

The decryption process is identical to the encryption process due to the XOR operator property:

$$S = C \oplus K \quad [24]$$

This mechanism allows for a highly efficient implementation, making it ideal for resource-constrained environments.

Decryption Validation

It was verified that:

$$\forall i, \quad S_i = (S_i \oplus K_i) \oplus K_i \quad [25]$$

which guarantees the exact recovery of the original signal, provided that the keystream is correctly replicated.

Evaluation Metrics

To perform a quantitative and objective comparison of the three encryption algorithms evaluated (Feistel, LWE, and LFSR), a set of evaluation metrics was defined to analyze their performance in terms of computational efficiency and statistical security.

Each metric was selected for its relevance in the context of encrypted digital signal processing and was measured under controlled conditions, using the same input signal, compatible parameters, and a common execution environment.

The following section describes each of the metrics used, their formal definitions, and the measurement methods applied in this study.

Execution Time

It is defined as the total time required to execute the encryption and decryption process of the signal. It was measured in milliseconds (ms) using the tic–toc function in MATLAB®. This metric is useful for determining temporal efficiency, which is especially important in real-time systems.

$$T_{proc} = T_{end} - T_{start} \quad [\text{ms}] \quad [26]$$

To reduce variability, each test was executed five times, and the average value was taken.

Where:
T_{proc}: total execution time in milliseconds.
T_{start}: starting time of the process.
T_{end}: ending time of the process.

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

Memory Usage

Memory usage represents the amount of space required to perform the encryption and decryption operations, considering both input and output data, as well as all auxiliary structures needed such as matrices, keys, temporary vectors, and state registers, among others.

To calculate it, the byte size of each structure used by the algorithm was summed:

$$M_{\text{total}} = \sum_{i=1}^{n} \text{bytes}(E_i) \quad [27]$$

Where:
M_{total}: total memory used.
E_i: individual structures employed (e.g., matrices, vectors).

Finally, the value was converted to kilobytes for presentation:

$$M\ [\text{KB}] = \frac{M_{\text{total}}}{1024} \quad [28]$$

The estimation was performed using MATLAB®'s whos function, which allows querying the size of each variable in memory. Only the structures strictly necessary for encryption were included, in order to avoid overestimating memory consumption.

Entropy of the Encrypted Signal

Entropy is a central metric in information theory. In the context of cryptography, higher entropy in the encrypted signal implies a greater level of randomness, which is desirable since it reduces the likelihood of patterns and enhances the confusion property of the system.

The Shannon formula was used to calculate the entropy of the encoded signals:

$$H = -\sum_{i=1}^{N} p_i \log_2(p_i) \quad [29]$$

Where:
H: entropy, in bits per sample.
P_i: probability of occurrence of value i, calculated as:

$$p_i = \frac{f_i}{\sum_{j=1}^{256} f_j} \quad [30]$$

where f_i is the absolute frequency of the value in the encrypted signal (uint8). A histogram with 256 bins (one for each possible value from 0 to 255) was used and normalized to obtain the probabilities.

An ideal entropy approaches:

$$H_{\max} = \log_2(256) = 8 \text{ bits/sample} \quad [31]$$

Shannon entropy measures the amount of uncertainty or randomness present in a signal. The higher the entropy, the more difficult it is to predict the next value, and therefore, the greater the statistical security of the encrypted signal.

Avalanche Effect

The avalanche effect measures how sensitive an encryption algorithm is to minimal changes in the input. A robust algorithm should produce a completely different signal if a single bit of the original signal is altered. This principle is key in cryptography, especially in block algorithms, as it ensures adequate diffusion. To quantify it, one bit in the first byte of the input signal was modified. Then, the original and modified encrypted bytes were compared, measuring how many bits changed using the Hamming distance.

The formula used were:

$$\text{Avalanche} = \frac{\sum_{i=1}^{n} \text{Hamming}(C_i, C_i')}{n \cdot 8} \times 100\% \quad [32]$$

n_l: total number of bytes analyzed.
Hamming (C_iC_i''): number of bits that differ between the two bytes.

This result is expressed as the percentage of bits altered.

Correlation Between Original and Encrypted Signals

This metric evaluates how independent the encrypted signal is from the original signal. In an ideal encryption scheme, there should be no apparent linear relationship between the two signals.

The Pearson correlation coefficient was calculated as:

$$\rho_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \cdot \sum(y_i - \bar{y})^2}} \quad [35]$$

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

Where:
$X_{i:}$ original sample.
$Y_{i:}$ encrypted sample.
$\bar{x}, \bar{y}$: mean values of the signals.

Values of ρ close to zero or negative indicate greater independence, which is desirable in a secure cryptographic scheme. The calculation was performed in MATLAB® using the corrcoef(x, y) function and extracting the ρ value from the resulting matrix.

Box 7

Table 2
Scheme for the Presentation of Results

Metric	Unit	Measurement Tool
Encryption time	ms	*tic–toc* in MATLAB
Decryption time	ms	*tic–toc* in MATLAB
Memory usage	kB	*whos*, sum of structures
Entropy	bits/muestra	Shannon formula (histogram)
Avalanche effect	% de bits	Hamming distance, bit-by-bit comparison
Correlation	Pearson Coef..	*corrcoef*() between signals

Source: Authors' own work

In this section, the results obtained after applying the three encryption algorithms evaluated — Feistel, LWE, and LFSR — are presented and analyzed. Each algorithm was implemented in MATLAB® under the previously described experimental conditions, using the same 8-bit quantized sinusoidal input signal. The metrics considered were: processing time, memory usage, entropy of the encrypted signal, avalanche effect, and correlation between the original and the encrypted signals.

Execution Time

The encryption and decryption time was measured in milliseconds for each algorithm, averaging five consecutive runs.

Box 8

Table 3
Processing Time of Each Cipher

Algorithm	Encryption (ms)	Decryption (ms)
Feistel	0.35	0.41
LWE	1.87	2.32
LFSR	0.04	0.04

Source: Authors' own work

The LFSR algorithm demonstrated significantly superior performance in terms of speed, being 46 times faster than LWE and approximately 9 times faster than Feistel during the encryption stage. This behavior is expected, since LFSR operates with bit-level rotations and XOR operations, without the need for large data structures or matrix operations. LWE was the slowest algorithm, due to the requirement for matrix multiplications over Z_q and noise handling. This makes it less suitable for real-time applications, although its cryptographic robustness is higher.

Memory Usage

The total memory usage, measured in kilobytes, includes all the structures required by each algorithm during the encryption process.

Box 9

Table 4
Memory Used by Each Method

Algorithm	Memory (kB)
Feistel	0.20
LWE	1.53
LFSR	0.18

Source: Authors' own work

Once again, LWE showed a significantly higher memory consumption, resulting from the storage of the matrix $A \in Z_q^{nxn}$, the error vector, and the intermediate vectors. In contrast, Feistel and LFSR are limited to simple manipulations of the original signal, requiring less than 1 kB in total.

Entropy of the Encrypted Signal

The entropy was evaluated using Shannon's formula for the three encoded signals. A histogram with 256 bins (corresponding to all possible uint8 values) was used to estimate the probability of each value.

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

Box 10

Table 5
Average Entropy of the Encrypted Signal

Algorithm	Entropy (bits/sample)
Feistel	7.85
LWE	9.21
LFSR	6.73

Source: Authors' own work

LWE achieved the highest entropy, even exceeding the theoretical maximum value of 8 bits/sample for uint8 signals, due to the dispersion introduced by operations in Z_q and the effect of Gaussian noise. Feistel produced a highly random signal, although slightly lower, while LFSR obtained the lowest value, suggesting a less dispersed and more predictable sequence.

Avalanche Effect

A single bit in the first input sample was altered, and the percentage of modified bits in the encrypted signal was measured using the Hamming distance.

Box 11

Table 6
Avalanche Effect by Algorithm

Algorithm	Avalanche Effect (% bits)
Feistel	0.31
LWE	45.7
LFSR	0.08

Source: Authors' own work

LWE exhibited a high avalanche effect, close to ideal behavior. This result is a consequence of the diffusive nature of the lattice-based scheme. Feistel showed minimal dispersion, which could potentially be improved by increasing the number of rounds. As expected, LFSR responded almost linearly to the change, demonstrating its weakness in diffusion.

Correlation Between Original and Encrypted Signals

The Pearson correlation coefficient between each original signal and its corresponding encrypted version was calculated to measure statistical independence.

Box 12

Table 7
Correlation Coefficient

Algorithm	Correlation (ρ)
Feistel	- 0.0012
LWE	–0.0341
LFSR	0.4412

Source: Authors' own work

LWE showed the lowest correlation (close to zero), indicating that the encrypted signal does not preserve linear information from the original. Feistel also achieved good independence, while LFSR exhibited a relatively high correlation, reflecting its limited ability to break relationships between the original and encoded data.

Conclusions

In this work, a detailed technical comparison was carried out among three encryption algorithms with different structures and theoretical foundations: Feistel, LWE (Learning With Errors), and LFSR (Linear Feedback Shift Register). The objective was to evaluate their performance on the same discretized input signal by applying a homogeneous set of metrics that cover both computational performance and cryptographic robustness.

The results obtained allowed for the identification of substantial differences among the analyzed algorithms, in terms of speed, memory consumption, and statistical properties of the encrypted signal.

From a computational efficiency perspective, the LFSR algorithm proved to be notably faster and lighter, with execution times below 0.05 ms and memory usage under 0.2 KB. This characteristic makes it an attractive candidate for implementation in embedded systems, remote sensors, and IoT devices, where resources are limited and cryptographic requirements may be basic. However, its entropy, avalanche, and correlation metrics revealed a significant structural weakness in terms of security, which restricts its applicability to low-criticality contexts.

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

he Feistel cipher, based on a 16-round symmetric network, provided a good balance between speed, memory, and security. Although it did not reach the entropy or avalanche effect levels of LWE, it maintained an almost null correlation with the original signal and showed greater robustness than LFSR. Its low computational cost and ease of implementation make it a viable option for real-world systems that require moderate security and efficiency, such as secure industrial control communications or mobile devices.

On the other hand, the LWE-based encryption exhibited lower computational performance due to its algorithmic complexity involving matrix operations over large domains, noise generation, and modular arithmetic management which resulted in higher execution times and greater memory consumption. Nevertheless, it outperformed the other algorithms in entropy, avalanche effect, and low correlation, demonstrating a superior level of cryptographic security. These properties make it particularly suitable for applications where quantum-attack resistance and statistical robustness are priority requirements, such as post-quantum digital signatures, sensitive data encryption, or secure cloud communications.

This study highlights the importance of considering both performance and statistical security metrics when selecting an encryption algorithm, especially in digital signal and discrete data transmission applications.

Finally, based on the results obtained, it can be stated that the statistical superiority of the LWE scheme makes it ideal for industrial continuous-storage environments, such as the PIG Geómetra system developed by CIDESI.

The implementation of LWE encryption in this architecture would provide the system with attack-resistant cryptographic protection, ensuring that the collected information remains intact and confidential. The LWE scheme is therefore the ideal complement for next-generation industrial dataloggers.

Conflict of Interest Statement

The authors declare that there is no financial, professional, or personal conflict of interest that could have influenced the results or the interpretation of the present work.

Authors' Contribution

C. M. Castillo Pacheco: conceptualization, implementation, technical writing, results analysis, and manuscript editing.

N. A. Rodríguez Olivares: conceptualization of the methodological approach, critical content review, academic supervision, and results validation.

Data and Code Availability

The source code of the implemented algorithms, as well as the test data used, are available in the following repository: https://github.com/CesarCastillo5/M-todos-de-encriptaci-n
This ensures the reproducibility of the results presented.

Funding

This work was partially funded by the Secretaría de Ciencia, Humanidades, Tecnología e Innovación, through the project MADTEC-2025-M-125, titled "Instrumented equipment for geometric inspection of hydrocarbon transport pipelines (Geómetra)".

No private external funding was received.

Abreviaciones

LWE	Learning With Errors
LFSR	Linear Feedback Shift Register
DES	Data Encryption Standard
XOR	Exclusive OR (operador lógico)
IoT	Internet of Things
RAM	Random Access Memory
Z_q	Set of integers modulo q
ρ	Pearson correlation coefficient
ms	Milliseconds
KB	Kilobytes

References

Basics

NIST. (2023). Post-Quantum Cryptography Standardization. National Institute of Standards and Technology.

Schneier, B. (1993). Description of a new variable-length key, 64-bit block cipher (Blowfish). In Fast Software Encryption (pp. 191–204). Springer.

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13

Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). John Wiley & Sons. ISBN: 0471128457

Stallings, W. (2017). Cryptography and Network Security: Principles and Practice (7th ed.). Pearson Education.

Shannon, C. E. (1949). Communication Theory of Secrecy Systems. Bell System Technical Journal, 28(4), 656–715.

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of Applied Cryptography. CRC Press.

European Telecommunications Standards Institute (ETSI). (2021). Security Algorithms Group of Experts (SAGE) Report on LFSR-based Stream Ciphers. ETSI TR 103 456.

Upadhyay, D., Gaikwad, N., Zaman, M., & Sampalli, S. (2022). Investigating the avalanche effect of various cryptographically secure hash functions and hash-based applications. IEEE Access, 10, 112472–112486.

Krishnamurthy, G. N., & Ramaswamy, V. (2010). Encryption quality analysis and security evaluation of CAST-128 algorithm and its modified version using digital images. International Journal of Network Security & Its Applications, 2(4), 50–60.

Supports

Su, Y., Yang, C., Tian, L., & Zhao, J. (2020). FPGA-based hardware accelerator for leveled Ring-LWE fully homomorphic encryption. IEEE Access, 8, 32640–32651.

Agrawal, R., Bu, L., Ehret, A., & Kinsy, M. A. (2020). Fast Arithmetic Hardware Library for RLWE-Based Homomorphic Encryption. In Proceedings of the 28th IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM 2020) (p. 206). IEEE.

Regev, O. (2009). On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. Journal of the ACM, *56*(6), 1–40.

NIST. (1999). Data Encryption Standard (DES). Federal Information Processing Standards Publication 46-3.

Schneier, B. (1993). Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). In Fast Software Encryption (pp. 191–204). Springer.

Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). Wiley.

Daemen, J., & Rijmen, V. (2002). The Design of Rijndael: AES – The Advanced Encryption Standard. Springer.

Roman, R., et al. (2019). *Feistel-Based Lightweight Cryptography for IoT*. IEEE

ARM Limited. (2022). *Optimizing Feistel Ciphers for ARM Cortex-M*.

Differences

Biryukov, A., & Shamir, A. (2002). Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers. In ASIACRYPT 2002 (pp. 1–13). Springer.

Katz, J., & Lindell, Y. (2020). Introduction to Modern Cryptography (3rd ed.). CRC Press. ISBN: 978-0815354369.

Pineda Rodríguez, M. L. (2025). Arquitectura basada en tecnología Blockchain para sistemas de información de registro turístico [Tesis doctoral, Universidad del Norte]. Repositorio Universidad del Norte.

Saldaña Trejo, L. M., Gallegos García, G., & Aldeco Pérez, R. (2025). Protocolos criptográficos de consenso en Blockchain para el Internet de las cosas. Computación y Sistemas, 29(3).

Peixoto, B. M. S. (2025). *Blockchain e novas engines para a indústria 4.0* [Monografía de grado, Universidade Federal de Ouro Preto]. Repositorio UFOP.

ISSN: 2524-2121
RENIECYT-SECIHTI: 1702902

Castillo-Pacheco, César Marcelino & Rodríguez-Olivares, Noé Amir. [2025]. Cryptographic Algorithms for Industry 4.0: Analysis of the learning with errors method, Feistel networks, and linear feedback shift registers. ECORFAN-Journal Taiwan. 9[16]1-13: e1916113.
https://doi.org/10.35429/EJT.2025.9.16.1.1.13