GENERACIÓN DE SERIES PSEUDORANDOM PARA CIFRAR DATOS DE CONSUMO DE ENERGÍA ELÉCTRICA

PSEUDORANDOM SERIES GENERATION TO ENCRYPT ELECTRICAL ENERGY CONSUMPTION DATA

Saúl Moreno López
Universidad Politécnica de Tulancingo, México
saul.moreno.lopez@outlook.com

Francisca Angélica Elizalde Canales
Universidad Politécnica de Tulancingo, México
francisca.elizalde@upt.edu.mx

Iván de Jesús Rivas Cambero
Universidad Politécnica de Tulancingo, México
ivan.rivas@upt.edu.mx

Juan Antonio Cárdenas Franco
Universidad Politécnica de Tulancingo, México
juan.cardenas@upt.edu.mx

Uriel Edgardo Escobar Franco
Universidad Politécnica de Tulancingo, México
uriel.escobar@upt.edu.mx

Recepción: 30/octubre/2020 **Aceptación:** 27/noviembre/2020

Resumen

En este artículo se presenta la implementación de un prototipo para el cifrado de datos usando un criptosistema de clave simétrica que cifra los datos mediante la aplicación de generadores eficientes de secuencias pseudoaleatorias que aprovechan el comportamiento caótico de una ecuación logística, es diseñado para generar flujo de claves aplicadas al cifrado/descifrado de una señal de consumo de energía eléctrica. El objetivo del cifrado es el de proporcionar privacidad y confidencialidad al usuario de sus datos de medición en el marco de redes eléctricas inteligentes. Se realizan pruebas experimentales utilizando una señal de energía eléctrica real, los resultados obtenidos demuestran que el proceso de cifrado/descifrado no afectará la eficiencia de codificación, manteniendo una tasa de bits y un bajo consumo de recursos computacionales. Para validar los resultados,

éstos se someten a un análisis de seguridad basado en valoración estadística del NIST (Instituto Nacional de Normas y Tecnología), pruebas que son superadas, lo que indica, que la información o los datos quedan criptográficamente protegidos. Se hace una comparación entre resultados simulados y Reales.
Palabras Clave: Criptografía, ecuación logística, pseudoaleatorio, seguridad.

Abstract

Implementation of a prototype for data encryption using a symmetric key cryptosystem is presented in this paper, data are encrypted by applying efficient generators of pseudo-random sequences that take advantage of the chaotic behavior of a logistic equation and it is designed to generate flow of keys applied to the encryption/decryption of an electrical energy consumption signal. The purpose of encryption is to provide privacy and confidentiality to the user of his measurement data within the framework of smart grids. Experimental tests are carried out using a real electrical energy signal, the results obtained show that the encryption / decryption process will not affect the encoding efficiency, maintaining a low bit rate and low consumption of computational resources. To validate the results, they are subjected to a security analysis based on statistical evaluation from the NIST (National Institute of Standards and Technology), tests that are passed, which indicates that the information or data is cryptographically protected. A comparison is made between simulated and real results.
Keywords: *Cryptography, logistic equation, Pseudo-random, security.*

1. Introducción

En las últimas décadas, la información digital ha sido ampliamente difundida a través de Internet y las redes inalámbricas debido a la rápida evolución de la industria multimedia y de comunicaciones. Con los avances tecnológicos actuales, el uso diario de los sistemas de información está aumentando, lo que los ha llevado a ser vulnerables a los ciberataques.

Por tanto, es fundamental utilizar mecanismos de seguridad para resguardar la información de este tipo de ataques. La herramienta más utilizada es la criptografía,

ya que este método se encarga de ocultar los datos a terceros y asegurar la confidencialidad a través del cifrado [Carlet, 2008].

Bajo la consideración de que, el caos es un comportamiento de un sistema dinámico que cambia de manera irregular en el tiempo; muchos métodos o esquemas de comunicación segura se han desarrollado para cifrar información basándose en sistemas discretos caóticos [Guan, 2017], [Mylrea, 2017]. Existe una relación cercana entre el caos y la criptografía porque los sistemas caóticos tienen características de ergodicidad, propiedades de mezcla, sensibilidad en parámetros de control y en las condiciones iniciales, que pueden considerarse análogos a las técnicas de difusión y confusión, integrados en muchos sistemas criptográficos [Jiménez, 2015], [Dazahra, 2018], [Bose, 2006].

Debido a las características específicas de los sistemas caóticos, los métodos de encriptación basados en caos parecen ser más eficientes para el uso práctico considerando su complejidad, velocidad y alta seguridad [Kong, 2015]. El cifrado es una de las medidas defensivas con que cuenta cualquier tecnología de la información que desee proteger una instancia. Los algoritmos de cifrado definen transformaciones de datos que los usuarios no autorizados no pueden revertir con facilidad. Se tiene a disposición varios algoritmos para elegir, incluidos DES, Triple DES, TRIPLE_DES_3KEY, RC2, RC4, RC4 de 128 bits, DESX, AES de 128 bits, AES de 192 bits y AES de 256 bits. Aunque ningún algoritmo único resulta idóneo para todas las situaciones [Zhen, 2018].

Los algoritmos criptográficos son la columna vertebral de la protección de datos altamente sensibles. La selección de un algoritmo criptográfico adecuado afectará dinámicamente la vida útil y el rendimiento de un dispositivo en términos de duración de la batería, memoria de hardware, latencia de cómputo y ancho de banda de comunicación. En desarrollos de entornos con recursos limitados, la tendencia está cambiando hacia diseño de algoritmos livianos [Mrabet, 2018].

Para abordar el problema de seguridad [Badra, 2017] presentan una distribución gradual donde el cifrado homomórfico se agrega a medidores inteligentes involucrados en la transferencia de datos desde una fuente a unidad colectora de tal manera que los resultados intermedios no se revelen a ningún dispositivo en la

ruta. También se trabaja en protocolos de preservación de la privacidad basados en encriptación homomórfica adicional [Borges, 2017] o enmascaramiento [Knirsch, 2018] en la infraestructura de medición que permite el cálculo de la suma de todos los valores de carga del hogar sin señalar valores individuales.

En [Tonyali, 2016] se propuso una ofuscación de datos para preservar la privacidad del consumidor y simultáneamente realizar la estimación del estado de distribución, donde la puerta de enlace de red de infraestructura de medición avanzada (AMI) calcula los vectores de ofuscación y multiplica el vector con un número aleatorio y lo distribuye a los medidores inteligentes mediante un uso compartido de llave. Por otro lado, [Rottondi, 2015], ha propuesto una infraestructura de privacidad amigable por medio de un algoritmo criptográfico que oculta el patrón de consumo de energía, basado en el intercambio secreto de Shamir esquema. [Tan, 2016] propuso un esquema de preservación de la privacidad basado en un seudónimo que tranquiliza la privacidad, integridad y autenticidad en AMI.

Dado que las tecnologías de la comunicación son susceptibles a ataques intencionados y debe proponerse la solución más adecuada en cada caso. Una forma de fortalecer la confidencialidad e integridad de la información es a través de cifrado, siendo la columna vertebral en la protección de datos altamente sensibles, pero la selección de un cripto-algoritmo afectará el rendimiento de un dispositivo en términos de memoria, latencia de cálculo y ancho de banda en la comunicación [Al-Haija, 2014]. Para atender la seguridad y fiabilidad de los datos resulta imprescindible analizar la seguridad de las redes de energía inteligentes; enfocándose al sistema de medición, describiendo los requisitos de seguridad y considerando las posibles amenazas y vulnerabilidades, con la finalidad de que se comprenda cómo los atacantes maliciosos pueden comprometer la seguridad, refiriendo las vulnerabilidades, así como los ataques más sofisticados y su impacto [Zapateiro, 2015]. Recientemente, se han introducido varios esfuerzos de investigación para superar los desafíos y encontrar soluciones apropiadas asociadas con la seguridad, especialmente seguridad de extremo a extremo [Mrabet, 2018], [Benslimane, 2017]. Preservar la privacidad los esquemas han avanzado significativamente en los últimos años, especialmente debido a la

necesidad de comunicación. La investigación se ha centrado en crear mecanismos de seguridad adecuados para el contexto de la inteligencia dispositivos de medida; sin embargo, las necesidades son variadas y crecientes. Además, la privacidad cotidiana está expuesta a intrusiones de aquellos que tienen propósitos maliciosos y poseen suficiente conocimiento para encontrar información delicada.

En investigaciones recientes, se han presentado varios mecanismos criptográficos para fortalecer la seguridad en dispositivos de medición y redes de energía inteligentes, como se revisó en [Desai, 2018]; sin embargo, los resultados obtenidos muestran la necesidad de esquemas novedosos para reducir la complejidad y los recursos computacionales en los trabajos revisados. De esta manera, el presente trabajo tiene como propósito la implementación de un algoritmo de "ofuscación de datos" en un sistema embebido de bajos recursos computacionales basado en un generador de bits pseudoaleatorio a través de semillas caóticas.

Probada la efectividad de la implementación del algoritmo de cifrado, se combinan dos funciones; ecuación logística y generador congruencial, para analizar la compensación entre recursos y seguridad. El objetivo principal es fortalecer la seguridad para preservar la privacidad contra ciber-ataques. Sin embargo, para todas las aplicaciones prácticas, el rendimiento y el costo de implementación también son factores a tener en cuenta al fortalecer la seguridad de los datos.

2. Métodos

Esta investigación se centra en el caso de la aplicación logística. Esta viene dada por la ecuación unidimensional dependiente de un solo parámetro μ, ecuación 1.

$$x_{t+1} = \mathrm{f}(\ x_t) = \mu x_t(1 - x_t) \qquad (1)$$

Donde x Є (0, 1) y es la variable de estado que determina las secuencias, μ Є (0, 4) y es el parámetro de control que determina el grado de no linealidad del mapa, de modo que al ser iterada a partir de un valor inicial $x_0$se va generando una órbita o trayectoria $\{x_1, x_2, x_3 \ldots\}$. La aplicación fue introducida por por Verhulst en 1845. Fue popularizada en un artículo de 1976 del físico May, como análoga en tiempo discreto a la ecuación diferencial logística para el crecimiento de una población.

May, en su artículo, enfatizó que incluso aplicaciones no lineales simples pueden presentar dinámicas muy complejas [Bose, 2006], [Xiao, 2009]. El hecho de que la iteración del cálculo para distintos valores del parámetro μ condujese a soluciones complejas, que parecían aleatorias en su comportamiento pese a tratarse de un modelo determinista muy sencillo causó gran impacto a nivel científico, y fue uno de los detonantes del estudio de lo que se llamaría teoría del caos. El diagrama de bifurcación con distintos valores en μ que se muestra en la figura 1, resume todos los comportamientos antes mencionados. El eje horizontal muestra los valores del parámetro μ, y el eje vertical muestra el valor x en el intervalo (0,1), además, el diagrama es un fractal [Zhang, 2013].

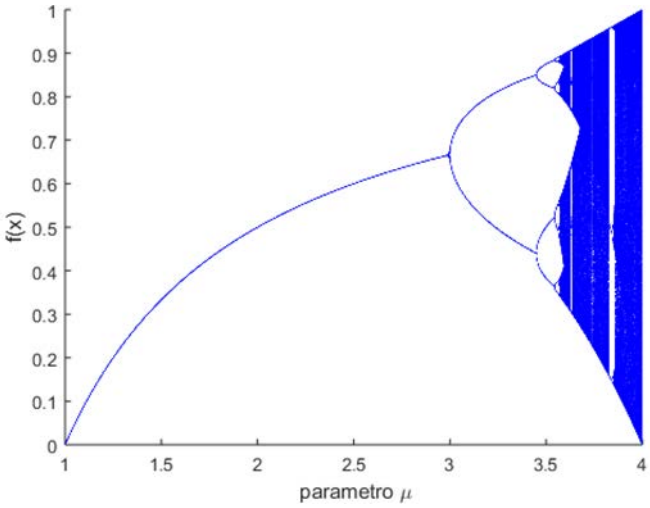
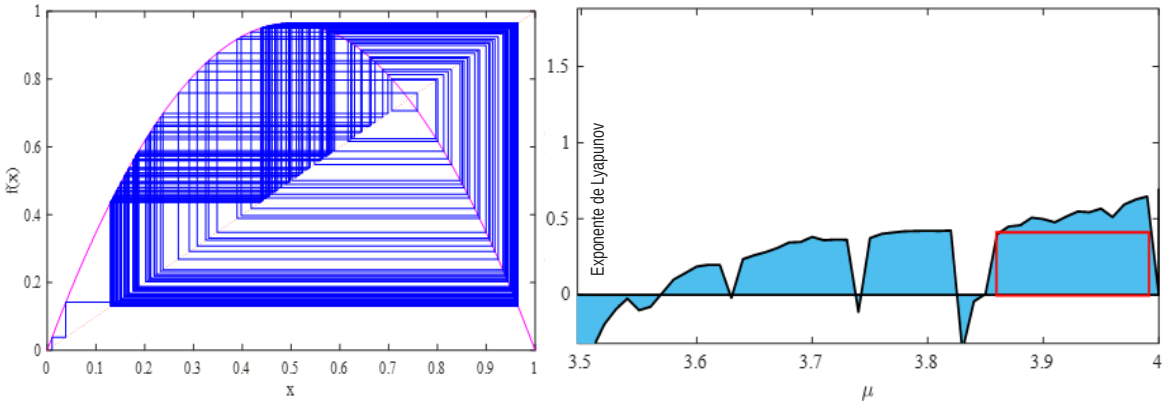


Figura 1 Diagrama de bifurcación.

El gráfico del diagrama de bifurcación básicamente representa el valor de μ contra los puntos donde la dinámica se ha concentrado, después de algunas iteraciones iniciales [Radwan, 2016]. En la figura 2 a), dado un valor del parámetro μ superior a 3.57 y $x = 0.32$, se muestra el comportamiento de las órbitas y la parábola; nótese que, en este caso desde la parábola, el procedimiento para encontrar los puntos fijos no tiene terminación. Esto significa presencia de un "atractor extraño", para el caso presente $\mu = 3.86$ y $x_t = 0.32$. Para asegurar secuencias impredecibles, es necesario utilizar un parámetro μ como semilla que se encuentre dentro de la zona en que el sistema se comporta de forma caótica como se aprecia en el diagrama de

bifurcación figura 1 y en el diagrama de trayectorias figura 2 a). Para garantizar que los parámetros utilizados sean los apropiados, se evalúan a través de los exponentes de Lyapunov [Sandri, 1996], que cuantifican el grado de sensibilidad a las condiciones iniciales, es decir, la inestabilidad local en un espacio de estados con el fin de delimitar el rango del parámetro que muestre un comportamiento impredecible, donde un valor positivo del exponente de Lyapunov indica divergencia exponencial de trayectorias cercanas como se observa en la figura 2.



a) Ecuación logística con $\mu = 3.86$ *y* $x = 0.32$ *b) Diagramas de exponentes de Lyapunov*

Figura 2 Zona de caos.

La generación de números pseudo-aleatorios juega un papel crítico en gran número de aplicaciones tales como, simulaciones numéricas, las comunicaciones o la criptografía. Las principales ventajas de tales generadores son la rapidez y las sucesiones de números aleatorios con periodos máximos. En la práctica, la generación de números pseudo-aleatorios no es trivial y la calidad aleatoriedad de la secuencia producida puede ser esencial en la elección de la aplicación [Zia, 2017]. Los generadores de números pseudoaleatorios son de vital importancia en muchas aplicaciones criptográficas para la generación de claves y códigos de acceso. Uno de los generadores más antiguos y sencillos es el generador de congruencia lineal, propuesto por D.H. Lehmer en 1949, que consiste en, a partir de un número inicial llamado semilla, generar una secuencia por recurrencia; cuya relación se muestra en la ecuación 2 [Miyazaki, 2008].

$$X_{n+1} = (\mathrm{a}X_n + c) mod\ m \qquad (2)$$

Donde X_n, es el valor inicial de las secuencias pseudoaleatorias, a es el multiplicador y c es el incremento; además debe tenerse en cuenta que los valores de a, X_n y c deben ser mayores a cero. Y la variable m que es el módulo, debe ser un número primo suficientemente mayor que los tres anteriores.

Este tipo de generador es computacionalmente rápido y de fácil implementación; sin embargo, posee propiedades no tan ideales, como la producción de secuencias de valores que se repiten con un período máximo de $m-1$, por otra parte, las secuencias producidas por un generador congruencial lineal son muy sensibles a cambios en sus parámetros, lo cual es una propiedad útil.

Los dos métodos descritos anteriormente se combinan en el diseño del algoritmo de cifrado propuesto, aprovechando las características principales de cada método. Dichas características son la velocidad de procesamiento y el bajo costo, en términos de recursos de hardware computacional requeridos.

El mapa logístico definido en la ecuación 1 muestra alta sensibilidad a las condiciones iniciales, parámetro fundamental para generar secuencias pseudoaleatorias. Los valores se citan en los intervalos $xt \in (0,1)$ y $\mu \in (3.85,4)$ y al evaluarse con los exponentes de Lyapunov se obtiene valores positivos, lo que es característico cuando el mapa está en régimen caótico [Rezk, 2019].

Dentro de los intervalos antes mencionados junto con las condiciones iniciales $\mu = 3.86$ y $x = 0.01999$, para la primera secuencia $\mu = 3.89$ y $a = 0.00499$ para la segunda secuencia; bajo estas condiciones iniciales se generan series de números pseudoaleatorios con gran impredecibilidad que se utilizan como semillas para complementar la clave de cifrado mediante la aplicación de la "técnica de confusión" que oculta las relaciones entre la información original, la cifrada y la clave generada.

En la figura 3, muestra el algoritmo, evidenciando los valores de las condiciones iniciales que alimentan las funciones de la ecuación logística y originan las secuencias utilizadas por el generador congruencial. Este diagrama representa el procedimiento seguido para generar dos secuencias (GNPR1 y GNPR2), a través del mapa logístico unidimensional. Las secuencias generadas se acoplan al generador congruencial para aumentar el nivel de pseudoaleatoriedad [Robinson, 2009.]; Los datos a cifrar corresponden a una señal de consumo de energía eléctrica

obtenida a través de un prototipo. El proceso de cifrado se realiza mezclando la señal de interés con las secuencias del generador congruencial a través del operador lógico de disyunción XOR; posteriormente la señal encriptada esta lista para ser enviada de forma inalámbrica a través de un canal probablemente inseguro.

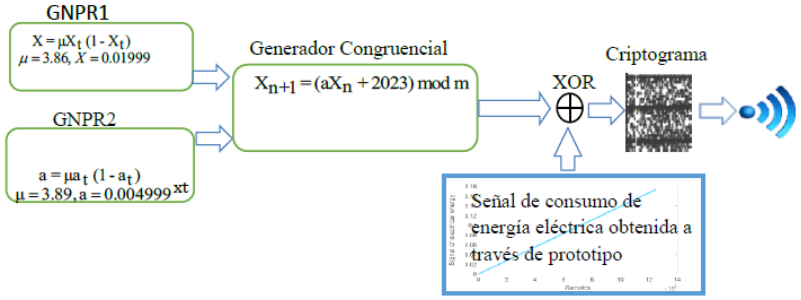


Figura 3 Diagramas del algoritmo generador pseudoaleatorio.

Otra etapa importante del proceso es el descifrado para la recuperación de la señal original, lo cual requiere realizar la operación de forma inversa como se muestra en la figura 4, donde el procedimiento de descifrado es muy similar al cifrado haciendo uso de los mismos valores iniciales que alimentan al generador congruencial.

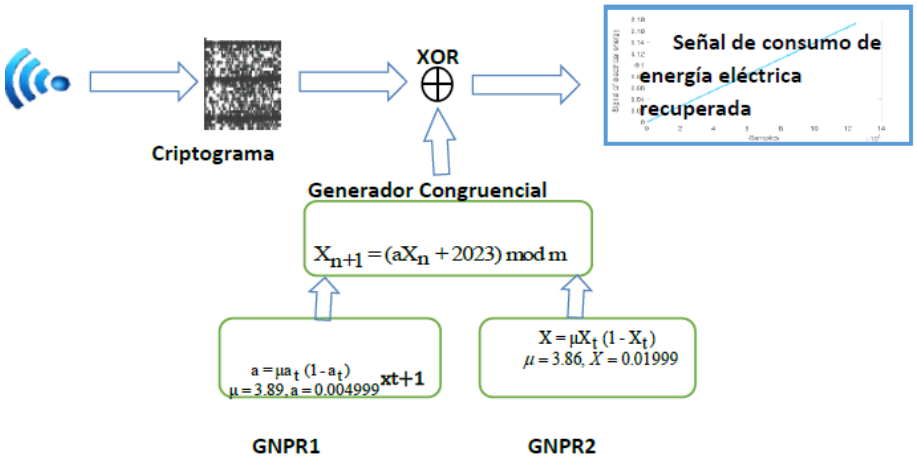


Figura 4 Diagrama de bloques del algoritmo de recuperación de señal cifrada.

Los algoritmos presentados en este trabajo ofrecen un alto grado de confidencialidad, ya que la información solo puede ser recuperada utilizando la

misma semilla usada en la generación de las secuencias que desempeña la función de clave para generar el sistema criptográfico.

3. Resultados

En esta sección se muestra la implementación del algoritmo de cifrado/descifrado sobre un prototipo que incluye la medición del consumo de energía eléctrica con el propósito de ser enviados de forma inalámbrica y fortalecer la seguridad y privacidad, dos temas clave para este tipo de sistemas [Soliman, 2017].

El desarrollo del prototipo incluye una etapa de medición de voltaje y corriente para el cálculo de potencia instantánea y consumo de energía eléctrica monofásica, lo anterior se hace sobre un circuito ideal resistivo para evitar problemas con el factor de potencia. Esta etapa se representa con el diagrama de bloques de la figura 5, en donde se ilustra el proceso de adquisición de las variables físicas a través de los sensores de voltaje (V) y corriente (I).

La implementación del prototipo se hace sobre la plataforma Arduino, ya que, entre sus principales características técnicas se destaca su bajo costo, tamaño reducido, las interfaces de comunicación que maneja, y además posee un conversor análogo digital y puertos con salida de PWM por hardware, cuenta con manejo de interrupciones externas e internas y una gran cantidad de librerías para el manejo de diferentes dispositivos.

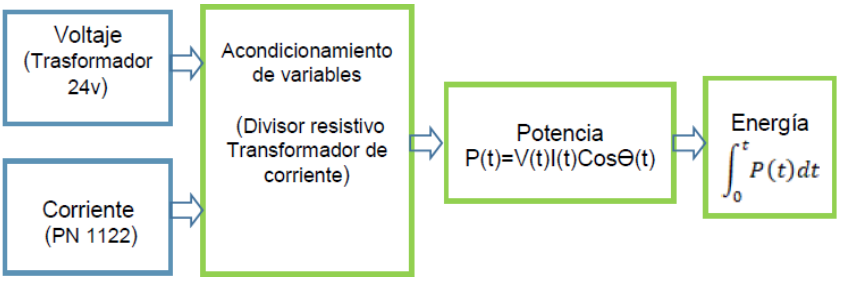


Figura 5 Diagrama de bloques del proceso de adquisición de variables físicas.

Se requiere un acondicionamiento de las señales para manejar valores entre 0 y 5 volts debido a que las entradas analógicas del Arduino no permiten valores negativos. El cálculo de la potencia se obtiene con la ecuación 3, cabe aclarar que

en las pruebas que aquí se reportan el ángulo θ es igual a cero por tener un circuito resistivo, es decir, factor de potencia igual a 1.

$$P(\mathrm{t}) = \mathrm{V(t)I(t)Cos\Theta(t)} \tag{3}$$

Donde la variable $P(t)$ representa la potencia instantánea (watts), $V(t)$ e $I(t)$, representan el voltaje y la corriente, respectivamente. El consumo de energía eléctrica se calcula con la integral de la potencia por el diferencial de tiempo, evaluando en el intervalo de tiempo de funcionamiento o de prueba, como lo muestra la ecuación 4.

$$E = \int_0^t P(t)dt \tag{4}$$

Donde E corresponde al consumo de energía eléctrica. En la adquisición de los datos se usa un periodo de muestreo de 50 milisegundos. Las pruebas realizadas para obtener la señal de consumo de energía, misma que es usada para el cifrado, se hacen sobre una lámpara incandescente de 100 Watts, figura 6.

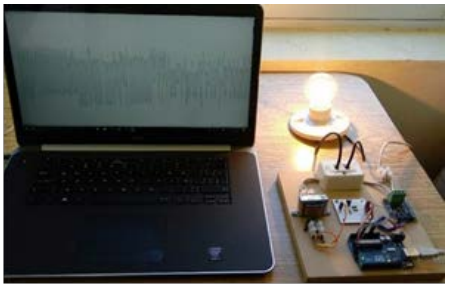


Figura 6 Prototipo de cálculo de consumo de energía eléctrica.

El esquema del sistema integrado para la creación del prototipo se muestra en la figura 7. Una vez que los datos están encriptados, se establece la comunicación con otros dispositivos a través de Bluetooth (transmisor HC-05) incorporado. Por otra parte, la figura 8 muestra la imagen del prototipo.

La etapa de adquisición de datos se subdivide en dos subetapas; uno para el acondicionamiento de la señal y el otro para la adquisición de datos. El rendimiento estadístico del criptosistema se evaluó mediante un conjunto de pruebas estadísticas, utilizando 125000 muestras de datos como se muestra en la figura 9 y estableciendo el intervalo de parámetros $\mu \in (3.86;\ 4)$ y $xt \in (0;\ 1)$.

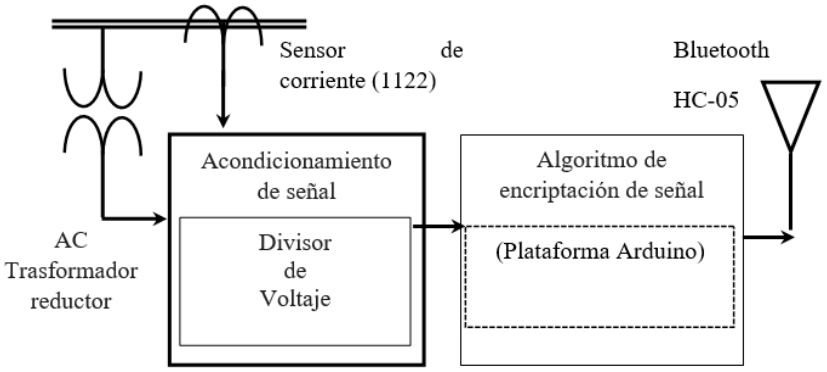


Figura 7 Diagrama de bloque de Sistema embebido.

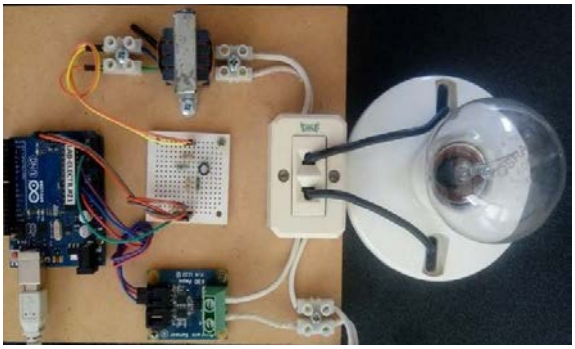


Figura 8 imagen de prototipo.

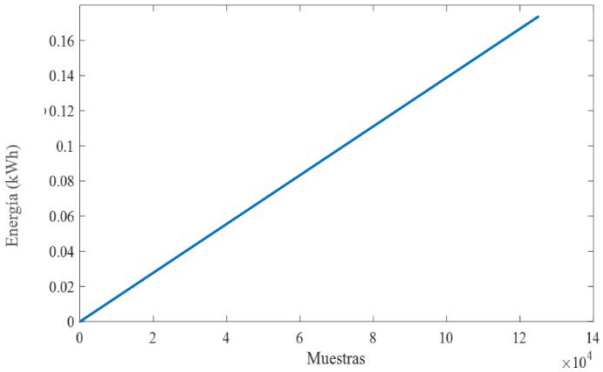


Figura 9 Señal de consumo de energía eléctrica.

La señal cifrada (Figura 10), evidencia una media de $1.8173\ x\ 10^3$ y métricas de varianza de $1.0860\ x\ 10^6$para la señal original, mientras que, para la señal cifrada, la media y la varianza se calcularon como $4.4981\ x\ 10^5$y $6.7240\ x\ 10^{10}$ respectivamente. Cada valor P correspondiente a una prueba en particular se presenta en la tabla 1 e indica las secuencias producidas por el algoritmo propuesto.

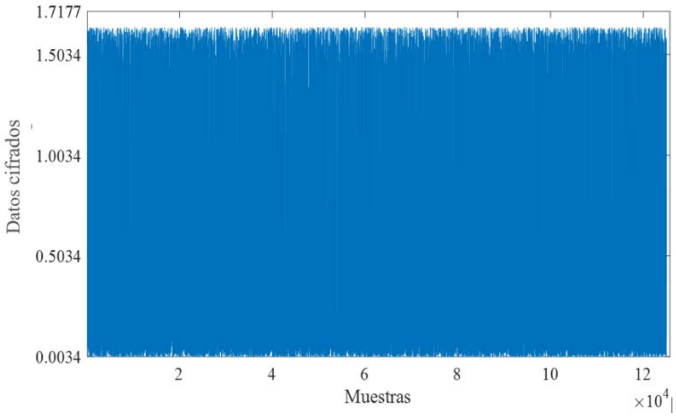


Figura 10 Señal de consumo de energía eléctrica cifrada.

Después de la representación gráfica los resultados de cifrado, éstos son evaluados bajo las pruebas estadísticas de aleatoriedad de 15 pruebas NIST como se muestra en la tabla 1 [Bassham, 2016]. El rendimiento estadístico del criptosistema se evaluó utilizando un conjunto de pruebas estadísticas, utilizando 125000 muestras de datos y configurando el intervalo de parámetro $\mu \in (3.86, 4)$ y condición inicial de $xt \in (0, 1)$.

Tabla 1 Pruebas Nist de los métodos utilizados para el algoritmo.

Pruebas	**Algoritmo Propuesto**			
	Simulado		**Implementado**	
	Valor_P	Estado	Valor_P	Estado
Entropía aproximada	0.8097	✓	0.3679	✓
Frecuencia de bloqueo	0.4917	✓	0.8712	✓
Sumas acumuladas (adelante)	0.4128	✓	0.2456	✓
Sumas acumuladas (inversa)	0.3129	✓	0.5297	✓
Transformada Discreta de Fourier	0.8043	✓	0.0067	×
Frecuencia	0.4065	✓	0.3200	✓
Complejidad lineal	0.7503	✓	0.3372	✓
Corridas	0.5048	✓	0.0470	✓
Coincidencia de plantillas no superpuestas	0.5094	✓	0.3896	✓
Coincidencia de plantillas superpuestas	0.3136	✓	0.1526	✓
Rango de una Matriz	0.8851	✓	0.0712	✓
Corridas por bloques	0.4369	✓	0.6248	✓
Excursión Aleatoria	0.5094	✓	0.7154	✓
Serial	0.3816	✓	0.9645	✓
Variante de Excursión Aleatoria	0.8772	✓	0.2589	✓

Los resultados obtenidos demuestran que todas las métricas NIST se logran bajo simulación utilizando el algoritmo de cifrado propuesto; mientras en el prototipo, la prueba FFT muestra un valor P bajo, que supone como resultado de una interferencia eléctrica en el circuito. La tabla 2, resume las métricas estadísticas del comportamiento de la señal resultante después del proceso de cifrado donde las propiedades importantes esperadas son la uniformidad e independencia en los datos.

Tabla 2 Concentrado de evaluaciones estadísticas al criptograma.

Coeficientes de correlación	0.000251
Entropía	0.8097
Error cuadrado medio	3.469111
Media	4.4981×10^{05}
Varianza	6.7240×10^{10}
Información mutua	0.000202

4. Discusión

Antes de realizar el diseño del hardware se selecciona el sistema de cifrado adecuado basado en el nivel de seguridad deseado y los recursos disponibles en la metodología de diseño para fortalecer la seguridad y mermar las amenazas de seguridad misma. Se trabaja en el diseño del desarrollo de cifrado de peso ligero apto para implementarse en un diseño de hardware de bajos recursos y limitado de energía. Dato que la implementación fue uno de los principales objetivos, la comparación del tiempo de procesamiento se probó en una computadora personal bajo Matlab R2015a, con un procesador Intel (R) Celeron 2 Core a 2.16 GHz de frecuencia, 4 GB en RAM, ejecutando Windows 10 Home O.S. El tiempo de procesamiento resultante fue de 0,5263 segundos, cifrando 125,000 muestras de datos de consumo de energía eléctrica. Finalmente, en la tabla 3, se muestra el tiempo de cifrado promedio comparativo tomado de algunas imágenes de Lena con diferentes tamaños. El tiempo de ejecución del algoritmo criptográfico aumenta a una tasa menor que la observada en Li et al. [Li, 2017]. El análisis de tiempo se realizó en una CPU Core 2 Duo de 2.26 GHz con una notebook de 4 GB de RAM usando Matlab; las mismas características que Li. [Li, 2017].

Tabla 3 Tiempo comparativo de cifrado.

Tamaño de la imagen (pixeles)	**Tiempo de cifrado(s)**	
	Algoritmo propuesto	(Li et al., 2017)
256 x 256	0.84	0.90
512 x 512	1.79	1.82
1024 x 1024	8.46	13.08
2048 x 2048	33.45	76.38

En investigaciones recientes se discute la implementación de hardware de un sistema de cifrado de peso ligero, incluyendo un estudio comparativo con otras implementaciones. Sin embargo, los diseños de comparación por lo general se aplican en diferentes tecnologías de proceso, por lo tanto, la comparación no es muy precisa. En principio, es muy difícil comparar objetivamente las implementaciones de software. Considerando que la implementación de software depende del estilo de codificación y la plataforma entre otras cosas.

5. Conclusiones

Este artículo presenta la implementación y evaluación de un algoritmo generador de bits pseudoaleatorio con cifrado caótico, basado en secuencias dinámicas. Estas secuencias se generan a partir de funciones unidimensionales de la ecuación logístico acoplado a un generador congruencial lineal cuyos parámetros constituyen la clave secreta para el sistema de cifrado.

Se presentó un prototipo que calcula el consumo de energía eléctrica y se implementa en éste el algoritmo de cifrado/descifrado. El prototipo lo caracteriza el bajo costo económico y computacional sin sacrificar el rendimiento requerido.

El algoritmo se evalúa bajo un entorno ideal con las pruebas NIST. Logrando todas las métricas NIST satisfactorias bajo simulación; sin embargo, en las pruebas sobre el prototipo la métrica de FFT cuyo propósito es detectar características periódicas (es decir, patrones repetitivos que están cerca uno del otro) fue rechazada derivado de la interferencia eléctrica en el circuito.

Los resultados presentados ofrecen un alto grado de confiabilidad, ya que la información solo se puede utilizar con la misma clave utilizada para generar el algoritmo. En este caso tiene un error cuadrático medio de 3.469111 en las pruebas

de sensibilidad, que indica qué tan lejos están los datos descifrados incorrectos de los datos originales. Se observó un tiempo de procesamiento de 0.5263 segundos en un Intel Celeron de 2.16 GHz.

6. Bibliografía y Referencias

[1] Al-Haija Q., Tarayrah M., Al-Qadeeb H. & Al-Lwaimi A., (2014). A Tiny RSA Cryptosystem based on Arduino Microcontroller Useful for Small Scale Networks.Procedia Computer Science, 34, pp.639-646.

[2] Badra M. & Zeadally S. (2017). Lightweight and efficient privacy-preserving data aggregation approach for the Smart Grid. Ad Hoc Networks, 64, pp.32-40.

[3] Bassham L., et al. (2018). A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST: https://www.nist.gov/publications/statistical-test-suite-random-and-pseudo random-number-generators-cryptographic.

[4] Benslimane Y. & BenAhmed K., (2017). Efficient End-to-End Secure Key Management Protocol for Internet of Things. International Journal of Electrical and Computer Engineering (IJECE), 7(6), p.3622.

[5] Borges de Oliveira F., (2018). on privacy-preserving protocols for smart metering systems. springer international pu.

[6] Bose R. & Pathak S., (2006). A novel compression and encryption scheme using variable model arithmetic coding and coupled chaotic system. IEEE Transactions on Circuits and Systems I: Regular Papers, 53(4), pp.848-857.

[7] Carlet C., (2008). On an improved correlation analysis of stream ciphers using multi-output Boolean functions and the related generalized notion of nonlinearity. Advances in Mathematics of Communications, 2 (2)

[8] Dazahra M., Elmariami F., Belfqih A. & Boukherouaa J., (2018). A Defense-in-depth Cybersecurity for Smart Substations. International Journal of Electrical and Computer Engineering (IJECE), 8(6), p.4423.

[9] Desai S., Alhadad R., Chilamkurti N. & Mahmood A., (2018). A survey of privacy preserving schemes in IoE enabled Smart Grid Advanced Metering Infrastructure. Cluster Computing, 22(1), pp.43-69.

[10] Guan Z., Si G., Wu J., Zhu L., Zhang Z. & Ma Y., (2017). Utility-Privacy Tradeoff Based on Random Data Obfuscation in Internet of Energy. IEEE Access, 5, pp.3250-3262.

[11] Jiménez Rodríguez M., et al., (2015). Sistema para codificar información implementando varias órbitas caóticas. Ingeniería, Investigation y Tecnología, 16(3), pp.335-343.

[12] Knirsch F., Eibl G. & Engel D., (2018). Error-Resilient Masking Approaches for Privacy Preserving Data Aggregation. IEEE Transactions on Smart Grid, 9(4), pp.3351-3361.

[13] Kong J., Ang L. & Seng K., (2015). A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments. Journal of Network and Computer Applications, 49, pp.15-50.

[14] Li C., Luo G., Qin K. et al., (2017). An image encryption scheme based on chaotic tent map. Nonlinear Dyn 87, 127–133.

[15] López Hernández J., Díaz Méndez A., Del Río Correa J., Cruz Irisson M. & Vázquez Medina R., (2012). A current mode CMOS noise generator using multiple Bernoulli maps. Microelectronic Engineering, 90, pp.163-167.

[16] Miyazaki Y., Tsuneda A., & Inoue T., (2008). Spreading Sequences with Negative Auto-correlations Generated by LFSRs Based on Chaos Theory of Modulo-2 Added Sequences. In ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications, pp. 541-544.

[17] Mrabet Z., Kaabouch N., Ghazi H., (2018). Cyber-security in smart grid: Survey and challenges. Computers & Electrical Engineering, 67, pp.469-482.

[18] Mylrea M., (2017). Smart energy-internet-of-things opportunities require smart treatment of legal, privacy and cybersecurity challenges. The Journal of World Energy Law & Business, 10(2), pp.147-158.

[19] Robinson C., (2009). Dynamical Systems, Boca Raton, Fla.CRC Press.

[20] Rottondi C. & Verticale G., (2015). Privacy-friendly load scheduling of deferrable and interruptible domestic appliances in Smart Grids. Computer Communications, 58, pp.29-39.

[21] Qin Z., Zhou E., Ding Y., Zhao Y., Deng F. & Xiong H., (2018). Data Service Outsourcing and Privacy Protection in Mobile Internet. Data Service Outsourcing and Privacy Protection in Mobile Internet.

[22] Radwan A., AbdElHaleem S. & Abd-El-Hafiz S., (2016). Symmetric encryption algorithms using chaotic and non-chaotic generators: A review. Journal of Advanced Research, 7(2), pp.193-208.

[23] Rezk A., Madian A., Radwan A. & Soliman A., (2019). Reconfigurable chaotic pseudo random number generator based on FPGA. AEU - International Journal of Electronics and Communications, 98, pp.174-180.

[24] Sandri M., (1996). Numerical calculation of lyapunov exponents. Mathematica Journal, 6(3):78–84.

[25] Soliman M. S., Alahmadi A. A., Maash A. A., & Elhabib M.O., (2017). Design and Implementation of a Real-Time Smart Home Automation System Based on Arduino Microcontroller Kit and LabVIEW Platform. Vol 3(18), pp. 7259-7264.

[26] Tan X., Zheng J., Zou C. & Niu Y., (2016). Pseudonym-based privacy-preserving scheme for data collection in smart grid. International Journal of Ad Hoc and Ubiquitous Computing, 22(2), p.120.

[27] Tonyali S., Cakmak O., Akkaya K., Mahmoud M. & Guvenc I., (2016). Secure Data Obfuscation Scheme to Enable Privacy-Preserving State Estimation in Smart Grid AMI Networks. IEEE Internet of Things Journal, 3(5), pp.709-719.

[28] Xiao D., Liao X. and Wei P., (2009). Analysis and improvement of a chaos-based image encryption algorithm. Chaos, Solitons & Fractals, 40(5), pp.2191-2199.

[29] Zapateiro De la Hoz M., Acho L. and Vidal Y., (2015). An Experimental Realization of a Chaos-Based Secure Communication Using Arduino Microcontrollers. The Scientific World Journal, 2015, pp.1-10.

[30] Zhang Y. & Xiao D., (2013). Double optical image encryption using discrete Chirikov standard map and chaos-based fractional random transform. Optics and Lasers in Engineering, 51(4), pp.472-480.