Congr. Int. Ing. Electrón. Mem. Electro 2012, vol.34, pp. 28-33 Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2012.zip
ISSN 1405-2172

ANÁLISIS Y MEJORAS EN ALGORITMOS DE ENCRIPTACIÓN EN COMUNICACIONES INALÁMBRICAS

Víctor Manuel Hinostroza Zubía(1) y Armando Bolívar(2)
Universidad Autónoma de Ciudad Juárez
Av. Del Charro # 450 norte Colonia Partido Romero Ciudad Juárez Chihuahua, México 32310
Teléfono: 656-6884800 al 09, ext. 4971 y 4674; Fax: 656-6884841
(1) vhinostr@uacj.mx y (2) abolivar@wistron.com

RESUMEN

En este trabajo se analizo y se modifico una versión básica del codificador Rijndael (AES), se realizaron análisis de eficiencia y de seguridad en este codificador reducido. Se revisó la teoría de las redes inalámbricas relacionada con los fundamentos de encriptación y los fundamentos matemáticos para encriptación. Los resultados obtenidos de la modificacionesnos muestran que al variar los parámetros del codificador se obtienen variaciones no muy grandes en el tiempo de codificación y se puede reducir considerablemente el consumo de energía. En este trabajo son comparadas distintas versiones del codificador, así como una comparación con respecto al codificador original que es tomado como base de comparación.

I. INTRODUCCIÓN.

AES (AdvancedEncryption Standard) es el estándar de encriptación más seguro y el cual es implementado en las técnicas de encriptación mas usadas en la actualidad. Este algoritmo está basado en el codificador a bloques Rijndael (solo se diferencia de este en la estandarización de un solo tamaño de bloque de texto) [2], [6]. Este codificador además de ser seguro es relativamente sencillo y eficiente.Sin embargo, existen ambientes, donde el consumo de potencia en el procesamiento es crítico para poder alargar la vida o autonomía de un dispositivo [9], [10]. Tomando como base esta premisa, en este trabajo se realizaron pruebas sobre una versión reducida del codificador Rijndaelpara reducir la potencia usada en su implementación y además para establecer guías básicas acerca de su seguridad.

Los dispositivos inalámbricos son utilizados para realizar cualquier tipo de actividades, algunas tan triviales como escuchar música y otras tan vitales como el monitoreo de signos vitales de un paciente. Por esto es importante dar una alternativa de protección cuando los dispositivos inalámbricos transmitan información sensible, para ello existen varios métodos de encriptación de información para comunicaciones inalámbricas. Estos algoritmos están basados en llaves de encriptación que en ocasiones carecen de la suficiente seguridad para algunas aplicaciones, ya que pueden ser quebrantados con relativa facilidad y rapidez. Otra de las limitantes que se tienen con estos ambientes es el consumo de energía, pues se puede aplicar algún algoritmo más complejo de encriptación como el utilizado por VPN's (Virtual Private Networks) corporativas, cuya seguridad es mucho mayor que los presentados por las comunicaciones comunes. Sin embargo, este nivel de seguridad conlleva un nivel elevado también de complejidad computacional y por consecuencia mayor consumo de energía en los componentes, lo cual podría reducir de manera significativa la vida de la batería de los dispositivos.

Además, a pesar de que los algoritmos de encriptación se manejan como estándares, algunos de ellos aun requieren ajustes y mejoramiento en su desempeño. Uno de los objetivos de este trabajo fue modificar una versión del algoritmo (AES) y tratar de mejorar su eficiencia computacional y consumo de energía para aplicaciones inalámbricas.

Actualmente la encriptación en las comunicaciones inalámbricas carece de la robustez que en ocasiones se requiere para ciertas aplicaciones. Aunque los algoritmos para una encriptación robusta existen, en las comunicaciones inalámbricas el costo de energía que se requiere para implementar estos algoritmos en algunas ocasiones es prohibitivo. Ya que el consumo de energía puede ser mayor al que el sistema en el que se implementaría pueda soportar y se tendría una reducción considerable en la vida de la batería de los dispositivos. El análisis de los algoritmos

actuales de encriptación de información y su adaptación para comunicaciones inalámbricas es un campo de acción que se estudio en este proyecto y la implementación de estas modificaciones para reducir el consumo de energía y mantener una seguridad aceptable en sistemas donde se transmiten datos sensibles.

En laparte2, se describe mas a profundidad el algoritmo AES, se explican sus características y como se lleva a cabo el proceso de la encriptación con este algoritmo. Asimismo, se proporciona una explicación detallada de cómo se implementa el algoritmo en aplicaciones prácticas. En la parte 3 se explican las modificaciones que se le hicieron al algoritmo y en la parte 4se muestran algunos de los resultados de este trabajo que se obtuvieron con estas modificaciones. Para finalizar, se presentan las conclusiones y recomendaciones. Finalmente se proporciona una lista de referencias.

II. ALGORITMO

El algoritmo Rijndael fue el candidato para AES, desarrollado por el Dr. Joan Daemen de ProtonWorld International y el Dr. VincentRijmen, un investigador postdoctoral en el departamento de Ingeniería Eléctrica de la KatholiekeUniversiteit de los Países Bajos. Rijndael es una red de transformación lineal de substitución (no Feistel) con múltiples vueltas, dependiente del tamaño de la llave. Las longitudes de la llave y del bloque son 128, 192 o 256 bits, no soporta tamaños arbitrarios y su tamaño de llave y bloque debe ser de una de estas tres longitudes.Utiliza una sola tabla de substitución que actúa en un byte de entrada para dar un byte de salida. Para propósitos de implementación, puede ser considerada como una tabla de búsqueda de 256 bytes. Rijndael se define por la ecuación 1 sobre el campo GF(28) donde M es una matriz y b es una constante. Un bloque de datos que va a ser procesado por Rijndael es particionado en un arreglo de bytes y cada una de las operaciones del codificador está orientada a bytes. Cada una de las diez vueltas de Rijndael realiza cuatro operaciones. En la primera etapa una tabla de substitución de 8 x 8 (tabla de substitución utilizada como componente no lineal) es aplicada a cada byte. La segunda y tercera etapa son etapas de mezcla lineal en donde cada renglón del arreglo es desplazado y las columnas mezcladas. En la cuarta etapa bytes de sub llaves son utilizadas en XOR en cada byte del arreglo. En la última vuelta, la mezcla de columnas es omitida.

$$S(x) = M(1/x) + b \quad (1)$$

Rijndael fue seleccionado como un estándar debido a su combinación de seguridad, desempeño, eficiencia, facilidad en la implementación y flexibilidad. Sus características son:buen desempeño en hardware y software en un amplio rango de ambientes computacionales;buen desempeño en modos de retroalimentación y no retroalimentación;el tiempo de configurar la llave es excelente;los requerimientos de memoria son bajos;fácil de defender en contra de ataques de tiempo o potencia (esta defensa se puede proveer sin un impacto significativo en el rendimiento).

Algunas de las desventajas de Rijndael son; que algunos opinan que las matemáticas internas son simples, casi rudimentarias. Si Rijndael fuera escrito como una fórmula matemática, sería mucho más sencilla que cualquier otro candidato de estándar. Otra crítica es que Rijndael evita cualquier tipo de técnica de obstrucción para esconder sus mecanismos de encriptación de sus adversarios. Finalmente, Rijndael utiliza dos tablas de substitución diferentes para la encriptación y desencriptación, en contraste a DES que utiliza la misma tabla de substitución para ambas operaciones, esto significa que una implementación requiere que Rijndael tenga ambas partes de encriptación y desencriptación y es el doble de grande que la implementación que solo hace una sola operación, lo cual tal vez sea un inconveniente en dispositivos que estén restringidos.

III. MODIFICACIONES AL ALGORITMO

La modificación del algoritmo consistió en la reducción de vueltas, se realizaron pruebas con 4, 5, 6, 7, 8 y 9 vueltas. Resultando obvio que el mejor tiempo se presentaba a menores vueltas, al utilizar solo 4 vueltas con un bloque y llave de 32 bits se obtuvo una mejora del 30% con respecto a su contraparte estipulada en la definición original de 14 vueltas. En las graficas mostradas en esta sección se observa la comparación de todas las posibles combinaciones de bloques y llaves para un algoritmo reducido y la especificación original,

se observa también que entre más grande es el bloque con el que se trabaje menor es el tiempo en que se realiza la encriptación.

Tabla 6.1: Tiempo promedio del algoritmo original.

Tamaño (bytes) Llave / Bloque	16	24	32
16 (encriptación)	12.7092	11.6727	11.1856
24 (encriptación)	12.7190	11.6749	11.2191
32 (encriptación)	12.4173	11.6357	1.1274
16 (desencriptación)	12.5916	11.4410	10.8595
24 (desencriptación)	12.5880	11.4269	10.8435
32 (desencriptación)	12.5296	11.4596	10.8779

Una vez obtenidos los resultados con el algoritmo original, se realizo la misma cantidad de corridas en diferentes versión del algoritmo reducido en vueltas, las pruebas realizadas fueron hechas en tamaños de llave (series en la Figura 1) de 16, 24, 32 bytes, tamaños de bloques (eje X) 16, 24 y 32 bytes y 4, 5, 6, 7, 8, 9 vueltas (sub-eje X). Los tiempos promedios se ven graficados en la Figura 3 y Figura 4.

Se realizaron cien corridas para cada combinación de valores (vueltas / tamaño de llave / tamaño de bloque), con cien corridas se logro obtener una desviación estándar de 0.1344 en promedio con respecto al valor esperado en cada caso lo cual fue considerado como suficiente para poder eliminar posibles variaciones que tuviera la computadora que fue utilizada para realizar las pruebas. Se encriptó un archivo de 2 MB y el archivo encriptado fue desencriptado, ambos tiempos (encriptación y desencriptación fueron almacenados y promediados).

Ya que los tiempos menores fueron obtenidos con la versión de menor vueltas, 4 vueltas, se analizaron los tiempos promedios en las diferentes configuración de 16, 24 y 32 bytes tanto en tamaño de llave como en tamaño de bloque para esta versión reducida, Figura 6.5.

IV. CONCLUSIONES

AES (y en consecuencia Rijndael) es el algoritmo de encriptación que actualmente se está adoptando en todas las aplicaciones donde se requiera un nivel de seguridad en los datos, esto en gran parte a la adopción de este estándar por la NIST del gobierno norteamericano. La apertura del diseño del codificador Rijndael permite hacer análisis detallados acerca del desempeño y posibles mejoras en el algoritmo.

Rijndael es un algoritmo que puede ser fácilmente implementado en la mayoría del hardware y puede además ser implementado de manera eficiente en software, además su nivel de seguridad es alto esto en parte debido al margen de seguridad que tiene su codificador, varias vueltas de codificación permiten tener este margen que para estándares de seguridad es necesario. Sin embargo este margen de seguridad tiene un precio cuando nos referimos a consumo de potencia de diversos dispositivos donde potencialmente (o ya actualmente) se pueda implementar Rijndael. Por este motivo se analizo el desempeño y nivel de seguridad de una versión reducida del codificador, aun cuando no se logro implementar esta versión modificada del algoritmo en ninguna aplicación real, el concepto y el análisis pueden ser utilizados para futuros trabajos.

Aun cuando conceptualmente el diseño del codificador Rijndael permite una difusión de datos aparentemente suficiente a partir de la segunda vuelta en este trabajo se centro el análisis la versión reducida de cuatro vueltas ya que dos vueltas lo harían demasiado susceptible a ataques. Si bien reducir las vueltas del codificador implica por si mismo reducir el margen de seguridad del codificador hay ocasiones en la que sea preferible tener esta reducción en la seguridad a cambio de una sobrecarga de procesamiento menor que permita consumir menos recursos.

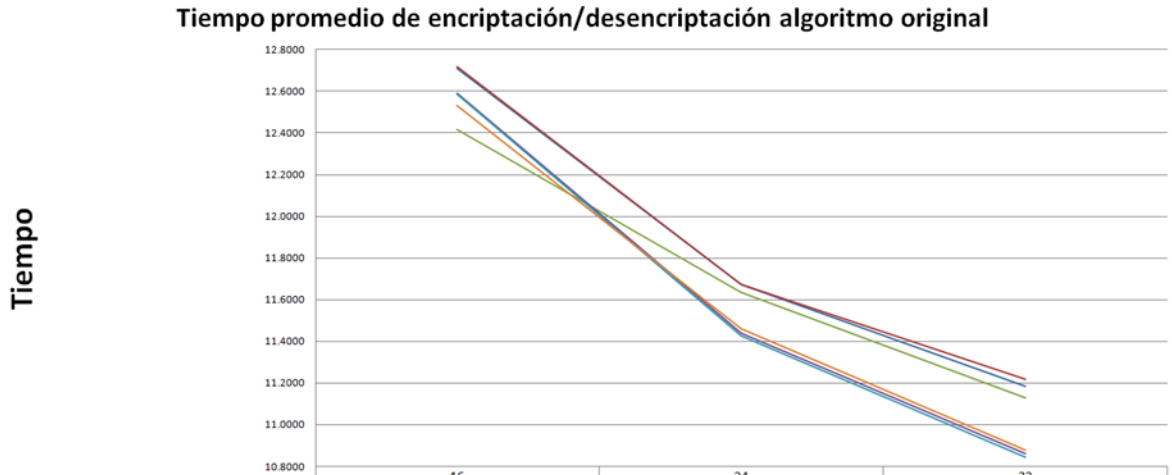
Otro punto a notar aquí es que si bien estas modificaciones pueden ser implementadas en un sistema “hecho en casa” o dentro del contexto de alguna otra investigación, este codificador esta fuera del estándar del AES por lo que no sería compatible con ninguna aplicación que maneje este codificador según lo estipula la FIPS. Es requerido realizar un

ISSN 1405-2172

criptoanálisis más detallado a estas modificaciones del algoritmo para poder determinar con mayor claridad y precisión la afectación de estas alteraciones al algoritmo del codificador.

REFERENCIAS:

1. B. Sklar, "Digital Communications: Fundamentals and Applications", 2da ed., 2001, Prentice Hall.
2. C. Swenson, "Modern Cryptoanalysis: Techniques for Advanced Code Breaking", 2008, John Wiley & Son.
3. D. Khadraoui, F. Herrmann, "Advances in Enterprise Information Technology Security", 2007, IGI Global.
4. R. L. Krutz, R. D. Vines, "The CEH Prep Guide: The Comprehensive Guide to Certified Ethical Hacking", 2008, John Wiley & Sons.
5. R. L. Krutz, R. D. Vines, "The CISSP Prep Guide: Gold Edition", 2003, John Wiley & Sons.
6. Y. Zhang, J. Zheng, M. Ma, "Handbook of Research on Wireless Security", 2008, IGI Global.
7. H. F. Tipton, M. Krause, "Information Security Management Handbook", 6ta ed., 2007, Auerbach Publications.
8. B. Carter, R. Shumway, "Wireless Security: End to End", 2002, John Wiley & Sons.
9. G. N. Selimis, A. P. Kakarountas, A.P. Fournaris, A. Milidonis, O. Koufopavlou, "A Low Power Design for Sbox Cryptographic Primitive of Advanced Encryption Standard for Mobile End-User", Journal of Low Power Electronics Vol. 3, 1-10, 2007.
10. K. J. Kumar, S. Salivahanan, K. C. Reddy, "Implementation of Low Power Scalable Encryption Algorithm", International Journal of Computer Applications (0975-8887), Vol. 11, No. 1, 2010.
11. P. Kitsos, O. Koufpavlou, G. Selimis, N. Sklavos, "Low Power Cryptography", Second Conference on Microelectronics, Microsystems and Nanotechnology, Journal of Physics: Conference Series 10, 343-347, 2005.
12. A. Garcia, H. Stichtenoth, "Topics in geometry, coding theory and cryptography", 2007, Springer.
13. M. Abdalla, D. Pointcheval, P. Fouque, "Applied Cryptography and Network Security: 7th International Conference", 1994, Springer.
14. R. E. Blahut, "Communications and cryptography: two sides of one tapestry", 1994, Springer.
15. H. Niederreiter, C. Xing, "Algebraic geometry in coding theory and cryptography", 2009, Princeton University Press.
16. O. N. Vasilenko, "Number-theoretic algorithms in cryptography", 2007, AMS Bookstore.



	16	24	32
Promedio tiempo encriptación llave (NK) - 16	12.7092	11.6727	11.1856
Promedio tiempo encriptación llave (NK) - 24	12.7190	11.6749	11.2191
Promedio tiempo encriptación llave (NK) - 32	12.4173	11.6357	11.1274
Promedio tiempo desencriptación llave (NK) - 16	12.5916	11.4410	10.8595
Promedio tiempo desencriptación llave (NK) - 24	12.5880	11.4269	10.8435
Promedio tiempo desencriptación llave (NK) - 32	12.5296	11.4596	10.8779

Figura 1: Tiempo promedio del algoritmo original.

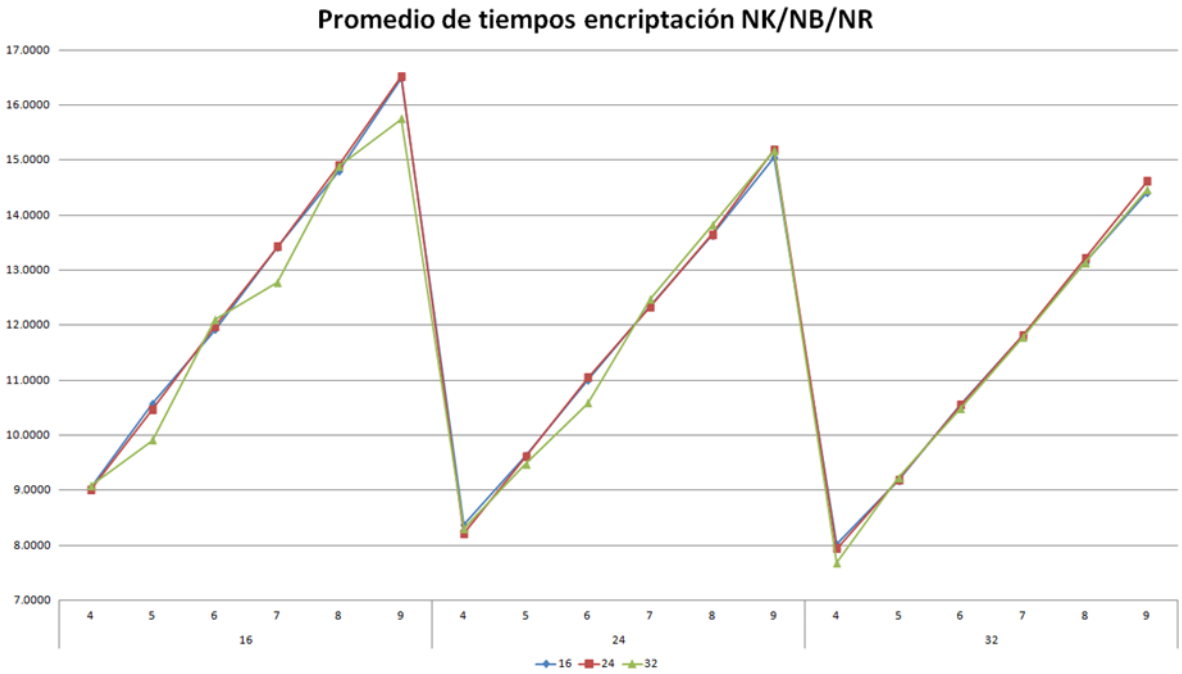
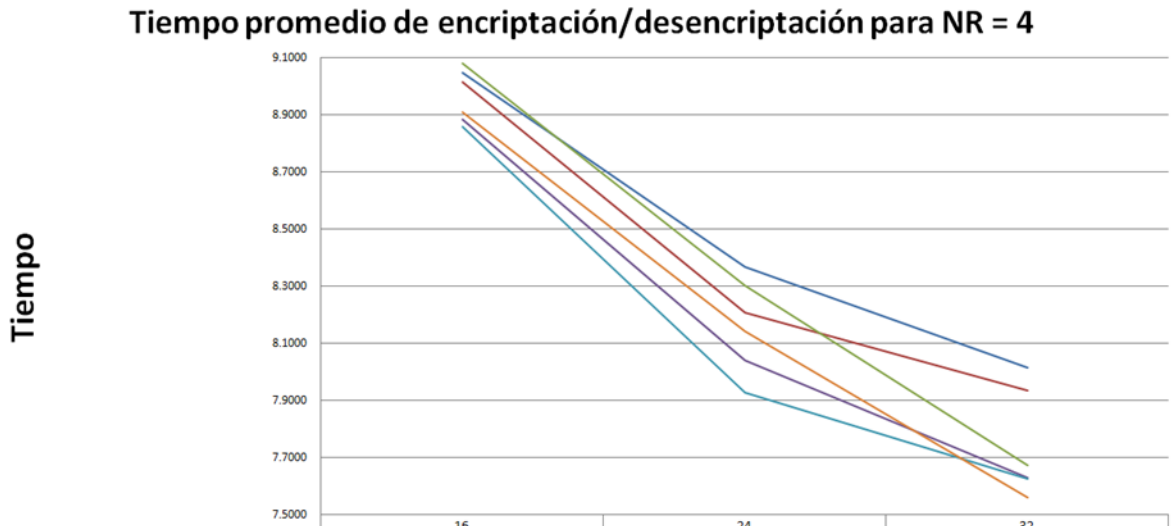


Figura 2. Promedio de tiempos de encriptación



	16	24	32
Promedio de tiempo encriptación (NK) - 16	9.0494	8.3688	8.0146
Promedio de tiempo encriptación (NK) - 24	9.0148	8.2101	7.9353
Promedio de tiempo encriptación (NK) - 32	9.0811	8.3035	7.6743
Promedio de tiempo desencriptación (NK) - 16	8.8836	8.0423	7.6299
Promedio de tiempo desencriptación (NK) - 24	8.8586	7.9280	7.6253
Promedio de tiempo desencriptación (NK) - 32	8.9109	8.1435	7.5626

Figura 3. Tiempo promedio de encriptación/desencriptación