UNIVERSIDAD NACIONAL
"SANTIAGO ANTÚNEZ DE MAYOLO"

ESCUELA DE POSTGRADO

"DISEÑO CRIPTOGRÁFICO SOBRE HARDWARE RECONFIGURABLE CIFRADO POR BLOQUES EMPLEANDO EL ESTÁNDAR DE ENCRIPTACIÓN AVANZADO, APLICADO AL ÁMBITO EDUCATIVO TECNOLÓGICO EN LA CIUDAD DE HUARAZ"

Tesis para optar el grado de Maestro
en Ciencias e Ingeniería
Mención en Auditoria y Seguridad Informática

FRANZ MARTIN VILLANUEVA COCHACHIN

Asesor: Mag. CESAR AUGUSTO NARRO CACHAY

HUARAZ – PERÚ

2017

UNIVERSIDAD NACIONAL "SANTIAGO ANTÚNEZ DE MAYOLO"

ESCUELA DE POSTGRADO

"DISEÑO CRIPTOGRÁFICO SOBRE HARDWARE RECONFIGURABLE CIFRADO POR BLOQUES EMPLEANDO EL ESTÁNDAR DE ENCRIPTACIÓN AVANZADO, APLICADO AL ÁMBITO EDUCATIVO TECNOLÓGICO EN LA CIUDAD DE HUARAZ"

Tesis para optar el grado de Maestro
en Ciencias e Ingeniería
Mención en Auditoria y Seguridad Informática

FRANZ MARTIN VILLANUEVA COCHACHIN

Asesor: Mag. CESAR AUGUSTO NARRO CACHAY

HUARAZ – PERÚ

2017

N° de Registro: T0574

MIEMBROS DEL JURADO

Doctor Alexander Pacheco Castillo Presidente

_____

Magister Erick G. Flores Chacón Secretario

_____

Magister Cesar A. Narro Cachay Vocal

_____

ASESOR

Magister Cesar Augusto Narro Cachay

AGRADECIMIENTO

- *A Dios por permitirme la vida y brindarme la sabiduría para cumplir todas mis metas.*
- *A mis padres por ser los cimientos y el motivo en la consecución de mis metas.*
- *Un agradecimiento al asesor Magister Cesar A. Narro Cachay, por su apoyo, así como la sabiduría que me transmitió para guiarme en el desarrollo del presente trabajo y llegar a la culminación del mismo.*
- *Un agradecimiento a la Escuela de Postgrado durante los años de estudio en mi formación profesional.*

A Dios, por permitirme la vida y darme unos maravillosos padres, quienes son la fuente de mi inspiración y brindarme su apoyo para desarrollarme como persona y buen profesional.

A mi querida hermana quien con sus atentos y cariñosos mensajes motivaron la consecución del presente trabajo.

A todos mis amigos, con quienes he compartido tantas alegrías y siempre estuvieron presente para brindarme sus consejos y apoyo.

ÍNDICE

RESUMEN

El propósito de la presente investigación fue la optimización del diseño criptográfico cifrado por bloques, empleando el estándar de encriptación avanzado sobre hardware reconfigurable, para su aplicación al ámbito educativo tecnológico en la Ciudad de Huaraz. La investigación es de tipo correlacional, explicativo, transversal, la población de estudio estuvo conformado por los usuarios de la red del Instituto de Educación Superior Tecnológico “Eleazar Guzmán Barrón”, con una muestra de 277 usuarios, los instrumentos utilizados para la recolección de la información fueron notas de campo, lista de cotejo, cuestionario de opinión, Se analizó la data obtenida mediante escalas de valoración, para un análisis estadístico rápido y ordenado, y la contrastación de la información mediante la prueba de chicuadrado.

Los resultados de la investigación, permitieron la optimización del diseño criptográfico cifrado por bloques empleando el estándar de encriptación avanzado sobre hardware reconfigurable, disminuyendo significativamente los altos valores en tiempo empleados durante la autenticación y trasmisión de datos.

Se concluye, que la implementación del modo cifrado por bloques (CCM) empleando AES sobre FPGA, alcanza una velocidad de 1.1617 Gbps, este último, superior a los antecedentes citados en la investigación, logrando un mejor desempeño que su contraparte implementada en software.

Los resultados obtenidos fueron aplicados en la red del Instituto de Educación Superior Tecnológico Público Eleazar Guzmán Barrón, de la Ciudad de Huaraz.

Palabras claves: Cifrado por bloques, matrices programables en campo, estándar de encriptación avanzado, lenguaje de descripción de hardware.

ABSTRACT

The purpose of the present investigation was the optimization of the encrypted cryptographic design by blocks, using the standard of advanced encryption on reconfigurable hardware, for its application to the technological educational field in the City of Huaraz. The research is correlational, explanatory, cross-sectional, the study population was made up of users of the Higher Institute of Technological Education "Eleazar Guzmán Barrón" network, with a sample of 277 users, the instruments used for the collection of the information were field notes, checklist, opinion questionnaire, the data obtained was analyzed by means of rating scales, for a quick and orderly statistical analysis, and the contrast of the information by means of the chi-square test.

The results of the research allowed the optimization of the cryptographic design encrypted by blocks using the standard of advanced encryption on reconfigurable hardware, decreasing significantly the high values in time used during the authentication and transmission of data.

It is concluded that the implementation of the encrypted mode by blocks (CCM) using AES over FPGA, reaches a speed of 1.1617 Gbps, the latter, higher than the background cited in the research, achieving a better performance than its counterpart implemented in software.

The results obtained were applied in the Eleazar Guzmán Barrón Institute of Higher Technological Public Education, of the City of Huaraz.

Key words: Block coding, field programmable matrices, advanced encryption standard, hardware description language.

I. INTRODUCCIÓN

Los avances tecnológicos y su creciente demanda en las diversas áreas de la ciencia, ha generado la denominada era de la información, trayendo consigo, una manera diferente de organizar el trabajo, aumentando así la productividad y la eficiencia de las empresas. En este contexto los medios informáticos se han convertido en herramientas casi indispensables en el día a día para la transferencia e intercambio de la información, desde operaciones tan sencillas como enviar correos electrónicos, realizar transacciones financieras, hasta operaciones tan complicadas como poner satélites en órbita, entre otras, conlleva a crear los mecanismos necesarios para mantener resguardada la información transmitida, y es hoy en día, la principal preocupación de los administradores de los servicios de red, para ello se han desarrollado diversos esquemas y protocolos de seguridad, entre ellos, y uno de los más importantes, se ubica el modo de cifrado por bloques, que en unión a otros protocolos puede aumentar considerablemente su grado de robustez, si este es empleado sobre la plataforma adecuada.

El presente proyecto, pretende diseñar un sistema para gestionar un grupo de usuarios, capaz de permitir el acceso a información privada mediante claves encriptadas. La agregación o desagregación al grupo de uno de sus usuarios provoca la redistribución de claves para alguno o todos ellos. La encriptación de estas claves se lleva a cabo mediante el algoritmo Advanced Encryption Standard (AES) y, para incrementar la eficiencia, éstas se organizan jerárquicamente. Además, para conseguir un mejor rendimiento, el sistema se implementará sobre un FPGA empleando el lenguaje VHDL.

1.1. Objetivos

1.1.1 Objetivo general.

Optimizar el diseño criptográfico sobre hardware reconfigurable del modo cifrado por bloques, empleando el estándar de encriptación avanzado, y aplicarlo al ámbito educativo, escogiendo como piloto a la Institución “Eleazar Guzmán Barrón” – Huaraz.

1.1.2. Objetivos específicos.

- Emplear el lenguaje de descripción de hardware VHDL para el diseño criptográfico sobre hardware reconfigurable del modo CCM empleando AES.
- Aplicar el diseño criptográfico sobre hardware reconfigurable del modo cifrado por bloques empleando el estándar de encriptación avanzado, en la red del Instituto Eleazar Guzmán Barrón - Huaraz.

1.2. Hipótesis.

1.2.1. Hipótesis alternativa.

H_1: El uso del estándar de encriptación avanzado sobre hardware reconfigurable del modo cifrado por bloques, permite optimizar el diseño criptográfico, y puede aplicarse al ámbito educativo tecnológico en la ciudad de Huaraz.

1.2.2. Hipótesis nula.

H_0: El uso del estándar de encriptación avanzado sobre hardware reconfigurable del modo cifrado por bloques, no permite optimizar el diseño criptográfico, y no puede aplicarse al ámbito educativo tecnológico en la ciudad de Huaraz.

1.3. Variables

1.3.1. Variable dependiente.

Estándar de encriptación avanzado.

1.3.2. Variable independiente.

Diseño criptográfico.

1.3.3. Matriz de consistencia.

<table>
<tr><th colspan="6">DISEÑO CRIPTOGRÁFICO SOBRE HARDWARE RECONFIGURABLE CIFRADO POR BLOQUES EMPLEANDO EL ESTÁNDAR DE ENCRIPTACIÓN AVANAZADO, APLICADO AL ÁMBITO EDUCATIVO TECNOLÓGICO EN LA CIUDAD DE HUARAZ</th></tr>
<tr><th>PROBLEMAS</th><th>OBJETIVOS</th><th>HIPÓTESIS</th><th>VARIABLES DE ESTUDIO</th><th>INDICADORES</th><th>METODOLOGÍA</th></tr>
<tr><td>GENERAL</td><td>GENERAL</td><td>GENERAL</td><td>INDEPENDIENTE</td><td>INDEPENDIENTE</td><td></td></tr>
<tr><td rowspan="3">¿Será posible optimizar el diseño criptográfico cifrado por bloques empleando el aes, sobre hardware reconfigurable y esta, ser aplicada al ámbito educativo tecnológico en la ciudad de Huaraz?</td><td>Optimizar el diseño criptográfico sobre hardware reconfigurable del modo cifrado por bloques, para comprobar la factibilidad de aplicación del estándar de encriptación avanzado, y aplicarlo al ámbito educativo, escogiendo como piloto a la Institución “Eleazar Guzmán Barrón” – Huaraz.</td><td rowspan="3">Hipótesis H1

❖ El uso del estándar de encriptación avanzado sobre hardware reconfigurable del modo cifrado por bloques, permite optimizar el diseño criptográfico, y puede aplicarse al ámbito educativo tecnológico en la ciudad de Huaraz.

Hipótesis H0

El uso del estándar de encriptación avanzado sobre hardware reconfigurable del modo cifrado por bloques, no permite optimizar el diseño criptográfico, y no puede aplicarse al ámbito educativo tecnológico en la ciudad de Huaraz.</td><td>Diseño Criptográfico sobre hardware reconfigurable.</td><td>Nivel de eficiencia frente a desarrollos en software.</td><td rowspan="3">Tipo de estudio
Correlacional,

Metodología
cuantitativa y cualitativa

Diseño de investigación
Aplicativo.

Población y muestra:
Docentes, administrativos y estudiantes del IESTP “Eleazar Guzmán Barrón” - Huaraz.

Técnica de recolección de datos
Cuestionario estructurado.</td></tr>
<tr><td>ESPECIFICO</td><td>DEPENDIENTE</td><td>DEPENDIENTE</td></tr>
<tr><td>- Emplear el lenguaje de descripción de hardware VHDL para el diseño criptográfico sobre hardware reconfigurable del modo CCM empleando AES.

- Aplicar el diseño criptográfico sobre hardware reconfigurable del modo cifrado por bloques empleando el estándar de encriptación avanzado, en la red del Instituto Eleazar Guzmán Barrón - Huaraz</td><td>Estándar de encriptación avanzado</td><td>Nivel de eficiencia en área y velocidad empleando VHDL para el diseño criptográfico.

Nivel de eficiencia en el empleo del estándar de encriptación avanzado en la red del IESTP “Eleazar Guzmán Barrón” – Huaraz.</td></tr>
</table>

Delimitación

Desde la óptica de **(Sabino, 1986)**[1], la delimitación habrá de efectuase en cuanto al tiempo y el espacio, para situar nuestro problema en un contexto definido y homogéneo.

Una vez justificada la investigación, es necesario plantear las limitaciones dentro de las cuales se realizara el presente proyecto.

- *Limitaciones de tiempo*. El presente estudio inicio su desarrollo en el mes de julio del año 2014, culminándose de manera satisfactoria, en el mes de febrero del año 2015. Obteniéndose los resultados previstos en el proyecto.
- *Limitaciones de espacio o territorio*. Los estudios fueron llevados a cabo en el ámbito académico, tomando como piloto al IESTP "Eleazar Guzmán Barrón" – Huaraz durante el año 2014, esto con la finalidad de cumplir los lineamientos y el marco legal en el desarrollo de la investigación.
- *Limitaciones de recursos*. Los recursos en su totalidad fueron cubiertos por el investigador, esto es: el equipo especializado en hardware, herramientas en software necesarias para la síntesis del diseño, encuestas y todos los consumibles para llevar a cabo el desarrollo del proyecto.

La población en estudio, estuvo delimitada al ambiente donde se llevó a cabo la investigación, especialmente a Docentes, Administrativos, y la comunidad Estudiantil que hacen usos de los recursos de Tecnologías de la Información y Comunicación.

[1] Sabino Carlos A. (1986) *El Proceso de Investigación*. Caracas: Editorial Panapo, p.53

Ética de la investigación.

Un punto importante de discusión en la actualidad es el lugar que la ética debe tener en la ciencia, y en las investigaciones científicas.

En principio, este tema se puede subdividir en dos: uno referente a la ética relacionada con la ciencia en sí, y otra que analiza la ética en las relaciones entre la ciencia y la sociedad.

(Bunge, 1996)[2], da una serie de hábitos que debería tener un buen científico como guía para evitar las faltas a la ética:

1. Honestidad intelectual (o "culto" a la verdad), el aprecio por la objetividad y la comprobabilidad, el desprecio por la falsedad y el autoengaño. La observancia de la honestidad intelectual exige:
2. La independencia de juicio, el hábito de convencerse por sí mismo con pruebas, y de no someterse a la autoridad. La honestidad intelectual y la independencia de juicio requieren, para ser practicadas, una dosis de independencia.
3. Coraje intelectual (y aún físico en ocasiones): decisión para defender la verdad y criticar el error cualquiera se a su fuente, y muy particularmente, cuando es un error propio. La crítica y la autocrítica practicadas con coraje infunden la toma de decisiones correctas.
4. Amor a la libertad intelectual, y, por extensión, amor por las libertades individuales y sociales que la posibilitan; concretamente, desprecio por la

[2] Bunge M. (1996) *Ética, Ciencia y Técnica*. M: Editorial siglo veintiuno editores, p.329 – 330

autoridad infundada – sea intelectual o política- y por todo poder injusto. La honestidad intelectual y el amor por la libertad llevan a afianzar el trabajo de una investigación.

5. Sentido de la justicia, que no es precisamente la servidumbre a la ley positiva – que nos imponen y que puede ser injusta- sino la disposición a tomar en cuenta los derechos y opiniones del prójimo, evaluando sus fundamentos respectivos.

II. MARCO TEORICO.

2.1. Antecedentes.

2.1.1 Antecedentes Nacionales.

a) **(Medina Aparcana, 2012)[3],** donde se centra en el estudio del algoritmo de criptografía con curvas elípticas sobre cuerpos p – ádicos propuestos por MaoZhi et al. La autora analiza algunos ataques al algoritmo, implementando para tal fin algunos ejemplos con el sistema de cálculo PARI/GP. Mediante el uso del grupo formal de las curvas elípticas definidas sobre $\mathbb{Q}_P$, encuentra un grupo finito especial al cual denomina *grupo criptográfico*. Finalmente usando la teoría de las aproximaciones, la autora encuentra una manera de expresar las coordenadas de los representantes de cada elemento del grupo criptográfico por aproximaciones finitas, y con estas últimas, da un algoritmo efectivo para el cálculo de la multiplicación puntual.

b) **(León Lomparte, 2005)[4]**, presenta la implementación de los algoritmos RSA en Java en un sistema denominado: Sistema de encriptación RSA. La autora, presenta una interfaz que permite la interacción del usuario con la generación de claves pública y privada, así como la encriptación y decriptación de archivos.

[3] Medina Aparcana, R. (2012) *Criptografía con Curvas Elípticas sobre cuerpos p-ádicos*. Lima: Universidad Nacional de Ingeniería, p. 5 – 6.

[4] León Lomparte, K. (2005) *Encriptación RSA de Archivos de Texto*. Lima: Universidad Pontificia Católica del Perú, p. 5 – 7

2.1.2 Antecedentes Internacionales.

a) **(Khoa & Zier, 2003)**[5], discuten una posible implementación del algoritmo AES, específicamente para el uso en el modo de cifrado por bloques (CCM) sobre hardware reconfigurable (FPGA). En este trabajo, se investiga la posibilidad de crear un sistema fuera del microprocesador para el algoritmo AES, de tal modo que los procesos puedan ser acelerados. Dicha implementación se realizó empleando una FPGA XILINX Spartan – II, a una frecuencia de 50 MHz.

b) **(Nazzar Abbas, 2004)**[6], siguiendo el trabajo de **(Khoa & Zier, 2003)**[2], en su tesis de grado, tiene como objetivo obtener implementaciones de algoritmos criptográficos basadas en dispositivos reconfigurables como los FPGAs, que tengan un alto desempeño sin tener que utilizar altos requerimientos de hardware. Para cumplir con tal objetivo el investigador selecciono tres algoritmos de acuerdo a su importancia en aplicaciones de seguridad: *Data Encription Standard* (DES), *Advanced Encryption Standard* (AES), ambos algoritmos de llave simétrica, y *Criptografía de curvas elípticas* como algoritmo de llave asimétrica. Los resultados de su trabajo mostraron la obtención de diseños para

[5] Khoa, V., & Zier, D. (2003) *FPGA Implementation AES for CCM Mode Encryption Using Xilinx Spartan – II*. USA: Editorial SPRING, p. 1 – 5

[6] Nazzar Abbas, S. (2004) *Implementación Eficiente de Algoritmos Criptográficos en Dispositivos de Hardware Reconfigurable*. México: Instituto Politécnico Nacional, p. 7 – 8

algoritmos criptográficos con un alto desempeño mediante un uso eficiente de los recursos de hardware.

c) Estas investigaciones, motivaron la profundización del tema deviniendo en la implementación de diversos algoritmos criptográficos sobre plataformas reconfigurables. Tales aseveraciones son demostradas en las investigaciones realizadas por **(Segredo, Zabala, & Bellora, 2004)**[7], que tienen como objetivo principal, medir la eficiencia del encriptador AES, sobre una plataforma FPGA Virtex XCV 100, logrando alcanzar una velocidad de 742 Mbps para el proceso de encriptado. Los resultados fueron comparados con implementaciones realizadas en software, notándose así, que las implementaciones en hardware eran superiores.

d) Sin embargo, con el avance de los procesadores, y a que estos últimos operan a grandes velocidades (en el orden de los Gigahertz), ha sido posible lograr velocidades comparables **(Gladman, s.f.)**[5], **(Karri & Kim, s.f.)**[6]. Finalmente los autores, concluyen que las implementaciones en hardware ofrecen mayor seguridad en mantener la privacidad de la clave, también ofrecen una instalación más sencilla, ya que no requiere emplear ningún procesador y software, y lo más

[7] Segredo, A., Zabala, E., & Bellora, G. (2004) *Diseño de Procesador Criptográfico Rijndael en FPGA*. Uruguay: Universidad ORT, p. 1 – 10

[5] Gladman, B. (s.f.) *Implementation of Rijndael in C++ and C*

[6] Karri, R., & Kim, Y. (s.f) *FPGA Implementation of AES Rijndael, Comparison of different Rijndael Implementations.*

importante, el consumo energético de una FPGA es mucho menor comparado con la de un procesador.

e) (Trejo, 2004)[7], tiene como objetivo realizar la implementación sobre una plataforma reconfigurable (FPGA) del modo CCM empleado AES, para lograr una aplicación eficiente y competitiva con las aplicaciones comerciales ya existentes. Los resultados de la investigación lograron alcanzar una velocidad de 1051 Mbps.

f) (Mali, Novak, & Biasizzo, 2005)[8], presentan la implementación en hardware del AES, para una aplicación de almacenamiento externo altamente fiable. El desarrollo fue llevado a cabo empleando una plataforma reconfigurable FPGA, demostrando asi la potencia y versatilidad de estos, para sus aplicaciones en Sistema embebidos. Los resultados logran alcanzar una velocidad de 182.86 Mbps para el proceso de cifrado y descifrado.

g) La investigación realizada por **(Badillo, Feregrino, Cumplido, & Morales, 2008)**[9], tiene como objetivo desarrollar una arquitectura basada en hardware, con alto rendimiento y eficiencia para una aplicación equilibrada, demostrando que el desarrollo sobre hardware es superior en medida a los desarrollos obtenidos en software, hacen también las comparativas a diferentes frecuencias de funcionamiento y

[7] Trejo, E. L. (2004) *Implementación eficiente en FPGA del modo CCM usando AES*. México: Instituto Politécnico Nacional, p 87 – 95

[8] Mali, M., Novak, F., & Biasizzo, A. (2005) *Hardware Implementation of AES Algorithm*. FEI STU, p. 265 - 269

[9] Badillo, I., Feregrino, C., Cumplido, R., & Morales, M. (2008) *FPGA Implementation and Performance Evaluation of AES – CCM Cores for Wireless Networks*. INAOE: México, p. 1 – 6

al número de recursos empleados (Bloques de memorias Ram, elementos lógicos y matrices lógicas), obteniendo como resultado 1.087 Gbps para una frecuencia de 86.34 MHz, y 1.951 Gbps para una frecuencia de 152.42 MHz. Con estos resultados demuestran que es posible emplear la menor cantidad de recursos sobre el hardware para obtener mejores prestaciones sobre el resultado.

h) Basado en estos antecedentes, **(Velásquez & Castaño, 2011)[10]**, desarrollan una infraestructura de clave pública, utilizando una plataforma reconfigurable FPGA. La arquitectura desarrollada se soporta en el criptosistema de curvas elípticas (ECC), además integrada por el algoritmo AES (Rijndael) para cifrado simétrico y SHA como algoritmo de integridad de la información. Las velocidades alcanzadas en este proceso llegaron hasta 660 Mbps para el cifrado y descifrado respectivamente.

Con el proyecto desarrollado por **(Trejo, 2004)[7],** y **(Velásquez & Castaño, 2011)[10]**, se pretende diseñar de una manera eficiente algoritmos criptográficos sobre plataformas reconfigurables. El aporte de estos antecedentes sustento las bases de diseño y la estrategia para poder cumplir con los objetivos planteados en el presente trabajo de investigación.

[10] Velásquez, F., & Castaño, J. (2011) *Implementaciones Criptográficas en FPGA* p.

[7] Trejo, E. L. (2004) *Implementación eficiente en FPGA del modo CCM usando AES* p.

[10] Velásquez, F., & Castaño, J. (2011) *Implementaciones Criptográficas en FPGA*. Bogotá: Visión Electrónica, p. 26 – 37

2.2. Bases teóricas

2.2.1 Criptografía.

En la denominada era de la información, la necesidad de proteger la información es cada vez más pronunciada. El intercambio de información sensible, no solo está limitada a las instituciones militares o gubernamentales, sino también al sector empresarial y privado. Ya que el intercambio de la información, sea por ejemplo una transacción bancaria, números de tarjetas de crédito a través de internet son ahora, prácticas comunes.

A medida que el mundo se va interconectando, la dependencia de los servicios electrónicos sigue en aumento. Este contexto conlleva a adoptar una postura que permita adecuar los medios de transmisión y almacenamiento para proteger la información con el fin de que esta, no sea interceptada y sufra modificaciones no autorizadas.

Un sistema de cifrado criptográfico, se hace necesario en este escenario, ya que hace posible ocultar el contenido real de la información, transformándola (Cifrado) antes de su transmisión y almacenamiento. El campo encargado de esta tarea es la criptografía.

(A. Menezes, 2001)[11], define a la *criptografía* como el estudio de las técnicas matemáticas relacionadas con aspectos de la inseguridad, tales como la confidencialidad, integridad, autenticación de la entidad, y

[11] A. Menezes, P. V. (2001) *Handbook of Applied Crytography*. Massachusetts Institute of Technology, p. 1 – 10

autenticación del origen de los datos. A continuación se describen dichos aspectos.

- **Confidencialidad**: Es un servicio que permite mantener el contenido de la información comprensible a todos los sistemas o usuarios autorizados. La confidencialidad tiene como propósito que las partes autorizadas puedan entender los datos intercambiados. La confidencialidad es una característica impuesta por el cifrado.
- **Integridad de datos**: Es un servicio que se ocupa de la modificación no autorizada de la información. Esta propiedad es atribuida a los datos que no han sido modificados, destruidos o perdidos de manera accidental o no autorizada.
- **Autenticación**: Es un servicio que se encuentra relacionado con la Identificación. Esta función se aplica a ambas entidades (emisor y receptor), y a la información en sí. La criptografía suele subdividirlo en dos clases: *Autenticación de la entidad y autenticación del origen de los datos*.
- **No repudio**: Es un servicio estrechamente relacionado con la autenticación, y que permite comprobar la participación de entidades. La diferencia esencial con la autenticación, es que la primera se produce entre las partes que establecen la comunicación y el servicio de no repudio, se produce frente a un tercero, existiendo así dos

posibilidades: *No repudio en origen* y *no repudio en el destino*. **(Costas Santos, 2010)**[12]

En la terminología de cifrado, el mensaje original se denomina *texto claro*, el cual pasará a codificarse, ocultando así al mensaje original, a este proceso se le denomina *cifrado*, para finalmente obtener el mensaje encriptado, el cual se llamara *texto cifrado*. El proceso contrario a esta operación recibe el nombre de *descifrado*. Estos procesos, por lo general hacen uso de una *llave* y un método de codificación para el cifrado y descifrado, de tal manera que el texto claro solo pueda ser descifrado conociendo la llave correcta.

La criptografía divide en dos categorías importantes este proceso: Criptografía de clave secreta y, clave pública. Ambas son métodos indispensables.

II.2.1.1. Criptografía de clave secreta

Criptografía de clave secreta o simétrica, hace uso de la misma llave para el cifrado y el descifrado. Dicha llave se mantiene en secreto y deben ser conocidos en ambos extremos (emisor y receptor).

La característica de este tipo de criptografía es la velocidad de los algoritmos para cifrar y descifrar datos con altos volúmenes. El cifrado simétrico más popular es el criptosistema DES (Data

[12] Costas Santos J. (2010) *Seguridad Informática*. Madrid : Editorial Ra – Ma, p. 19 – 29

Encrytpion Standard) que emplea 56 bits, siendo esta última reemplazada por la variante Triple DES (3DES) a manera de brindar mayor seguridad al emplear el doble número de bits efectivos. A manera de incrementar la seguridad, **(Joan & Rijmen, 2002)**[13] proponen un nuevo algoritmo de cifrado simétrico denominado "*Rijndael*", este último pronunciado por el NIST (National Institute of Standards and Technology), en octubre del 2000 como el nuevo estándar de cifrado avanzado (AES) Debido a su mejorado nivel de seguridad. **(NIST, 2001)**[14]

Los algoritmos simétricos se pueden dividir en cifrado por flujo y cifrado por bloques.

Cifrado por flujo

Es un tipo de algoritmo de cifrado simétrico, donde los datos de entrada son encriptados bit a bit (a veces en bytes), el cifrado del siguiente bit depende del estado actual. Algunos ejemplos de cifradores por flujo son: SEAL, TWOPRIME, WAKE, RC4, A5, entre otros.

Cifrado por bloques

Este tipo de cifrado simétrico, toma como entrada un bloque de longitud fija (texto plano) y lo transforma en otro bloque de la misma longitud (texto cifrado), bajo la acción de una llave secreta

[13] Joan, D., & Rijmen, V. (2002) *The Design of Rijndael, AES – The Advanced Encryption Standard*. New York: Editorial Springer, p. 1 – 7

[14] NIST. (2001) *Advanced Encryption Standard (AES).* p. 1 – 6

proporcionada por el usuario. El proceso de descifrado se realiza aplicando ingeniería inversa al bloque de texto cifrado empleando la misma llave. Los cifrados por bloque modernos suelen emplear una longitud de bloque de 128 bits. La longitud de bloque oficial para *Rijndael* es de 128 bits, sin embargo el algoritmo puede procesar también bloques de tamaño de 192 y 256 bits.

Algunos ejemplos de cifradores por bloques son: Serpent, RC6, MARS, IDEA, Twofish, entre otros.

II.2.1.2. Criptografía de clave pública

A diferencia de la criptografía de clave secreta, la criptografía de clave pública, emplea algoritmos asimétricos con una clave diferente para el proceso de cifrado y descifrado, la clave de descifrado no se deriva fácilmente de la clave de cifrado.

La clave pública se encuentra disponible para todos en el extremo del emisor, sin embargo una clave privada o secreta es conocida solo por el receptor del mensaje. Un elemento importante al sistema de claves públicas y privadas, es que estas, están relacionadas de tal manera que solo la clave pública se utiliza para cifrar los mensajes, y solo la clave privada correspondiente sirve para descifrarla. Los algoritmos de clave pública más populares son RSA (basado en la factorización de números largos), ElGamal (basado en el problema del logaritmo discreto) y McEliece (basado en código de corrección de errores).

2.2.2. Tecnología inalámbrica

La comunicación inalambrica es el proceso de transmitir información en un medio electromagnetico a distancia a través del aire, en lugar de hacerlo por medio de cables u otro conducto físico. **(Nichols & Lekkas, 2002)** [15]

Una red inalambrica es una extensión a las redes alambricas para brindar movilidad haciendo uso de la tecnología de radio frecuencia.

Podemos dividir a las redes inalambricas en dos grandes grupos: Las de corto alcance y largo alcance. En la redes de corto alcance podemos situar a las redes de area local (*LAN*), estas últimas se encuentran limitadas a extensiones pequeñas, por ejemplo hogares, oficinas, escuelas, a ellas tambien se unen las redes *PAN* (*Personal Area Network*), conformadas por computadoras interconectadas a distancias muy cortas.

Las redes inalambricas descritas anteriormente operan en espectros de frecuencia que no se encuentran bajo licencia y que estan reservados para su uso Industrial, Cientifico y Medico (*ISM*). Las frecuencias disponobles para esta banda varian en cada País, siendo las más utilizadas la banda de 2.4 GHz, 5.7 GHz y 40 GHz, estas frecuencias permiten el uso de las redes inalambricas sin recurrir a pagos por derechos de uso de frecuencias.

[15] Nichols, K. & Lekkas, C. (2002) *Wireless Security Models, Threats and Solutions*. USA: Editorial McGraw – Hill, p. 1

Por otra parte, las redes de largo alcance estan diseñadas para proveer una acceso a la información en zonas geograficamente mas alejadas, abarcando ciudades, paises y continentes. Estas redes de largo alcance son denominadas Redes Inalambricas de Áreas Amplias (*WWAN*). Un ejemplo de este tipo de redes lo conforma los enlaces satelitales, ya que estos son capaces de cubrir extensas zonas geograficas.

La diversisdad de tecnologías para las redes inalambricas abarca un amplio campo de estudio. La presente tesis se orienta a la tecnologia inalambrica WLAN.

II.2.2.1. Tecnología Inalámbrica WLAN

(Andreu, Pellejero, & Lesta, 2006)[16], argumenta que las redes WLAN, se han convertido en un elemento clave en el aumento de la productividad de la empresa, ofreciendo ventajas como la ubicuidad y flexibilidad.

Este tipo de redes es la que mayor crecimiento ha tenido durante los últimos años. La adopción de estándares por parte de la industria, y el correspondiente desarrollo de estas redes, ha derivado en la implementación de soluciones WLAN en muchos de los segmentos de los mercados, incluyendo pequeñas oficinas, hogares, grandes empresas, plantas de manufactura, lugares públicos como aeropuertos, centros comerciales. **(Mallik,**

[16] Andreu, F., Pellejero, I., & Lesta, A. (2006) *Redes WLAN Fundamentos y Aplicaciones de Seguridad*. Barcelona: Editorial Marcombo, p. 1 – 4

2003)[17]. La adopción de esta red inalambrica ha conllevado al desarrollo de multiples estandares, la presente tesis aborda el estandar IEEE 802.11, por ser este último, el mas utilizado. Es importante conocer sus especificaciones y la seguridad que brindan.

En la Tabla 2.1, se pueden pareciar las redes y coberturas de los servicios inalambricos.

Estándar IEEE 802.11

Aprobado en el año 1997, fue el primer estándar para la red inalámbrica WLAN, emplea los mismos protocolos que una red Ethernet y permite la comunicación sin cables empleando el espectro electromagnético.

Dentro del estándar IEEE 802.11, podemos encontrar los siguientes: **802.11b, 802.11g, 802.11n**, las cuales operan sobre la banda de los 2.4 GHz. El más utilizado es el **802.11b (Budris, 2011)[18]**, capaz de transmitir a velocidades de hasta 11 Mbits/s. Este estandar soporta hasta 32 usuarios por access point y hasta tres canales simultaneos en la misma ubicación.

El estandar **802.11g**, es mas rápido ya que soporta velocidades teoricas de hasta 54 Mbits/s, ofrecen compatibilidad con el estandar **802.11b**. El estandar mas reciente **802.11n** (trabajan en

[17] Mallik, M. (2003) *Mobile and Wireless Design Essentials*. Editorial WILEY, p. 3 – 5

[18] Budris, P. (2011) *Administrador de Redes Windows*. USERS, p. 31 – 33

las bandas de 2.4 GHz y 5 GHz)**,** fue ratificado en el año 2009, con velocidades reales de hasta 600Mbits/s. Por otra parte, el nuevo estandar **802.11ac**, que opera en la banda de los 5 GHz, ofrece velocidades superiores a 1Gbit/s. **(Perahia & Stacey, 2013)**[19]

Tabla 2.1. Redes y coberturas de servicios inalámbricos.

Red	**Cobertura**	**Función**	**Costo**	**Velocidad**	**Tecnologías**
WPAN (Red de área Personal)	Inferior a 10 m	Reemplazo de cables	Muy bajo	0.1 – 4 Mbps	IrDA Bluetooth 802.15
WLAN (Red inalámbrica de área local)	Inferiores A 100m	Alternativa De LANs	Medio	1 – 600Mbps	802.11b,g,n
Red	**Cobertura**	**Función**	**Costo**	**Velocidad**	**Tecnologías**
WWAN (Red Inalámbrica de amplia cobertura)	Cobertura Amplia	Extensión De LANs	Alto	8Kbps – 2 Mbps	GSM TDMA CDMA GPRS EDGE WCDMA
Redes Satelitales	Cobertura global	Extensión de LANs	Muy alto	2Kbps – 2Mbps	TDMA FDMA CDMA

Nota. Recuperado de "*Mobile and Wireless Design Essentials"* de Mallik, M.,2003: WILEY.

[19] Perahia, E., & Stacey, R. (2013) *Next Generation Wireless LANs 802.11n and 8021.ac*. United Kingdom: Editorial Cambridge University Press, p. 17

Como se había descrito anteriormente, las redes inalámbricas son susceptibles a ataques, y estos pueden agruparse en dos grandes grupos: Activos y pasivos.

Los ataques pasivos ocurren cuando un usuario no autorizado obtiene acceso al canal de comunicación, logrando así el robo de información en su totalidad o partes. Este tipo de ataques es muy común debido al modo de transmisión de la información, puesto que en teoría cualquiera que tenga los receptores adecuados, y este dentro del rango de transmisión puede acceder a la información.

Por otro lado los ataque activos, ocurren cuando el intruso modifica maliciosamente los datos transmitidos, ocasionando así ataques que pueden impedir que la red se comunique, este ataque es conocido como "*Denegación de Servicio*".

Para frenar estos ataques, es necesario contar con esquemas de protección que garanticen la integridad de la información. A continuación se presentan algunos esquemas empleados en la actualidad para garantizar el intercambio de información en las redes inalámbricas.

Protocolos de seguridad

La seguridad en cualquier red, incluidas las WLAN, pueden ser comprometidas en dos aspectos: **autenticación y cifrado**. Los mecanismos de autenticación se emplean para identificar un

usuario inalámbrico ante un punto de acceso y viceversa, mientras que los mecanismos cifrados, aseguran que no sea posible decodificar el tráfico de usuario. (**Andreu, Pellejero, & Lesta, 2006)**[16]

Protocolo de seguridad WEP

WEP (Wired Equivalent Privacy), es un protocolo de cifrado a nivel de enlace, contenido en las especificación del Estándar original IEEE 802.11.

Cuando la IEEE crea la especificación 802.11, implementa también WEP, con el intento de proveer los mecanismos de seguridad de autenticación y cifrado de datos, esto era necesario debido a que las redes inalámbricas no contaban con la protección física de las redes alámbricas.

El protocolo WEP, fue concebido inicialmente para brindar seguridad equivalente a puntos de acceso cableados, inicialmente fue diseñado para una llave de 40 bits, desarrollándose posteriormente WEP2 incrementándose la longitud de la llave a 104 bits. **(Erickson, 2003)**[20]

[16] Andreu, F., Pellejero, I., & Lesta, A. (2006) *Redes WLAN Fundamentos y Aplicaciones de Seguridad*. Barcelona: Editorial Marcombo, p. 45

[20] Erickson, J. (2003) *HACKING the art of exploitation*. San Francisco: Editorial No Starch Press, p. 434

A continuación, se explica el modo de funcionamiento de este protocolo:

El cifrado se realiza en base a paquetes, cada paquete es esencia un mensaje (texto plano) a enviar.

Denominemos *M* al mensaje a enviar, calculándose la suma de verificación (*checksum*) para que la integridad de dicho mensaje pueda ser verificada posteriormente. El *Checksum*, se calcula empleando un código de redundancia cíclica (CRC, por sus siglas en ingles), de 32 bits, a la cual se le denomina **CRC32**. La suma de verificación se denominará CS, entonces:

CS = CRC32 (M). Este resultado es agregado al final del mensaje, lo que forma el texto claro *P*.

La idea básica de un algoritmo CRC consiste en tratar al mensaje como un numero binario muy grande, y dividirlo con otro número binario constante (polinomio generador), al residuo de esta división se le realiza la suma de verificación. El receptor puede realizar la misma división y comparar el residuo con la suma de verificación recibida.

Una vez realizado el cálculo del CRC, el siguiente paso es cifrar *P*, empleando el *RC4*, que es un cifrador por flujo de datos. Este cifrador inicializa con un valor denominado *semilla*, que genera un flujo de llaves, el cual es una cadena de longitud arbitraria de bytes pseudoaleatorios, WEP emplea un vector de inicialización

(*IV*), para el valor de la semilla. El *IV* consiste en una cadena de 24 bits que es generada para cada paquete.

Independiente del valor elegido del *IV*, este es agregado al inicio del al llave del WEP. Los 24 bits de *IV* son incluidos. Es decir, cuando nos referimos a una llave de 64 o 128 bits, en realidad los tamaños son 40 y 104 bits respectivamente con los 24 bits de *IV*. La unión del *IV* y la llave WEP, forman una semilla la cual se denomina *S*.

Tabla 2.2. Composición de la semilla *S*.

IV (24 bits)	Llave (40 y 104 bits)

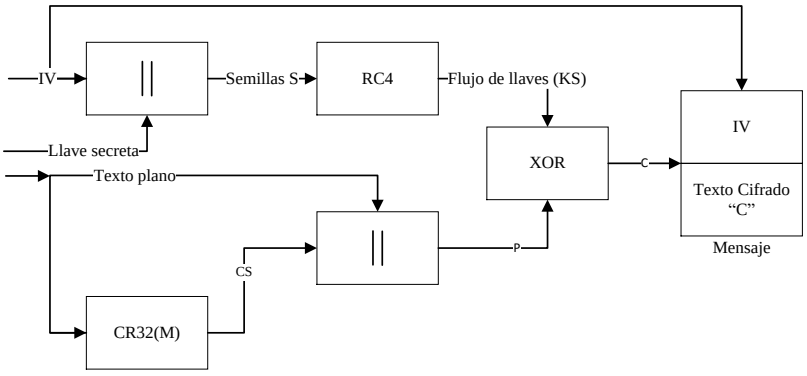
Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 16, México: IPN.

La semilla generada se introduce al *RC4*, el cual genera el flujo de llaves (*KS*). Este flujo se somete a la operación XOR con el texto plano *P*, para así producir el texto cifrado *C*, finalmente se le agrega a *C* el *IV*, y este resultado es enviado a través de la red inalámbrica hacia el receptor.

Por otra parte, en el lado del receptor, para la recuperación de un mensaje cifrado con WEP, se realiza el proceso invertido. Esto se logra separando el *IV* (vector de inicialización), del mensaje y se concatena a la llave *K* para producir la semilla *S*. En este punto si tanto el emisor como receptor tiene la misma semilla, este se

ingresa al *RC4* para producir el mismo flujo de llaves, el cual mediante la operación XOR con el resto de mensaje cifrado producirá el texto claro original, el cual es el mensaje *M* concatenado con *CS*. Finalmente, el receptor calcula CS_t = *CRC32 (M)*, si $CS_t = CS$, se da conformidad del mensaje recibido y este es aceptado, caso contrario el mensaje sufrió modificaciones durante la transmisión.

Figura 2.1. Proceso de cifrado del protocolo WEP.



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 17, México: IPN.

II.2.2.2. Propuestas de seguridad en redes inalámbricas

Como se analizó, WEP, no es un protocolo seguro, ha sido necesario el desarrollo de nuevos métodos que permitan salvaguardar la información que es transmitida por las redes inalámbricas. Estos esquemas de seguridad se encuentran agrupados en el estándar IEEE 802.11i.

Acceso protegido Wi – Fi (WPA)

La principal diferencia en cuanto al cifrado entre WEP y WAP radica en la gestion de las llaves y en el vector de inicialización.

WPA pretende resolver las limitaciones de WEP, mediante el empleo del protocolo **TKIP** (Temporal Key Integrity Protocol). Esto garantiza un vector de inicialización ampliado (doble tamaño que en WEP, 48 bits) con reglas de secuencia y la mezcla de dicho vector de inicialización por paquete, hace que sea mucho más robusta frente a ataques de las redes WLAN **(Andreu, Pellejero, & Lesta, 2006)**[16].

El empleo de este nuevo protocolo conlleva a una actualización de firmware en algunas tarjetas. Los principales requerimientos son descritos en **(Cam-Winget, Housley, Wagner, & Walker, 2006)**[21], estos se detallan a continuación.

- Los sistemas desarrollados deben ser actualizables en software o firmware.

- Impedir que la implementación WEP actual sea modificada.d

- El rendimiento no debe verse afectado por las modificaiones realizadas.

[16] Andreu, F., Pellejero, I., & Lesta, A. (2006) *Redes WLAN Fundamentos y Aplicaciones de Seguridad*. Barcelona: Editorial Marcombo, p. 100 – 101

[21] Cam-Winget, N., Housley, R.,Wagner, D.,& Walker, J. (2006) *Communications of the ACM*, p. 35-39

La adición de una función hash en WPA permite defenderse del ataque FMS **(Fluhrer, Mantin, & Shamir, 2001)**[22], en unión a un código de integridad de mensaje (*MIC*) y un manejador de llaves basado en el 802.11X, que impide la reutilización de llaves y su distribución. **(Vebjorn, Havard, & Hole, 2004)**[23]. En la Figura 2.2, se aprecia el proceso de encapsulamiento.

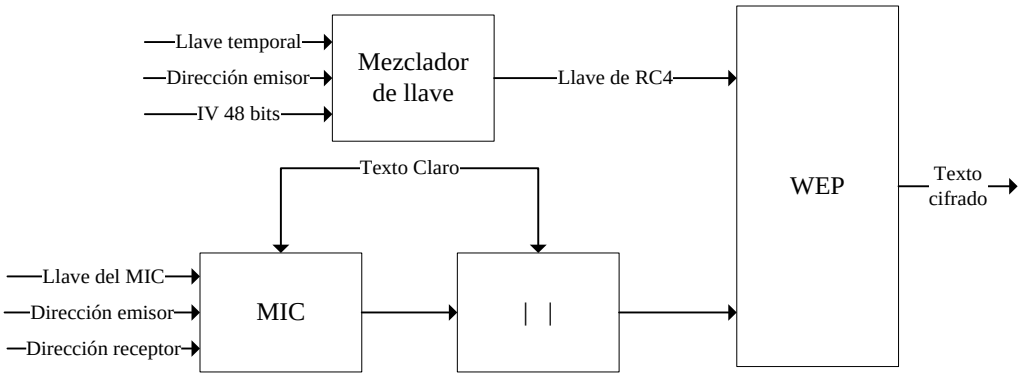
La llave temporal (TK) de 16 bytes es obtenida del esquema de manejo de llave durante la autenticación, y es introducida a la función hash junto con la dirección (6 bytes), del emisor (TA) y los 48 bits del *IV*, frecuentemenete llamado contador de secuencia del TKIP. La función hash proporciona una llave para el RC4 de 16 bytes donde los tres primeros bytes son derivados del *IV*. La llave es utilizada solo para una trama del WEP, debido a que el *IV* es implementado como un contador que se incrementa con cada paquete, por lo que la llave es tambien utilizada como una llave por paquete. El *IV* es tambien utilizado como una defensa contra los ataques de reenvio, por lo que el receptor no aceptará paquetes con un valor de *IV* menor o igual a los recibidos con anterioridad. **(Vebjorn, Havard, & Hole, 2004)[23].**

[22] Fluhrer, S., Mantin, I., & Shamir, A. (2001) *Weaknesses in the key Scheduling Algorithm of the RC4*. p. 1 – 2

[23] Vebjorn, Havard, & Hole (2004) *Weaknesses in the temporal key hash of WPA*. P. 76 – 83

[23] Vebjorn, Havard, & Hole (2004) *Weaknesses in the temporal key hash of WPA*. P. 76 – 83

Figura 2.2. Encapsulamiento WPA.



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 22, México: IPN.

Como se menciono, WPA requiere una actualización en el firmware en la tarjeta de red (considerando una expansión de WEP).

Según **(Cam-Winget, Housley, Wagner, & Walker, 2006)**[21] Las deficiencias de WPA radican en el empleo de dos llaves, una de 128 bits empleada por el mezclador de llave para la genaración de la llave de cifrado, y la otra de 64 bits utilizada por el *MIC*.

EL CCMP

CCMP, son la siglas de *Counter – Mode – CBC – MAC Protocol*. Del mismo modo que el WPA, fue pensado para sustituir al WEP, teniendo la ventaja de ser implementado sin considerar el hardware existente. Para tal fin, el Estándar de Encriptación

[21] Nancy Cam-Winget, R. Housley, D. Wagner, J. Walker *Communications of the ACM* p. 35-39

Avanzado (*AES*), ha sido elegido como el algoritmo para el cifrado.

El CCM, diseñado por **(Doug, Housley, & Ferguson, 2002)**[24], esta concebido para cumplir con los siguientes requisitos:

- Proporcionar proteccion de la integridad al encabezado y contenido de la información.
- Permitir la paralelización para incrementar el rendimiento.
- Implementaciones reducidas en tamaño.
- Evitar los modos de operación que esten bajos derechos de autor.

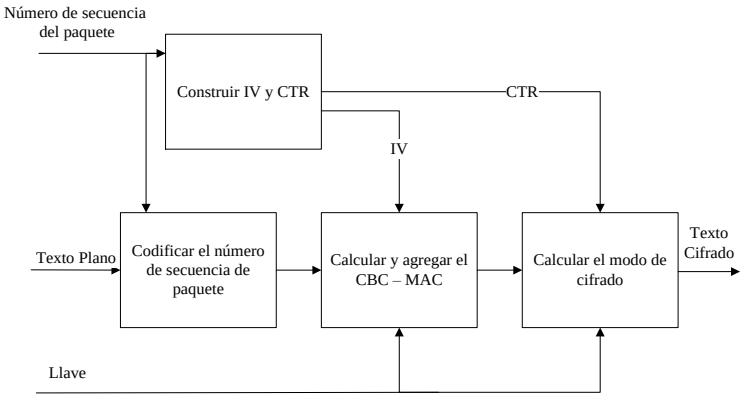
El CCM, emplea el conteo (*CTR*) para el proceso de cifrado y el CBC – MAC para la autenticación. En ambos casos utilizan unicamente la primitiva de cifrado del AES en el emisor y receptor. El empleo de la misma llave tanto para la confidencialidad como para la integridad, es inseguro, por tal motivo el CCM lo evita, al garantizar que el espacio de modo de conteo jamas se mezcle con el vector de inicialización del CBC – MAC. Al respecto, **(Cam-Winget, Housley, Wagner, & Walker, 2006)**[21], indica que la intuición del CCM, es que si el AES se comporta como una permutación pseudoaleatoria,

[24] Doug, W., Housley, R., & Ferguson, N. (2002) *Counter with CBC – MAC (CCM)* p. 2

[21] Nancy Cam-Winget, R. Housley, D. Wagner, J. Walker *Communications of the ACM* p. 35 – 39

entonces la salida del cifrador en cada uno de los modos deberá ser independiente. La Figura 2.3, muestra el proceso del CCMP.

Figura 2.3. Proceso del CCMP.



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 23, México: IPN.

En la tabla 2.3, se muestra la comparativa del WEP con otros protocolos de seguridad.

Tabla 2.3. Diferencias en los esquemas de seguridad IEEE 802.11

	WEP	**WPA**	**CCM**
Cifrado tamaño de llave	RC4 40 ó 104 bits para el cifrado	RC4 128 bits para cifrado de y 64 bits para autenticación	AES 128 bits
Vida útil de la llave	24 bits de protección en el IV	48 bits de IV	48 bits de IV
Llave por paquete	Concatenación IV con la llave base	Función de mezclado del TKIP	No se necesita

	WEP	WPA	CCM
Integridad del encabezado del paquete	Ninguna	Dirección origen y destino protegidas por la función hash	CCM
Función de integridad	CRC32	Función hash	CCM
Detección ataques reenvió	Ninguna	Secuencia del IV	Secuencia del IV
Maneja llaves	Ninguna	IEEE 802.1X	IEEE 802.1X

Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 24, México: IPN.

2.2.3. Seguridad en hardware reconfigurable.

(Wollinger & Christof, 2003)[25], indica que el diseño de un sistema digital está condicionada a muchos criterios, y en gran parte al área de aplicación. Así mismo los aspectos referentes al algoritmo, velocidad y costos asociados debemos de tener en cuenta aspectos como la seguridad física (recuperación de la llave y manipulación del algoritmo), flexibilidad (sin importar los parámetros del algoritmo).

Las grandes ventajas que ofrecen las FPGAs, son la propiedad de reconfiguración que estos poseen cuando son usados en aplicaciones criptográficas. Pese a que existen diversos trabajos sobre

[25] Wollinger, T., & Christof, P. (2003) *How Secure Are FPGAs in Cryptographic Aplications* p. 91 - 100

implementaciones criptograficas sobre esta plataformas, pocos estudios se han centrado en la seguridad que estos brindan.

La resistencia de los FPGAs a ataques físicos o del sistema, son mas peligrosos que los ataques a sus porpios algoritmos, por tal motivo las FPGAs no son completamente seguras, pero brindan niveles aceptables de seguridad cuando son empleadas en infraestructuras adecuadas. **(Wollinger, Guajaro, & Paar, Security on FPGAs: State - of - the - art implementations and attacks , 2004)**[26].

2.2.4. Computación reconfigurable.

Al respecto, **(Moriello, 2001)**[27], define a la computación reconfigurable como una nueva idea en la filosofía de la computación, en la cual algunos dispositivos electrónicos de propósito general se diseñan para llevar a cabo una serie de tareas específicas, pero pueden ser reconfigurados por el usuario para ejecutar otras tareas, indica también que estas máquinas se acercarán más a los humanos.

Los investigadores americanos **(Villasenor & Mangione-Smith, 1997)**[28], de la universidad de California, realizaron un análisis de las opciones al momento de realizar los diseños, encontrando dos opciones: los circuitos integrados de propósito general, que son dispositivos versátiles y económicos pero a su vez lentos, y los circuitos integrados

[26] (Wollinger, Guajaro, & Paar, Security on FPGAs: State - of - the - art implementations and attacks , 2004) *ACM Transactions on Embedded Computing Systems (TECS)*. p. 534 – 574

[27] Moriello, S. (2001) *Inteligencias Sintéticas*. Buenos Aires: Editorial ALSINA, p. 32

[28] Villasenor, J., & Mangione – Smith, W. (1997) *Configurable Computing* P. 55 – 59

de propósito específico, que son dispositivos mucho más rápidos pero más costosos y se encuentran diseñados específicamente para realizar una tarea precisa y concreta. Bajo esta perspectiva, es posible producir un circuito integrado más diminuto, de mayor velocidad y menor consumo que un microprocesador convencional; la única limitación pese a la de ser costoso, radica en la de ser incapaz de realizar otra tarea distinta de aquella para la cual fue diseñado.

Estas limitaciones, permitieron a los investigadores tomar ambas características de las dos opciones, planteando así, una tercera alternativa, y esta es: "*el conjunto de compuertas programables en campo (FPGA)".*

II. 2.4.1. Compuertas programables en campo (FPGA).

(Nichols & Lekkas, 2002)[15], se refieren al hardware reconfigurable, al tipo de circuitos integrados mayoritariamente conocidos como FPGAs, hacen comparaciones con otros dispositivos como los Dispositivos de Lógica Programable (PLDs por sus siglas en ingles) que se aproximan a la misma definición pero no a las mismas capacidades. Los investigadores, indican que este tipo de dispositivos son también reconfigurados por el diseñador. Cada nueva configuración es realizada nuevamente en

[15] Nichols, K. & Lekkas, C. (2002) *Wireless Security Models, Threats and Solutions*: Editorial McGraw – Hill, p.245 – 247

fracciones de segundo y con esto se logra que la FPGA sea capaz de llevar a cabo una función totalmente nueva.

(Nazzar Abbas, 2004)[3], define a una FPGA como un circuito integrado que pertenece a la clase de dispositivos programables, el cual consiste en miles de bloques básicos, denominados Bloques Lógicos de Configuración (CLBs por sus siglas en ingles), los cuales están conectados a través de interconexiones programables.

En cuanto a nuestro tema de trabajo, las bondades de las FPGAs son enormes. Al respecto **(Nazzar Abbas, 2004)**[3], menciona las ventajas que tienen los FPGAs para los algoritmos criptográficos:

- Fácil adaptación a cifrados simétricos que contienen operaciones bit a bit, para ser adaptados a la estructura CLB del FPGA.

- El carácter iterativo de los algoritmos criptográficos, permite generar bucles iterativos (Iterative loop IL). En una red de alta velocidad, en lugar de implementar una ronda, *n* rondas son replicadas y se disponen los registros entre estas, para controlar el flujo de datos, esta última operación recibe el nombre de *bucle desenrollado* o *tubería* (pipeline).

[3] Nazzar Abbas, S. (2004) *Implementación Eficiente de Algoritmos Criptográficos en Dispositivos de Hardware Reconfigurable*. México: Instituto Politécnico Nacional, p. 9

[3] Nazzar Abbas, S. (2004) *Implementación Eficiente de Algoritmos Criptográficos en Dispositivos de Hardware Reconfigurable* México: Instituto Politécnico Nacional , p. 10 – 11

- La lógica provista por los FPGA, son útiles para estas estrategias de diseño, debido a la alta velocidad y densidad del dispositivo.

- La sustitución, es una de las operaciones más fundamentales en la mayoría del cifrado por bloques, tal como DES o Rijndael, esto ocasiona un excesivo consumo de memoria. Gracias a las modernas familias de FPGA, y a que estas vienen equipadas con 280 o más BRAM (block rams) de 4K cada una, y un doble puerto BRAM, que puede ser configurado como un solo puerto BRAM con datos independientes y de rápido acceso, es decir, lectura y escritura, pueden ser ejecutadas independientemente en cada puerto.

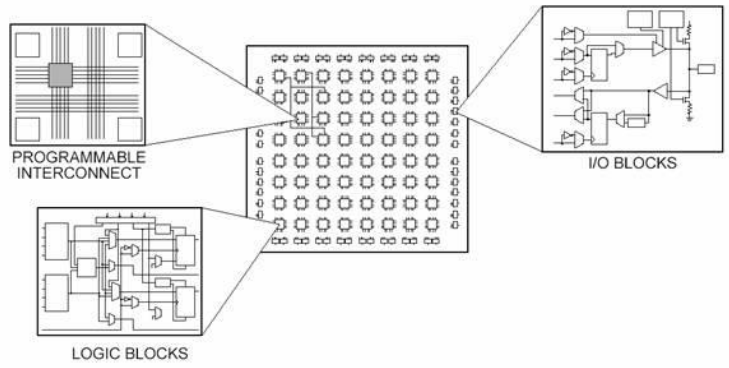
- Los FPGAs, son ideales para la depuración de su diseño, especialmente si estas se sintetizan.

- El empleo de diferentes algoritmos criptográficos para aplicaciones semejantes, suelen tener problemas de compatibilidad. Una configuración dinámica para cualquier algoritmo criptográfico en una FPGA, es una solución práctica a este problema.

- Facilidad de integración en la plataforma, y la modificación directa en la arquitectura, son ventajas valiosas de las plataformas FPGA. La Figura 2.4, muestra la organización

de los bloques de configuración lógica (CLBs), las interconexiones programables y los bloques de E/S de datos **(National Instruments, 2011)**[29].

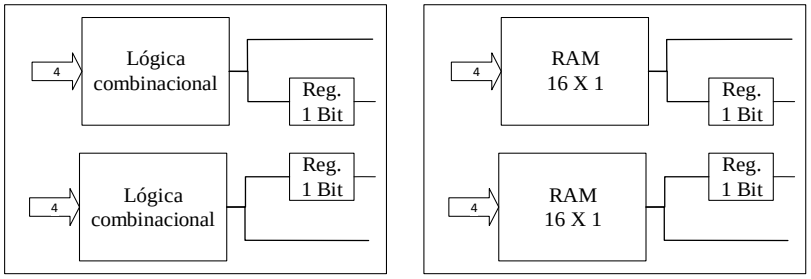
Figura 2.4. Estructura interna de un FPGA.



Nota. Recuperado de *"FPGAs a Fondo"*, (12, 03, 2015) Recuperado de http:// http://www.ni.com/white-paper/6983/es/

Es importante mencionar que un CLB, puede ser configurado de dos modos: Modo lógico o modo de memoria. La Figura 2.5, muestra las dos configuraciones de los CLBs.

Figura 2.5. Modo lógico (izquierda) y modo memoria (derecha)



Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 10, México: IPN.

[29] National Instruments (2012) *FPGAS a fondo*. p. 1

El funcionamiento y cantidad de CLBs, se encuentran definidos por las especificaciones de los fabricantes.

Entre los fabricantes más importantes de FPGA tenemos: Virtex, Spartan, Altera, Actel, Quicklogic, Logic Cell Array y Lucent.

Ventajas y/o desventajas del hardware reconfigurable.

El uso de hardware reconfigurable para algoritmos criptográficos, presentan grandes beneficios, entre ellos podemos mencionar:

- Fácil adaptación a cifrados simétricos que contienen operaciones bit a bit, para ser adaptados a la estructura CLB del FPGA.

- El carácter iterativo de los algoritmos criptográficos, permite generar bucles iterativos (Iterative loop IL). En una red de alta velocidad, en lugar de implementar una ronda, *n* rondas son replicadas y se disponen los registros entre estas, para controlar el flujo de datos, esta última operación recibe el nombre de *bucle desenrollado* o *tubería* (pipeline).

  La lógica provista por los FPGA, son útiles para estas estrategias de diseño, debido a la alta velocidad y densidad del dispositivo.

- La sustitución, es una de las operaciones más fundamentales en la mayoría del cifrado por bloques, tal como DES o Rijndael, esto ocasiona un excesivo consumo de memoria. Gracias a las modernas familias de FPGA, y a que estas vienen equipadas con 280 o más BRAM (block rams) de 4K cada una, y un doble

puerto BRAM, que puede ser configurado como un solo puerto BRAM con datos independientes y de rápido acceso, es decir, lectura y escritura, pueden ser ejecutadas independientemente en cada puerto.

- Los FPGAs, son ideales para la depuración de su diseño, especialmente si estas se sintetizan.

- El empleo de diferentes algoritmos criptográficos para aplicaciones semejantes, suelen tener problemas de compatibilidad. Una configuración dinámica para cualquier algoritmo criptográfico en un FPGA, es una solución práctica a este problema.

- Facilidad de integración en la plataforma, y la modificación directa en la arquitectura, son ventajas valiosas de las plataformas FPGA.

Frente a todas estas ventajas, las FPGAs, son aún difíciles de emplear en dispositivos de bajo costo, tales como teléfonos móviles, asistentes digitales, equipos de localización, etc, debido a su tamaño físico y al alto consumo energético. Sin embargo modelos recientes, pueden ser integrados fácilmente en sistemas de comunicación de altas prestaciones, tales como: analizadores de espectro, transmisores, receptores, repetidores, y equipos médicos de alta complejidad.

2.2.5. Operaciones fundamentales con algoritmos criptográficos.

Los algoritmos de clave secreta o simétrica, poseen los bien conocidos principios matemáticos y criptográficos, es posible realizar permutaciones, sustituciones, rotaciones, operación XOR bit a bit. Estas características hacen que sea un cifrado rápido.

Por su parte, los algoritmos de clave pública o asimétrica, están basados en problemas matemáticos más difíciles de resolver, estos son la suma y sustracción modular, multiplicación modular, rotación de longitud variable, etc. Estas propiedades dan lugar a que este algoritmo sea fuerte, pero más difícil de realizar en la práctica, ocupando mayor espacio y tiempo de procesamiento. Este motivo ocasiona que el algoritmo no sea empleado para archivos de gran tamaño, más bien, son empleados para operaciones criptográficas importantes como: intercambio de llaves, firmas, verificación, etc.

En la Tabla 2.4, se muestra un resumen de las operaciones realizadas en varios algoritmos criptográficos (Nazzar Abbas, 2004)[3]**.** Como se puede observar en la Tabla 2.4, la mayoría de los algoritmos criptográficos, poseen la propiedad de operaciones bit a bit como XOR, AND / OR. Estas características proporcionan una lógica simple para su implementación sobre plataformas de hardware.

[3] Nazzar Abbas, S. (2004) *Implementación Eficiente de Algoritmos Criptográficos en Dispositivos de Hardware Reconfigurable*. México: Instituto Politécnico Nacional, p. 5

Otra característica importante en los algoritmos criptográficos es la longitud de palabra, esta última, recomendada por diversas normas internacionales con el fin evitar posibles ataques por fuerza bruta.

Tabla 2.4. Operaciones de algoritmos criptográficos simétricos.

Adición o sustracción modular	Blowfish, CAST, FEAL, GOST, IDEA, WAKE, RC4, RC5, RC6, TEA, SAFER K-64, Twofish, SEAL, TWOPRIME.
XOR bit a bit	Blowfish, CAST, DEAL, TWOPRIME, FEAL, A5, IDEA, GHOST, RC4, RC5, SAFER, SEAL, Twofish, DES, WAKE, LOKI97, LOKI91, Rijndael, MISTY TEA, MMB, RC6, K-64
AND / OR bit a bit	MISTY
Rotaciones de longitud variable	CAST, Madryga, RC5, RC6
Rotaciones de longitud fija	DEAL, DES, CAST, FEAL, GOST, Serpent, RC6 Twofish.
Multiplicación modular	CAST, IDEA, RC6, MMB, Rijndael.
Sustitución	Blowfish, DEAL, DES, LOKI91, LOKI97, Twofish, Rijndael.
Permutación	DEAL, DES, ICE, LOKI91, LOKI97
Turnos no circulares	Serpent, TEA

Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 5, México: IPN.

En **(NIST, 2001)**[14], se recomienda el uso de una longitud de palabra como mínimo de 80 bits.

II.2.5.1. Aplicaciones criptográficas importantes

En la actualidad, son muchas las empresas que venden productos de seguridad empleando algoritmos criptográficos. Estos productos vienen siendo empleados por organizaciones militares o gubernamentales, jugando así un papel importante en las comunicaciones seguras. Las aplicaciones potenciales para los algoritmos criptográficos pueden ser clasificadas en dos categorías **(Nichols & Lekkas, 2002)**[15].

1. Procesamiento de gran volumen de datos en tiempo real y potencialmente a altas velocidades. Ejemplos: Conversaciones telefónicas, telemetría, video conferencias, transmisiones de audio/video codificado, entre otros.
2. Procesamiento de bajo volumen de datos en tiempo real y moderadamente a altas velocidades. Ejemplos: comercio electrónico, transacciones de comercio móvil, transmisión de números de tarjetas de crédito, cuentas bancarias, pagos electrónicos, y micro-buscadores basados en WAP, entre otros.

[14] NIST. (2001) *Advanced Encryption Standard (AES)*

[15] Nichols, K. & Lekkas, C. (2002) *Wireless Security Models, Threats and Solutions*: Editorial McGraw – Hill, p. 243 – 247

En la Tabla 2.5, muestra un breve resumen de las aplicaciones y velocidades de transferencia haciendo uso de una línea VDLS (Línea de abonado digital de alta velocidad) **(ETSI Technical Specification Access transmission systems on metallic access, 2015)**[30] de la categoría 1. La Tabla 2.5, muestra que para aplicaciones de alta demanda (gran volumen de información), el flujo de datos oscila entre 384 Kbps – 24 Mbps para la subida (del servidor de red hacia el usuario), y 64 Kbps – 3 Mbps, para la descarga (del usuario hacia las instalaciones del servidor).

Tabla 2.5. Algunas aplicaciones criptográficas importantes.

Aplicación	**Upstream**	**Downstream**
Educación a distancia	384Kbps-1.5Mbps	384Kbps-1.5Mbps
Teletrabajo	1.5Mbps-3.0Mbps	1.5Mbps-3Mbps
TV Digital	6.0Mbps-24.0Mbps	64Kbps-640Kbps
Acceso a internet	400Kbps-1.4Mbps	128Kbps-640Kbps
Alojamiento Web	400Kbps-1.5Mbps	400Kbps-1.5Mbps
Video conferencia	384Kbps-1.5Mbps	384Kbps-1.5Mbps
Video en demanda	6.0Mbps-18Mbps	64Kbps-128Kbps
Video interactivo	1.5Mbps-6.0Mbps	128Kbps-1.5Mbps
Telemedicina	6.0Mbps	384Kbps-1.5Mbps
TV de alta definición	16Mbps	64Kbps

Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 7, México: IPN.

El proceso de implementar dichas aplicaciones, independientemente si se emplea software con algoritmos criptográficos, o bien software

[30] ETSI Technical Specification Access Transmission Systems on Metallic Access (2015)

sobre procesadores embebidos de alta velocidad, las limitaciones recaen sobre el intervalo de 400 Kbps, ya que los procesadores de propósito general, no son capaces de generar ganancias a tales frecuencias para los algoritmos criptográficos. Sin embargo haciendo uso de plataformas de hardware (FPGAs), es posible lograr con facilidad velocidades superiores a 400 Kbps, entre estas plataformas tenemos a los conocidos ASICs (circuitos integrados de aplicación específica) y los FPGA (arreglo de compuertas programables en campo). Con estas plataformas es posible lograr factores de rendimiento en el orden de Gbps.

2.2.6. Implementación de algoritmos criptográficos.

El enfoque de implementación de los algoritmos criptográficos, basan su desarrollo sobre dos criterios: velocidad y costos. **(Nazzar Abbas, 2004)**[3]. En tal sentido, el gran reto del diseño radicará en la tasa de flujo de los datos, ya que se espera que esta, se realice en el menor tiempo (velocidad) y con recursos computacionales limitados (costos). En una red de alta velocidad, donde se va a procesar gran flujo de datos en tiempo real, no supone un buen candidato para el software, en tal sentido, entra a tallar el hardware como una alternativa frente a este problema. Soluciones en hardware basadas en VLSI (integración a escala muy grande), permiten obtener altas tasas de velocidad, pero conlleva demasiado tiempo el desarrollo de la aplicación. Cualquier

[3] Nazzar Abbas, S. (2004) *Implementación Eficiente de Algoritmos Criptográficos en Dispositivos de Hardware Reconfigurable*. México: Instituto Politécnico Nacional, vii

cambio, por más mínimo que sea, involucra que el diseño se complique o hasta incluso sea imposible de realizar. Esta limitación se agudiza más cuando se desea trabajar con dispositivos móviles. Sin embargo, existe una plataforma basada en hardware, que supera las dificultades de los circuitos VLSI, permitiendo aún mayores velocidades y prestaciones. Esta plataforma completamente reconfigurable denominada FPGA (arreglo de compuertas programables en campo), ofrece soluciones rápidas en poco tiempo y con un alto grado de flexibilidad. La Tabla 2.6, se muestran las comparativas entre las opciones disponibles para la implementación de algoritmos criptográficos.

Tabla 2.6. Plataformas Software, VLSI y FPGA.

	Software	**VLSI**	**FPGA**
Tamaño	Pequeño (depende)	Grande	Pequeño
Costo	Bajo	Alto costo	Bajo costo
Velocidad	Bajo	Muy elevado	Elevado
Memoria	Buena	Buena	Buena
Flexibilidad	Altamente flexible	Sin flexibilidad	Altamente flexible
Tiempo de comercialización	Bajo	Muy alto	Bajo
Consumo	Depende	Bajo	No tan bajo
Prueba/verificación	Sencillo	Dificultoso	Sencillo
Configuración en ejecución	Ninguna	Ninguna	Permitido

Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 8, México: IPN.

Implementaciones basadas en software presentan bajos costos, pero son lentos, implementaciones basadas en VLSI son muy rápidos, pero su desarrollo resulta demasiado grande y son poco flexibles. Los dispositivos reconfigurables (FPGA), son económicos, rápidos y muy flexibles.

2.2.7. Estándar de cifrado avanzado (AES).

Breve historia

En al año 1997, se publican los requerimientos finales para desarrollar el Estándar de cifrado Avanzado (AES), entre estos requerimientos se encontraban la capacidad de ser empleados con longitudes de bloque de 128 bits y longitudes de llave con 128, 192 y 256 bits.

El propósito de este nuevo estándar radicaba en la búsqueda de un cifrador tan seguro como el 3-DES, pero mucho más eficiente. **(NIST, 2001)**[14]

En octubre del 2000, la NIST pronuncia a *Rijndael* **(Daemen & Vincent, 1999)**[31] como el ganador. Este nuevo estándar, reemplaza al DES (cifrado estándar de datos), y su variante 3-DES.

II.2.7.1. Características básicas del AES

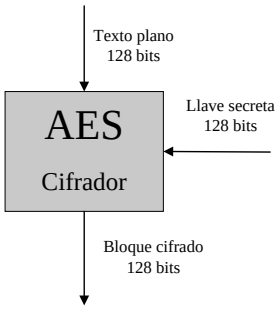
Es un tipo de cifrado simétrico por bloques, con una capacidad por bloque de 128 bits y capacidades de llaves de 128, 192 y 256 bits. Se trata de un sistema de cifrado iterativo, esta característica

[14] NIST. (2001) *Advanced Encryption Standard (AES)*

[31] Daemen, J., Vincent, R., (1999). *AES Proposal: Rijndael*. p. 4

permite que el cifrado y descifrado estén compuestas por la misma función de redondeo básico. En la Figura 2.6, se ilustra el diagrama de bloque para el cifrado mediante AES, para este caso, se ha empleado un bloque de texto de 128 bits y una llave de la misma longitud.

Figura 2.6. Diagrama de bloque del Cifrador AES



Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 69, México: IPN.

La distribución de los datos de entrada y la llave respectivamente, se organizan en matrices, están se muestran en las Figuras 2.7 y 2.8.

Figuras 2.7 Distribución Matriz de entrada

P_0	P_4	P_8	P_{12}
P_1	P_5	P_9	P_{13}
P_2	P_6	P_{10}	P_{14}
P_3	P_7	P_{11}	P_{15}

Figuras 2.8 Distribución llave de entrada

K_0	K_4	K_8	K_{12}
K_1	K_5	K_9	K_{13}
K_2	K_6	K_{10}	K_{14}
K_3	K_7	K_{11}	K_{15}

Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 36, México: IPN.

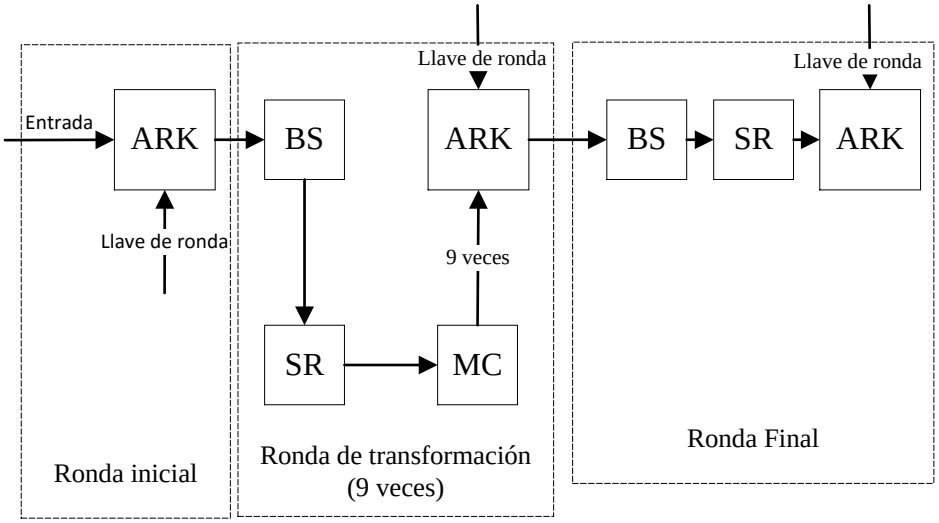
Rondas del AES

En AES, el número de rondas de cifrado depende del tamaño de la clave, estos son 10, 12, 14, para llaves de 128, 192 y 256 bits respectivamente. Los datos son agrupados internamente en forma cuadrada a la cual se denomina ***estado***.

Para el caso de un bloque de texto de 128 bits y una llave de la misma longitud, AES inicia con la suma inicial de la llave, denominada ***AddRoundKey***, seguida de las nueve rondas de transformación y finalmente la ejecución de la ***FinalRound***. La ronda inicial, y cada una de las siguientes toman como entrada a la matriz de estado y una llave de ronda. La llave de ronda para la *í-esima* ronda es denotada como ***ExpandedKey[i]***, y el ***ExpandedKey[0]*** para denotar la llave de entrada para la ronda inicial. La obtención del ***ExpandKey*** para la llave de cifrado se denomina ***KeyExpansion***.

Dentro de la ronda de transformación, se ubican cuatro bloques denominados ***SubBytes***, ***ShiftRows***, ***MixColumns***.

Figuras 2.9. Proceso de encriptado AES



Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 69, México: IPN.

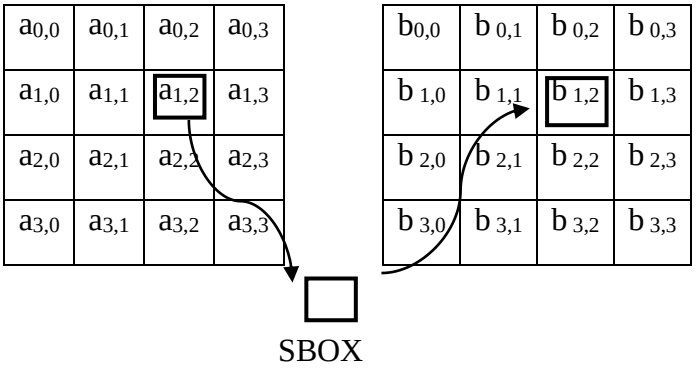
A continuación, se describen los bloques que conforman el proceso de cifrado. La primera etapa conformada por ***AddRoundKey*** que es la suma inicial de la llave.

SubBytes (BS)

La sustitución de bytes (SubBytes), es la única transformación no lineal del cifrador, y está basada en bloques pequeños, la cual consiste en la aplicación de una caja de sustitución (*S-BOX)* a los bytes de la matriz de estado con el propósito de considerar todas las combinaciones, la caja *S*, es una tabla de 256 bytes.

En la Figura 2.10, se aprecia el proceso de transformación por *SubBytes* a una matriz de estado.

Figuras 2.10. SubBytes a matriz de estado

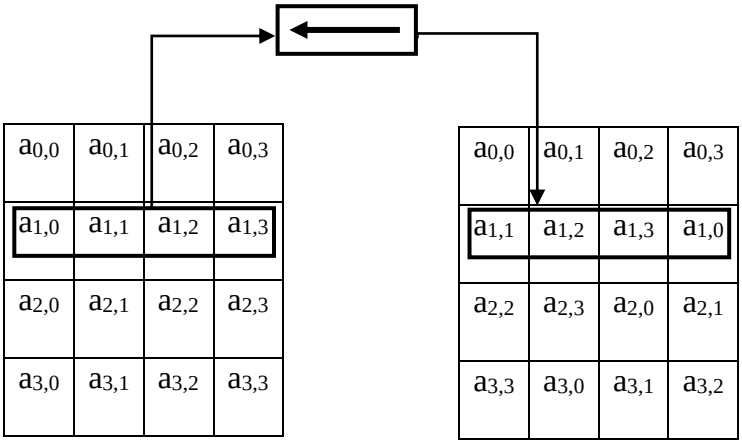


Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 70, México: IPN.

ShiftRows (RS)

El corrimiento de filas (ShiftRows), consiste en una transposición de bytes que realiza corrimientos circulares con diferentes desplazamientos a los renglones de la matriz de estado. El renglón 0 es desplazado C_0 bytes, el 1, C_1 bytes, el 2, C_2 bytes y el 3, C_3 bytes.

Figuras 2.11. ShitRows a la matriz de estados.



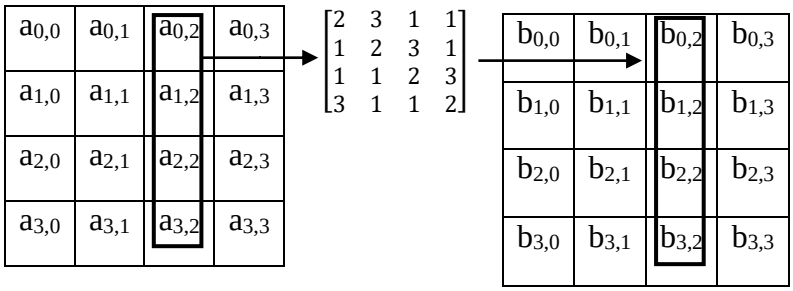
Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 72, México: IPN.

MixColumns (MC)

El mezclado de columnas (MixColums), es una permutación basada en bloques sobre la matriz de estado realizada columna a columna, esta operación puede ser descrita por una multiplicación de la matriz de estado, por una matriz constante. Todas las operaciones deben ser realizar en campos finitos binarios $GF(2^8)$.

En la Figuras 2.12, se muestra la operación MixColums con la matriz de estado en el proceso de cifrado AES.

Figuras 2.12. MixColumns con la matriz de estado.

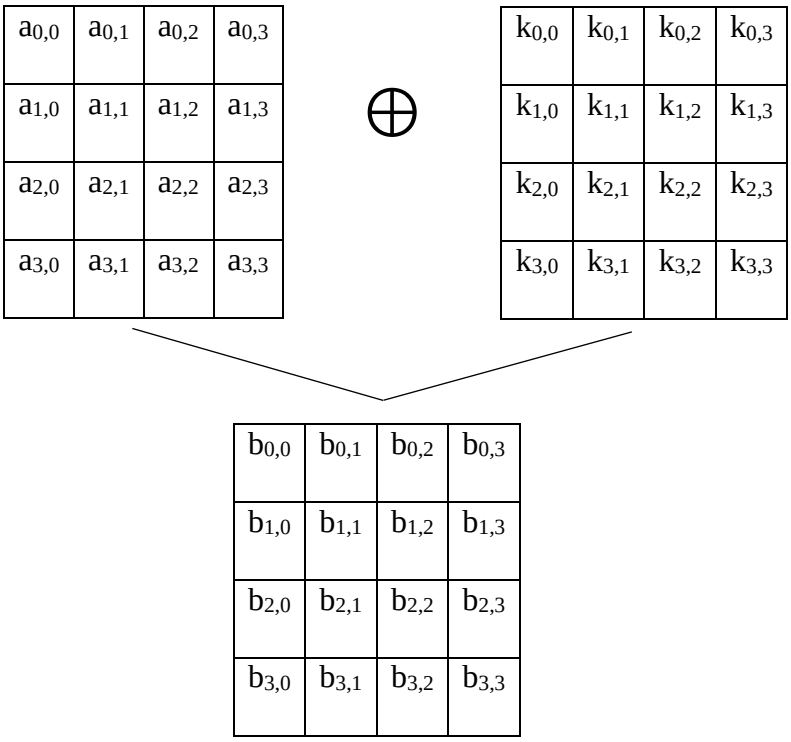


Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 73, México: IPN.

AddRoundKey (ARK)

La suma de llave (AddRoundKey), consiste en la combinación de la llave de ronda con la matriz de estado mediante la operación XOR. Una llave de ronda es denotada por *ExpendedKey[i]*, $0 \leq i \leq 10$. La llave de ronda tiene la misma longitud de la matriz de estado (128 bits). En la Figuras 2.13, se muestra el proceso de AddRoundKey en el cifrado AES.

Figuras 2.13. Proceso AddRoundKey en el cifrado AES



Nota. Recuperado de *"Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigurable"* de Nazzar, A., 2004, p. 73, México: IPN.

2.2.8. Modo de operación CCM

Introducción al modo de operación CCM

CCM, es el nombre para denotar a *Counter with CBC-MAC* (Contador con cifrado encadenado por bloques). Este modo, combina dos modos de operación: el CTR (Contador) y el CBC-MAC (cifrado encadenado por bloques). Se encuentra diseñado para su uso en cifradores de 128 bits, tal como es el caso del AES.

El modo CCM, provee una buena seguridad y rendimiento, tanto en software como en hardware (Jonsson, 2002)[32]. Este modo de operación fue propuesto en el año 2002, por Whiting et al **(Doug, Housley, & Ferguson, 2002)[24]**. Su artículo original fue enviado al Instituto Nacional de Estándares y Tecnología de los Estados Unidos (NIST), para ser adoptado como un modo de operación genérico. La IEEE, lo ha adoptado en su estándar 802.11 en su versión "i".

En **(Jonsson, 2002)[32]**, describe las siguientes ventajas del modo de operación de CCM:

- El CCM utiliza únicamente la operación para el ciframiento del cifrador por bloques, esto aplica tanto para el proceso de cifrado como descifrado del CCM. Esta característica hace atractivo al CCM, para aplicaciones en las cuales se desea un tamaño de código pequeño, además de ser posible utilizar una función que genere permutaciones pseudoaleatorias, la cual no sea necesariamente reversible, lo que convierte al CCM en uno de los modos más versátiles.

- Esta basado en dos tecnologías (CTR y CBC-MAC) ampliamente utilizadas y documentadas alrededor del mundo, lo que permite ahorrar tiempo en implementaciones, pues existen estrategias

[32] Jonsson, J. (2002) *on the Security of CTR + CBC – MAC*. p. 1

[24] Doug, W., Housley, R., & Ferguson, N. (2002) *Counter with CBC – MAC (CCM)*

[32] Jonsson, J. (2002) *on the Security of CTR + CBC – MAC*. p. 2

eficientes reportadas para su implementación en diferentes plataformas de desarrollo.

- Maneja mensajes en los cuales una parte de la información se desea únicamente autenticar y no cifrar, lo cual es realizado de manera sencilla sin incurrir en una perdida en la eficiencia del sistema. Para algunos otros modos de operación es necesario implementar mejoras para poder realizar esta operación.

- Todos los derechos intelectuales para el CCM han sido liberados para el uso público (Doug, Housley, & Ferguson, 2002)[24].

Así mismo, (Struick, 2003)[33] también enumera ciertas desventajas cruciales, las cuales son:

- Los criterios por parte de la *NIST* para la selección del CCM no fueron claros.

- Definido únicamente para cifradores de 128 bits.

- Es necesario saber con anterioridad la longitud de datos de entrada.

- El modo original del CCM no define la opción de brindar únicamente confidencialidad, sino que tiene que existir autenticación de datos de manera obligatoria, lo cual no es útil o necesario si una entidad externa brinda la autenticación de la información.

[24] Doug, W., Housley, R., & Ferguson, N. (2002) *Counter with CBC – MAC (CCM)*

[33] Struick, R.(2003) *Comments NIST Draft Pub 800 – 38 C Technical Report, Certicom*. p. 2 – 3

- Susceptibilidad a ataques cuando se emplean campos de longitud variable.

II.2.8.1. Parámetros del CCM

Se compone de dos parámetros, el primer parámetro denominado M, y el segundo, L.

El primer parámetro M, representa el tamaño de bytes del campo de autenticación, los valores válidos para este son: 4, 6, 8, 10, 12, 14 y 16, y estos dependerán estrechamente del grado de compromiso entre la expansión del mensaje y la probabilidad de que un atacante puede modificar el mensaje sin ser detectado. Su codificación es *(M-2)/2*.

El segundo parámetro L, representa el tamaño del campo de longitud (*l(m)*), sus valores están comprendidos entre los 2 y 8 bytes, estos valores dependerán del compromiso entre la extensión máxima del mensaje y el largo del *Nonce.* Su codificación es *L – 1*. **(Doug, Housley, & Ferguson, 2002)[24]**, resume el significado es estos parámetros en la Tabla 2.7.

La Tabla 2.7, muestra el valor del tamaño de campo de autenticación, y el tamaño del campo de longitud del mensaje, con su correspondiente codificación.

[24] Doug, W., Housley, R., & Ferguson, N. (2002) *Counter with CBC – MAC (CCM)* p. 2

Tabla 2.7. Parámetros del CCM

Nombre	Descripción	Tamaño del campo	Codificación del campo
M	Bytes del campo de autenticación	3 bits	*(M – 2) / 2*
L	Bytes en el campo de longitud del mensaje	3 bits	*L – 1*

Nota. Recuperado de "FPGA Implementation AES for CCM Mode Encryption Using Xilinx Spartan – II" by Khoa, V., 2003, *Advanced Cryptography*, p. 2, Copyright 2003 de SPRING.

II.2.8.2. Entradas del modo de operación CCM

Para que el emisor sea capaz de enviar un mensaje, debe proporcionar la siguiente información.

- Una llave de cifrado *K* adecuada para el cifrador por bloques.

- Un número de inicialización *Nonce* (*N*) con una longitud de 15 – L bytes. El valor de *nonce* debe ser único, esto quiere decir que para cada N empleado con la llave *K* no debe de repetirse.

- El mensaje *m*, que es una cadena de *l(m)* bytes. Donde: $0 \leq l(m) \leq 2^{8L}$

- Datos adicionales (opcionales) *a*, que consisten en una cadena de *l(a)* bytes donde: $0 \leq l(a) \leq 2^{64}$. Se debe de tener en cuenta, que estos datos son autenticados, pero no cifrados. La Figura 2.14, muestra el diagrama de bloques del modo de operación del CCM, emisor (a) y receptor (b).

Figura 2.14 *(a)*. Emisor del modo de operación CCM

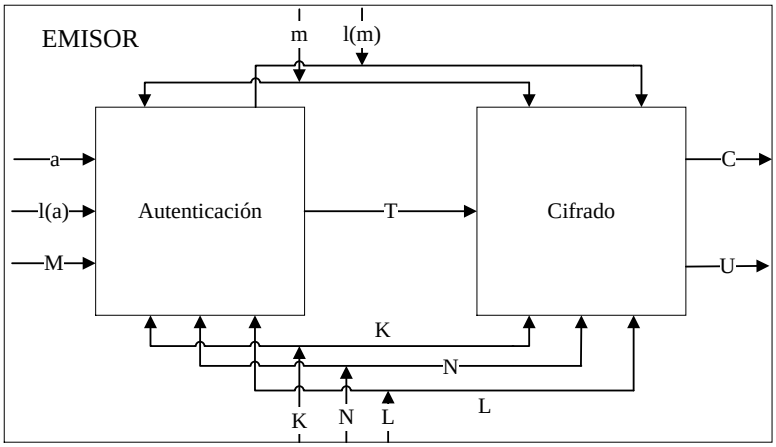
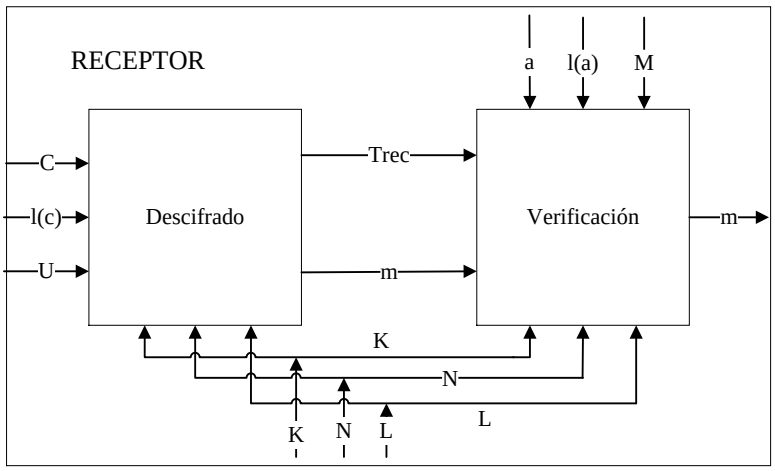


Figura 2.14 *(b)*. Receptor del modo de operación CCM



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 48, México: IPN.

II.2.8.3. Autenticación

Este primer paso durante el proceso del CCM, sirve para realizar la autenticación del mensaje, se hace necesario el empleo del CBC – MAC para la obtención del campo de autenticación *T*, esto debido a que CBC – MAC procesa información múltiplos de *kb*

fija, lo cual puede no necesariamente coincidir con los datos de entrada del CCM.

Para cumplir con lo mencionado, se define una secuencia de bloques $B_0 B_1 B_n$ y se calcula el CBC – MAC de estos.De acuerdo a **(Doug, Housley, & Ferguson, 2002)[24]** el primer bloque (B_0) se forma como se aprecia en la Tabla 2.8.

Tabla 2.8. Estructura bloque B_0

Byte n°	0	1 … 15 – L	16 – L … 15
Contenido	Bandera	*Nonce (N)*	l(m)

Nota. Recuperado de "FPGA Implementation AES for CCM Mode Encryption Using Xilinx Spartan – II" by Khoa, V., 2003, *Advanced Cryptography*, p. 2, Copyright 2003 de SPRING.

l(m), se codifica con el bit más significativo. El contenido del byte bandera para B_0, se muestra en la Tabla 2.9.

Tabla 2.9. Estructura de bandera para B_0

Bit n°	7	6	5	4	3	2	1	0
Contenido	Reservado	A data	M			L		

Nota. Recuperado de "FPGA Implementation AES for CCM Mode Encryption Using Xilinx Spartan – II" by Khoa, V., 2003, *Advanced Cryptography*, p. 2, Copyright 2003 de SPRING.

[24] Doug, W., Housley, R., & Ferguson, N. (2002) *Counter with CBC – MAC (CCM)*. p. 3

De la Tabla 2.9, el bit 7 se reserva para futuras expansiones y siempre debe estar puesto en 0. Adata = 0, si l(a)= 0 ó Adata = 1, si l(a) > 1.

Formado el bloque B_0, se deben formar los subsiguientes bloques y aplicarles el CBC – MAC. Para esta tarea se siguieron los siguientes pasos.

1. Del bloque B_1 al B_k se concatena l(a) (codificado), la información adicional *a* y divide el resultado con un bloque de 16 bytes, en este caso se agregará ceros de ser necesario.
2. Formado B_1 a B_k con los datos (opcionales), para la autenticación, es necesario agregar bloques del mensaje, estos bloques están constituidos mediante la división de *m* bloques de 16 bytes, agregando ceros de ser necesario. Si el mensaje *m* es una cadena vacía, no se agregan bloques en este paso.

Concluido estos pasos, el resultado es la secuencia de bloques B_0, $B1,\ldots B_n$. El CBC – MAC se calcula con la expresión:

$$X_1 = E_k(B_0)$$

$$X_{i+1} = E_k(X_i \oplus B_i)\ for\ i = 1, \ldots, n$$

$$T = firstMbytes(X_{n+1})$$

E( ) : Operación de cifrado con el cifrador por bloques

T : Campo de autenticación *M* bytes.

(B_n), es sumado mediante la operación XOR con X_n, este resultado es cifrado con el cifrador por bloques. De ser necesario, el texto

cifrado es truncado con la función firstMbytes para obtener *T* de la longitud deseada.

Las ecuaciones descritas pueden ser implementadas con el algoritmo 2.1.

1. Sean: $B_0, B_1, \ldots, B_n$ los bloques de autenticación
2. $X_1 = E_k(B_0)$
3. **for** i = 1 to *n* **do**
4. $X_{i+1} = E_k(B_i \oplus X_i)$
5. **end for**
6. $T = firstMBytes(X_{n+1})$

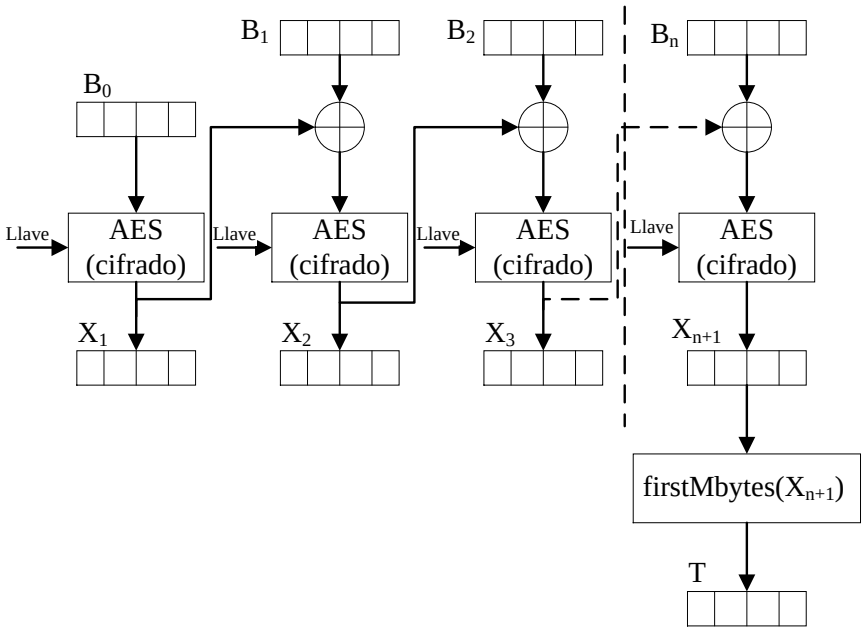
Algoritmo 2.1. Algoritmo de autenticación.

Como se puede apreciar, el algoritmo de implementación del CBC – MAC es sumamente sencillo, una característica importante de este algoritmo es emplear el encadenamiento, con esto, cada texto nuevo cifrado depende del anterior, por ello cualquier modificación mínima del mensaje ocasionará que el CBC – MAC, sea calculado erróneamente, y así detectar alteraciones intencionadas y/o accidentales.

Esta característica del encadenamiento, impide la paralelización en la etapa de implementación debido a la dependencia de datos existente, esto lo hace un algoritmo más lento.

En la Figura 2.15, se muestra el proceso de autenticación del CCM.

Figura 2.15. Proceso de autenticación en CCM



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 51, México: IPN.

II.2.8.4. Cifrado

Para el cifrado de un mensaje en el modo CCM, se emplea el modo de conteo (*CTR* por sus siglas en inglés).para ello se define un flujo de bloques.

$$S_i = E(K, A_i)\ for\ i = 0,1,2, \dots , \qquad (1)$$

(Doug, Housley, & Ferguson, 2002)[24], indica que los bloques A_i se conforman como la Tabla 2.10.

[24] Doug, W., Housley, R., & Ferguson, N. (2002) *Counter with CBC – MAC (CCM)*. p. 3

Tabla 2.10. Estructura del bloque A_i

Byte N°	0	1 … 15 – L	16 – L … 15
Contenido	Banderas	Mientras tanto N	Contador i

Nota. Recuperado de "FPGA Implementation AES for CCM Mode Encryption Using Xilinx Spartan – II" by Khoa, V., 2003, *Advanced Cryptography*, p. 3, Copyright 2003 de SPRING.

Los valores de *i* se codifican de forma que el byte más significativo representa una función de conteo establecida por el usuario, y dicho conteo se incrementa conforme se crean los bloques A_i.

Para cada bloque A_i, el campo de las banderas se encuentra formado por los valores vistos en la Tabla 2.11.

Tabla 2.11.Estructura de las banderas de A_i

Bit n°	7	6	5	4	3	2	1	0
Contenido	Reservado	Reservado	0			L		

Nota. Recuperado de "FPGA Implementation AES for CCM Mode Encryption Using Xilinx Spartan – II" by Khoa, V., 2003, *Advanced Cryptography*, p. 3, Copyright 2003 de SPRING.

Los bits reservados serán empleados para expansiones futuras, para ello deben estar puestos en cero, los bits 5, 4, 3, también son puestos a cero para diferenciarlos de los B_0, asegurando así, dominios diferentes para la autenticación y cifrado.

Los pasos para llevar a cabo el cifrado, son los siguientes.

1. Se cifra el mensaje mediante la operación XOR de *m* con los l(m) bytes de la concatenación se S_1, S_2, S_3, … , obteniéndose el texto cifrado *C*. S_0, no es empleado para cifrar el mensaje.
2. El parámetro de autenticación *T* es cifrado con S_0, truncándose a la longitud deseada, es decir el valor de *M*. Se muestra la ecuación de este procedimiento.

$$U = T \oplus firstMbytes\,(S_0) \quad (2)$$

Los elementos resultantes del proceso de cifrado son: el texto cifrado *C* seguido por el parámetro de autenticación cifrado *U*. estos son enviados al receptor para su proceso de descifrado y verificación.

Se muestra el algoritmo 2.2, de cifrado de un mensaje *m*. Es necesario indicar que para el cifrado no se emplean los datos *a*, ya que solo se emplean para autenticar y no para el cifrado.

Condición: *m* dividido en bloques de 16 bytes, formando m_0, m_1, … , m_n, *k* y *N*.

1. $Dado: A_0, A_1, \dots, A_n$ *los bloques de conteo*
2. $S_0 = E_k(A_0)$
3. $\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ n\ \boldsymbol{do}$
4. $S_i = E_k(A_i)$
5. $C_{i-1} = S_i \oplus m_{i-1}$
6. $\boldsymbol{end\ for}$
7. $U = T \oplus firstMBytes(s_0)$

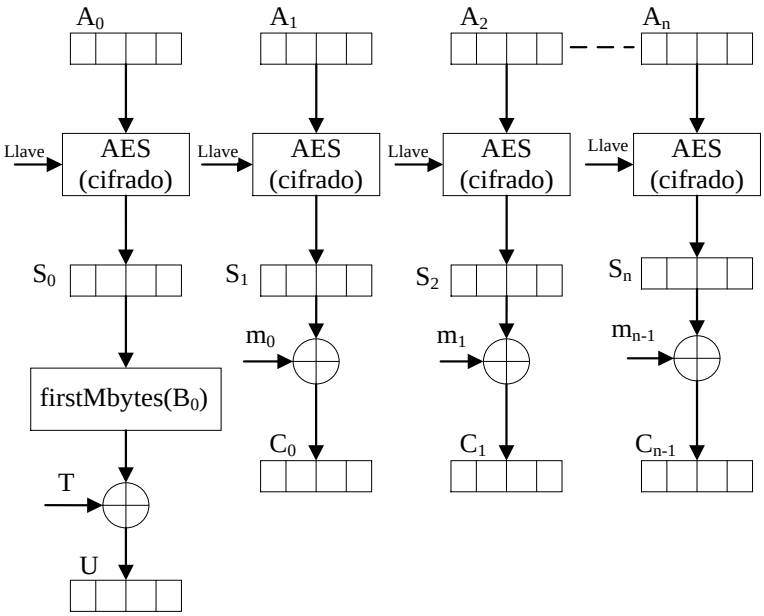
Algoritmo 2.2. Proceso de cifrado

De igual manera que le proceso de autenticación del CBC – MAC, el proceso de cifrado posee una implementación sencilla. Algo importante a recalcar, es que a diferencia del proceso de autenticación, el cifrado de la información, si puede ser paralelizado, puesto que no existe dependencia entre los datos. (Doug, Housley, & Ferguson, 2002)[24], indica que al cifrar el campo de autenticación *T*, se evitan los ataques por colisión al CBC – MAC. Si el cifrador por bloque se comporta como una permutación pseudoaleatoria, entonces el bloque de llave es indistinguible de una cadena aleatoria, esto hace imposible obtener una información útil de los resultados del CBC – MAC.

La Figura 2.16, muestra el diagrama de bloques del proceso de cifrado, mostrándose la interacción de los datos y los pasos necesarios para obtener la cifra.

[24] Doug, W., Housley, R., & Ferguson, N. (2002) *Counter with CBC – MAC (CCM)*. p. 5

Figura 2.16. Proceso de cifrado CCM



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 54, México: IPN.

Las propiedades de seguridad indican que el ***nonce (número aleatorio)***, no debe repetirse para garantizar que todos los bloques A_i de entrada del CTR y todos los bloques B_0 utilizados durante el tiempo de vida de la llave sean diferentes, y así evitar fisuras de seguridad.

II.2.8.5. Descifrado

Para el proceso de descifrado, el receptor requiere la siguiente información.

- Llave *K* del cifrador

- El *nonce (número de inicialización) N*

- El texto cifrado *C*

El proceso de descifrado inicia con el cálculo del flujo de llaves para el CTR (bloques S_i). Para ello se emplea la Ecuación (3), con estos bloques para obtener el mensaje original.

$$m_{i-1} = S_i \oplus C_i \ for\ i = 1, \dots, n \qquad (3)$$

En el proceso de descifrado, también es necesario obtener el parámetro de autenticación T_{rec} a partir de U, para lo cual es necesario aplicar la siguiente Ecuación (4).

$$T_{rec} = U \oplus firstMBytes(S_0) \qquad (4)$$

El algoritmo 2.3, muestra el proceso de descifrado.

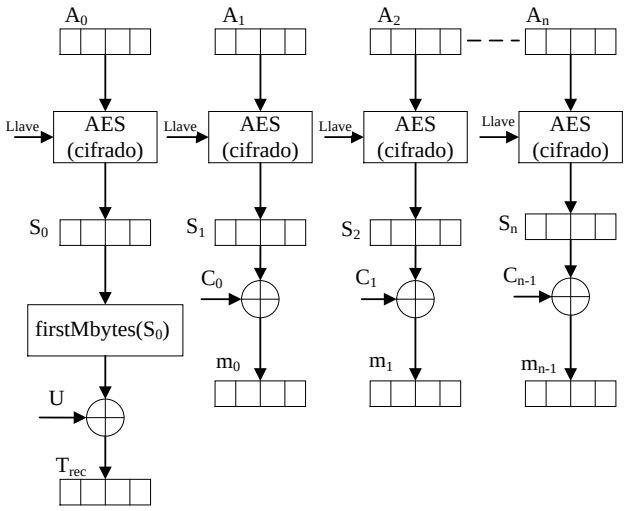
$Requiere$: C $dividido\ en\ bloques\ de$ 16 $bytes$,
$formando\ C_0, C_1, \dots, C_n, k, T\ y\ N$

1. $Sean\ A_0, A_1, \dots, A_n\ los\ bloques\ llave\ para\ el\ cifrado$
2. $S_0 = E_k(A_0)$
3. $\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ n\ \boldsymbol{do}$
4. $\quad S_i = E_k(A_i)$
5. $\quad m_{i-1} = S_i \oplus C_{i-1}$
6. $\boldsymbol{end\ for}$
7. $T_{rec} = U \oplus firstMBytes(S_0)$

Algoritmo 2.3. Algoritmo de descifrado

En la Figura 2.17, se puede apreciar el diagrama de bloques del proceso de descifrado, el proceso es similar al cifrado, con la diferencia que el descifrado utiliza los datos del cifrado recibidos y obtener así el mensaje original.

Figura 2.17. Proceso de descifrado CCM



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 55, México: IPN.

II.2.8.6. Verificación.

Descifrados el mensaje *m* y el parámetro de autenticación T_{rec}, se hace necesario realizar el proceso opuesto a la autenticación, a este proceso se le denomina verificación.

Este proceso es importante debido a que con ello, se garantiza que los datos no hay sufrido modificaciones accidentales y/o malintencionadas durante la transmisión.

En el modo CCM, el proceso de verificación es llevado de la siguiente manera:

1. Calculado el mensaje *m*, y los datos adicionales para autenticar *a*, se calcula el CBC – MAC, para obtener el parámetro de autenticación T_{calc}.

2. Se verifica la información comparando T_{rec} y T_{calc}. Si estos valores son iguales, entonces la información no sufrio cambios y puede ser utilizada, caso contrario, es desechada y se solicita reenvió de la misma. El algoritmo 2.4, muestra el proceso de verificación

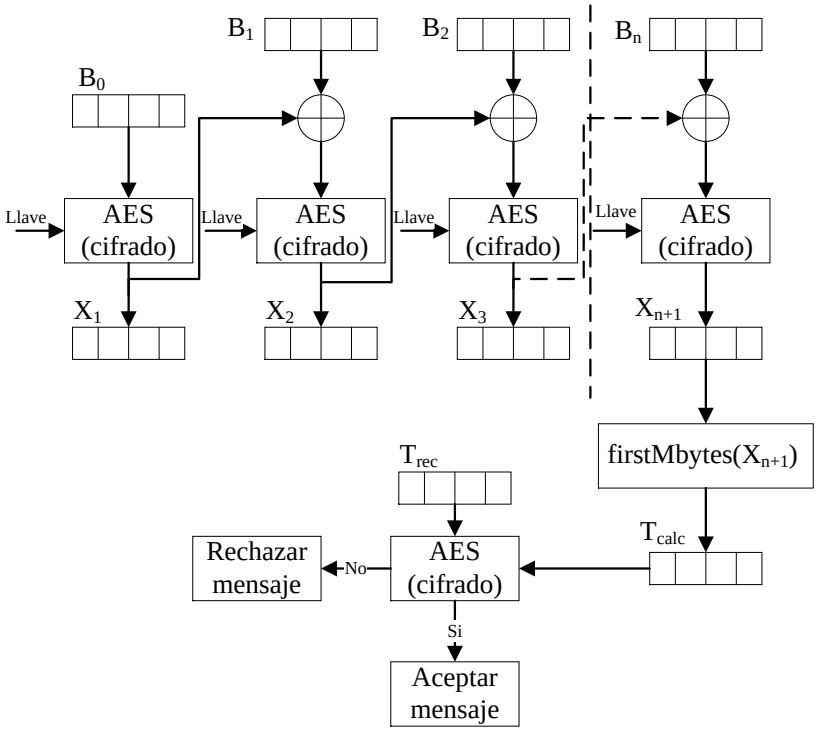
Requiere: $T_{rec}, k, m\ descifrado$

1. $Sean\ B_0, B_1, \dots, B_n\ los\ bloques\ de\ autenticación$
2. $X_i = E_k(B_0)$
3. $\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ n\ \boldsymbol{do}$
4. $X_{i+1} = E_k(B_i \oplus X_i)$
5. $\boldsymbol{end\ for}$
6. $T_{calc} = firstMBytes\ (X_{n+1})$
7. $T_{rec} = U \oplus firstMBytes(S_0)$
8. $Información\ aceptada\ y\ utilizada$
9. $\boldsymbol{else}$
10. Se desechan los datos y se solicita reenvío.
11. $\boldsymbol{end\ if}$

Algoritmo 2.4. Algoritmo del proceso de verificación

La Figura 2.18, muestra el diagrama de bloques, del proceso de verificación de la información recibida.

Figura 2.18. Proceso de verificación CCM



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 57, México: IPN.

II.2.8.6. Seguridad del modo CCM

(Jonsson, 2002)[32], menciona que la mejor fortaleza del CCM resuelve el problema de garantizar que la modificación del texto cifrado sea difícil para un oponente, esto a través de un parámetro de autenticación, a diferencia de otros modos de operación, los cuales realizan sumas de verificación. Menciona que el parámetro de autenticación disminuye la eficiencia de este modo de operación, pero al contrario brinda mayor nivel de seguridad.

El autor centra lo descrito en dos aspectos fundamentales.

[32] Jonsson, J. (2002) *on the Security of CTR + CBC – MAC*. p. 7

1. **Privacidad**

   Debe ser imposible para un usuario no autorizado obtener la información a partir del texto cifrado, sin conocer la llave.

2. **Autenticidad**

   Debe ser imposible para un usuario no autorizado generar texto cifrado valido sin tener acceso a la llave.

Finalmente **(Jonsson, 2002)**, indica que los nivel de seguridad proporcionados por el modo CCM, frente a la probabilidad de un atacantes externo, se encuentran dentro de los límites establecidos.

2.2.9. Implementación del CCM en (FPGA)

II.2.9.1. Plataforma para el desarrollo

El presente trabajo de tesis fue desarrollado empleando las siguientes herramientas.

- Herramientas de desarrollo IDE provistas por el fabricante ALTERA (ahora parte de Intel), estas son el Quartus II, en su versión 13.0, y la herramienta de simulación ModelSim – Altera 10.1d, proporcionada también por el mismo fabricante, ambas desarrolladas para el Sistema Operativo Windows.

- El Chip empleado fue el Cyclone IV (EP4CE22F17C6N), gracias a sus características de área y velocidad. Se emplea el módulo de desarrollo y educación ***DE0 – nano***, producido y

comercializado por **terasic**. Las características de esta placa de desarrollo, se resumen en Tabla 2.12.

- La programación de la plataforma fue llevada a cabo mediante diagrama de bloques funcionales, esto con la finalidad de facilitar los diseños y reducir los tiempos de implementación. Dado que los diagramas de bloques simplifican la tarea de implementación, es necesario aclarar que la síntesis es llevada a cabo mediante el lenguaje de programación VHDL (lenguaje de descripción de hardware de circuitos de muy alta velocidad).

En la sección 2.9.2, se brinda una descripción acerca del lenguaje de programación VHDL. Todos los resultados reportados en el presente trabajo, fueron extraídos a través de la herramienta de síntesis ModelSim – Altera 10.1d.

Tabla 2.12. Características del FPGA empleado.

Cyclone IV - EP4CE22F17C6N			
Elementos lógicos (LEs)	Pines de E/S	Memoria	SDRAM
22320	154	608256 bits	32 MB

Nota. Recuperado de "Your FPGA Platform Partner", (20, 08, 2017). Recuperado de http:// http://www.mouser.pe/ProductDetail/Intel-Altera/EP4CE22F17C6N/?qs=jblrfmjbeiGZz%252bplBRwiDQ==

II.2.9.2. Lenguaje VHDL

(Mano, 2003)[34], define a los lenguajes de descripción de hardware como lenguajes que describen el hardware de los sistemas digitales en forma textual. Se parecen a los lenguajes de programación, pero están orientados específicamente a la descripción de las estructuras y el comportamiento del hardware.

(Alfonso, Soto, & Fernández, 2002)[35], a su vez indica que VHDL (lenguaje de descripción de hardware para circuitos integrados de gran velocidad), es un lenguaje de descripción de hardware, que puede emplearse para modelar, documentar, simular, verificar y sintetizar un sistema digital. En este sentido abarca el ciclo completo de diseño, excepto el trazado físico o *layout*.

Este lenguaje ha sido adoptado por la IEEE como un estándar (J., 1990)[36] indica que VHDL está diseñado para satisfacer necesidades durante el proceso. Estas se indican a continuación.

1. Permite la descripción de la estructura del diseño, esto permite representar la manera de cómo se encuentran interconectados.

[34] Mano, M., M. (2003) *Diseño Digital*. México: Editorial Prentice Hall, p. 99 – 100

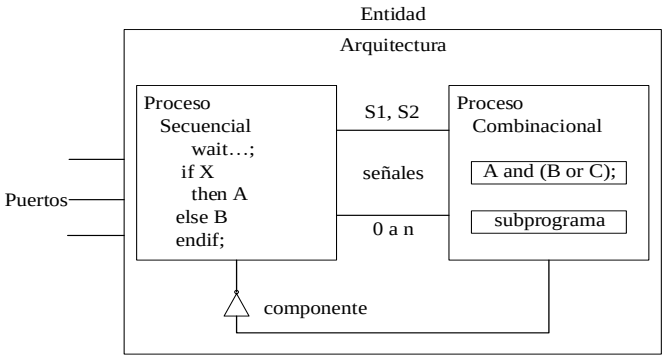
[35] Alfonso, S., Soto, E., & Fernández (2002) *Diseño de Sistemas Digitales con VHDL*. España: Editorial Paraninfo, p. 1 – 4

[36] J., A. P. (1990) *The VHDL CookBook*. Australia: Computer Science University of Adelaide, p. 1-1

2. Permite que las funciones sean especificadas mediante el uso de estructuras de lenguajes de programación conocidas por los programadores.
3. Es posible realizar la simulación del diseño antes de ser fabricado, esto permite a los desarrolladores verificar sus diseños, tiempos de respuestas y compararlos con alternativas, sin incurrir en costos del prototipo en hardware.

VHDL divide las entidades (circuitos), en una parte visible o externa y otra, oculta o interna (algoritmo de la entidad o penetración). Este concepto de vista externa e interna es central para un diseño de sistemas en VHDL. (Sánchez Santiago, 2003)[37], Indica que una vez definida la entidad para un diseño, esta puede reutilizarse en tantos diseños como sea necesario. La Figura 2.19, muestra el diseño de hardware en VHDL.

Figura 2.19 Estructura de un diseño de sistemas en VHDL.



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 65, México: IPN.

[37] Sánchez Santiago, M. (2003) *Implementación en Hardware – Software del algoritmo criptográfico DES*. México: Instituto Politécnico Nacional, p. 31

Una entidad (diseño) VHDL, tiene uno o más puertos de entrada, salida o bidireccionales, los cuales son conectados a sistemas vecinos, formando así *procesos y componentes* interconectados, todos operando concurrentemente. Toda entidad está conformada por una *arquitectura* particular, constituidas por construcciones VHDL tales como operaciones aritméticas, asignaciones de señal.

En el lenguaje VHDL, los circuitos de modelo secuencial en procesos independientes, se encuentran conformados por *flip-flops* y *latches*, y los circuitos combinacionales, por compuertas lógicas. Los *procesos* pueden definir y llamar (instanciar), a *subprogramas* (diseños secundarios), esto hace posible la reutilización de diseños poco complicados para la ejecución de un proceso de mayor complejidad, los cuales serán definidos por el usuario.

De acuerdo a **(Sánchez Santiago, 2003)**[37], el nivel de abstracción, VHDL está divida en tres categorías:

1. **Comportamiento**: Se describe el circuito electrónico mediante el comportamiento funcional o algorítmico del diseño, expresado en un proceso secuencial VHDL.

[37] Sánchez Santiago, M. (2003) *Implementación en Hardware – Software del algoritmo criptográfico DES*. México: Instituto Politécnico Nacional, p. 32

2. **Flujo de datos**: Los datos son analizados como un flujo a través de la entidad, desde la entrada hacia la salida. La operación queda definida por la transformación de estos datos.

3. **Estructural:** Es la categoría más cercana al hardware, los componentes de diseño se interconectan entre sí, y esta expresada por instancias.

Debido a la categoría estructural y sus ventajas, el presente proyecto fue llevado a cabo empleando este nivel de abstracción.

II. 2.9.3. Ciclo de diseño con FPGAs.

Para el desarrollo del presente proyecto de tesis, se procedió de acuerdo a los lineamientos descritos en **(Argüelles Cruz, Ascencio Roman, & Villalobos Baigorria, 2001)**[38], los cuales se describen a continuación por etapas.

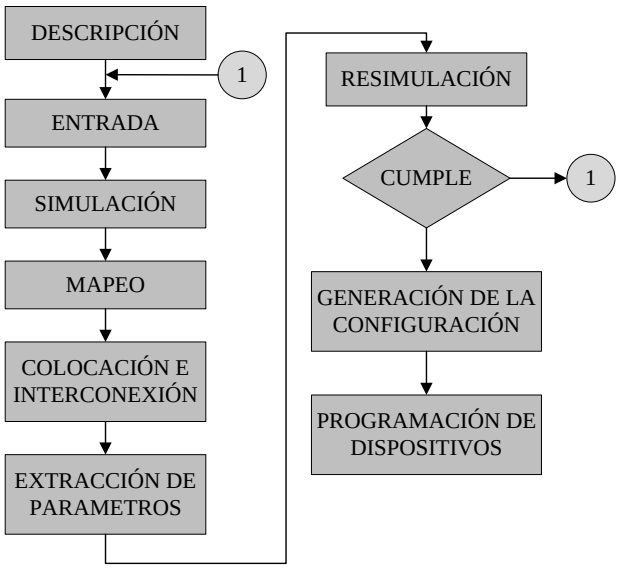
1. La introducción o descripción del diseño.
2. La simulación del diseño para verificar su funcionamiento.
3. El mapeo del diseño en la arquitectura del FPGA a utilizar.
4. La colocación e interconexión del diseño en FPGA.
5. La extracción de parámetros, una vez que el diseño ha sido conectado.
6. Una nueva simulación para verificar los tiempos de propagación de las señales involucradas en el diseño.

[38] Arguelles Cruz, A., Ascencio Roman, J., & Villalobos Baigorria, J. (2001) *Diseño de Sistemas Digitales Usando FPGA* p, 15 – 20

7. La generación del formato de configuración del dispositivo FPGA.
8. La configuración o programación del dispositivo.
9. La prueba del producto para observar la existencia de algún funcionamiento no deseado.

La Figura 2.20, muestra el diagrama de flujo en el ciclo de diseño con FPGAs.

Figura 2.20. Ciclo de diseño con FPGAs



Nota. Recuperado de “Diseño de sistemas digitales”, de Argüelles, C., Ascencio, R., y Villalobos, B., 2001, *Polibits,* p. 17.

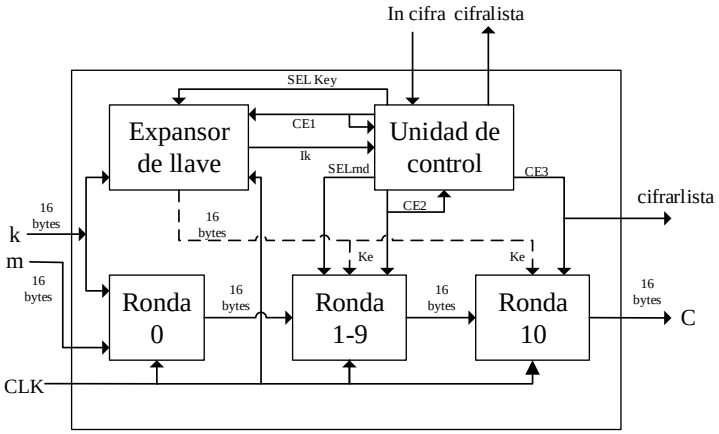
Para la implementación del prototipo se tuvo en cuenta las siguientes consideraciones.

1. CCM solo emplea la primitiva AES, por tal motivo solo se emplea esta primitiva.
2. Basado en tarjetas inalámbricas comerciales, que acepta un máximo de 1048 bytes en el buffer de salida, se considera para el presente trabajo un tamaño máximo de mensaje de 1024 bytes.
3. La información opcional de autenticación tiene una longitud de 32 bytes, esto basado en el tamaño de trama típica del IEEE 802.11.
4. Los datos necesarios para el CCM son provistos por una entidad externa, cuando esta la requiera.
5. Debido a que la autenticación no puede ser paralelizada, se recurre a la implementación mediante aproximaciones iterativas.

2.2.10. Implementación del AES

AES en modo de cifrado, posee tres componentes básicos: expansión de la llave, la unidad de control y los tres tipos de rondas del AES. La Figura 2.21, muestra la arquitectura general del AES.

Figura 2.21. Arquitectura del algoritmo AES.



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 68, México: IPN.

La expansión de llave, es la encargada de suministrar la ronda correspondiente al cifrador, La unidad de control es la responsable de sincronizar los procesos de generación de llave y cifrado, para así obtener un texto cifrado valido cuando el proceso termine. Este texto cifrado se denomina "cifrar lista".

Para comenzar el cifrado, "In cifra", debe de estar en un estado alto durante un pulso de reloj.

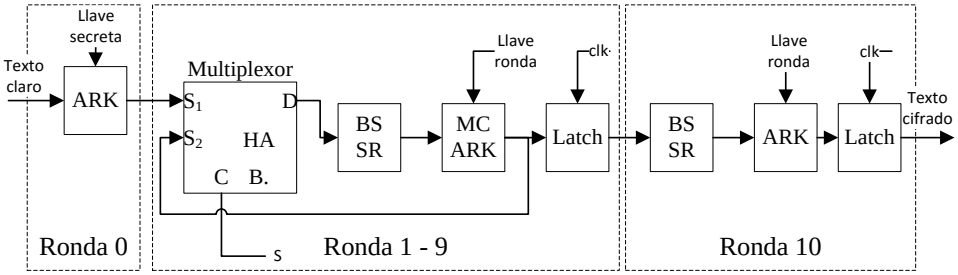
A la salida, se proporciona el texto cifrado correspondiente, el cual será utilizado en la implementación del CCM como cifrador por bloques.

A continuación se describe la implementación de cada uno de estos bloques.

II.2.10.1. Implementación de rondas del AES

Para la implementación de las rondas del AES, se hizo necesario implementar cada paso que conforma la ronda. En la Figura 2.22, se puede apreciar la arquitectura diseñada para la implementación de las 10 rondas del AES.

Figura 2.22. Implementación de rondas del AES

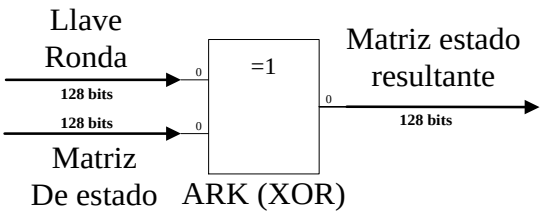


Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 69, México: IPN.

Implementación del ARK

El *AddRoundKey* consiste únicamente en la suma de la matriz de estado con la matriz de la llave de ronda mediante una XOR. Para este fin la entrada de los datos se da en formato de matrices. La Figura 2.23, muestra la implementación del ARK, en este punto es necesario aclarar que este proceso solo se aplica en la ronda 0 y la ronda 10.

Figura 2.23. Implementación del bloque ARK



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 69, México: IPN.

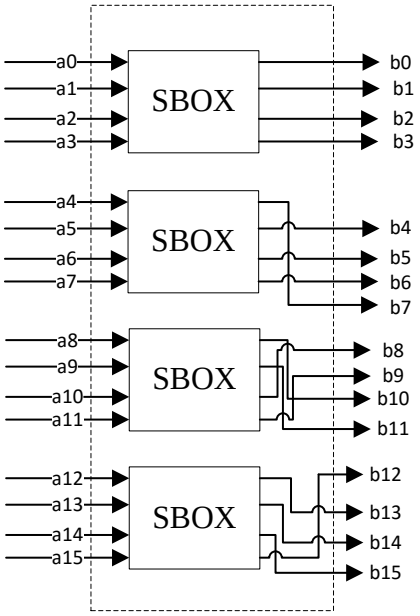
Implementación BS y SR

La implementación de *ByteSubstitution* y *ShiftRows* se realizó de manera simultánea. Se tuvieron en cuenta los siguientes pasos:

1. Se empleó un bloque de memoria RAM con una capacidad de 256 bytes para el almacenamiento de los *SBOX*.
2. Con la memoria configurada se construye un componente, se implementó un componente que emplea 4 bloques de memoria RAM, y realiza la sustitución de los 4 bytes simultáneamente. Asi mismo, este componente es reutilizado para la implementación del expansor de llaves.
3. Para la sustitución de toda la matriz, se recurre al paso anterior creándose 4 de estos componentes, para así formar un componente encargado de la sustitución de los 16 bytes de la matriz de estado.
4. Dado que S*hifRows* es únicamente una reorganización de los bytes, se realiza un redireccionamiento de las salidas del *SBOX* al orden especificado por los *offsets*.

La Figura 2.24, muestra el proceso de implementación del BS y SR así como también el redireccionamiento de los bytes sustituidos en la *SBOX*.

Figura 2.24. Implementación de *SuBbytes* y *ShiftRows*



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 70, México: IPN.

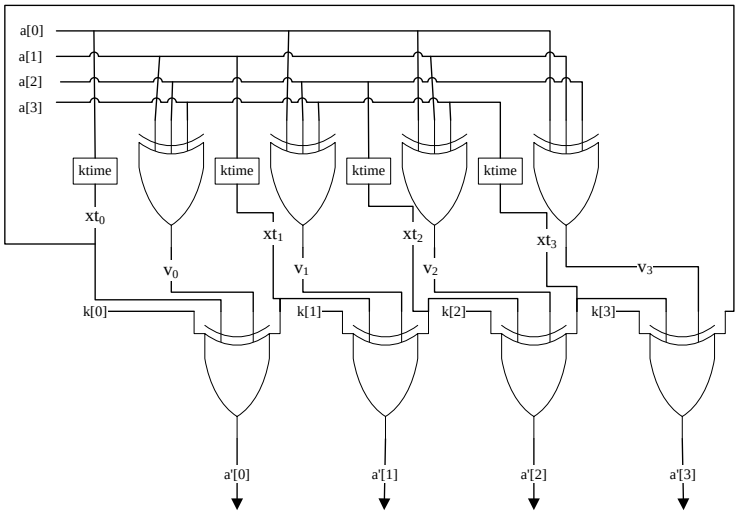
De la figura, a_i, representa el byte *i* de la matriz de estado de entrada, mientras que b_i, la matriz de estado resultante.

Implementación del MC y ARK

La implementación del *Mix Columns* y *AddRoundKey*, se realizó en un solo paso con la finalidad de optimizar el tiempo de procesamiento. Para la implementación de este proceso, se emplean las compuertas XOR, y las tablas de consulta implementadas mediante los bloques de memoria RAM. La

Figura 2.25, muestra la implementación de este proceso. *xtime()* se representa por un cuadro que simboliza la consulta a la memoria RAM, para la obtención del dato buscado.

Figura 2.25. Implementación de MixColumns y AddRoundKey

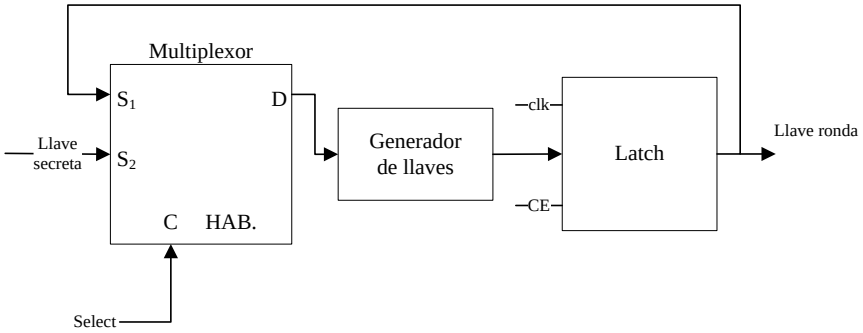


Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 71, México: IPN.

Implementación de la expansión de llave

La implementación de la expansión de llave es llevada a cabo de manera iterativa, donde el proceso *generador de llave* se repite 10 veces para suministrar las llaves de ronda correspondientes. El multiplexor se encarga de la selección de la llave a utilizar en el generador de llaves. Al inicio se emplea la llave secreta, en los pasos subsiguientes, se retroalimentan las llaves calculadas en el paso anterior. La Figura 2.26, muestra la arquitectura de la expansión de llave, así como los diferentes componentes que la conforman.

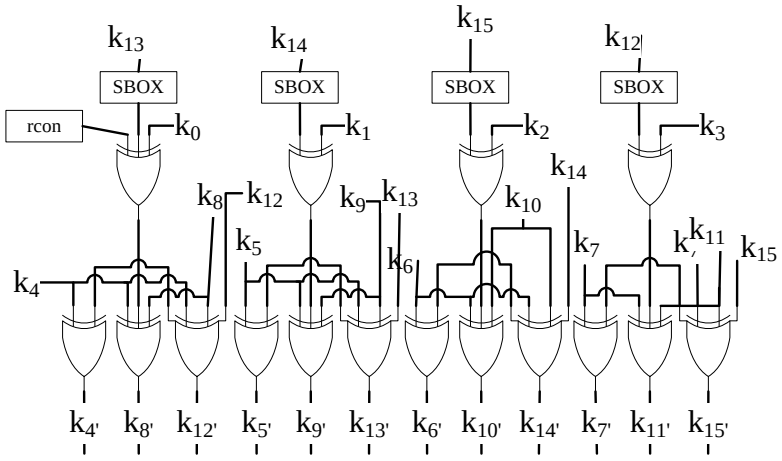
Figura 2.26, Implementación de la expansión de llaves.



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 71, México: IPN.

El generador de llaves fue llevada a cabo empleando compuertas XOR, tal como se muestra en la Figura 2.27.

Figura 2.27. Implementación de la expansión de llave.



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 84, México: IPN.

De la Figura 2.27, puede observarse el uso de las funciones *SBOX* y *rcon*, las cuales son tablas de consulta implementadas mediante un bloque de memoria RAM.La tabla *SBOX*, se encuentra

conformado por 256 bytes, organizados en una matriz de 16 x 16 bytes. Por otra parte, *rcon* es una matriz de 4 x 8 bytes, la cual almacena el cálculo de las constantes de ronda. Se calcula del siguiente modo.

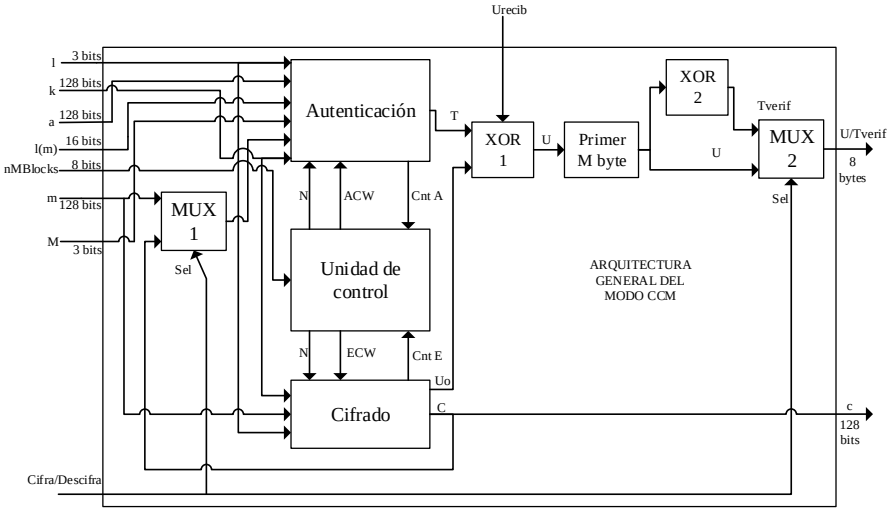
$$r_{con}(i) = 2^{(i-4/4)} \; en \; GF\,(2^8) \qquad (4)$$

El *latch* final, se emplea para almacenar los resultados parciales con la finalidad de simplificar la sincronización entre los componentes restantes del AES.

2.2.11. Implementación del CCM

II.2.11.1. Arquitectura general del CCM.

Figura 2.28. Arquitectura general del modo CCM



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 73, México: IPN.

Como se aprecia en la Figura 2.28, los bloques que conforman el sistema final, y los más importantes son la unidad de control,

bloque autenticación y bloque de cifrado. La unidad de control es la encargada de llevar a cabo los procesos de sincronización a través de las palabras de control ACW y ECW.

Tabla 2.13. Parámetros y entradas del CCM

Nombre	Descripción	Tamaño campo	Cod. del campo
Parámetros			
M	Bytes del campo de autenticación	3 bits	(M-2)/2
L	Bytes en el campo de longitud del mensaje	3 bits	L – 1
Entradas			
K	Llave para el AES	16 bytes	Cadena de bits
N	Numero aleatorio (Nonces)	15 – L Bytes	Cadena de bits
m	Mensaje a cifrar y enviar	Max. 1024 bytes	Cadena de bits
a	Información adicional a autenticar (no se cifra)	32 bytes	Cadena de bits
Auxiliar			
T	Campo de autenticación sin cifrar	M bytes	Cadena de bits
Salidas			
U	Campo de autenticación cifrado	16 bytes	Cadena de bits
C_i	Bloque *i-ésimo* del texto cifrado	16 bytes	Cadena de bits

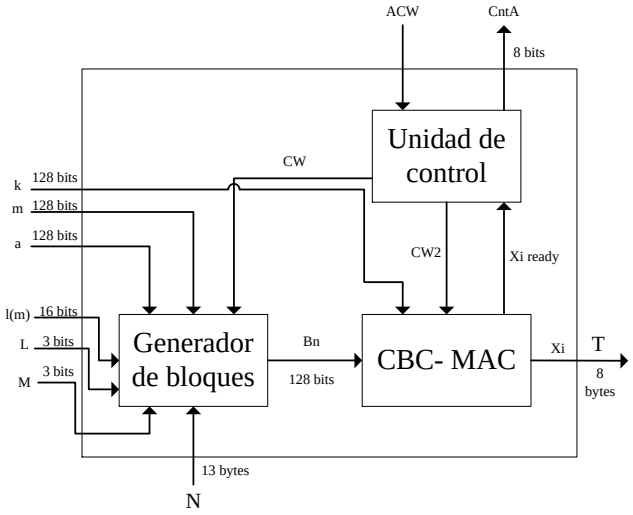
Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 74, México: IPN.

II.2.11.2. Implementación de autenticación del modo CCM

El bloque de autenticación se presenta en la Figura 2.29, donde se muestran los componentes que la conforman: Generador de bloques, CBC –MAC y la unidad de control para la autenticación.

Los bloques que conforman al módulo de autenticación son tres: generador de bloques, CBC – MAC, y la unidad de control encargada de la autenticación.

Figura 2.29. Arquitectura del Módulo de autenticación.



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 75, México: IPN.

A continuación se describen los bloques que conforman la arquitectura del módulo de autenticación.

Implementación generador de bloques

El generador de bloques se encarga de construir los bloques B_0, B_1,…, B_n. Estos bloques son generados con la información

proveniente del usuario, así como los parámetros del sistema. Para la implementación, se recibe una palabra de control (2 bits) proveniente de la unidad de control de autenticación. En la Tabla 2.14, se describe la palabra de control, y el **algoritmo 2.5**, muestra la implementación del generador de bloques.

Tabla 2.14, Palabras de control.

Valor	**Significado**
00	Genera B_0 sin datos de autenticación adicionales (Adata = 0)
01	Genera B_0 con datos de autenticación adicionales (Adata = 1)
10	Genera B_i a partir del mensaje a enviar (m)
11	Genera B_i con los datos de autenticación adicionales (a)

Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 76, México: IPN.

$Requiere: CW, m, a, l(m), nonce, M, L, contador$

1. $\boldsymbol{if}$ *cambio en CW y contador* $\boldsymbol{then}$
2. $\quad \boldsymbol{if}\ CW = 00\ o\ CW = 01\ \boldsymbol{then}$
3. $\quad\quad \boldsymbol{if}\ CW = 00\ \boldsymbol{then}$
4. $\quad\quad\quad Bloque \leftarrow B_0\ con\ Adata = 0$
5. $\quad\quad \boldsymbol{else}$
6. $\quad\quad\quad Bloque \leftarrow B_0\ con\ Adata = 1$
7. $\quad\quad \boldsymbol{end\ if}$
8. $\quad \boldsymbol{else\ if}\ CW = 10\ \boldsymbol{then}$
9. $\quad\quad Bloque \leftarrow B_i\ en\ base\ al\ mensaje\ m$
10. $\quad\quad\quad \boldsymbol{else\ if}\ CW = 11\ \boldsymbol{then}$
11. $\quad\quad Bloque \leftarrow$

$B_i\ en\ base\ a\ los\ datos\ de\ autenticación\ a$

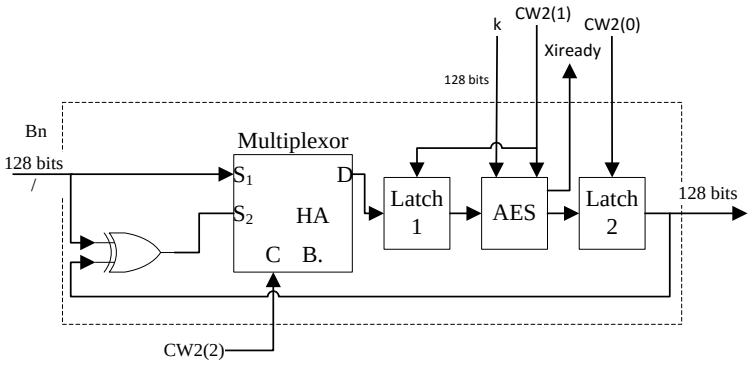
12. ***end if***
13. ***end if***
14. $BlockGen \leftarrow Bloque$

Algoritmo 2.5 Algoritmo generador de bloques de autenticación

El CBC – MAC

El CBC – MAC, es el componente básico de la autenticación, y es el encargado de llevar a cabo la autenticación de la información proporcionada por el generador de bloques. La Figura 2.30, muestra la arquitectura de la implementación del CBC – MAC, la cual consta de 4 componentes: Operación XOR, un multiplexor, el cifrador AES y dos latches.

Figura 2.30. Arquitectura de implementación del CBC – MAC



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 77, México: IPN.

Implementación de la unidad de control para la autenticación

Esta etapa, es la encargada de sincronizar los procesos que intervienen en la autenticación de la información. El **algoritmo 2.6**, muestra el proceso para la implementación de la unidad de control, en la Tabla 2.15, se muestran sus valores y significados.

$Requiere: CW, xiReady, estado$

1. $estado = inicial$
2. $\boldsymbol{for}\ i = 0\ to\ i = n\ \boldsymbol{do}$
3. $\quad \boldsymbol{if}\ xiReady = '1'\ y\ CW = '1'\ \boldsymbol{then}$
4. $\quad\quad \boldsymbol{if}\ estado = inicial\ \boldsymbol{then}$
5. $\quad\quad\quad CWout \leftarrow GenB0\ \{Generar\ B_0\}$
6. $\quad\quad\quad estado = iniciado$
7. $\quad\quad \boldsymbol{else\ if}\ i < 3\ \boldsymbol{then}$
8. $\quad\quad\quad CWout \leftarrow GenB1y2\ \{Generar\ B_1\ y\ B_2\ con\ a\}$
9. $\quad\quad \boldsymbol{else}$
10. $\quad\quad\quad\quad CWout \leftarrow GenBk\ \{Generar\ B_k\ con\ m\}$
11. $\quad\quad\quad \boldsymbol{end\ if}$
12. $\quad\quad \boldsymbol{end\ if}$
13. $\quad \boldsymbol{end\ for}$

Algoritmo 2.6. Algoritmo de la unidad de control para autenticación.

Tabla 2.15. Descripción de elementos de la unidad de control de autenticación

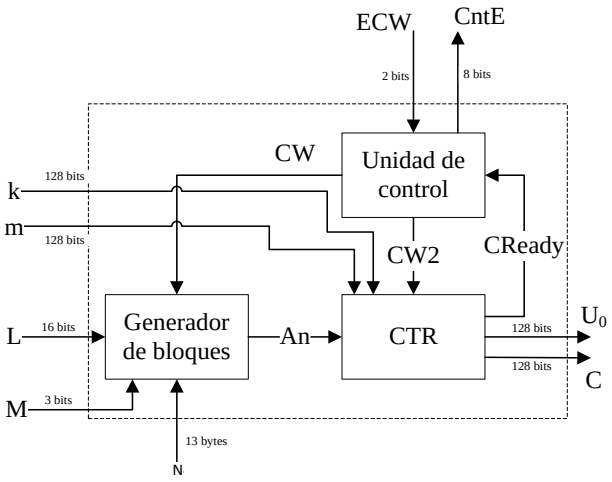
Valor	**Significado**
Bits De control para el generador de bloques	
00XXX	Genera B_0 sin datos de autenticación adicionales (Adata = 0)
01XXX	Genera B_0 con datos de autenticación adicionales (Adata = 1)
10XXX	Genera B_i a partir del mensaje a enviar (m)
11XXX	Genera B_i con los datos de autenticación adicionales (a)
Bits de control para el CBC – MAC	
XX1XX	Control del multiplexor, se usa B_i
XX0XX	Control del multiplexor, se usa $B_i \oplus X_i$
Valor	**Significado**
XXX0X	Control del latch1 y AES, no cifrar y desactivar latch1
XXX1X	Control del latch1 y AES, cifrar y activar latch1
XXXX0	Control del latch2, desactivar
XXX1	Control del latch2, activar

Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 79, México: IPN.

II.2.11.3. Implementación del cifrado CCM

La Figura 2.31, muestra la arquitectura diseñada para el cifrado CCM, se parecían los componentes que la conforman: Un generador de bloques, el cifrador CTR, y una unidad de control. Se detalla la implementación de cada uno de estos componentes.

Figura 2.31. Arquitectura Cifrado CCM



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 80, México: IPN.

Generador de bloques para el cifrado.

Esta etapa se encarga de proporcionar los bloques $A_0, A_1, A_2, \ldots, A_n$. Los bloques se generan a partir de la función de conteo, proporcionados por la unidad de control del cifrado. Al **algoritmo 2.7**, muestra este procedimiento.

El generador de bloques es habilitado a través de una palabra de control proveniente de la unidad de control del cifrado. Si es ´1´, el proceso es ejecutado, caso contrario no se realiza ninguna operación.

$Requiere: nonce, L, Contador\ y\ CW$

1. $\boldsymbol{if}\ CW = 1\ \boldsymbol{then}$
2. $\quad i \leftarrow contador$
3. $\quad \boldsymbol{if}\ i \geq 0\ \boldsymbol{then}$
4. $\quad\quad construye\ A_i$

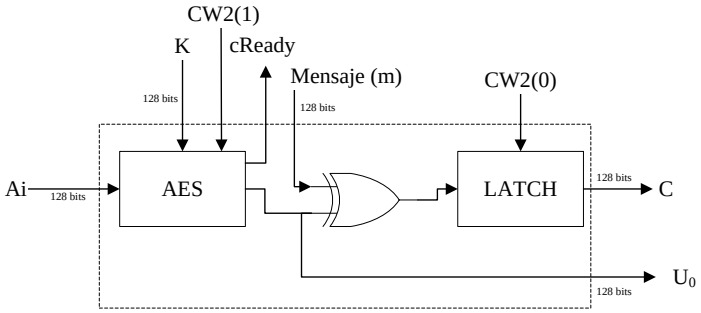
5. ***end if***
6. ***end if***
7. $Salida \leftarrow A_i$

Algoritmo2.7. Algoritmo generador de bloques para cifrado

Cifrado con el modo CTR

Formado el bloque A_i, la información es cifrada mediante el modo CTR. La Figura 2.32, muestra el diagrama de bloques para la implementación de este modo de operación.

Figura 2.32. Diagrama de bloques del módulo CTR



Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 81, México: IPN.

Como puede apreciarse de la figura, A_i, es cifrado con AES para obtener el S_i, con el cual se cifra el mensaje (m) mediante la operación XOR.

La salida U_0 corresponde a S_0, este último es utilizado para cifrar el parámetro de autenticación T, por este motivo no ingresa a la operación XOR con el mensaje, ni es almacenado en el latch al final del circuito. En la Tabla 2.16, se describen los valores de la palabra de control para el cifrado.

Tabla 2.16. Valores de la palabra de control para el cifrado

Valor	**Significado**
0XX	Generador de bloques deshabilitado
1XX	Generador de bloque habilitado
X1X	Iniciar cifrado con AES
X0X	Detener cifrado con AES
XX0	Latch deshabilitado
Valor	**Significado**
XX1	Latch habilitado

Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 82, México: IPN.

Unidad de control para cifrado.

La unidad de control recibe una palabra de control general, esta última consta de dos bits, e indica el procedimiento que debe de llevarse a cabo. En la Tabla 2.17, se indican los valores de la unidad de control. Es necesario indicar, que la unidad de control se encarga de proporcionar el índice del bloque a generar, por tal motivo en la implementación, se realiza el conteo de los bloques, incrementado en uno el valor del contador cada vez que se ha terminado de procesar el bloque anterior.

El pin RESET, funciona en todo momento, y su función es reiniciar todo el sistema a un valor inicial. El **algoritmo 2.8**, muestra la implementación de la unidad de control para el cifrado.

$Requiere: CWGen, RESET, CE, cReady$

1. $\boldsymbol{if}\ RESET = 1\ \boldsymbol{then}$
2. $contador\ Aux \leftarrow 0$
3. $CW \leftarrow Desactivar\ procesos$
4. $\boldsymbol{else\ if}\ CE = 1\ \boldsymbol{then}$
5. $CW \leftarrow Activar\ procesos$
6. $\boldsymbol{if}\ CW\ Gen = 1\ y\ cReady = 1\ \boldsymbol{then}$
7. $contador\ Aux+ = 1$
8. $\boldsymbol{else}$
9. $contador\ Aux = 0$
10. $\boldsymbol{end\ if}$
11. $\boldsymbol{end\ if}$
12. $contadorSalida \leftarrow contadorAux$

Algoritmo 2.8. Algoritmo de implementación de la unidad de control.

II.2.11.4. Unidad de control general.

La implementación de una unidad de control general permite de manera adecuada que los procesos de autenticación y verificación sincronicen el intercambio de información entre ellos.

La Tabla 2.17, muestra los valores de cada bit de la palabra de control.

Tabla 2.17. Valores de los bits para la unidad de control general

Valor	Significado
1XX	Utilizar el valor incrementado de la función de conteo
0XX	Utilizar el valor inicial de la función de conteo
X0X	Proceso de cifrado deshabilitado
X1X	Proceso de cifrado habilitado
XX0	Deshabilitar el proceso de autenticación
XX1	Habilitar el proceso de autenticación

Nota. Recuperado de *"Implementación eficiente en FPGA del modo CCM usando AES"* de Trejo, E., 2004, p. 84, México: IPN.

II.2.11.5. Implementación del descifrado y verificación.

Para este procedimiento se diseñó un bloque extra, el cual puede ser apreciado en la Figura 2.28.

Este diseño permitió controlar las operaciones de descifrado y verificación de manera independiente que las descritas anteriormente, ya que los procesos de descifrado y verificación son los opuestos al cifrado y autenticación respectivamente.

Por lo descrito, es posible la ejecución de ambos procesos únicamente con la modificación de la cifra *C* enviada por el emisor, en lugar del mensaje *m*, mientras que por su parte, la verificación utiliza el mensaje descifrado.

El multiplexor 1, tiene como objetivo realizar la selección de la entrada correspondiente para la autenticación/verificación, por lo que si la entrada de selección es 0, se utilizará el mensaje *m* a cifrar, por lo tanto se realiza la autenticación de la información (realizado por el emisor). En caso contrario, se utiliza la información proporcionada por el módulo de cifrado, el cual estará realizando el descifrado de la información recibida.

El proceso de verificación, es llevada exactamente igual a la autenticación, con la diferencia que al final del procesamiento de la información, mediante la compuerta XOR2, se valida la información recibida, el multiplexor2 brinda a su salida el valor de *U* calculado.

2.3. Definición de términos.

AES Estándar de encriptación avanzada. También conocido como Rijndael, es un esquema de cifrado por bloques empleado en criptografía simétrica

ASIC Circuito integrado de aplicaciones específicas, es un circuito integrado hecho a la medida para un uso en particular.

BUFFER Espacio de memoria en un equipo digital reservado para el almacenamiento temporal de la información.

CLB Bloque lógico configurable, es la arquitectura que compone a una matriz de compuertas programables en campo.

CCM	Contador con CBC-MAC , es un modo de operación de cifrado por bloques, diseñado para brindar autenticación y confidencialidad
CCMP	(Protocolo modo CCM), Es un protocolo de cifrado para LAN inalámbrica, y es un estándar de la IEEE 802.11i
DSP	Procesador digital de señales, es un equipo electrónico optimizado para aplicaciones que requieran operaciones numéricas a muy alta velocidad.
ECC	Criptosistema de curvas elípticas, es una aproximación a la criptografía de clave pública basada en la estructura algebraica de las curvas elípticas sobre campos finitos.
EDA	Automatización para el diseño electrónico, herramientas utilizadas en el diseño de sistemas electrónicos
FPGA	Matriz de compuertas programables en campo
GPS	Sistema de posicionamiento global
IEEE	Instituto de ingenieros eléctricos y electrónicos
LAN	Red de área local
LCA	Matriz de celdas lógicas
MAC	Código de autenticación de mensaje.
MD5	Message-Digest algorithm 5, es una algoritmo de reducción criptográfico.
PLD	Dispositivo lógico programable
VLSI	Integración a escala muy grande en circuitos electrónicos
VHDL	Lenguaje de descripción de hardware

WEP	Privacidad equivalente a cableado en redes inalámbricas
WLAN	Red de área local inalámbrica
WPA	Acceso Wi-Fi protegido.
Wi-Fi	Fidelidad sin cables, es un mecanismo de conexión de dispositivos electrónicos de manera inalámbrica.

III. METODOLOGIA

3.1. Tipo y diseño de Investigación.

Por el alcance de la investigación, la cual consiste en mejorar la seguridad en las redes inalámbricas del IESTP “Eleazar Guzmán Barrón – Huaraz”, el presente estudio es de tipo **correlacional**, ya que se evaluó el modo de operación CCM, y AES como cifrado por bloques; para un esquema sobre hardware reconfigurable (FPGA) que permitió comprobar la factibilidad de empleo del AES, y como esta, fue aplicada a la seguridad en la redes de datos del Instituto Tecnológico.

Se empleó un enfoque **cuantitativo**, ya que se pretendió conocer mediante la recolección de datos, el fenómeno en estudio, encontrándose las soluciones para la misma.

a. De acuerdo a la orientación

La investigación de acuerdo a la orientación, es **aplicada**, puesto que se orientó a lograr un incremento en los conocimientos tecnológicos, destinado a procurar soluciones a problemas prácticos.

b. De acuerdo a la técnica de contrastación

La investigación es de tipo **explicativa**, en la medida que los datos fueron obtenidos por observación de fenómenos condicionados por el suscrito y utilizando la experimentación.

Se expone los recursos utilizados, tales como equipos, instalaciones, formatos, encuestas, etc., asimismo los procedimientos de recolección de datos y análisis de información que fueron practicados para obtener los

resultados. Se indica el lugar donde se realizó el trabajo y se reportó los datos necesarios y suficientes para que otros investigadores puedan repetir el trabajo o simplemente, puedan verificar las condiciones en que fue realizado el experimento o la metodología seguida. Se presenta con claridad los tratamientos, las variables respuesta o parámetros de evaluación, el diseño estadístico empleado y el número de repeticiones. Se explica cómo estuvo constituida la unidad experimental.

Esta parte del proceso de investigación consiste en procesar los datos (dispersos, desordenados, individuales) obtenidos de la población objeto de estudio durante el trabajo de campo, y tiene como fin generar resultados (datos agrupados y ordenados), a partir de los cuales se realizará el análisis según los objetivos e hipótesis o preguntas de la investigación realizada o de ambos.

El procesamiento de datos se efectuó mediante el uso de herramientas estadísticas con el apoyo de la computadora, utilizando alguno de los programas estadísticos que hoy fácilmente se encuentran en el mercado.

Para efectuar un procesamiento de datos **Cuando existen datos que merecen tratamientos estadísticos** se siguió los siguientes pasos:

a. Se obtuvo la información de la población o muestra objeto de la investigación.
b. Se definió las variables o criterios para ordenar los datos obtenidos del trabajo de campo.

c. Se definió las herramientas estadísticas y el programa de computadora que va a utilizarse para el procesamiento de datos.

d. Se introdujo los datos en la computadora y activó el programa para que procese la información.

e. Se imprimió los resultados.

3.2. Plan de recolección de la información y/o diseño estadístico.

3.2.1 Población.

La población, la conformaron los usuarios que emplean las redes de datos inalámbricos en el Instituto de Educación Superior Tecnológico Eleazar Guzmán Barrón – Huaraz. En este punto es necesario aclarar que la población, se mantiene constante a través del tiempo.

Para el presente trabajo de tesis, se organizó a la población en grupos, tal como se puede apreciar en la tabla 3.1. Esta población la conformaron, el personal administrativo, quienes acceden al manejo de información sensible (registro académico, controles internos y Sistema de video vigilancia), personal Docente (acceso al portal web Institucional, páginas web con intercambio de claves y páginas de consulta), la comunidad estudiantil (acceso al portal web Institucional, páginas web con intercambio de claves y páginas de consulta), personal de servicio (Sistema de video vigilancia).

La población en estudio, fueron todos los usuarios que hacen uso de la redes en el interior de Instituto Tecnológico “Eleazar Guzmán Barrón – Huaraz”. A continuación, se presenta las características de la población:

Tabla 3.1. Distribución de la población por áreas

Área	Usuarios	Porcentaje
Personal oficina	11	1.21 %
Personal servicio	9	1.00 %
Docentes	78	8.63 %
Estudiantes	806	89.16 %
TOTAL	**904**	**100 %**

3.2.2. Muestra

Con los datos obtenidos en la tabla 3.1, se procedió al cálculo de la muestra, para este fin se empleó la formula descrita en la tabla 3.2.

Tabla 3.2. Coeficientes para el cálculo de la muestra

$n = \frac{N}{[(E^2)(N-1)]+1}$		
n	=	Tamaño de muestra
E	=	Coeficiente de error
E	=	5 %
N	=	Población

Nota. Recuperado de "*Muestreo Estadístico Diseño y Aplicaciones"*, de Vivanco, M., 2005, Santiago de Chile: UNIVERSITARIA.

Obtenido los datos de la población, en la Tabla 3.2, se procedió a calcular el tamaño de la muestra, obteniéndose como resultado una muestra de **n=277**,

Como se puede ver de la tabla 3.1, la mayor parte de la población está conformada por los estudiantes de las diversas Carreras Profesionales, siendo estos últimos la mayoría de los encuestados.

Por otra parte, el personal Administrativo, Docente y de Servicio, conforman una cantidad pequeña con respecto a los estudiantes, por este motivo, son seleccionados en su totalidad para el estudio correspondiente mediante los instrumentos de recolección de información.

3.3. Instrumentos de recolección de la información.

Tabla 3.3. Instrumentos para la recolección de la información

Notas de campo	Fueron obtenidas directamente de la observación, correspondientes al uso en equipos de tecnologías de la información en ambientes inalámbricos.
Lista de cotejo	Permitieron apreciar el uso de las redes inalámbricas, y la seguridad de estas en el Instituto "Eleazar Guzmán Barrón – Huaraz"
Cuestionario de opinión	Correspondientes a la seguridad en el intercambio de archivos por medios inalámbricos.
Escalas de actitud	Correspondientes a materia de seguridad en redes inalámbricas, y la comodidad del usuario al emplearlas.

Las herramientas descritas permitieron al investigador tener un conocimiento de las realidad de en cuanto a materia de seguridad se refiere. La encuesta realizada, permitió realizar el análisis estadístico, y así obtener la información necesaria para la ejecución del presente trabajo.

3.4. Plan de procesamiento y análisis estadístico de la información.

a. Validación y edición

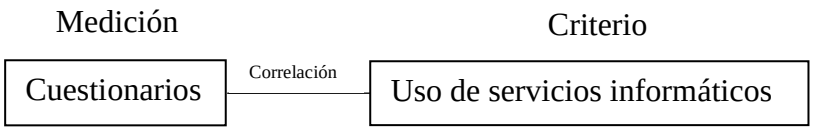
La validez de criterio, se estima al correlacionar la medición con el criterio externo (puntuaciones del instrumento frente a las

puntuaciones en el criterio), y este coeficiente es tomado como coeficiente de validez.

Basado en esta definición, podemos ejemplificar la validez de criterio empleada para el análisis de los datos adquiridos en el presente trabajo.

La figura 3.1. Muestra la validez de criterio para el análisis de los datos.

Figura 3.1. Validez de criterio para el análisis de los datos



b. codificación

Es necesario realizar un consolidado, ya que varias respuestas pueden ser interpretadas como si se tratase de lo mismo, por tal motivo, se procedió a la determinación de los códigos que serán asignadas a cada una de las categorías consolidadas. Finalmente se realizó un enlistado consolidado de todas las respuestas, y se hizo una introducción real de los datos. La Tabla 3.4, muestra el formato empleado para la codificación de los datos.

Tabla 3.4. Formato empleado para la codificación de datos

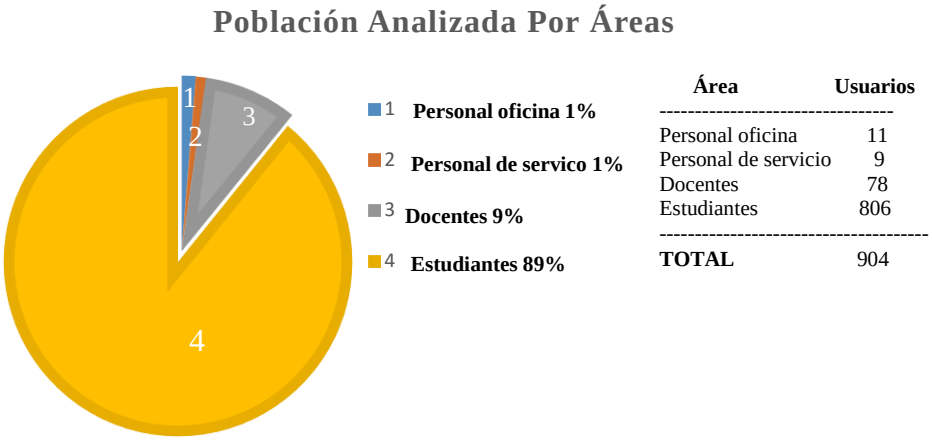
Categoría	codificación
No existente	0
Inicial	1
Repetible	2
Definido	3
Administrado	4
Optimizado	5

El análisis fue llevado a cabo mediante el modelo basado en el análisis de riesgos). Estas permitieron realizar un consolidado de los datos a los entrevistados, y así evitar sesgos.

c. Tabulación y análisis estadístico.

Se empleó plantillas elaboras en Excel (plantillas elaboradas basadas en el modelo de riesgos) , para el análisis de los riegos en el uso de las redes inalámbricas en el interior del Instituto Superior Tecnológico Público “Eleazar Guzmán Barrón” – Huaraz. En la Figura 3.2, se muestra la distribución de la población del Instituto Superior Tecnológico “Eleazar Guzmán Barrón”

Figura 3.2. Distribución de la población del IESTP “EGB” por áreas



La Figura 3.3, muestra los resultados obtenidos, en materia de seguridad de las redes inalámbricas del IESTP “EGB” – Huaraz específicamente en *Seguridad física y del entorno*, antes del desarrollo de los algoritmos criptográficos. Los resultados son comparados con un estado ***óptimo*** al que se pretende llegar.

Figura 3.3. Seguridad física y del entorno

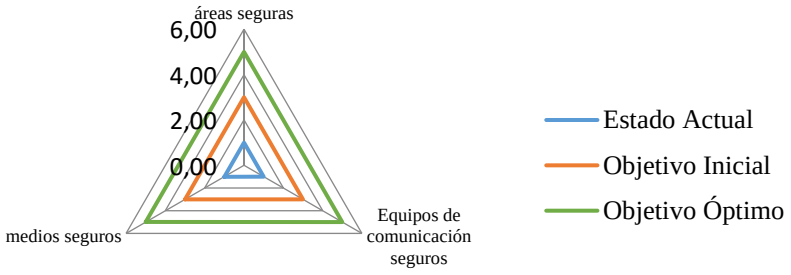


Figura 3.4. Gestión de la comunicación y operaciones

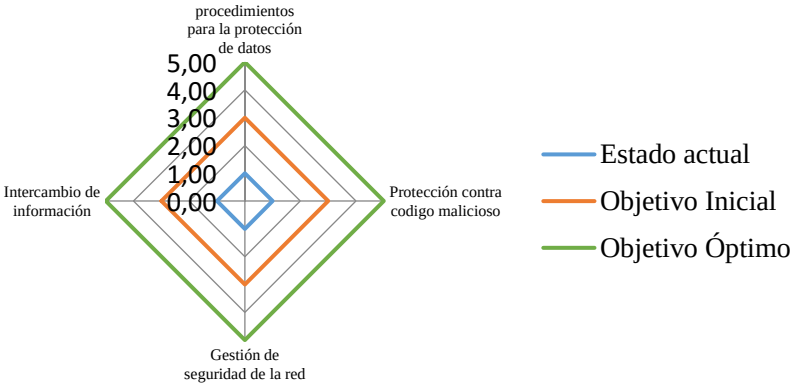
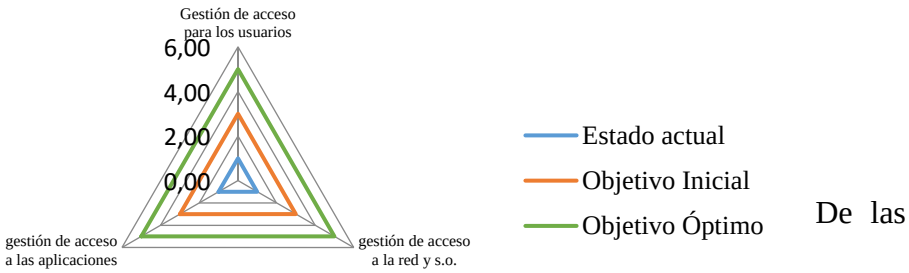


Figura 3.5. Control de acceso



De las figuras obtenidas, se realizó un resumen donde se muestra el estado previo a la implementación de los algoritmos criptográficos empleando el estándar de encriptación avanzado sobre plataformas reconfigurables en las redes inalámbricas del IESTP "EGB" – Huaraz. Las Figuras 3.6 y 3.7, muestran también los valores del objetivo inicial al que se pretende llegar con el desarrollo del presente proyecto de investigación, los gráficos muestran también un objetivo óptimo.

Figura 3.6. Niveles de seguridad en la red del IESTP "EGB"

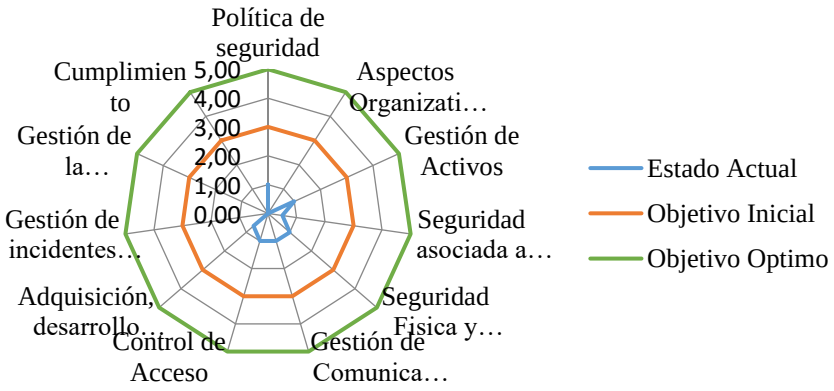
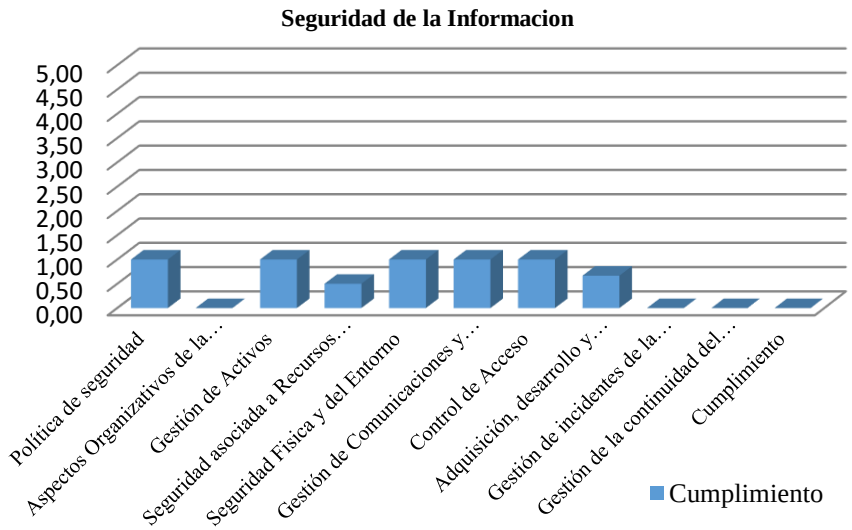


Figura 3.7. Escala de seguridad en la red del IESTP – Huaraz.



Para la obtención de datos y análisis estadístico, se procedió a automatizar los procesos, para tal fin, se recurrió al uso de una aplicación desarrollada en Microsoft Excel.

IV. RESULTADOS.

4.1. PRESENTACION DE RESULTADOS

Los resultados fueron determinados atendiendo básicamente el rendimiento del algoritmo ejecutado y su aplicación al ámbito tecnológico en la ciudad de Huaraz, tomando como piloto al Instituto Superior Tecnológico "Eleazar Guzmán Barrón" del mismo modo, se consideraron también los resultados comparativos en la implementación del algoritmo criptográfico desarrollado frente a otras implementaciones citadas en la presente investigación.

4.1.1 Medida de rendimiento de la FPGA.

Fueron medidos en dos aspectos, espacio empleado y velocidad de procesamiento desarrollada.

- **Área:** Es el espacio ocupado por el diseño, son medidas en slices de CLBs, en los modernos FPGAs, existen otros recursos como los BRAMs, multiplicadores, etc , si bien estos recursos no ocupan espacio debido a que son recursos dedicados, deben ser mencionados, caso contrario no justificaría el ahorro en el espacio en el diseño. Si el resultado es pequeño, este es un diseño económico.
- **Velocidad de procesamiento (throughput)**

  Es necesario medir el rendimiento en tiempo para comprobar la eficiencia del diseño, este valor se denomina throughput, y para los algoritmos criptográficos, se calcula a partir de la siguiente expresión:

$$Throughput = \frac{Frecuencia\ límite\ x\ número\ de\ bits}{número\ de\ rondas}\ (bits/s)\ ..\ (4.1)$$

En la Tabla 4.1, se muestra los resultados para la transmisión de datos sin cifrado, cifrado WEP a 104 bits y el cifrado empleando AES – CCM, todos ellos basados los controladores (software).

4.1.2. Herramientas para la medición del rendimiento en hardware.

Los resultados obtenidos fueron medidos mediante la herramienta de síntesis integrada *Quartus 13.0 sp1*, provista por el fabricante de FPGA *ALTERA (actualmente parte de Intel)*, Las simulaciones correspondientes fueron realizadas con el software ModelSim Altera 10.1d, en su versión académica.

4.1.3. Análisis de las pruebas con tarjetas inalámbricas comerciales.

En este punto, se harán comparativas de las pruebas realizadas en software y hardware con la tarjeta inalámbrica del fabricante D-Link DWL – G520, para la transmisión de datos entre los terminales configurados (Cliente – servidor). Se muestran los resultados obtenidos y las estimaciones de tiempo necesarias para llevar a cabo cada situación propuesta. Se hicieron las comparativas de los resultados obtenidos con el modo de operación CCM empleando AES desarrollado en el presente trabajo de investigación. Se empleó el cifrado WEP a 104 bits, y el cifrado AES con CCM, este procedimiento se desarrolló ejecutando la transmisión de los datos un determinado número de veces, esto con la finalidad de comprobar el tiempo necesario para los intentos y verificar la existencia de pérdidas de paquetes durante la transmisión.

Para las implementaciones **en software**, la Tabla 4.1, concluye que el cifrado CCM empleando AES es el más lento frente a las otras opciones, esto es debido a que en buen medida la operación de cifrar con AES, es significativamente más lenta que hacerlo con el cifrado RC4. Todas

Tabla 4.1. Tiempos y velocidades para las opciones de cifrado en software

Número de veces	Sin cifrar (software)	WEP 104 bits (software)	AES – CCM (software)	WEP 104 bits (Mbps) (software)	AES - CCM (Mbps) (software)
10 000	3.45 s	4.78 s	5.12 s	136.53	69.13
20 000	7.56 s	8.96 s	9.75 s	138.84	78.58
30 000	9.89 s	12.35 s	14.49 s	136.90	74.02
40 000	13.18 s	15.69 s	19.75 s	145.63	73.55
50 000	18.29 s	21.68 s	24.19 s	148.94	76.20
100 000	35.45 s	45.38 s	47.72 s	153.98	74.00
200 000	69.98 s	89.78 s	94.56 s	147.60	75.76
300 000	117.37 s	126.19 s	148.46 s	145.03	74.89
400 000	159.67 s	175.64 s	199.86 s	145.37	74.77
500 000	197.51 s	219.39 s	259.45 s	146.78	75.05

4.1.4. Resultados de la implementación del AES en hardware.

Los resultados mostrados en la Tabla 4.2, muestran que el diseño realizado en el presente trabajo, es económico en cuanto a espacio se refiere, sin embargo el diseño consume considerable cantidad de bloques de memoria RAM, este último permite realizar operaciones de consulta con un acceso más rápido.

De la expresión (4.1), obtenemos el resultado del throughput generado en hardware del estándar de encriptación avanzado (AES).

Periodo mínimo de reloj: ***9 ns*** (empleando PLL)

$$f_{max} = \frac{1}{9\ ns} = 111.11\ MHz \ldots (4.2)$$

Para nuestro diseño se procesa el AES a 128 bits, con un texto cifrado por cada 12 ciclos de reloj, entonces de 4.1, se obtiene para el AES, un throughput de:

$$Throughput_{AES} = \frac{111.11\ MHz\ x\ 128\ bits}{12} = 1185.17\ Mbps \ldots (4.3)$$

4.1.5. Resultados de la implementación del modo CCM en hardware.

Del mismo modo que en los resultados presentados para el cifrado AES, con la ayuda de la herramienta de síntesis *Quartus II*, se muestra los resultados obtenidos para este tipo de cifrado.

La Tabla 4.2, muestra los valores de slices (compuertas), y BRAMs (bloques de memoria RAM), empleados para cada uno de los bloques que conforman al modo de operación CCM. Se encuentran también incluidos los resultados correspondientes al cifrado AES, motivo por el cual, los bloques de autenticación y cifrado emplean 56 bloques de memoria cada una.

Tabla 4.2. Recursos empleados para la implementación de AES Y CCM

AES		**CCM**		
Recurso	**Valor empleado**	**Bloque**	**Logic Elements (LEs)**	**BRAMs**
Puertos de E/S	148	Autenticación	1020	56
BRAMs	56	Cifrado	659	56
Logic Elements (LEs)	598 (448+150)	Unidad control y hardware extra	412	N/A
		Operación CCM completo	2091	112
Periodo mínimo de reloj	9 ns	Periodo mínimo de reloj		9 ns

Al igual que el AES, el periodo del reloj es 9 ns, entonces para el modo cifrado por bloques (CCM) se tomaron las siguientes consideraciones:

- Cada bloque es procesado en 12 ciclos de reloj, en CCM, se emplean 67, por tanto el total toma 804 ciclos de reloj.

- El total de bits a procesar es de 66 x 128 bits, lo que hace un total de 8848 bits, no se considera el bloque *overhead*.

Bajo estas consideraciones, el throughput obtenido para el modo cifrado por bloques (CCM), es:

$$Throughput_{CCM} = \frac{66\ x\ 128\ bits}{804\ \ bits\ x\ 9\ ns} = 1.1675\ Gbps \ldots (4.4)$$

Como puede observarse, el valor obtenido del throughput es relativamente alto, con respecto a los antecedentes citados en el presente trabajo, logrando en este sentido la optimización del diseño criptográfico cifrado por bloques empleando el estándar de encriptación avanzado sobre hardware reconfigurable, y estos resultados ser aplicados al ámbito educativo tecnológico en la ciudad de Huaraz.

4.1.6. Comparativa de resultados.

Con la finalidad de comprobar la eficiencia del diseño planteado en el presente trabajo, se comparó los resultados obtenidos sobre hardware reconfigurable con las implementaciones en software. En este sentido, se comparan los resultados de nuestro diseño sobre hardware reconfigurable en (4.3) y (4.4), y su implementación en software con la tarjeta D-Link DWL – G520 en 4.1. La cantidad de bits procesados tiene la expresión de la ecuación (4.5), y el tiempo necesario para el procesamiento, la ecuación (4.6). El valor de overhead (*ov*) es 48 bytes por cada paquete de1024 bytes.

$$nbits = n_{veces}\, x\, (1024 + ov)\, x\, 8 \ldots (4.5)$$

$$t_{proc} = \frac{nbits}{Throughput} \ldots (4.6)$$

Tabla 4.3. Resultados comparativos entre software y hardware.

Número de veces	AES – CCM (software)	AES – CCM (hardware)
10 000	1,24 s	0,07 s
20 000	2,18 s	0,15 s
30 000	3,48 s	0,22 s
40 000	4,66 s	0,29 s
50 000	5,63 s	0,37 s
100 000	11,59 s	0,73 s
200 000	22,64 s	1,47 s
300 000	34,35 s	2,2 s
400 000	45,88 s	2,94 s
500 000	57,14 s	3,67 s

4.2. PRUEBA DE LA HIPOTESIS.

El desarrollo del presente trabajo de investigación permitió corroborar la hipótesis planteada inicialmente, esto debido a que el desarrollo de algoritmos criptográficos sobre hardware reconfigurable del modo cifrado por bloques empleando el estándar de encriptación avanzado, fue capaz de lograr los objetivos previstos, referentes a la optimización de aplicación del estándar de encriptación avanzado (AES).

V. DISCUSIÓN

Se analizó y discutió los resultados obtenidos en la sección anterior, contrastando así, la hipótesis de trabajo presentada. Posteriormente se analizó los resultados obtenidos en el presente trabajo de investigación comentándose las diferencias o similitudes halladas.

5.1. Desarrollo de algoritmos criptográficos y su capacidad de mejorar la seguridad en las redes de información.

Se expuso, en toda su dimensión que es factible la optimización del estándar de encriptación avanzado sobre hardware reconfigurable del modo cifrado por bloques, verificándose tales resultados sobre la red del IESTP "Eleazar Guzmán Barrón" – Huaraz.

Basándose en esta información, la hipótesis de trabajo planteada en la presente investigación, relacionaba la aplicación de algoritmos criptográficos y su optimización con el estándar de encriptación avanzado (AES), aplicando los resultados al ámbito educativo, para comprobar la eficiencia del diseño criptográfico.

Los algoritmos criptográficos proporcionan la seguridad necesaria para la transferencia de información sobre diversos medios de comunicación. Como se pudo observar durante la ejecución del presente trabajo de investigación, los algoritmos criptográficos tienden a debilitarse (tal como es el caso de el algoritmo criptográfico WEP), esto debido en gran medida a la simplicidad de las operaciones matemáticas que sustentan el funcionamiento del algoritmo. En este

sentido, se analizó, los modernos algoritmos criptográficos con los que se disponen hoy en día, encontrando en la bibliografía una notable mejora en los niveles de seguridad proporcionados por estos últimos.

En tal sentido, nuestra hipótesis de trabajo posee los argumentos necesarios para comprobar la optimización del estándar de encriptación avanzado sobre hardware reconfigurable, y los resultados ser empleados en la red del IESTP Eleazar Guzmán Barrón – Huaraz.

5.2. Plataformas para el desarrollo de algoritmos criptográficos.

Se ha expuesto, sobre las diversas plataformas existentes para la implementación de algoritmos criptográficos, y como estos pueden desempeñarse de manera eficiente dicha tarea.

Teniendo como referencia esta información, la hipótesis de trabajo, seleccionaba la plataforma más eficiente para el desarrollo de algoritmos criptográficos.

Las plataformas estudiadas, poseen diversas características, de las cuales podemos mencionar sus ventajas y desventajas frente a ellas.

Las plataformas basadas en software proporcionan en baja medida una solución a la problemática, ya que la velocidad de procesamiento bajo esta alternativa es muy baja, lo cual involucra una tasa baja de trasferencia de información.

Se estudiaron también, las plataformas basadas en hardware, estas mostraron un alto desempeño en la ejecución de los algoritmos criptográficos, mostrando en los resultados velocidades muy elevadas

y tiempos muy reducidos de procesamiento, lo cual ocasiona que el sistema resultante posea una eficiencia elevada frente a su contraparte desarrollada en hardware. Podemos mencionar también la adaptabilidad de esta plataforma si se implementase en su interior cualquier otro algoritmo criptográfico en el menor tiempo posible, este factor es el más sobresaliente frente todas las plataformas estudiadas.

En este sentido, la hipótesis de trabajo posee los argumentos necesarios para justificar el empleo de plataformas basadas en hardware para atender la problemática en el área de estudio.

5.3. Comparativas con los antecedentes.

Se hace una comparativa con los antecedentes citados en el desarrollo del presente trabajo de investigación, y estos son medidos en cuanto a velocidad, área de diseño y fabricantes. Véase la tabla 4.4.

Las comparativas son realizadas con los antecedentes citados en la investigación, especialmente los que desarrollaron algoritmos criptográficos sobre hardware reconfigurable.

Tabla 4.4. Comparativas de la investigación con los antecedentes.

ANTECEDENTES		**Aporte de nuestra investigación**
Autor	**Alcance**	
Medina Aparcana	Se realiza un análisis de los ataques al algoritmo de curvas elípticas sobre cuerpos *p* – ádicos, para optimizar el desarrollo de un algoritmo rápido y efectivo implementado en software.	Se demuestra que es factible aplicar el AES con el modo CCM sobre hardware reconfigurable (FPGA), mejorando aún más la eficiencia que su contraparte en software.
León Lomparte	Realiza la implementación en software del algoritmo RSA sobre JAVA para cifrar y descifrar archivos.	El empleo del algoritmo de estándar de encriptación avanzado (AES) sobre FPGA, proporciona una manera más eficiente y segura en el proceso de cifrar y descifrar información que su contraparte en software.
Khoa & Zier	Desarrollan la implementación del AES en FPGA con un sistema fuera del microprocesador para la aceleración de los procesos a una frecuencia de 50 MHz.	El AES con el modo CCM, sobre FPGA, fue desarrollado fuera del procesador, aliviando así la carga de procesamiento a una frecuencia de 111.11 MHz.
Nazzar Abbas	Demuestran la factibilidad de implementación de algoritmos criptográficos simétricos y asimétricos con alta eficiencia sobre hardware reconfigurable, implementan el Data Encription Standard (DES), estándar de encriptación avanzado (AES), (ambos simétricos), y criptografía de curvas elípticas (asimétricos).	El estándar de encriptación avanzado (AES), con el modo cifrado por bloques (CCM), fue desarrollado satisfactoriamente sobre hardware reconfigurable (FPGA), con una eficiencia en **velocidad de 1.17 Gbps**. El desarrollo del algoritmo AES con el CCM, fue sobre la plataforma reconfigurable (Altera Cyclone EP4 ep4ce22f17c6n). El procesamiento fue llevado a cabo fuera del procesador de la tarjeta inalámbrica D – Link DWL – G520, puesto que nuestro desarrollo, se encargó de dicha tarea aumentando así considerablemente la velocidad de procesamiento
Segredo Zabala & Bellora	Implementan el AES sobre hardware reconfigurable (Virtex XCV 100), logrando una velocidad de procesamiento de 742 Mbps	
Gladman, Karri & Kim	Los autores concluyen que las implementaciones en hardware ofrece mayor seguridad en mantener la privacidad de la clave con una instalación sencilla y alta eficiencia frente a implementaciones en software.	
Trejo	El autor implementa el estándar de encriptación avanzado sobre plataformas reconfigurables, logrando una velocidad de 1051 Mbps	
Mali, Novak & Biasizzo	Los autores realizan la implementación del estándar de encriptación avanzado sobre hardware reconfigurable para una aplicación de almacenamiento externo con una velocidad de 182.86 Mbps.	

ANTECEDENTES		Aporte de nuestra investigación
Autor	**Alcance**	
Badillo, Feregrino, Cumplido & Morales	Los autores demuestran que las arquitecturas basadas en hardware muestran mejor rendimiento y eficiencia frente a desarrollos obtenidos en software. Se hacen comparativas del rendimiento obtenido en recursos empleados (memoria RAM, elementos lógicos y matrices lógicas. Se obtiene una velocidad de 1.087 Gbps a una frecuencia de 86.34 MHz, y 1.951 Gbps a 152.42 MHz para los algortimos DES, RSA.	El estándar de encriptación avanzado (AES), con el modo cifrado por bloques (CCM), fue desarrollado satisfactoriamente sobre hardware reconfigurable (FPGA), con una eficiencia en **velocidad de 1.17 Gbps**. El desarrollo del algoritmo AES con el CCM, fue sobre la plataforma reconfigurable (Altera Cyclone EP4 ep4ce22f17c6n). El procesamiento fue llevado a cabo fuera del procesador de la tarjeta inalámbrica D – Link DWL – G520, puesto que nuestro desarrollo, se encargó de dicha tarea aumentando así considerablemente la velocidad de procesamiento
Velásquez y Castaño	Desarrollan una infraestructura de clave pública sobre hardware reconfigurable para el criptosistema de curvas elípticas integrando el algoritmo AES para cifrado simétrico y SHA como algoritmo de integridad de la información lográndose velocidades de 660 Mbps.	

VI. CONCLUSIONES

En esta sección se presentan las conclusiones obtenidas. En primer lugar se muestra un resumen de los resultados alcanzados en el presente trabajo, seguidamente se brindan las conclusiones obtenidas analizando dichos resultados, y orientando estos últimos a la aplicación de los algoritmos criptográficos al ambiente Tecnológico.

6.1. Resultados obtenidos

El desarrollo del presente trabajo, permitió obtener los siguientes resultados.

- El algoritmo de cifrado por bloques empleado AES, posee una estructura iterativa, lo cual permite una fácil implementación bajo hardware reconfigurable, y así ser competitiva con otras aplicaciones similares desarrolladas en la literatura citada tanto en hardware como software.
- La operación del modo CCM, el cual es empleado en el estándar 802.11i, fue realizada de manera eficiente, mostrando en el software de medición empleado, resultados muy competitivos con soluciones disponibles comercialmente.
- Se realizó el análisis comparativo mediante tarjetas inalámbricas comerciales del fabricante D – Link, para poder ver el grado de eficiencia de la implementación en hardware reconfigurable, notándose así una eficiencia mayor que su contraparte lograda en software.
- Se realizó los procedimientos para la paralelización de los procesos de autentificación y cifrado del modo CCM, con esto se logró una reducción significativa en el tiempo de procesamiento de la información. Así mismo, las operaciones de cifrado y descifrado fueron llevados en un solo diseño.

6.2. Conclusiones de la investigación.

- Durante el desarrollo del presente trabajo, se optimizo el uso del estándar de encriptación avanzado del modo cifrado por bloques (CCM) sobre hardware reconfigurable (FPGA), con una clave de cifrado de 128 bits (para poder trabajar con el CCM), obteniéndose una eficiencia en velocidad de 1.1617 Gbps. En este sentido, la investigación permitió comprobar que el diseño de algoritmos criptográficos en especial el AES con el CCM no solo pueden ser implementadas en software, sino que estas tienen un mejor desempeño si son implementadas en hardware.
- El lenguaje de descripción de hardware (VHDL), permitió de manera fluida desarrollar el algoritmo de cifrado por bloques empleando el estándar de encriptación avanzado, optimizando no solo área de diseño, sino también velocidad de procesamiento, gracias a la paralelizacion de la data a procesar.
- Con la optimización obtenida en el desarrollo del algoritmo criptográfico sobre hardware reconfigurable del modo cifrado por bloques empleando el estándar de encriptación avanzado, se notó un incremento notable de la eficiencia, sobre todo si los diseños son realizados sobre hardware, esto lo corroboró las pruebas realizadas en la red del IESTP Eleazar Guzmán Barrón – Huaraz, lográndose una fluidez en el intercambio de información cifrada.

VII. RECOMENDACIONES

Dentro de los límites de la presente investigación, siempre se desea que exista una mejora continua del mismo; por tanto se recomienda a futuros investigadores continuar con la investigación sobre esta interesante temática, logrando así la complementación con mayores aplicaciones a diversas ramas donde la seguridad de la información es un factor importante. Entre estas citamos:

- El Desarrollo de un procesador criptográfico dedicado, permitiendo la concurrencia del procesamiento de datos para el cifrado a fin de incrementar los niveles de eficiencia en cuanto a velocidad y área, tal como fueron descritos en el desarrollo de la presente investigación.
- El empleo de los algoritmos sobre hardware reconfigurable a redes inalámbricas WLAN donde se desea brindar altos niveles de seguridad (intercambio de llaves, manejo de información sensible, etc.), brindando interoperabilidad sin modificar el hardware existente.
- La aplicación de los algoritmos criptográficos sobre hardware reconfigurable a las comunicaciones móviles (red de telefonía móvil), desarrollando Gateway o puertas de enlace (APN), basadas en plataformas reconfigurables para incrementar los niveles de seguridad durante el intercambio de información, apoyando en este sentido a implementaciones desarrolladas en software.
- Finalmente se recomienda el empleo del hardware reconfigurable al ámbito educativo, para mostrar la potencialidad en el empleo de estas herramientas. Podemos citar como ejemplo: el desarrollo de todo tipo de circuitos digitales

e incluso la implementación de procesadores embebidos, este último origina el desarrollo de aplicaciones con altas prestaciones (procesador digital de señales, analizadores de espectro, equipamiento biomédico), los cuales son aplicables a diferentes campos de la Medicina e Ingeniería.

VIII. REFERENCIAS BIBLIOGRAFICAS.

1. Sabino, C. (1986). *El Proceso de Investigación.* Caracas: Panapo.

2. Bunge, M. (1996). *Ética, Ciencia y Técnica*. México: Sudamericana

3. Medina, R. (2012). *Criptografía con Curvas Elípticas sobre cuerpos p-ádicos*. Lima: Universidad Nacional de Ingeniería

4. Léon K. (2005). *Encriptación RSA de archivos de texto*. Lima: Universidad Pontificia Católica del Perú.

5. Khoa, V., & Zier, D. (2003). FPGA Implementation AES for CCM Mode Encryption Using Xilinx Spartan-II. *ECE 679, ADVANCED CRYPTOGRAPHY, OREGON STATE UNIVERSITY*, 1-5.

6. Nazzar Abbas, S. (2004). *Implementación Eficiente de Algoritmos Criptográficos en Dispositivos de Hardware Reconfigurable.* Tesis Doctoral, Instituto Politécnico Nacional, Departamento de Ingeniería Eléctrica, México.

7. Segredo, A., Zabala, E., & Bellora, G. (2004). *DISEÑO DE UN PROCESADOR CRIPTOGRÁFICO RIJNDAEL EN FPGA.* Artículo Científico, Universidad ORT , Facultad de Ingeniería, Uruguay.

8. Gladman, B. (s.f.). *Implementation of Rijndael in C++ and C*. Recuperado el 21 de Febrero de 2017, de http://www.nist.gov/cgi-bin/exit_nist.cgi?url=http://fp.gladman.plus.com/cryptography_technology/rijndael/

9. Karri, R., & Kim, Y. (s.f.). *FPGA Implementation of AES Rijndael, Comparison of different Rijndael Implementations*. Recuperado el 21 de Febrero de 2017, de http://eeweb.poly.edu/dream-it/publications/Rijndael.pdf

10. Trejo, E. L. (2004). *Implementación eficiente en FPGA del modo CCM usando AES.* Tesis de Maestría, Instituto Politécnico Nacional, Departamento de Ingeniería Eléctrica , México.

11. Mali, M., Novak, F., & Biasizzo, A. (2005). Hardware Implementation Of AES Algorithm. *ELECTRICAL ENGINEERING*, 265-269.

12. Badillo, I., Feregrino, C., Cumplido, R., & Morales, M. (2008). FPGA Implementation and Performance Evaluation of AES-CCM Cores for Wireless Networks. *2008 International Conference on Reconfigurable Computing and FPGAs*.

13. Velásquez, F., & Castaño, J. (2011). IMPLEMENTACIONES CRIPTOGRÁFICAS EN FPGA. *Visión Electrónica*.

14. A. Menezes, P. V. (2001). *Hanbook of Applied Cryptography.* Boca Raton FL: CRC Press.

15. Costas Santos, J. (2010). Seguridad Informática. Madrid: Ra-Ma.

16. Joan, D., & Rijmen, V. (2002). *The Design of Rijndael, AES - The Advanced Encryption Standard.* New York: Springer.

17. NIST. (2001). Recuperado el 11 de Marzo de 2015, de Announcing the ADVANCED ENCRYPTION STANDARD (AES). Federal Information Standards Publication: http://csrc.nist.gov/archive/aes/

18. Nichols, K., & Lekkas, C. (2002). *Wireless Security Models, Threats and Solutions* (Primera ed.). McGraw - Hill .

19. Andreu, F., Pellejero, I., & Lesta, A. (2006). *Redes WLAN Fundamentos y Aplicaciones de Seguridad.* Barcelona: Marcombo.

20. Mallik, M. (2003). *Mobile and Wireless Design Essentials.* WILEY.

21. Budris, P. (2011). Administrador de Redes Windows. *USERS*, 31 - 33.

22. Perahia, E., & Stacey, R. (2013). *Next Generation Wireless LANs 802.11n and 802.11ac.* United Kingdom: Cambridge University Press.

23. Erickson, J. (2003). *HACKING the art of exploitation.* San Francisco: No Starch Press.

24. Cam-Winget, N., Housley, R., Wagner, D., & Walker, J. (2006). Security Flaws in 802.11Data Link Protocols. *In Communications of the ACM*, 35-39.

25. Fluhrer, S., Mantin, I., & Shamir, A. (2001). Weaknesses in the Key-Scheduling Algorithm of the RC4. *Eighth Annual Workshop on Selected Areas in Criptography*, 1-2.

26. Vebjorn, M., Havard, R., & Hole, K. (2004). Weaknesses in the Temporal Key Hash of WPA. *ACM SIGMOBILE Mobile Computing and Communications Review*, 76-83.

27. Doug, W., Housley, R., & Ferguson, N. (2002). Counter with CBC-MAC (CCM). *NIST*.

28. Wollinger, T., & Christof, P. (2003). How Secure Are FPGAs In Cryptographic Aplications. En *Field - Programable Logic and Applications* (págs. 91-100).

29. Wollinger, T., Guajaro, J., & Paar, C. (2004). Security on FPGAs: State - of - the - art implementations and attacks. En *ACM Transactions on Embedded Computing Systems (TECS)* (págs. 534-574).

30. Moriello, S. (2001). *Inteligencias Sinteticas.* Buenos Aires: ALSINA.

31. Villasenor, J., & Mangione-Smith, W. (1997). *Configurable Computing. Scientific American.*

32. National Instruments. (2012). Recuperado el 12 de Marzo de 2015, de FPGAs a fondo: http://www.ni.com/white-paper/6983/es/

33. ETSI Technical Specification Access transmission systems on metallic access. (2015). Recuperado el 12 de Marzo de 2015, de ETSI Technical Specification Access transmission systems on metallic access: http://www.etsi.org/deliver/etsi_ts/101200_101299/10127001/01.02.01_60/ts_10127001v010201p.pdf

34. Daemen, J., & Vincent, R. (1999). Recuperado el 14 de Marzo de 2015, de http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf

35. Jonsson, J. (2002). On the Security of CTR + CBC-MAC. *Proceedings of Select Areas in Cryptography - SAC, 2595*.

36. Struick, R. (Octubre de 2003). *Comments NIST Draft Pub 800 - 38C. Technical Report, Certicom.* Obtenido de http://csrc.nist.gov/CryptoToolkit/modes/comments/

37. Mano, M. M. (2003). *Diseño Digital* (Tercera ed.). Mexico: Prentice Hall.

38. Alfonso, S., Soto, E., & Fernández, S. (2002). *Diseño de Sistemas Digitales con VHDL* (Primera ed.). España: Paraninfo.

39. J., A. P. (1990). *The VHDL Cookbook* (Primera ed.). Australia: Computer Science University of Adelaide.

40. Sánchez Santiago, M. (Julio de 2003). *Implementación en Hardware-Software del algoritmo criptográfico DES.* Tesis de Maestría, Centro de investigación Mexicano, CINVESTAV, Mexico.

41. Argüelles Cruz, A., Ascencio Roman, J., & Villalobos Baigorria, J. (2001). Diseño de Sistemas Digitales Utilizando FPGA. *Polibits*, 15-20.

42. (2004). Recuperado el 18 de Abril de 2015, de The madwifi project: http://madwifi-project.org/wiki/UserDocs/Distro/RedHat.

43. Vivanco, M. (2005). *Muestreo Estadístico Diseño y Aplicaciones*. Santiago de Chile: UNIVERSITARIA

ANEXOS

Detalle de las pruebas realizadas

Requerimientos del sistema

Se emplearon dos tarjetas de red de la marca D – Link, instaladas en dos computadoras de escritorio con las siguientes características:

- Sistema operativo: Fedora 21 Workstation 32 bits
- Procesador: Pentium 4
- Memoria RAM: 2 GB
- Disco duro: 180 Gb

Instalación de los drivers en el entorno Linux

Para trabajar correctamente fue necesario descargar los drivers necesarios, para que la tarjeta se comporte como un punto de acceso inalámbrico, para tal fin se empleó los drivers *madwifi (Atheros ar521x),* dado a que la tarjeta inalámbrica *D – Link DWL – G520,* posee un chipset Atheros.

El software permite operar a la tarjeta inalámbrica en modo AP (punto de acceso), su uso también proporciona varias ventajas, entre ellas podemos mencionar soporte completo para *WPA* y el estándar *IEEE 802.11*, es este último se encuentran los algoritmos criptográficos más robustos (CCMP, AES con CBC-MAC), los cuales reemplazan a *TKIP*.

Para el proceso de instalación, será necesario instalar el archivo *madwifi*-i.386 (x86 en nuestro caso, por emplear una plataforma de 32 bits), así

mismo también se debe instalar el módulo kernel: *madwifi-kmdl-*.rpm* y *madwifi-hal-kmdl-*.rpm*

Bajo Fedora, se empleó la siguiente instrucción:

- *apt install madwifi madwifi-kmdl-`uname-r`*, donde *uname-r*, hace referencia al *kmdl* a descargar.

Para crear una interfaz en modo estación ath0, se emplea los siguientes comandos:

- *install ath0 /sbin/modprobe ath_pci; /usr/bin/wlanconfig ath0 create wlandev wifi0 wlanmode sta*

  remove ath0 /usr/bin/wlanconfig ath0 destroy; /sbin/modprobe -r ath_pci options ath_pci autocreate=sta

Para iniciar los servicios

 # this references ifcfg-ath0-home

/etc/init.d/wireless start home

 # this references ifcfg-ath0-work-division-b

/etc/init.d/wireless start work-division-b

 # this references ifcfg-ath0-$DEFAULT_LOCATION

/etc/init.d/wireless start

Se configura los parámetros que identificaran al punto de acceso en ath0:

 # Atheros Communications, Inc.|AR5212 802.11abg NIC

DEVICE=ath0

ONBOOT=no

BOOTPROTO=dhcp

TYPE=wireless

ESSID="Prueba"

KEY=rendimiento

MODE=Managed

RATE=auto

IWPRIV="authmode 2"

Software para la realización de las pruebas

Cabe mencionar que en la bibliografía empleada, se recurrió en gran medida al uso de sockets empleando el esquema cliente-servidor, en tal sentido siguiendo con esta tendencia, el presente trabajo se desarrolló empleando dicha metodología. La aplicación fue programada en GNU-C, obteniéndose dos archivos que se detallan a continuación.

Programa servidor

Esta aplicación se ejecuta en una computadora y es utilizado como "servidor". Sus tareas son las siguientes:

- Abre un socket para establecer conexión con el cliente.
- Aceptada la conexión proveniente del cliente, recibe la información con un *buffer* de transmisión de 1024 bytes.

- Envía un mensaje de confirmación al cliente cuando se recibió un *buffer* para que continúe la transmisión.
- Cuando la transmisión ha concluido, envía un mensaje “datos recibidos”
- El servidor vuelve a la espera de una conexión entrante.

Programa cliente

Esta aplicación es ejecutada en otra computadora y se emplea como “cliente”. Se describe sus principales funciones:

- Crear un proceso denominado hijo, el cual llevará a cabo la tarea de enviar la información.
- Iniciar el contador de tiempo de transmisión.
- Ejecutar el proceso hijo, y este se encargara de las siguientes funciones:
- Abrir un socket para establecer una comunicación con el servidor por el puerto correspondiente.
- Establecida la conexión, realiza la tarea de transmisión.
- Enviada la información contenida en un *buffer* espera el mensaje de confirmación del servidor para leer el siguiente bloque de información.
- Al recibir el mensaje de confirmación, se continua con el envió de información hasta enviar el *buffer*, el número de veces especificado en los parámetros de entrada.
- Al terminar el proceso de envió, espera el mensaje de confirmación final “datos recibidos”.

Al finalizar, el proceso hijo genera un archivo reporte especificado en los parámetros.

Reporte

El archivo de reporte muestra un formato sencillo, en este solo se almacena el número de *Kb* enviados y el tiempo que duro la transmisión. Al momento de ejecutar el *programa cliente*, se abre el archivo reporte indicado, y en caso este no existiese, se recurre a dos opciones: escribir los datos al final de este, o de lo contrario crear un reporte nuevo.

Hoja de recolección de datos.

Se recurrió al uso de una plantilla en Excel para cuantificar el estado del estudio posterior a los diseños criptográficos, así mismo se muestra también de manera gráfica los valores obtenidos formato de recolección de datos.

Figura AN01. Adquisición de la información basada en el modelo de madurez en la red inalámbrica del IESTP “Eleazar Guzmán Barrón” – Huaraz.

DATA SECURITY MATURITY MODEL[SM]

Instructions: In each row where there is an 'x' in a yellow cell, delete that 'x' and type an 'x' in the cell in that same row that best approximates the status of the business unit or organization you are assessing.

Security Maturity Levels >	0: No existente	1: Inicial	2: Repetible	3: Definido	4: Administrado	5: Optimizado
Maturity Level Description >	*No existe evidencia del estándar o practica en la compañía.*	*La organización tiene practicas hechas a la medida pero inconsistentes.*	*La organización tiene un enfoque coherente pero no documentado.*	*Se tiene un enfoque coherente y documentado pero no medido.*	*Los procesos son medidos frecuente mente y se realizan mejoras.*	*La organización ha refinado su cumplimiento con el nivel de las mejores prácticas.*
process consistency	none	ad hoc	consistent	consistent	consistent	consistent
process documentation	none	none	minimal, high-level	detailed	detailed	detailed
business objectives	not met	not met	partially met	mostly met	fully met	value added
process measurement	none	none	none	ad hoc	routine	systemic
policy enforcement	none	none	none	ad hoc	routine	systemic
process improvement	none	ad hoc	ad hoc	ad hoc	routine	systemic
process benchmarking	none	none	none	ad hoc	ad hoc	routine
Corresponding Level of Risk of a Data Breach or Regulatory Noncompliance	*Very high across the organization*	*High across the organization, and very high in key parts of the organization*	*Moderate across the organization, with some pockets of high risk*	*Moderate across the organization.*	*Low across the organization.*	*Remote across the organization.*

ISO #	Domains
5	Política de seguridad
6	Aspectos Organizativos de la seguridad de la información
7	Gestión de Activos
8	Seguridad asociada a Recursos Humanos
9	Seguridad Fisica y del Entorno
10	Gestión de Comunicaciones y operaciones
11	Control de Acceso
12	Adquisición, desarrollo y mantenimiento de Sistemas de Información
13	Gestión de incidentes de la seguridad de la Información
14	Gestión de la continuidad del negocio
15	Cumplimiento

Figura AN02. Estado de los activos de información del IESTP “Eleazar Guzmán Barrón” – Huaraz, basados en el modelo de madurez.

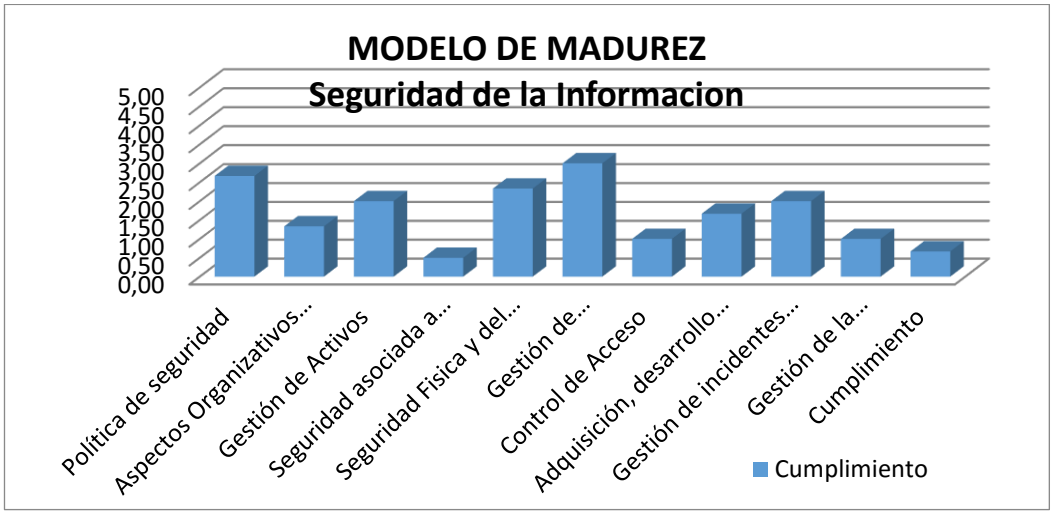


Figura AN03. Políticas de seguridad de la información en el IESTP “Eleazar Guzmán Barrón” – Huaraz.

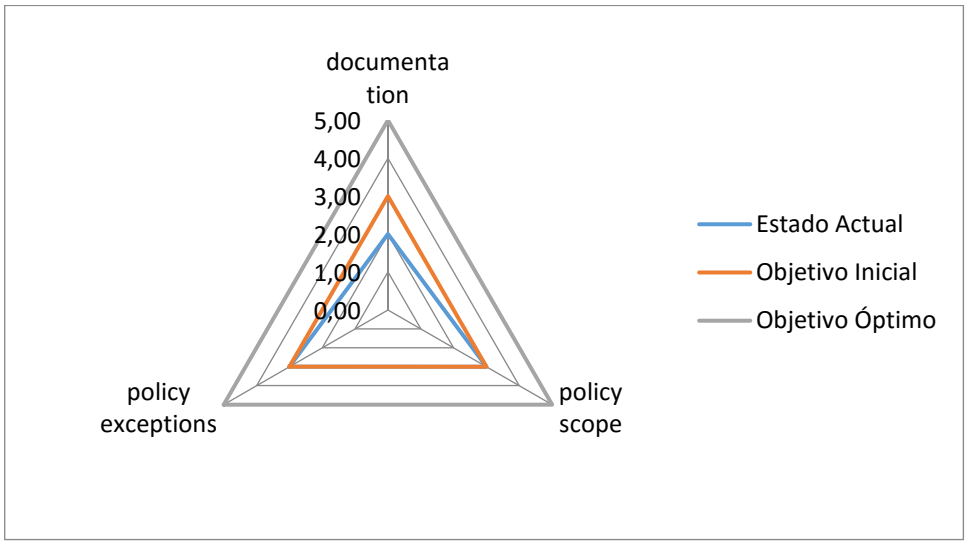


Figura AN04. Organización de la información en las instalaciones del IESTP “Eleazar Guzmán Barrón” – Huaraz.

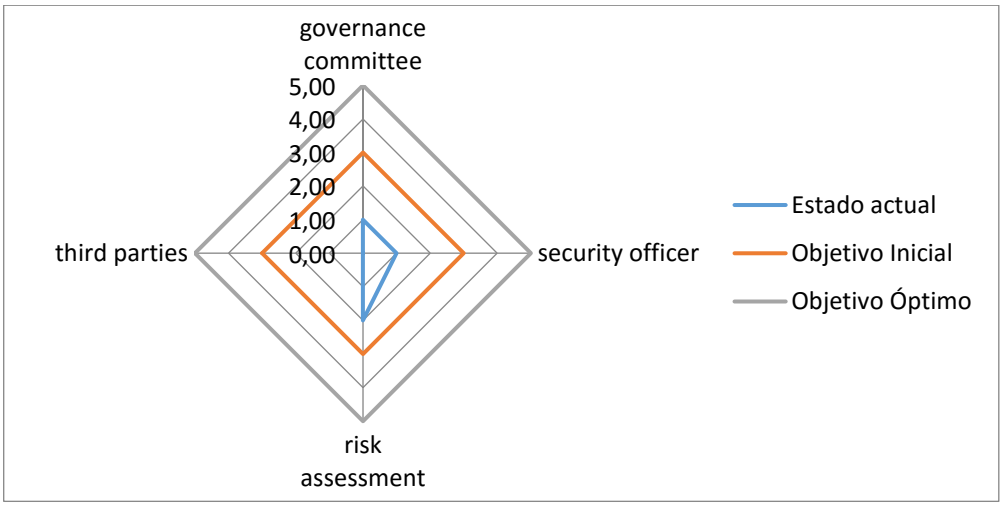


Figura AN05. Gestión de la información en las instalaciones del IESTP “Eleazar Guzmán Barrón” – Huaraz.

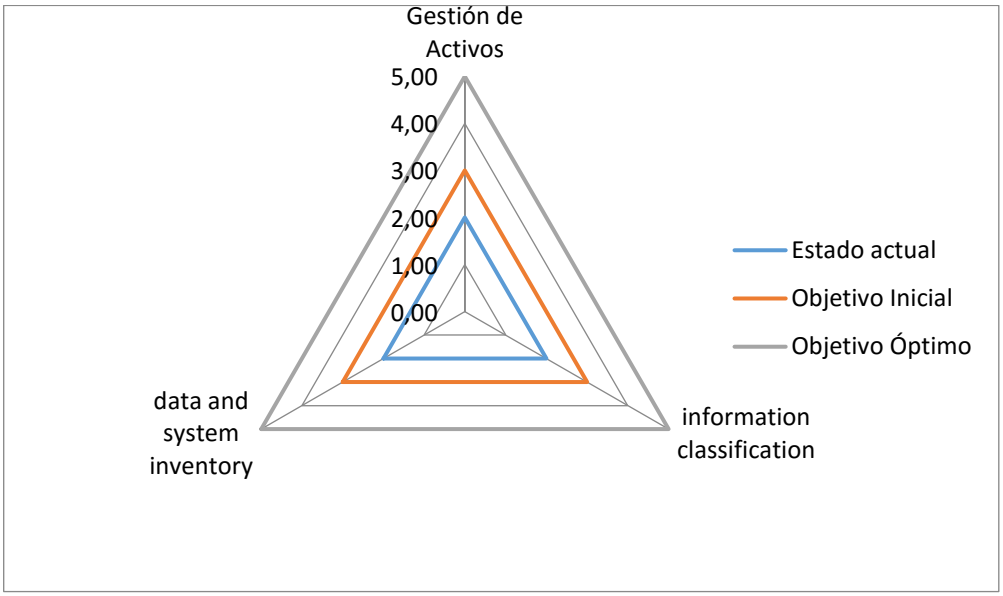


Figura AN06. Seguridad por parte de los usuarios en la red inalámbrica del IESTP “Eleazar Guzmán Barrón” – Huaraz.

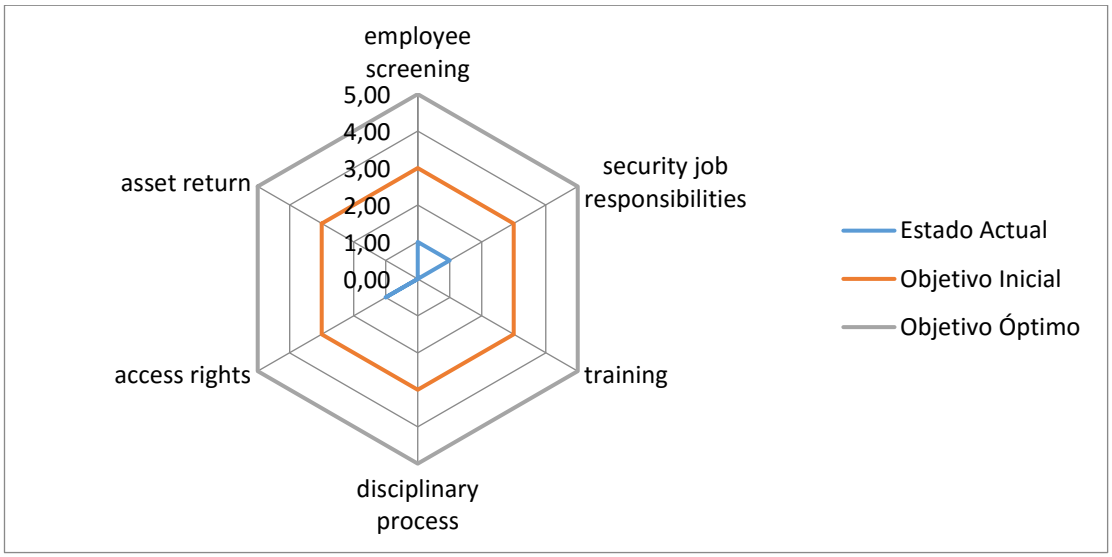


Figura AN07. Seguridad física y del entorno en la red inalámbrica del IESTP "Eleazar Guzmán Barrón" – Huaraz.

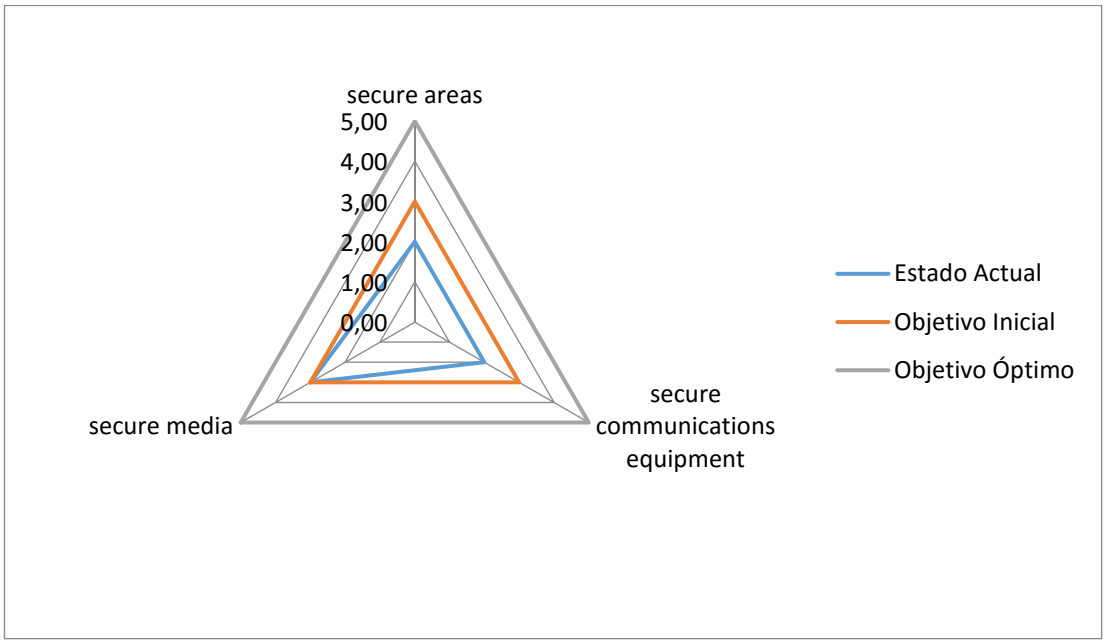


Figura AN08. Procedimientos operacionales para la protección en la red inalámbrica del IESTP "Eleazar Guzmán Barrón" – Huaraz.

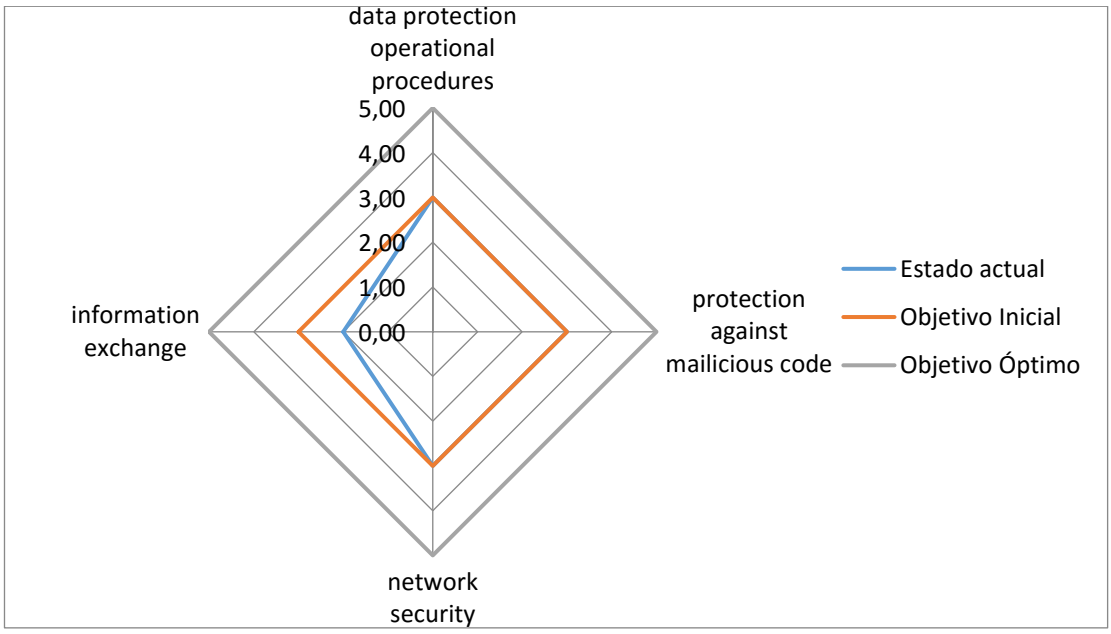


Figura AN09. Control de acceso en la red inalámbrica del IESTP “Eleazar Guzmán Barrón” – Huaraz.

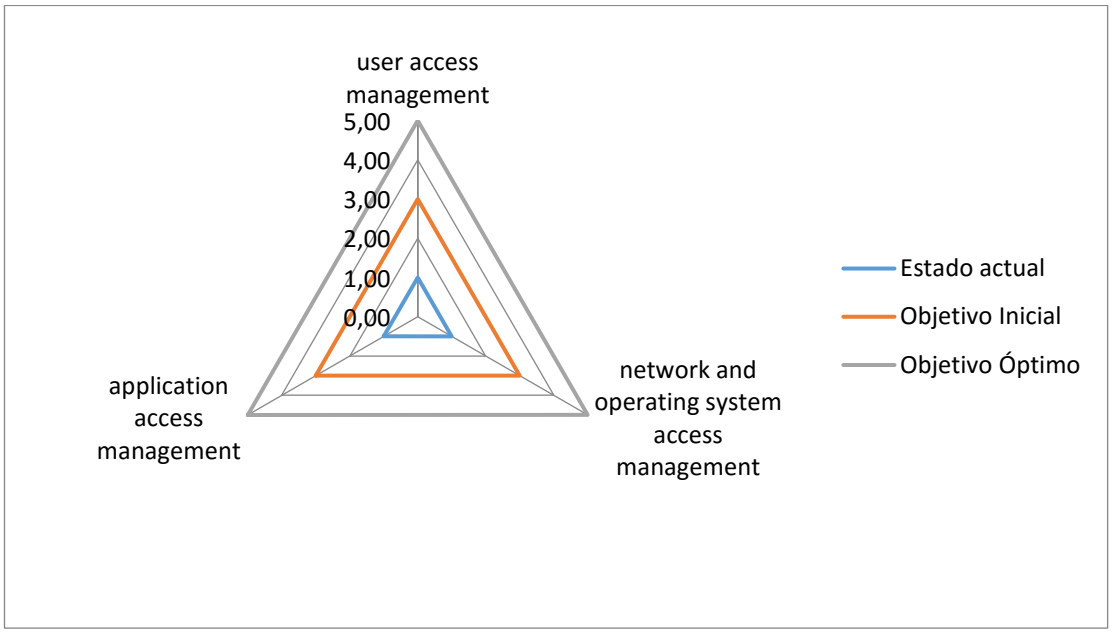


Figura AN10. Estado de los sistema de información en la red inalámbrica del IESTP “Eleazar Guzmán Barrón” – Huaraz.

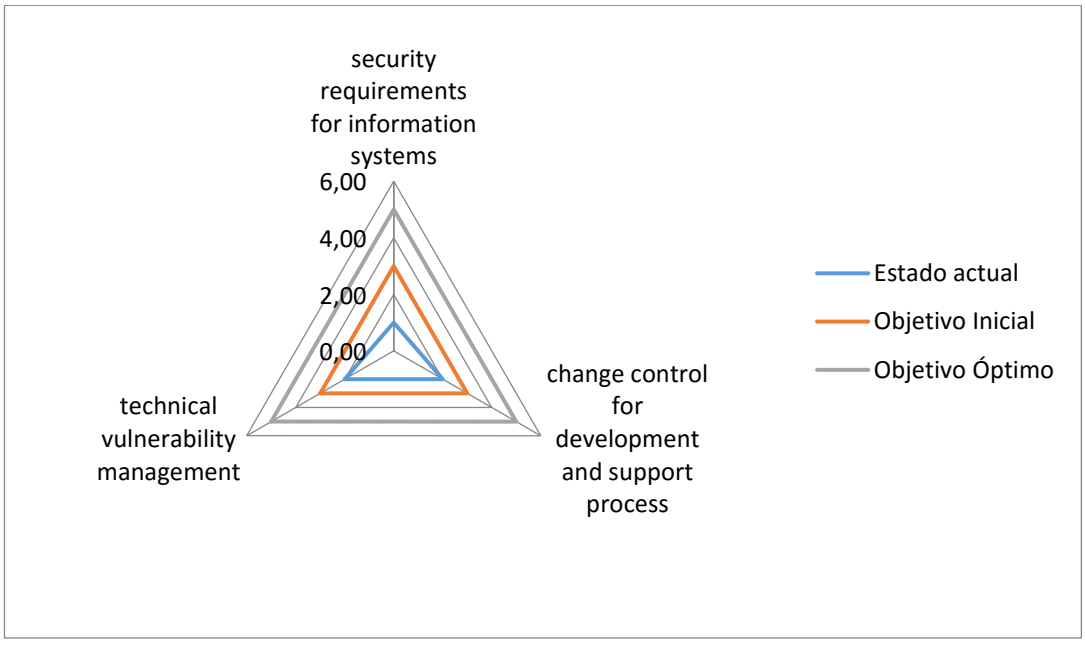


Figura AN11. Gestión de incidentes de la información en la red inalámbrica del IESTP “Eleazar Guzmán Barrón” – Huaraz.

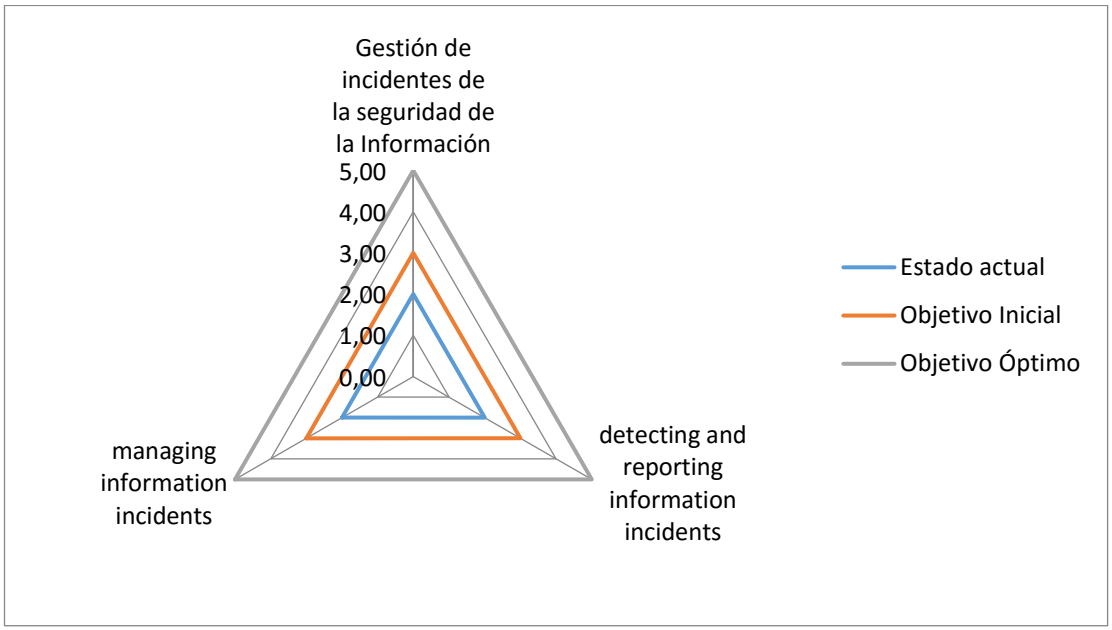


Figura AN13. Pantalla de trabajo del software QUARTUS 13.0 SP1.

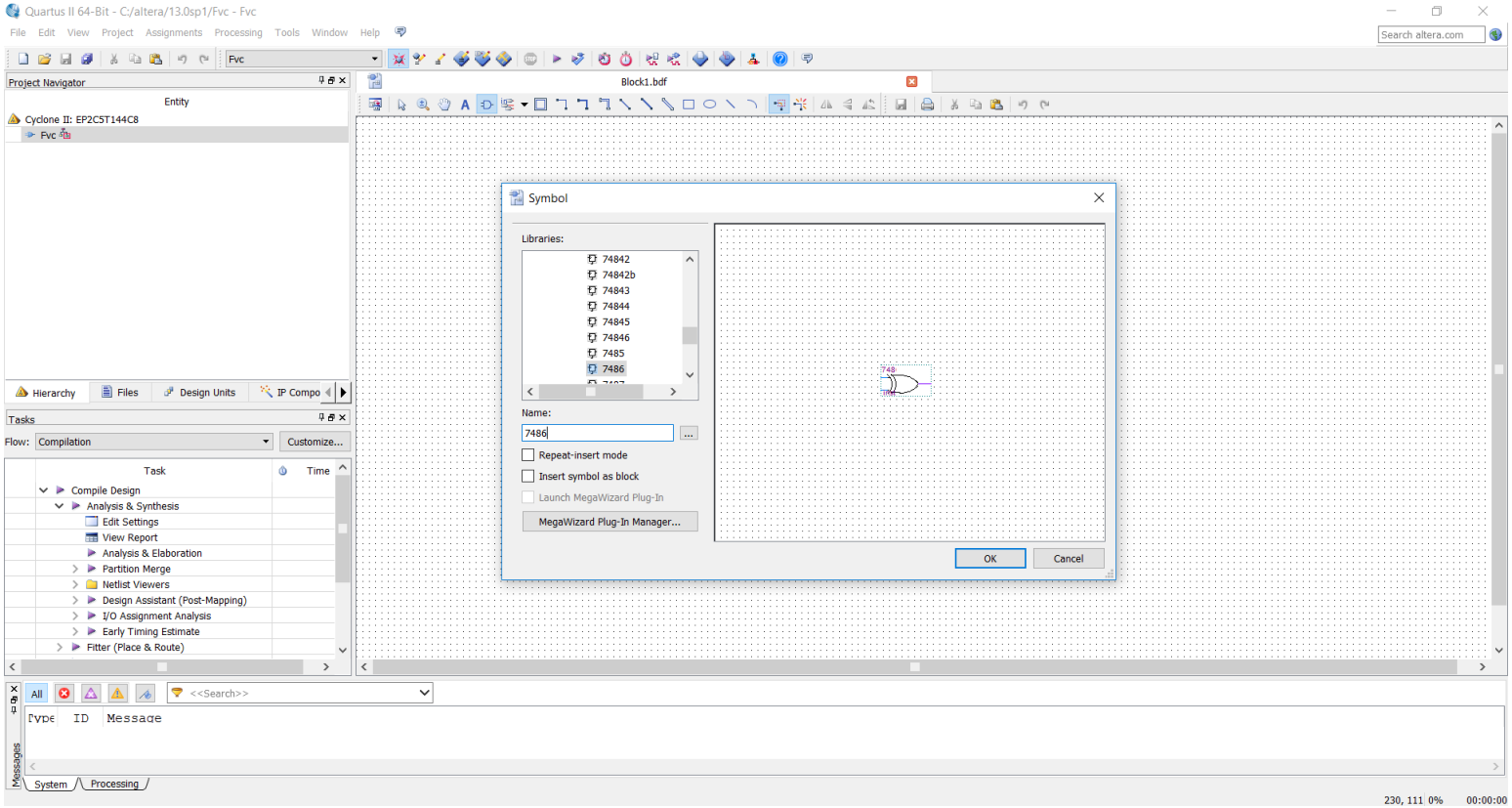


Figura AN15. Software ModelSim Altera 10.1d y diagrama de tiempos realizados durante el proceso de síntesis y diseño.

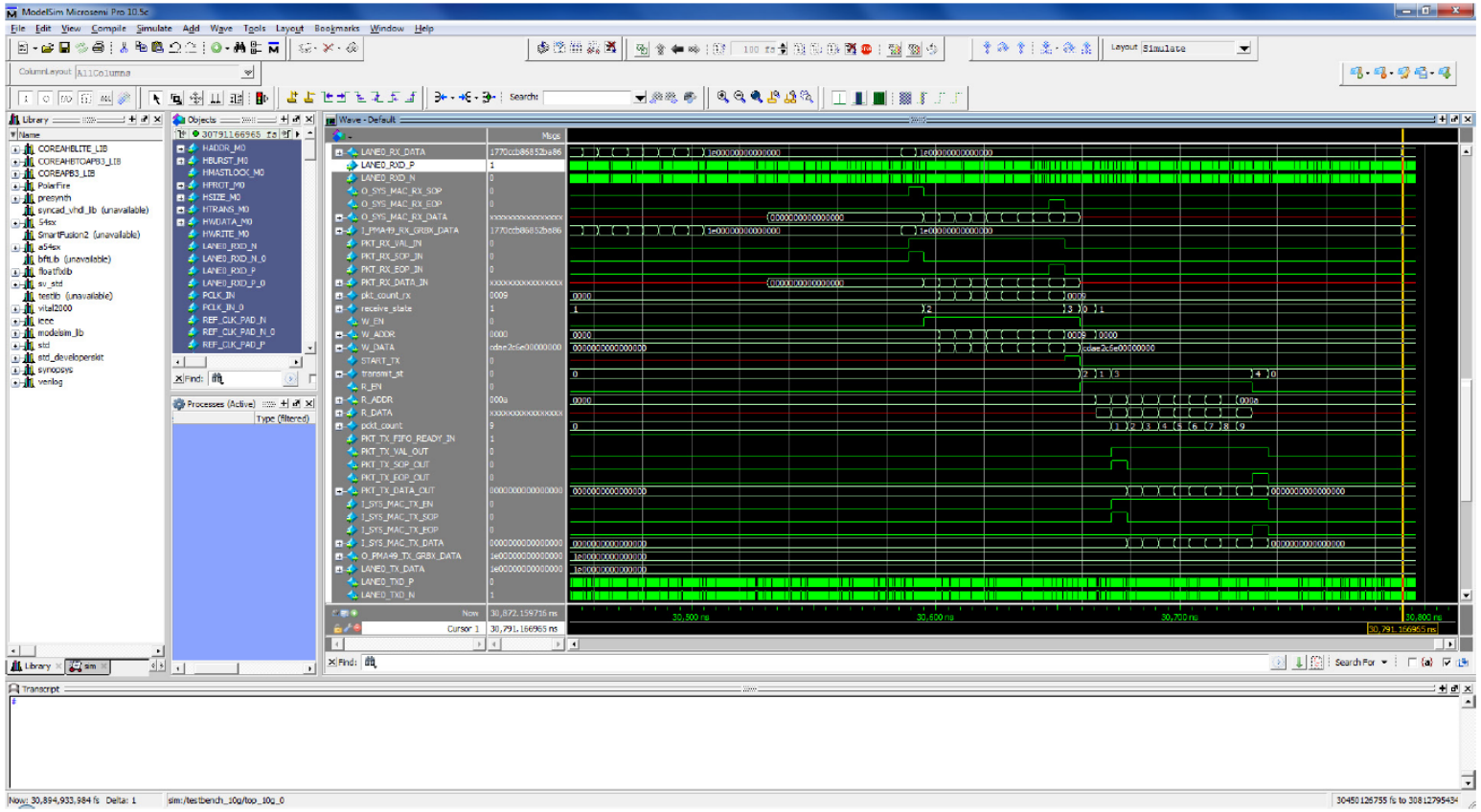


Figura AN16. Pantalla de configuración para el traslado de código a la memoria no volátil dela FPGA.