Unidad Ejecutora: *DIIT*

Título del proyecto de investigación:

Desarrollo de un Framework para evaluar la performance de distintos Algoritmos Criptográficos Livianos para ser instalados en dispositivos RFID de bajo costo.

Código del proyecto: *C2-ING-031*

Programa de acreditación: *CyTMA2*

Director del proyecto: *Mg. Jorge Esteban Eterovic*

Co-Director del proyecto: *Esp. Marcelo José Cipriano*

Integrantes del equipo:

Fecha de inicio: *01-01-2016*

Fecha de finalización: *31-12-2017*

Informe final de proyecto

Sumario:

1. Resumen y palabras clave

El desarrollo de la Internet de las Cosas (IoT: Internet of Things), dará lugar al despliegue de millones de objetos inteligentes que interactuarán entre sí a través de Internet. Cuando se habla de IoT, en realidad de lo que se está hablando es de la conectividad a través de Internet entre objetos para una gran diversidad de propósitos, por ejemplo: aplicaciones médicas, militares, científicas, etc., que hoy en día se realiza con dos tecnologías: RFID (Radio Frequency IDentification: Identificación por Radiofrecuencia) y WSN (Wireless Sensor Network: Redes Inalámbricas de Sensores).

Nuestro estudio hará foco en la adopción de la tecnología RFID, que representa un gran desafío en términos de seguridad y privacidad por los escasos recursos de procesamiento disponibles. Esto la condiciona negativamente al momento de querer implementar los métodos criptográficos tradicionales.

El uso de soluciones criptográficas tradicionales supone una aproximación correcta desde un punto de vista puramente teórico. Sin embargo, las primitivas criptográficas estándar tales como las funciones hash, código de autenticación de mensajes, cifradores de bloque/flujo, etc., exceden las capacidades de los sistemas RFID. Por lo tanto, la opción es el uso de Criptografía Ligera.

La Criptografía Ligera o Liviana (LICRYPT: Lightweight Cryptography) es un nuevo campo de investigación que apunta a estudiar nuevos métodos criptográficos con el fin que puedan utilizarse en objetos inteligentes, optimizados en función de la capacidad de procesamiento de los chips y del consumo de energía que se requiere en base a la longitud del código y del tamaño de la memoria RAM.

Los sistemas RFID pueden ser clasificados de acuerdo con el costo de las etiquetas, en sistemas RFID con etiquetas de alto costo y sistemas RFID con etiquetas de bajo costo. Nuestra investigación se centrará en el uso de estas últimas. El estudio y análisis del estado del arte nos ha permitido identificar a las soluciones criptográficas ligeras como las adecuadas para estos dispositivos limitados.

El objetivo de este proyecto de investigación es desarrollar un Framework para evaluar la performance de distintos Algoritmos Criptográficos Livianos para ser instalados en dispositivos RFID de bajo costo y asegurar las condiciones de seguridad y privacidad de las comunicaciones entre los objetos y la nube en el marco de Internet de las Cosas.

Se ha desarrollado una metodología innovadora, implementada en un software que corre en una máquina virtual que simula la capacidad de un dispositivo IoT, que mediante la realización de Pruebas de Carga permite la evaluación de algoritmos criptográficas. Esto permite la selección del algoritmo más con mejor performance tanto para el cifrado como para el descifrado.

Esta propuesta permite cubrir la ausencia de una métrica unificada adoptada por la Comunidad Científica Internacional sobre de la medición del desempeño y rendimiento de algoritmos de cifrado, de manera que la decisión de implementar una aplicación de cifrado que provea Confidencialidad a un conjunto de datos o canal de comunicaciones pueda ser tomada considerando su desempeño y rendimiento, sumado a la seguridad y robustez de esta. Se seleccionaron los algoritmos Clefia y Present como ejemplos a evaluar.

Palabras clave: Criptografía Ligera, Dispositivos RFID, Seguridad y Privacidad en Internet de las cosas, Algoritmo Clefia, Algoritmo Present.

2. Organización del Informe de Avance

- **Introducción:**

  - **Selección del Tema**

    Cuando se habla de Internet de las Cosas, pensamos en el concepto que se refiere a la interconexión digital de objetos cotidianos a través de Internet, pero Internet de las Cosas va mucho más allá, y llegará un momento en el tiempo en el que se conectarán a Internet más “cosas u objetos” que personas.

    Con la Internet de las Cosas, todo lo real se convierte en virtual, lo que significa que cada persona y las cosas tienen una ubicación en Internet. Estas entidades virtuales pueden producir y consumir servicios y colaborar entre sí.

    La manera en que estos objetos pueden comunicar o recibir información es a través de sensores que, en algunos casos, pueden visualizarse. Pero no siempre es posible notar su presencia. Dentro de la conexión de los objetos con los sistemas de información, dos son las tecnologías clave que ya se están insertando en diversos sectores de la industria para acercar la Internet de las Cosas a la realidad. Estas tecnologías son la identificación por radiofrecuencia (RFID) y las redes de sensores inalámbricas (WSN).

    Un sistema de RFID está constituido por un lector, denominado interrogador y un transpondedor o etiqueta, que generalmente está provisto de un microchip con una antena. Existen diferentes tipos de sistemas RFID, pero por lo general el lector envía ondas electromagnéticas con una señal a la cual la etiqueta debe responder.

    Las etiquetas pasivas no cuentan con una fuente de energía propia. Captan su energía del campo generado por el lector y la utilizan para proveer de energía a los microcircuitos del chip. Luego, el chip modula las ondas que la etiqueta envía al lector, y éste a su vez las convierte en información digital.

    Las etiquetas activas poseen una fuente de energía propia y emiten su señal. Los sistemas de ubicación en tiempo real no responden a las señales del lector, pero en cambio emiten sus señales a intervalos fijos. Los lectores captan esas señales y luego se utiliza algún software para calcular la ubicación de la etiqueta.

    Las etiquetas y lectores de RFID deben sintonizarse en la misma frecuencia para que puedan comunicarse entre sí. Los sistemas de RFID utilizan muchas frecuencias diferentes, pero por lo general las más comunes son las frecuencias bajas (alrededor de 125 KHz), las frecuencias altas (13.56 MHz) y las frecuencias ultra altas o UHF (860 a 960 MHz). Es más, algunas aplicaciones utilizan las microondas (2.45 GHz).

    Cuando hablamos de la tecnología RFID UHF aparece asociada la palabra RAIN (RAIN RFID). La palabra RAIN es un acrónimo derivado de RAdio frequency IdentificatioN y describe la relación existente entre el RFID UHF y la nube de datos, donde se almacena toda la información generada por los lectores UHF y puede ser compartida vía Internet.

    RFID UHF es ahora una tecnología madura, las soluciones se pueden implementar de manera confiable, permite lectura de largo alcance, tiene una

etiqueta pasiva y es de bajo costo. Entonces, solo resta darles seguridad y privacidad a las comunicaciones.

En general, los sistemas RFID pueden ser clasificados de acuerdo con el costo de las etiquetas en sistemas RFID con etiquetas de alto costo y sistemas RFID con etiquetas de bajo costo. Nuestra investigación se centrará fundamentalmente en estas últimas. El estudio y análisis del estado del arte nos ha permitido identificar a las soluciones criptográficas ligeras como las adecuadas para estos dispositivos limitados.

El uso de soluciones criptográficas estándar supone una aproximación correcta desde un punto de vista puramente teórico. Sin embargo, las primitivas criptográficas estándar (funciones hash, código de autenticación de mensajes, cifradores de bloque/flujo, etc.) exceden las capacidades de los sistemas RFID con etiquetas de bajo costo y por tanto son incapaces de ofrecer confidencialidad y demás servicios criptográficos a las comunicaciones de este tipo.

Por lo tanto, es necesario el uso de Criptografía Ligera ya que la mayoría de los chips RAIN RFID tienen una pequeña cantidad de memoria que va de 64 a 512 bits, llegando en algunos chips especiales a 2k bits.

La International Telecommunication Union (ITU), en su "Informe sobre la Internet de las Cosas" califica a la tecnología RFID como “un pivote que habilitará el Internet de las Cosas", permitiendo la conversión de los "objetos cotidianos" en "inteligentes". Sin embargo, sin bases sólidas de seguridad, es posible que estos objetos sean pasibles de ataques. Estas amenazas podrían llegar a ser cada vez más perjudiciales que cualquiera de sus potenciales beneficios.

Las investigaciones sobre algoritmos criptográficos siguen avanzando y cada día se generan nuevos algoritmos para las claves de autenticación. La investigación académica en general y la Asociación Internacional de Investigaciones en Criptografía (IACR-International Association for Cryptologic Research) en particular, impulsan la definición de distintos mecanismos que proporcionarán un nivel de seguridad y privacidad adecuados a las limitaciones del hardware de las etiquetas RFID.

En los últimos años se han producido grandes avances en el área de la “Criptografía Ligera” como así también los temas de privacidad, protección de datos personales y seguridad en las comunicaciones electrónicas teniendo en cuenta las amenazas específicas de la tecnología RFID. La Criptografía Ligera aplicada a RFID es un nuevo campo de investigación que apunta a estudiar los métodos criptográficos con el fin que puedan implementarse en objetos inteligentes.

Cabe aclarar que, aunque el presente proyecto de investigación se ha centrado en el estudio de aplicaciones de tipo RFID, no queda agotada la investigación en esta área, sino que también puede ser aplicado al estudio en sistemas y plataformas donde sus recursos son limitados como por ejemplo dispositivos y sistemas móviles, vehículos no tripulados autónomos o conducidos a distancia, aparatos de e-Salud[1], Domótica, dispositivos “vestibles” o wereables[2].

[1] También conocida como e-Health. Es un área de la aparatología médica. Consiste en el implante de dispositivos inteligentes en el paciente, como son los marcapasos con conexión inalámbrica, bombas de insulina y demás, utilizados para seguimiento, gestión, prevención y tratamiento médico.

La Criptografía Ligera puede estar orientada al Hardware o al Software, determinándose parámetros para evaluar y medir las implementaciones que apliquen a este tipo de criptografía. Por ejemplo, para implementaciones en hardware se estudian el tamaño de los chips y el consumo de energía que se requiere. Para implementaciones en software, en cambio, se analizan la longitud del código y el uso y consumo de memoria RAM.

En Criptografía Ligera se han desarrollado algoritmos de clave pública, algoritmos de clave privada, algoritmos de cifradores de bloque y algoritmos de cifradores de flujo, funciones de hash y mecanismos de autenticación, tal como ocurre en la criptografía tradicional.

Pero hasta el presente, los investigadores no se han puesto de acuerdo en un criterio determinado para clasificar a un algoritmo criptográfico como ligero. Lo que sí está claro es que las técnicas criptográficas involucradas tienen que usar la mínima cantidad de recursos posibles de los objetos en los que se las aplicará.

Muchos son los avances a nivel de criptografía que se están realizando, pero aún no se ha comprobado cuáles algoritmos ligeros se desempeñan de manera eficiente a la hora de implementarlos en los dispositivos RFID de bajo costo.

- **Definición del Problema**

La Criptografía Ligera ofrece un conjunto de propiedades que hace que diferentes algoritmos de cifrado puedan trabajar en contextos de Hardware y Software con pocos recursos. Sin embargo, el desempeño de estos en un determinado ambiente de trabajo se desconoce.

Hasta tanto no se los instale en dichos sistemas y no sean empleados, se desconoce su rendimiento. Pudiendo incluso decidirse la remoción de un algoritmo instalado dado su pobre desempeño en dicho sistema, con el consiguiente desperdicio de tiempo, recursos humanos y económicos.

Esta propuesta permite, dado un determinado sistema, realizar la selección del algoritmo más apto y performante. Mediante las pruebas de carga y dado su comportamiento, se podrá determinar el candidato más adecuado. Esto permitiría predecir la eficiencia del criptosistema, ahorrando tiempo y recursos.

- **Justificación del Estudio**

La identificación por radio frecuencia o RFID - Radio Frequency IDentification, es una tecnología que permite identificar automáticamente un objeto gracias a una onda que emite la etiqueta y que transmite por radiofrecuencia los datos identificativos del mismo, siendo esta identificación normalmente unívoca.

Actualmente, bajo las siglas RFID, se agrupan tecnologías que sirven para identificar objetos mediante ondas de radio

[2] Dispositivos implantados en diferentes telas y prendas de vestir, como por ejemplo la instalación de GPS en zapatillas para la ubicación de personas. O remeras deportivas para el monitoreo cardíaco, etc.

La tecnología RFID plantea nuevas oportunidades de mejora de la eficiencia y de la usabilidad de los sistemas de uso cotidiano. Estas mejoras, que afectan a muchas facetas de la vida, desde lo personal a lo profesional, en ocasiones plantea nuevos riesgos para la seguridad y nuevos retos para evitarlos.

Para identificar los nuevos riesgos que se plantean, hay que tener en cuenta los dos tipos de usuarios de esta tecnología:

- Entidades que utilizan RFID para optimizar sus procesos internos de gestión, de almacén, de inventario, de producción, de gestión de personal, de seguridad, etc.

- Entidades que ofrecen un servicio a usuarios internos de la organización, como control de accesos, o a usuarios particulares, como venta de productos, prestación de servicios, etc.

En ambos casos, existen riesgos derivados de las características de la tecnología que son comunes, aunque las aplicaciones sean tan dispares en sus beneficiarios y objetivos. Estos riesgos comunes tienen que ver con ataques o averías que afectan al servicio, ya sea interrumpiéndolo, alterándolo o permitiendo realizar algún tipo de fraude. Éstos constituyen riesgos de la seguridad.

Por otro lado, existe también la posibilidad de que la tecnología se use de forma maliciosa para acceder de forma fraudulenta a información personal de los usuarios del sistema. Este segundo tipo de riesgo está principalmente asociado a los sistemas que dan servicio a usuarios y puede tener una repercusión muy importante para las organizaciones responsables. Éstos son riesgos que afectan la privacidad.

Además de estos dos tipos de riesgos, existen otros que pueden condicionar el uso de la tecnología RFID, como por ejemplo, se vuelve a plantear el debate sobre los riesgos de exposición de los seres humanos a las radiaciones.

Dada su importancia, los riesgos derivados del uso de RFID deben ser afrontados con mucha atención. En la prevención de estos riesgos, todos los participantes de la tecnología tienen responsabilidad, desde los encargados de desplegar la tecnología, los organismos de control que deben velar por los ciudadanos, hasta los propios usuarios.

En cada uno de sus ámbitos, los agentes implicados deben desarrollar políticas activas. El papel de las autoridades es clave en esta tarea, creando legislación, normativas y recomendaciones. También el de las instituciones de investigación mediante soluciones técnicas que mejoren la seguridad.

Las organizaciones proveedoras que utilizan la tecnología deben aplicar estas recomendaciones y desarrollar buenas prácticas. Por último, los usuarios deben conocer y exigir sus derechos para preservar su privacidad.

Los riesgos para la seguridad de la tecnología RFID son aquellos derivados de acciones encaminadas a deteriorar, interrumpir o aprovecharse del servicio de forma maliciosa. Con este tipo de actos se perseguirá un beneficio económico o bien un deterioro del servicio prestado.

Los riesgos para el servicio se concretan en los tipos de “ataques” más habituales que puede sufrir una instalación, cada uno de ellos con una finalidad y un impacto diferente. La forma más simple de ataque a un sistema RFID es evitar la comunicación entre el lector y la etiqueta, pero también existen otras formas de ataque más sofisticadas, cuyo blanco son las comunicaciones en radiofrecuencia:

1. Aislamiento de etiquetas: El ataque más sencillo a la seguridad en RFID consiste en impedir la correcta comunicación lector-etiqueta. Esto se puede conseguir introduciendo la etiqueta en una “jaula de Faraday” o creando un campo electromagnético que interfiera con el creado por el lector. Este ataque puede ser utilizado para sustraer productos protegidos por etiquetas RFID. También puede ser una medida de protección de usuarios ante lectores de etiquetas ilegales. Un ejemplo muy relevante de este caso es el del pasaporte electrónico, para el que existen fundas especiales con hilos de metal que crear una “jaula de Faraday” (una jaula de Faraday es un espacio cerrado revestido metálicamente que imposibilita la influencia de los campos eléctricos exteriores en el interior de este: las ondas de radio no pueden acceder al interior de la jaula) evitando lecturas incontroladas de su información.

2. Suplantación: Este ataque consiste en el envío de información falsa que parece ser válida. Por ejemplo, se podría enviar un código electrónico de producto (EPC) falso, cuando el sistema espera uno correcto. Este tipo de ataque puede servir para sustituir etiquetas lo cual puede permitir la obtención de artículos caros con etiquetas suplantadas de productos más baratos. Además, aplicado a la cadena de distribución, puede llevar a un fraude de grandes dimensiones por la sustitución de grandes volúmenes de mercancías. Este ataque puede utilizarse en otros entornos, como puede ser el de telepeaje.

3. Inserción: Este ataque consiste en la inserción de comandos ejecutables en la memoria de datos de una etiqueta donde habitualmente se esperan datos. Estos comandos pueden inhabilitar lectores y otros elementos del sistema. La finalidad de este tipo de ataque será la desactivación del sistema o la invalidación de parte de sus componentes, permitiendo algún tipo de fraude, o una denegación de servicio.

4. Repetición: Consiste en enviar al lector RFID una señal que reproduce la de una etiqueta válida. Esta señal se habrá capturado mediante escucha a la original. El receptor aceptará como válidos los datos enviados. Este ataque permitirá suplantar la identidad que representa una etiqueta RFID.

5. Denegación de Servicio (DoS): Este tipo de ataque, satura el sistema enviándole de forma masiva más datos de los que este es capaz de procesar, por ejemplo, colapsando la funcionalidad de backscattering o señal de retorno de la tecnología RFID. Asimismo, existe una variante, el RF Jamming, mediante el cual se consigue anular o inhibir la comunicación de radiofrecuencia emitiendo ruido suficientemente potente. En ambos, casos, se invalida el sistema para la detección de etiquetas. Con este ataque se consigue que los objetos etiquetados, escapen al control del sistema en su movimiento. Puede ser utilizado para la sustracción de mercancía a pequeña o gran escala.

6. Desactivación o destrucción de etiquetas: Consiste en deshabilitar las etiquetas RFID sometiéndolas a un fuerte campo electromagnético. Lo que hace este sistema es emitir un pulso electromagnético que destruye la sección más débil de la antena, con lo que el sistema queda inutilizado. Si se dispone de los medios técnicos necesarios, se pueden inutilizar las etiquetas de protección antirrobo de los productos, favoreciéndose así su sustracción. Este ataque también se puede utilizar en los sistemas utilizados para la cadena de distribución.

7. Clonación de la tarjeta RFID: A partir de la comunicación entre una etiqueta y el lector, se copian dichos datos y se replican en otra etiqueta RFID para ser utilizados posteriormente.

8. Riesgo de ataque mediante inyección de lenguaje de consultas SQL: Por medio de la comunicación entre la etiqueta y el lector, se pasa lenguaje SQL hacia el soporte físico que lee la etiqueta, el cual, debido a dicho ataque, ejecuta las órdenes incluidas en la etiqueta y esto puede ser introducido en una base de datos.

9. Código malicioso (malware): Otro posible riesgo de la tecnología RFID consiste en la infección y transmisión de códigos maliciosos incluidos dentro de etiquetas RFID. Para ello el código malicioso debe entrar en la etiqueta, lo que supone un hecho complicado, dado que la capacidad de almacenamiento de algunas etiquetas no es muy grande.

10. Spoofing: Caso particular para etiquetas activas (de lectura y escritura). En este caso se escriben datos reales en una etiqueta RFID para suplantar la información original. Es más habitual en las etiquetas RFID de las prendas de vestir. Se puede suplantar la información tantas veces se quiera, siempre que éstas sean de lectura.

11. Ataques Man in the Middle (MiM): Vulnera la confianza mutua en los procesos de comunicación y reemplaza una de las entidades. Ya que la tecnología RFID se basa en la interoperabilidad entre lectores y etiquetas es vulnerable a este tipo de ataques.

12. Inutilización de etiquetas: Si se somete la etiqueta RFID a un fuerte campo electromagnético, ésta se inhabilita. Esta técnica es usada para sustraer productos ya que, si se dispone, por ejemplo, de una antena altamente direccional, se pueden inutilizar las etiquetas del producto.

La posibilidad de estos ataques y la facilidad técnica de uso de alguno de ellos se debe a lo maduro de esta tecnología que permite el acceso a lectores y grabadores de RFID a un precio muy accesible.

Por este motivo, la tecnología debe mejorar aún más, aumentando el nivel de protección de accesos y minimizar las posibilidades de fraude. Una de las características que la tecnología debe mejorar para aumentar la seguridad es la capacidad de memoria y procesamiento de las etiquetas, la cual limita las posibilidades de implementar mecanismos avanzados de seguridad y cifrado.

Para evitar de una forma sencilla la modificación de la información en las etiquetas es recomendable utilizar las de sólo lectura, o no escribir los datos directamente en ellas. De esta manera se incluye un código en la etiqueta y el

resto de la información se traslada a una base de datos que disponga de mayores medidas de seguridad.

Los métodos de autenticación previos al borrado o desactivación de las etiquetas pueden evitar estas acciones no autorizadas. El cifrado en las etiquetas es otra buena práctica si éstas contienen información privada.

En el caso de medidas de seguridad para el lector, se pueden llevar a cabo técnicas de autentificación para realizar la comunicación entre lector y la etiqueta evitando así la falsificación de identificadores de lector.

Muchos de estos ataques se pueden prevenir o directamente evitar con el uso de técnicas criptográficas que permitan realizar el cifrado, la autenticación y asegurar la integridad de la información que se trasmite entre la etiqueta y el lector.

Como la capacidad de los dispositivos RFID es muy limitada, se deberá recurrir al uso de Algoritmos de Criptografía Ligera.

- **Limitaciones**

  Se trabajará con dos de los algoritmos de Criptografía Ligera disponibles para validar el framework a desarrollar. Se han seleccionado los algoritmos Clefia y Present por sus propiedades y cualidades. Cabe aclarar que el diseño del framework permite el uso de cualquiera de los algoritmos criptográficos ligeros disponibles.

- **Alcances del Trabajo**

  Desarrollar un Framework para evaluar la performance de los Algoritmos Criptográficos Livianos, definiendo las metodologías y los indicadores necesarios.

  Validar el Framework con un caso de estudio simulando el funcionamiento de los dos Algoritmos Criptográficos Livianos seleccionados.

- **Objetivos**

  El objetivo de este proyecto de investigación es desarrollar un Framework para evaluar la performance de distintos Algoritmos Criptográficos Livianos para ser instalados en dispositivos RFID de bajo costo y asegurar las condiciones de seguridad y privacidad de las comunicaciones entre los objetos y la nube en el marco de Internet de las Cosas.

- **Hipótesis**

  El avance de la Internet de las Cosas originará el despliegue de millones de objetos inteligentes que interactuarán entre sí y con Internet. La tecnología RFID, por definición insegura, será primordial en este escenario.

Los algoritmos criptográficos rara vez son estudiados comparativamente, para evaluar el rendimiento relativo entre ellos. Por ende, los usuarios de estos podrán descubrir si el algoritmo elegido y adoptado para ser aplicado a su sistema se comporta de la forma esperada y lo más eficientemente posible cuando el mismo esté operativo.

Lamentablemente esta evaluación se realiza de manera empírica cuando ya está instalado y operando. Si el algoritmo no se comporta como se esperaba, el proceso de recambio es tan costoso en tiempo y recursos que muchas veces se opta por dejarlo funcionando, sacrificando tal vez la performance completa del sistema.

Se sustentan las siguientes hipótesis de trabajo:

- H1: Se puede estudiar y evaluar el rendimiento y performance comparativa entre 2 o más algoritmos de cifrado, obteniendo un ranking de estos, de acuerdo con su comportamiento en las diferentes pruebas de carga a las que son sometidos.

- H2: La performance puede ser evaluada en un entorno virtual que asemeje al entorno real donde el criptosistema deba funcionar.

En este contexto, es fundamental desarrollar un Framework para evaluar la performance de distintos Algoritmos Criptográficos Livianos para ser instalados en dispositivos RFID de bajo costo que asegure las condiciones de seguridad y privacidad.

- **Desarrollo:**

  - **Material y Métodos**

    Para poder obtener resultados coherentes y confiables se debe tener en cuenta que los algoritmos deben ser ejecutados bajo idénticas condiciones, es decir que deben correrse en los mismos entornos de hardware y software.

    Esto implica: un mismo tipo de procesador e igual cantidad de ellos, idéntico Sistema Operativo, misma cantidad de memoria RAM, compilador y archivos de entrada. Para ello podría usarse siempre la misma computadora donde se realizarán los test. Sin embargo, no se puede tener un control y registros exactos sobre ella.

    No siempre los algoritmos se ejecutarán en idénticas condiciones. Por ejemplo, aplicaciones latentes que se activen durante la prueba podrían introducir errores de medición, entre otras posibilidades.

    También existe la posibilidad que el algoritmo deba ser ejecutado en una plataforma con determinadas características, propiedades y limitaciones de hardware, software o ambas. Podría ocurrir que el algoritmo criptográfico elegido para dotar de seguridad a tal dispositivo no se desempeñe adecuadamente o de la manera esperada dadas las limitaciones particulares del dispositivo donde se desea su ejecución.

    Una disminución en la velocidad, capacidad de cómputo, memoria de trabajo o cualquier otro aspecto de la plataforma podría provocar una dificultad no prevista que afecte negativamente el desempeño del mismo.

    Por ello se propone que los test se ejecuten en “entornos virtualizados” que permiten tener un mayor control de las condiciones del equipo, del ecosistema informático y de las demás variables, como así también permitir las pruebas en un contexto lo más semejante posible sobre el cual el algoritmo será instalado.

    Las distintas actividades planteadas en el Gantt informado en el formulario FPI-002 - Protocolo de presentación del proyecto, fueron las siguientes:

    1) Realizar un relevamiento exhaustivo de los principales algoritmos criptográficos ligeros existes.

    2) Estudiar y determinar qué algoritmos se podrían utilizar para dispositivos RFID de bajo costo.

    3) Estudiar los diferentes sistemas de RFID en función de sus frecuencias: bajas, altas y ultra altas (UHF).

    4) Redactar el informe de avance.

    5) Desarrollar un Framework para evaluar la performance de los Algoritmos Criptográficos Ligeros definiendo las metodologías y los indicadores necesarios.

    6) Validar el Framework con un caso de estudio simulando el funcionamiento de los algoritmos.

7) Redactar el informe final.

La programación de estas actividades se muestra en la Tabla 1:

	e	f	m	a	m	j	j	a	s	o	n	d	e	f	m	a	m	j	j	a	s	o	n	d
1. Revisión bibliográfica	x	x	x																					
2. Estudio de los Algoritmos			x	x	x	x	x																	
3. Estudio distintos RFID							x	x	x	x														
4. Redacción de Informe											x	x												
5. Desarrollo del Framework													x	x	x	x	x							
6. Validación del Framework																	x	x	x	x	x	x		
7. Redacción del Informe final																						x	x	x

Tabla 1: Gantt del Proyecto.

- **Lugar y Tiempo de la Investigación**

  El trabajo de investigación se desarrollará en el Laboratorio PRAMIN, del Departamento de Ingeniería e Investigaciones Tecnológicas de la UNLaM y en la biblioteca central de la UNLaM.

  El tiempo de desarrollo del proyecto se ha planificado en 2 (dos) años. Las tareas se llevan a cabo en base al Diagrama de Gantt descripto en el punto anterior.

- **Descripción del Objeto de Estudio**

  Con Internet de las Cosas, todo lo real tendrá su imagen especular en el mundo virtual: cada persona y los dispositivos que posea tendrán una “ubicación” en Internet. Estas entidades pueden producir y consumir servicios e información, como así también, colaborar entre sí con un objetivo en común.

  IoT se sustenta fundamentalmente en 2 tecnologías que han demostrado un importante crecimiento en los últimos años. Ellas permiten alcanzar el nivel de interconectividad requerido:

  - Redes Inalámbricas de Sensores.
  - Identificación por Radiofrecuencia.

  La Unión Internacional de Telecomunicaciones en su "Informe sobre la Internet de las Cosas” califica a la tecnología RFID como un "pivote para la IoT permitiendo la conversión de los objetos cotidianos en inteligentes"

  Dada la cantidad y naturaleza de la información que muchos de estos dispositivos colectan y transmiten, sumado a las limitaciones de hardware propias de estos aparatos; es posible que sean blanco de ataques y posean vulnerabilidades que esperan ser explotadas. Estas amenazas podrían llegar a

ser tanto o más perjudiciales que cualquiera de los beneficios que ofrecen su uso.

Hasta el año 2003 se han publicado numerosos artículos sobre RFID centrados en la seguridad. Pero la gran mayoría de estas propuestas no abordan de forma realista las fuertes limitaciones computacionales, de circuitos electrónicos y de consumos, entre otros, de este tipo de dispositivos.

A pesar de que, desde un punto de vista teórico, estas propuestas tenían sentido, en general no era posible su aplicación en un gran número de etiquetas y dispositivos RFID de bajo costo ya que estaban basadas en primitivas criptográficas estándar que excedían las capacidades de las etiquetas, y no se sugería el uso de primitivas criptográficas ligeras.

El advenimiento en años recientes de este nuevo campo de investigación y aplicación, llamado Criptografía Ligera o Liviana persigue el estudio de nuevos métodos criptográficos con el fin que puedan utilizarse en objetos inteligentes, particularmente adecuados a las limitaciones de los dispositivos que se emplean en IoT, pues los algoritmos tradicionales no pueden funcionar adecuadamente en dichos entornos.

Los algoritmos livianos pueden estar optimizados para entornos de Hardware, Software e incluso pueden tener buenos rendimientos en ambos. Se pueden encontrar algoritmos de: Clave Pública, Clave Privada, Block Ciphers, Stream Ciphers, Hash y mecanismos de Autenticación.

En 2003, Vajda et al. [10] publicaron el primer artículo en el que se proponía el uso de criptografía ligera. Al año siguiente, Juels [11] introdujo el concepto de criptografía minimalista y un año después el de seguridad y privacidad [12]. En 2005, hubo numerosas propuestas basadas en el uso de funciones resumen (Hash).

Sin embargo, esta área de investigación recientemente ha atraído cierto interés, especialmente desde la publicación de varios protocolos ligeros en el Informe Final del eSTREAM Project [9], organizado por la EU ECRYPT Network (European Network of Excellence in Cryptology), publicado en mayo de 2008.

Los primeros trabajos sobre seguridad en dispositivos RFID fueron publicados en la conferencia RFIDSec, considerada el evento anual más importante en esta área del conocimiento. Sin embargo, a menudo no se especificaba para qué clase de etiqueta eran adecuadas las propuestas.

La clase de la etiqueta determina un gran número de parámetros tales como las operaciones soportadas en la misma o el tipo de ataques frente a los que debe ser resistente.

Por lo tanto, no todas las etiquetas soportan el mismo tipo de operaciones. Por ejemplo, la criptografía estándar de clave pública es aplicable para las etiquetas de alto costo, pero excede las capacidades de las etiquetas de bajo costo. Además, cada clase de etiqueta tendrá un nivel de seguridad diferente.

En la Tabla 2 se muestran las etiquetas RFID mas difundidas:

Tipo de Tarjeta	Frecuencia	Protocolo	Tamaño EEPROM
Estándar UHF	860-960MHz	ISO18000-6C EPC Gen2	EPC 128bits User 32-512bits
Estándar UHF/HF	860-960MHz y 13.56MHz	ISO18000-6C /EPC Gen2 ISO14443A Mifare Classic	UHF (EPC 128bits, User 32-512bits), HF (1K Bytes)
MIFare UltraLight	13.56MHz	ISO14443A	512bits
MIFare Classic 1K	13.56MHz	ISO14443A	1Kb

Tabla 2: comparación entre tarjetas y dispositivos RFID.

Con respecto al nivel de seguridad, los ataques se pueden dividir en activos y pasivos. Las etiquetas de bajo costo deben ser resistentes frente a ataques pasivos y las etiquetas de costo moderado y alto frente a ataques pasivos y activos.

Por último, cabe mencionar que recientemente Chien [13] propuso una clasificación alternativa de las etiquetas basada en las operaciones soportadas en las mismas

En lo que respecta al cifrado, se llaman Algoritmos de Clave Privada o Simétricos a aquellos algoritmos que requieren que el emisor de un mensaje o comunicación cifrada, y el receptor, compartan la misma clave. Tal clave debe ser ingresada al algoritmo para que, con ella, se pueda cifrar/descifrar.

A su vez este tipo de algoritmos se clasifican de acuerdo con la forma con la que procesan los bits del mensaje:

- Cifrando bit a bit: Cifradores en Cadena o Cifradores en Flujo (Stream Ciphers)
- Cifrando un grupo de bits de longitud fija por vez: Cifradores en Bloque (Block Ciphers).

El tamaño de los bloques es un aspecto muy importante a tener en cuenta. Un valor pequeño puede disminuir la seguridad del algoritmo e incluso hacerlo vulnerable. En la actualidad los tamaños de bloques están acotados entre los 64 y 128 bits.

Los algoritmos Criptografía Ligera o Liviana se encuentran en la norma ISO/IEC 29192:2012 [14]. Además, la norma mencionada propone algunos criterios e indicadores para que un algoritmo sea considerado “liviano” o “ligero”:

- área del chip medido en GE (Gate Equivalent)

- consumo de energía,
- cantidad de líneas de código,
- tamaño de RAM,
- ancho de banda de la comunicación
- tiempo de ejecución

Dado que cada dispositivo tiene sus propias limitantes y aun conociendo en detalle estos indicadores para cada algoritmo, no es fácil elegir el más adecuado para ser implementado. Entre otras razones se puede mencionar la falta de consideración del funcionamiento en conjunto del algoritmo y su ecosistema. La realización de estas pruebas directamente sobre el sistema físico tiene costo económico y temporal, que podría ser importante, de acuerdo con los limitantes del problema.

- **Descripción de Población y Muestra**

Se creará, en una computadora personal, el entorno de virtualización adecuado al Smart Device que se pretende simular.

De los diferentes sistemas que permiten realizar Máquinas Virtuales (VM) se elegirá el que más control del entorno permita, ya que los resultados serán mejores cuanto más fiel sea la virtualización y mayor el control que de la misma se tenga.

Dado el perfil y tipo de dispositivo en el que se busca implementar el criptosistema, se puede seleccionar un conjunto de los mismos que mejor se adapten a tal dispositivo.

Esto se logra mediante el estudio de las especificaciones técnicas de los algoritmos. Para los algoritmos orientados al Software, deben programarse respetando las características que los autores indicaron.

Incluso algunos de ellos ofrecen a la comunidad los códigos de sus criptosistemas, por ejemplo, los que se publicaron en el Proyecto e-Stream [9].

Se tomarán como muestra dos algoritmos Criptográficos Livianos con algunas de sus variantes más usadas.

- **Diseño de la Investigación**

Para poder obtener resultados coherentes y confiables se debe tener en cuenta que los algoritmos deben ser ejecutados bajo idénticas condiciones, es decir que deben correrse en los mismos entornos de hardware y Software.

Esto implica: un mismo tipo de procesador e igual cantidad de ellos, idéntico Sistema Operativo, misma cantidad de memoria RAM, igual compilador y archivos de entrada. Para ello podría usarse siempre la misma computadora donde se realizarán los test. Sin embargo, no se puede tener un control y registros exactos sobre ella.

No siempre los algoritmos se ejecutarán en idénticas condiciones. Por ejemplo, aplicaciones latentes que se activen durante la prueba podrían introducir errores de medición, entre otras posibilidades.

La propuesta de solución al problema planteado es llevar adelante pruebas a los algoritmos criptográficos por medio de un software. Este permitirá la evaluación del rendimiento virtualizando el ecosistema donde el algoritmo se ejecutará.

Al cabo de las mismas, se procederá a un análisis estadístico del rendimiento de cada algoritmo que permita la elección del más adecuado por sus prestaciones. Para poder llevar adelante la metodología propuesta se deben realizar las etapas que se muestran en la Figura 1:

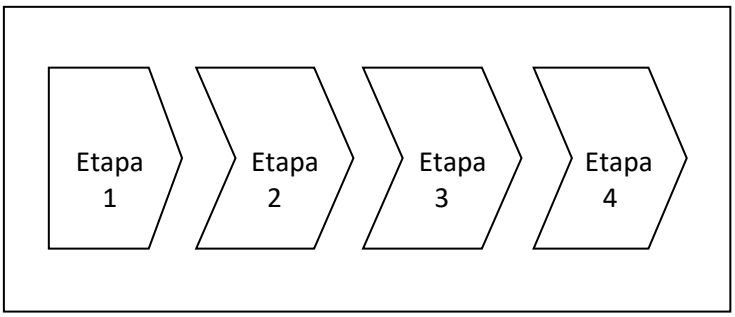


Figura 1: Etapas del Proyecto de Investigación.

1. Creación del Entorno de Virtualización y del Programa Evaluador.
2. Selección de los criptosistemas candidatos.
3. Ejecución de las Pruebas de Carga por medio de un Algoritmo Marco o Framework.
4. Análisis de los resultados de las pruebas.

- **Instrumentos de Recolección y Medición de Datos**

Etapa 1: Creación del Entorno de Virtualización y del Programa Evaluador.

Es de suma relevancia la definición de ciertos conceptos acerca de la tecnología a usar para llevar adelante esta etapa, como: el sistema operativo que se virtualizará, el software para virtualizar, el perfil del hardware que se desea simular, el programa en el que el algoritmo se ejecutará en la Máquina Virtual.

Como primer paso se realizó un estudio de los sistemas operativos disponibles en el mercado para la Internet de las Cosas, dado que no cualquier sistema operativo IoT es adecuado para cualquier uso de IoT.

Las características principales que diferencian a estos sistemas operativos son: bajo requerimiento de memoria, capacidades de tiempo real, eficiencia en el consumo energético, operación agnóstica del hardware (debe soportar una amplia variedad de plataformas de hardware para simplificar la interconectividad), amplio soporte de redes y protocolos (WiFi, celular, bluetooth, etc.), seguridad estricta y un ecosistema para desarrollo de aplicaciones que acelere la creación de soluciones y reduzca el tiempo de salida al mercado.

La evolución en esta clase de sistemas operativos está tomando dos caminos divergentes. Por un lado, están los RTOS (sistemas operativos de tiempo real) tradicionales, que se orientan a dispositivos que exigen que la información se

procese sin demoras introducidas por buffers. Los RTOS se usan en los dispositivos IoT más sofisticados, empleados en ambientes tales como el industrial, el aeronáutico o el cuidado de la salud.

Por el otro lado, están los sistemas operativos menos sofisticados, aunque no por eso menos capaces, que agregan el beneficio de un menor consumo energético y menores necesidades de recursos.

Es altamente probable que la tasa de crecimiento de esta segunda clase de sistemas operativos sea muy superior a la de los RTOS; al menos en la primera generación de dispositivos IoT.

Entre las posibles opciones, se evaluaron las siguientes:

- Linux (versions open-source: FreeRTOS)
- Linux RIOT
- Linux TinyOS
- Linux Yocto
- Android Brillo
- Ubuntu Core para IoT
- Windows 10 IoT Core

La elección recayó en Windows 10 IoT Core porque posibilita crear o migrar sketches de Arduino Wiring capaces de correr en los dispositivos soportados por IoT Core, lo cual incluye a Raspberry Pi 2, 3 y Minnowboard Max.

Como segundo paso. Se determinó que existen diferentes productos de software para virtualizar. Algunos de los más conocidos son VMWare Workstation Player y Oracle VirtualBox. Algunas de las características destacables y compartidas por ambos son: gratuidad, rendimiento y la posibilidad de poder ser ejecutados sobre un entorno Windows, MacOS o Linux.

Finalmente se decidió optar por Oracle VirtualBox. Esta elección se debió a que posee un mejor rendimiento, de acuerdo con diferentes pruebas realizadas por entidades relevantes y una interfaz más amigable y de mayor facilidad de uso. Asimismo, es el software del que se dispone en la UNLaM.

Para correr la Plataforma de Pruebas a efectos de realizar el Testing del Framework se instaló el sistema operativo Windows 10 IoT Core en la Máquina Virtual Oracle VirtualBox.

En la Imagen 1 se muestra el cuadro de diálogo para crear la máquina virtual con el sistema operativo Windows 10 IoT Core:

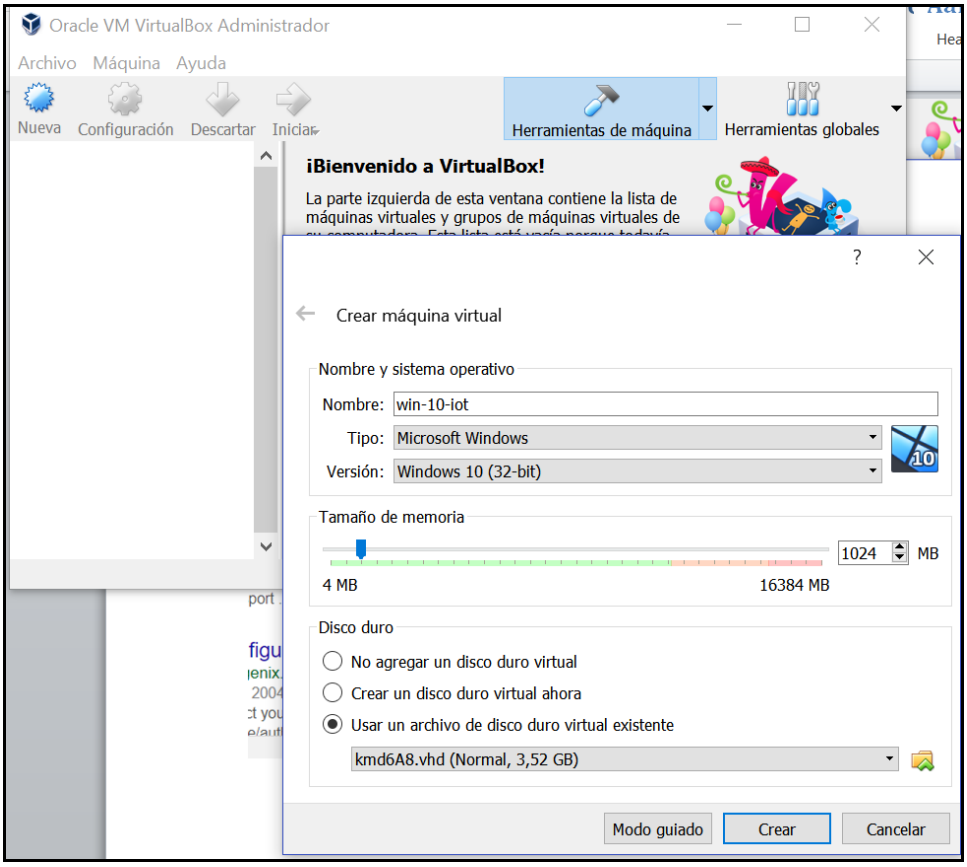


Imagen 1

En las opciones de “Configuración” – “Seleccionar”, de debe habilitar la opción EFI (solo para SO especiales), según se muestra en la Imagen 2:

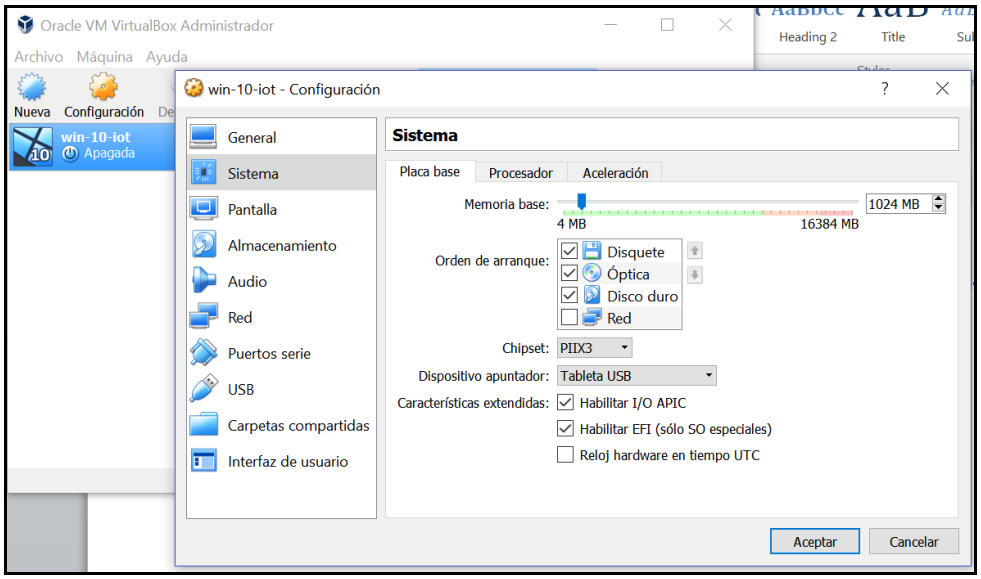


Imagen 2

La opción “Red” debe estar en el modo “Puente”. Si no se reconoce ninguna placa de red, hay que probar con más opciones, siempre en modo “Puente”, como se muestra en la Imagen 3:

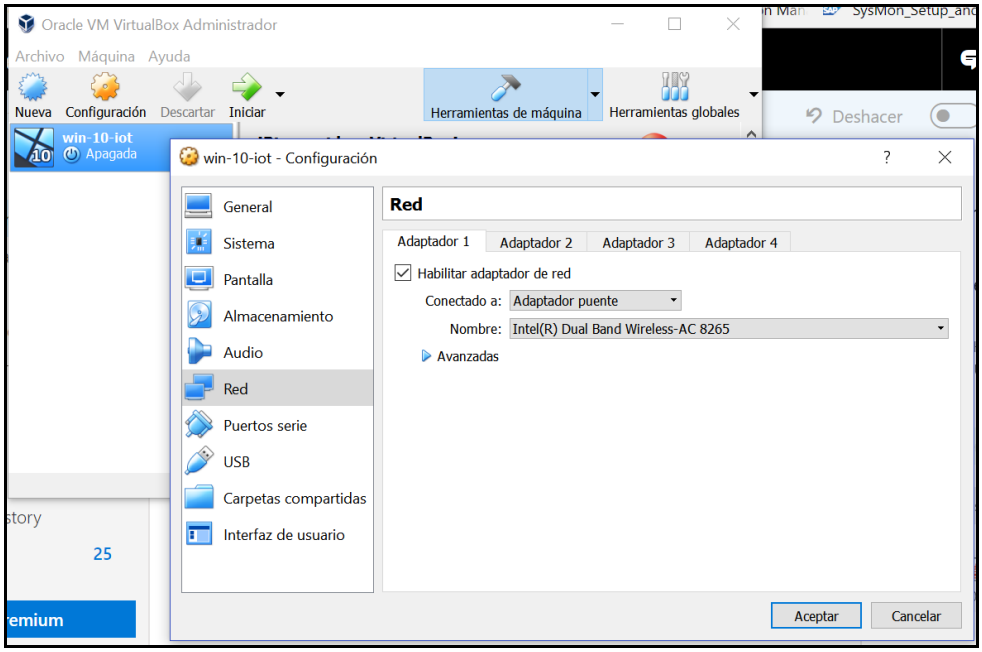


Imagen 3

El tipo de adaptador se puede cambiar, si la placa de red no se reconoce por defecto, tal como se muestra en la Imagen 4:

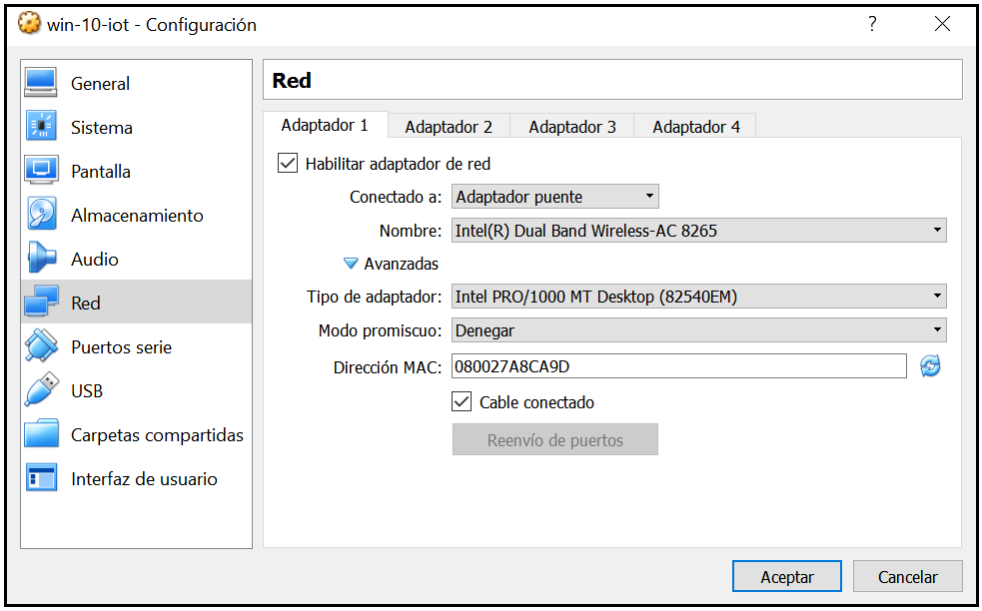


Imagen 4

El tercer paso fue tomar la decisión de la elección del lenguaje del Programa Evaluador donde la Metodología propuesta finalmente será implementada. En particular para realizar este trabajo de investigación y teniendo en cuenta que más adelante se seguirán profundizando los estudios en futuros trabajos, se decidió por desarrollar el Programa Marco o Framework en lenguaje C.

Este lenguaje es uno de los más populares y difundidos de la actualidad por su ductilidad, expresividad y eficacia para llevar adelante todo tipo de algoritmos. Es capaz de producir programas reducidos en extensión y notablemente veloces en su ejecución. Por consiguiente, se logran mediciones de tiempo más exactas.

Asimismo, permite un manejo de código a más bajo nivel que otros lenguajes y simultáneamente una sintaxis sencilla y mayor facilidad en la lectura e interpretación del código, por ser un lenguaje estructurado.

Etapa 2: Selección de los criptosistemas candidatos.

Esta decisión debe responder al análisis de las características descriptas en el punto: "*Descripción del Objeto de Estudio*", de acuerdo con la norma ISO/IEC 29192-1:2012. Es importante que se elijan los algoritmos con mejor rendimiento, teniendo en cuenta las métricas presentadas en la norma y de acuerdo con el software/hardware donde se ejecutarán.

Esta Metodología se presenta para hacer un análisis comparativo de 2 algoritmos y así permitir la elección del mejor de ellos. En caso de que ambos tengan un rendimiento estadísticamente semejante y ninguno se distinga en comportamiento del otro, cualquiera de ellos podría seleccionarse.

Esto permite evitar seleccionar un algoritmo que una vez instalado y en ejecución no se desempeñe de acuerdo con lo previsto, dadas las condiciones limitadas particulares de la plataforma de trabajo del mismo. Seleccionar al mejor algoritmo para tal contexto de Software y Hardware permitiría importantes ahorros de recursos económicos y tiempo por parte del fabricante del producto.

Serán objeto de estudio los algoritmos de Criptografía Ligera Clefia y Present. Los algoritmos Clefia y Present son Cifradores en Bloque públicos y de uso libre. A su vez la Norma ISO/IEC 29192 los reconoce e incluye en su listado de algoritmos criptográficos ligeros.

El algoritmo Clefia fue desarrollado por investigadores de Sony Corporation en el año 2007 [1] [3]. El mismo es un algoritmo de Cifrado en Bloques de 128 bits y con la posibilidad de usar claves de 128, 198 y 256 bits de longitud.

En principio fue creado para dotar de Autenticación y Protección de Derechos de Autor en sistemas DRM (Digital Rights Management) de la empresa. Sus autores también proponen otras aplicaciones de este algoritmo [4], como ser: la generación de números pseudoaleatorios, ser parte componente de funciones hash o la generación de claves en cadena o flujo.

De manera esquemática el algoritmo se divide en dos partes diferentes que procesan la clave y la información a cifrar, llamadas Key Scheduling Part y Data Processing Part respectivamente. Ver Figura 2.

Emplea una Estructura o Red de Feistel que divide la información a cifrar en bloques de 128 bits. A su vez cada bloque es dividido en 4 sub-bloques de 32 bits y los hace transitar por 4 caminos o "ramas".

Esto permite el uso de funciones f de menor tamaño respecto de una Estructura de Feistel de 2 "ramas". Al ser estas más pequeñas el efecto de difusión que se persigue no es tan bueno, por lo que se aumenta el número de vueltas para compensar y mantener alta la seguridad.

Además, Clefia tiene otras propiedades que hacen que pueda ser implementado tanto en Hardware como en Software.

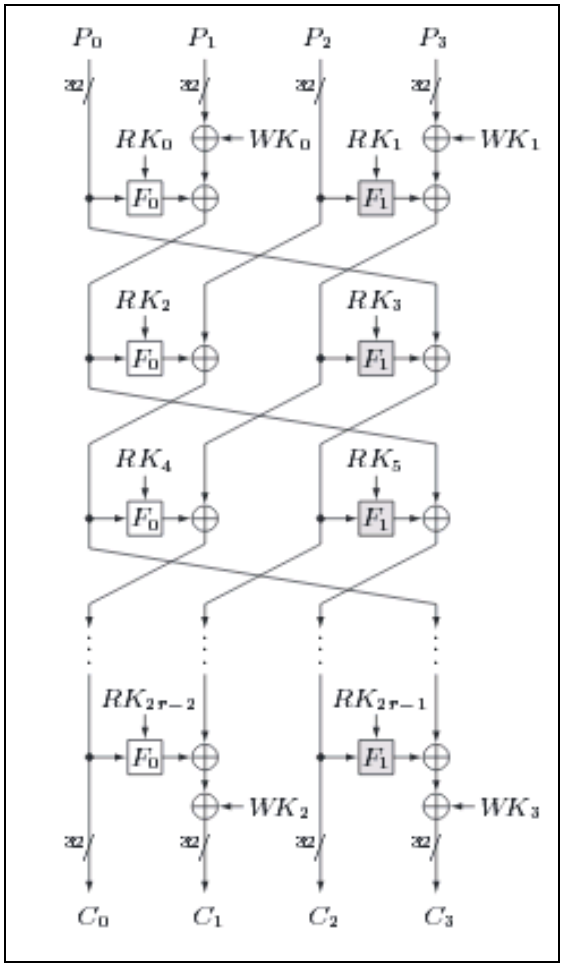


Figura. 2: detalle de diseño del algoritmo Clefia

Present [5] fue desarrollado, en colaboración, por científicos de Horst-Görtz-Institute for IT-Security, Ruhr-University Bochum (Alemania), Technical University Denmark, DK-2800 Kgs. Lyngby (Dinamarca) y France Telecom R&D, Issy les Moulineaux (Francia) en el año 2007 y presentado en el congreso Cryptographic Hardware and Embedded Systems - CHES 2007 [6].

Present es un algoritmo de Cifrado en Bloques de 64 bits, con dos tamaños de claves posibles: 80 bits y 128 bits de longitud, diseñado para ser implementado en Hardware. Ver Figura 3.

El algoritmo se destaca por su tamaño compacto, alrededor de 2,5 veces más pequeño que el algoritmo AES [7] [8], ya que fue específicamente diseñado para ser implementado en hardware y así optimizar su desempeño.

Present está emparentado de manera lejana con AES, pues como él emplea una red SPN (Substitution Permutation Network).

Por cada bloque de texto claro se aplican 31 rondas consistentes en una etapa o capa de Mezclado de Clave, una etapa de Confusión (S-Box) y una etapa de Difusión) (P-Box) tal como propone Claude Shannon, padre de la Teoría Matemática de la Información. Al final se realiza una última ronda que sólo mezcla la clave.

Aunque ha sido diseñado para Hardware el algoritmo es muy veloz si se lo implementa en Software. Es por ello que se lo puede usar en cualquiera de los dos entornos.

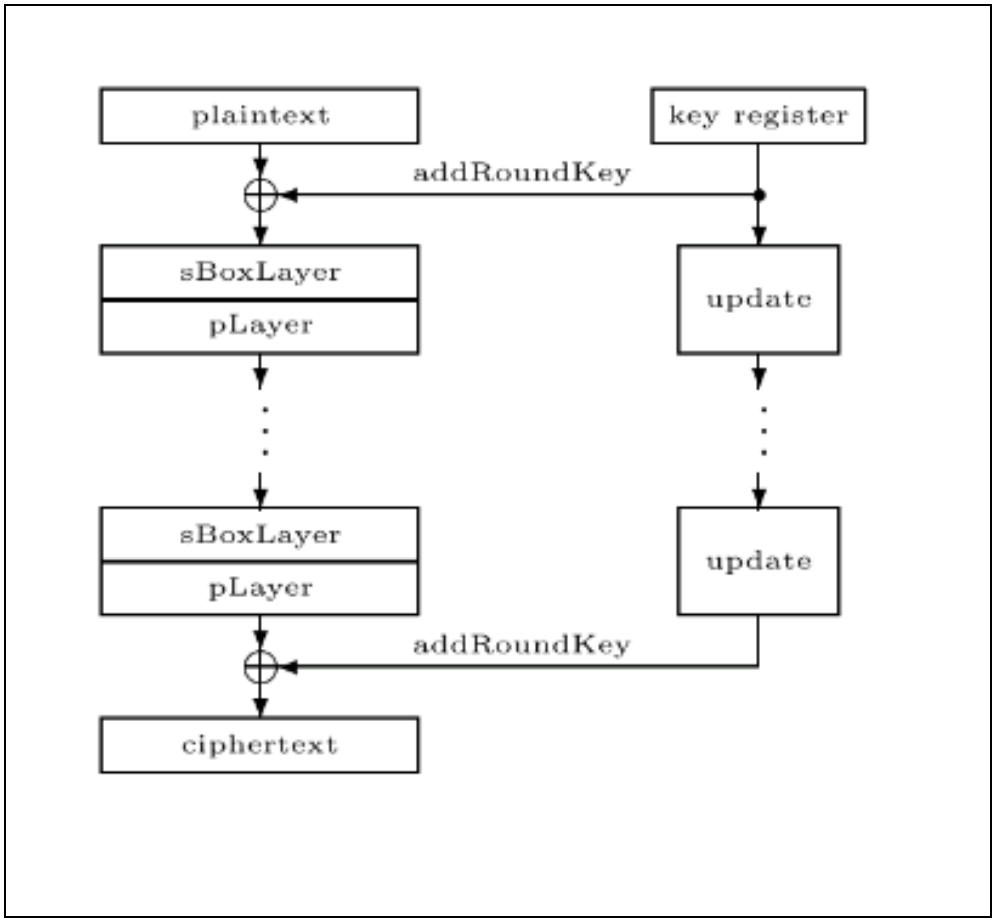


Figura 3: detalle de diseño del algoritmo Present.

Etapa 2: Ejecución de las Pruebas de Carga por medio de un Algoritmo Marco o Framework.

Implementando un Algoritmo Marco o Framework se deberán ejecutar las Pruebas de Carga, tomar el tiempo en que cada algoritmo tarda en hacerlas y registrarlo en una tabla para su posterior análisis estadístico.

Para tal fin se necesita que se tenga previamente definido:

- La Codificación de los algoritmos candidatos en el lenguaje que se determine para la prueba de los mismos.
- El Tamaño de las pruebas (en bits) para llevar adelante.
- La Cantidad de pruebas que se desean realizar a cada algoritmo.

Se propone que estos últimos 2 ítems sean valores que correspondan a potencias de 2, dado que mediante secuencias binarias equilibradas en 1`s y 0`s se persigue que las pruebas se lleven adelante con muestras de bits con las mejores propiedades estadísticas de equiprobabilidad y distribución.

En la Figura 4 se muestra, en forma esquemática, el diseño del algoritmo evaluador:

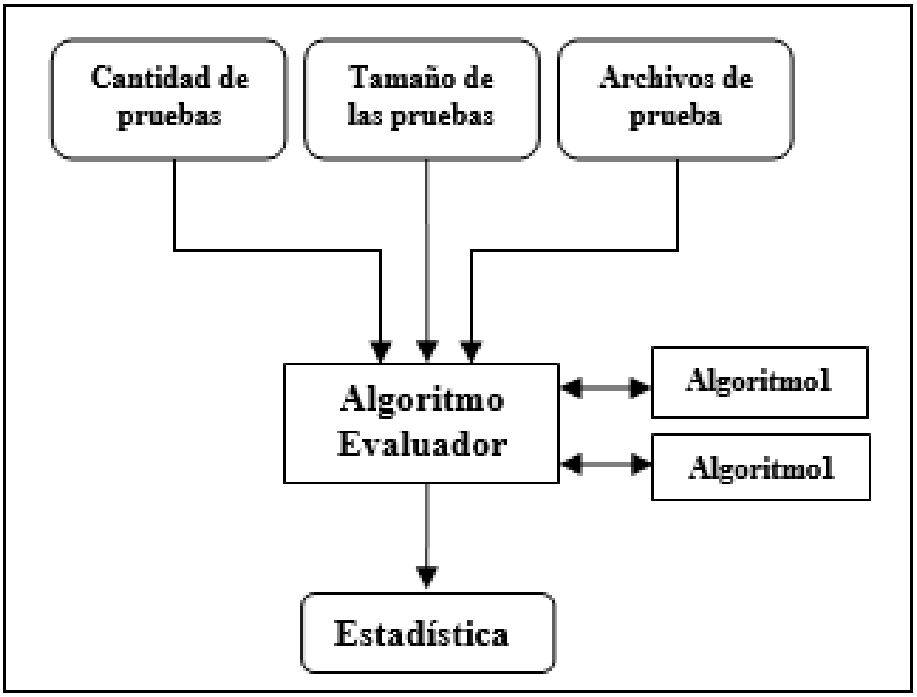


Figura 4: diagrama en bloques del framework

Donde los archivos de prueba servirán para ser cifrados y descifrados la cantidad de veces que se especifique en la variable: “Cantidad de Pruebas”.

Dichos archivos estarán formados por secuencias seudoaleatorias de bits producidos por LFSRs basados en Polinomios Primitivos en GF(2) y de tamaños variables, acordes a los tamaños de los archivos de prueba posibles.

Se propone el uso de LFSR´s y no NLFSR´s pues estos últimos no garantizan las propiedades de ciclos máximos y la distribución equiprobable de 1´s y 0´s que se persigue.

Si el LFSR tuviera una longitud L, entonces la cantidad de bits de la secuencia máxima que produciría sin ciclar sería:

$$|S| = 2^L - 1 \qquad (1)$$

Donde S es la secuencia obtenida, |S| la cantidad de bits de la secuencia y, L es la longitud del registro. En la Figura 5 se muestra el esquema de un LFSR de 9 estados internos:

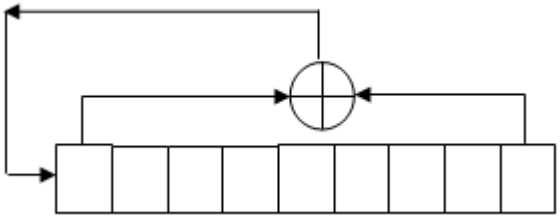


Figura 5: esquema de un LFRS.

Por propiedad de los LFSR basados en polinomios primitivos, la cantidad de 1´s en la secuencia es mayor por un valor, que la cantidad de 0´s. Para compensar y que tengan la misma cantidad, se debe agregar un 0 en la parte de la secuencia con L-1 ceros consecutivos. La secuencia ahora obtenida recibe el nombre de Secuencia de De Bruijn. Esta modificación corrige la fórmula (1) así:

$$|S| = 2^L \quad (2)$$

Entonces se tiene:

$$L = \log_2(T) \quad (3)$$

Siendo L la longitud de los LFSR a usar para construir los archivos de prueba y T el tamaño de los archivos de prueba.

Además de los Archivos de Prueba y los tamaños de dichos archivos, también el Programa Evaluador podrá ejecutar los algoritmos candidatos de acuerdo con un orden que responda a una secuencia seudoaleatoria. Este aspecto del proceso permite reducir al mínimo cualquier factor incontrolable que pueda interferir con la obtención de los tiempos de ejecución.

Nuevamente el uso de LFSR´s permite acercar al máximo la ejecución de los algoritmos lo más cercana al azar. Por ejemplo, haciendo que el “candidato 1” se ejecute, por ejemplo, cuando el bit de salida del LFSR elegido sea 0 y el “candidato 2” cuando el bit sea 1.

Se tiene entonces que:

$$L = 1 + \text{Log}_2 P \quad (4)$$

Donde L es la longitud del LFSR “selector” y P la cantidad de las pruebas que se desean realizar.

Por ejemplo, si P=1024 entonces la longitud del registro será de 11. De acuerdo con la ecuación (2), respetando la corrección sugerida anteriormente, se tiene que la longitud de la secuencia será de 2048 bits. La mitad de ellos, serán 1024 ceros y 1024 unos. Dado que al salir 0 se ejecuta un algoritmo y el otro al salir 1, entonces cada uno de ellos se ejecutó la cantidad deseada de veces.

A efectos de evaluar el rendimiento de cada algoritmo Criptográfico Ligero, el Programa Evaluador toma el tiempo en que cada algoritmo tarda en ejecutar el cifrado/descifrado de los archivos de carga.

Una vez que se han terminado las pruebas, el programa efectuará un análisis estadístico asumiendo que estos experimentos pertenecen a una variable de distribución discreta. Tal estudio busca hallar los tiempos medios de cifrado y descifrado para cada tamaño de archivo de prueba.

$$t_m = \frac{1}{p} \sum_{i=1}^{m} t_i \quad (5)$$

Donde *tm* es el tiempo medio (de cifrado o descifrado), *p* la cantidad de pruebas ejecutadas para cada algoritmo y cada *ti* es el tiempo de ejecución de la prueba *i*.

$$s = \sqrt{\frac{1}{p-1}\sum_{i=1}^{m}(t_i - t_m)^2} \qquad (6)$$

Donde el argumento de la raíz recibe el nombre de Varianza de la Distribución de Probabilidad Discreta y *s* el Desvío Estándar, *tm* es el tiempo medio, *p* la cantidad de pruebas ejecutadas para cada algoritmo y cada *ti* es el tiempo de ejecución de la prueba *i.*

El programa calcula los valores de las ecuaciones (5) y (6) para cada algoritmo y para cada proceso de cifrado y descifrado. Finalmente, el programa emite los valores de *tm* y *s*. El programa seleccionado será aquel con los menores valores *tm* y *s*.

Con estas premisas se diseñó y desarrolló un Programa Evaluador específicamente para la realización de las pruebas, que permita comparar el rendimiento de dos algoritmos Criptográficos Ligeros.

Luego de instalada la máquina virtual, con el correspondiente sistema operativo, se instaló el Programa Evaluador que hemos denominado: “Comparador Criptográfico”. El mismo funcionará como plataforma de las pruebas de carga del Framework. Se describe a continuación un ejemplo de funcionamiento del mismo.

Se tienen 2 algoritmos candidatos para asegurar las comunicaciones en determinado sistema que utilizará tarjetas tipo “MIFare Classic 1K” con transferencia de archivos de 1Kb (1024 bits) de longitud.

Para ello se ha decidido que cada uno de ellos se ejecute 1.048.576 veces, (es decir 220) para minimizar cualquier factor que pueda interferir con la toma de las mediciones de tiempo. Sólo habrá un tamaño de Archivo de Prueba de 1024 bits.

Se alimentará al Programa Evaluador con el valor de la cantidad de pruebas (p=1024), la cantidad de archivos de prueba según la fórmula (1) y la longitud del tamaño de las pruebas (1024 bits).

Con la información cargada, entonces el programa calculará la fórmula (3) y seleccionará el LFSR de tamaño adecuado para generar el contenido del archivo de prueba y lo generará. En este caso elegirá un LFSR de longitud 10 para generar los 1024 bits del archivo. Debe programarse la corrección mencionada anteriormente en la fórmula (2).

Luego el programa usará la fórmula (4) y seleccionará el LFSR acorde a la misma. Con él generará el “selector”. En este caso el valor de L del mismo será de 21. Por lo que la longitud de dicha secuencia asciende a 2.097.152, la mitad de ella (1.048.576) corresponderá a la ejecución de cada algoritmo.

El programa llamará a cada algoritmo asignándole el archivo de carga para cifrado y luego para descifrado, tomando el tiempo en cada proceso y

almacenándolo en una lista. Repetirá el proceso hasta que no queden bits en el selector. Al finalizar el proceso, computará los valores de las fórmulas (5) y (6) para cada algoritmo y para cada proceso. Al cabo de lo cual emitirá un informe con los valores calculados.

En el caso de estudio, se hicieron corridas del programa para los algoritmos Clefia y Present, cifrando y descifrando 6 “archivos de prueba” cuyos tamaños fueron de 16 kb, 32 kb, 64 kb, 128 kb, 256 kb y 512 kb.

Luego se ejecutan los algoritmos seleccionados en secuencia pseudo-aleatoria, utilizando también un LFSR para definir el orden de ejecución. De esta forma la cantidad de cifrados y descifrados es idéntica para ambos algoritmos.

La forma de uso es la siguiente:

- Se ejecuta el archivo "ComparadorCripto.exe" siguiendo las instrucciones en pantalla.
- Una vez terminado el procesamiento, se muestran los resultados en pantalla.
- Se generará un archivo con el reporte de los resultados en una carpeta creada ad-hoc. Se debe tener presente que este archivo se sobrescribe con cada ejecución.
- Para poder utilizar este programa, los algoritmos de cifrado deben estar previamente compilados.

El archivo "Configuracion.ini" del programa permite especificar los algoritmos que se van a ejecutar, con su correspondiente línea de comando para cifrado y para descifrado.

Las entradas de este archivo poseen el siguiente formato:

[NOMBRE]
cifrado=COMANDO1
descifrado=COMANDO2

Donde:

- "NOMBRE" es el nombre del algoritmo o modo de ejecución que aparece dentro del programa.
- "COMANDO1" es la línea de comando que se ejecutará al cifrar, incluyendo el ejecutable del algoritmo.
- “COMANDO2” es la línea de comando para descifrar.
- Se debe tener en cuenta que, al escribir la línea de comando, es necesario reemplazar el nombre del archivo de entrada por "#INPUT" y el de salida por "#OUTPUT". Por ejemplo: "Present.exe #INPUT #OUTPUT E 80".

El programa “Comparador Criptográfico” además de entregar los “archivos de prueba” a las aplicaciones criptográficas, registra el tiempo en el que los algoritmos a testear tardan en cifrar/descifrar dicho archivo y almacena los tiempos de ejecución.

- **Confiabilidad y Validez de la Medición**

  Se validó el programa del Framework con un caso de estudio simulando el funcionamiento de los dos Algoritmos Criptográficos Livianos seleccionados.

  En las corridas del programa “Comparador Criptográfico”, se usaron las siguientes variantes de los algoritmos Clefia y Present:

  - Present de 80 bits
  - Present de 128 bits
  - Clefia ECB de 128 bits
  - Clefia ECB de 192 bits
  - Clefia ECB de 256 bits
  - Clefia CBC de 128 bits
  - Clefia CBC de 192 bits
  - Clefia CBC de 256 bits

  Del estudio estadístico de los datos obtenidos en la ejecución de cada algoritmo en sus distintas variantes, con las diferentes cargas de trabajo a los que se los sometió en cada prueba, se obtuvieron los resultados del comportamiento de los mismos.

- **Métodos de Análisis Estadísticos**

  Se escogieron los tamaños de archivos de 16 kb, 32 kb, 64 kb, 128 kb, 256 kb y 512 kb, debido a las características propias del diseño de los algoritmos seleccionados, ya que los mismos no fueron creados para cifrar grandes volúmenes de datos.

  Los archivos de prueba contienen secuencias pseudoaleatorias binarias generadas por un LFSR, a los efectos de evitar que cualquier sesgo en la distribución de los bits pueda afectar las pruebas.

  Para minimizar los efectos y perturbaciones que los algoritmos puedan sufrir al ejecutarse, las pruebas se realizaron un gran número de veces. Este valor lo deberá determinar el responsable de la selección. El reducido tamaño de los archivos de prueba permite que los test se ejecuten un número elevado de veces. Por ejemplo, en nuestro análisis se ejecutaron 1024 cifrados y 1024 descifrados.

  Además, la Plataforma de Pruebas se puede disponer de manera aleatoria la elección tanto del tamaño del archivo a cifrar/descifrar como así también del algoritmo y su variante. De esta manera se puede asegurar la mayor aleatoriedad posible en la realización de las pruebas y por consiguiente la minimización de sesgos.

  Cabe aclarar que, para algunos algoritmos, previamente a su corrida de cifrado o descifrado, se le deberán realizar tareas de puesta a punto, como por ejemplo ejecutar procesos para la obtención de las subclaves, key whitening u otras actividades. El key whitening consiste en realizar pasos que combinan datos con porciones de la clave, normalmente con operaciones XOR, antes o después de las rondas de cifrado.

Este tiempo previo puede o no considerarse en la prueba. Queda a criterio del evaluador la inclusión o no de los mismos en la medición de los tiempos. Se sugiere que el proceso se ajuste lo más posible al que finalmente se realizará en el dispositivo IoT de destino.

- **Resultados**

  Luego de realizar un relevamiento exhaustivo de los principales algoritmos criptográficos ligeros existentes; estudiar y determinar qué algoritmos se podrían utilizar para dispositivos RFID de bajo costo; estudiar los diferentes sistemas de RFID en función de sus frecuencias: bajas, altas y ultra altas (UHF), tal como figura en la planificación del proyecto, se ha determinado el tipo de entorno de virtualización a utilizar y los algoritmos que serán utilizados para probar el mismo.

  Ellos son los sistemas Clefia y Present. La selección de estos algoritmos radica en la amplia gama de aplicaciones que tienen, sus propiedades criptológicas livianas y la posibilidad de utilizarse en dispositivos RFID.

  Los resultados obtenidos en las distintas corridas del programa ““Comparador Criptográfico” son los siguientes:

  ==========
  Peso del archivo de prueba (en bits): 131072
  ==========
  Present 80 bits

  Cantidad de cifrados: 1024
  Cantidad de descifrados: 1024
  Cifrado - Tiempo Medio: 0.05190723
  Cifrado - Desvío Estándar: 0.00732517
  Descifrado - Tiempo Medio: 0.04851465
  Descifrado - Desvío Estándar: 0.00488897
  ==========
  Present 128 bits

  Cantidad de cifrados: 1024
  Cantidad de descifrados: 1024
  Cifrado - Tiempo Medio: 0.05193652
  Cifrado - Desvío Estándar: 0.00739388
  Descifrado - Tiempo Medio: 0.04854004
  Descifrado - Desvío Estándar: 0.00492310
  ==========
  Clefia ECB 128 bits

  Cantidad de cifrados: 1024
  Cantidad de descifrados: 1024
  Cifrado - Tiempo Medio: 0.01317383
  Cifrado - Desvío Estándar: 0.00573333
  Descifrado - Tiempo Medio: 0.01300098
  Descifrado - Desvío Estándar: 0.00580061
  ==========
  Clefia ECB 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.01524414
Cifrado - Desvío Estándar: 0.00293777
Descifrado - Tiempo Medio: 0.01521777
Descifrado - Desvío Estándar: 0.00290051
==========
Clefia ECB 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.01739551
Cifrado - Desvío Estándar: 0.00501657
Descifrado - Tiempo Medio: 0.01724219
Descifrado - Desvío Estándar: 0.00482701
==========
Clefia CBC 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.01283887
Cifrado - Desvío Estándar: 0.00604677
Descifrado - Tiempo Medio: 0.01319824
Descifrado - Desvío Estándar: 0.00563736
==========
Clefia CBC 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.01520508
Cifrado - Desvío Estándar: 0.00293942
Descifrado - Tiempo Medio: 0.01519531
Descifrado - Desvío Estándar: 0.00294905
==========
Clefia CBC 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.01771582
Cifrado - Desvío Estándar: 0.00534959
Descifrado - Tiempo Medio: 0.01740234
Descifrado - Desvío Estándar: 0.00500618
==========
Peso del archivo de prueba (en bits): 262144
==========
Present 80 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.10118164
Cifrado - Desvío Estándar: 0.00782058
Descifrado - Tiempo Medio: 0.09474609
Descifrado - Desvío Estándar: 0.00414332
==========

Present 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.10132227
Cifrado - Desvío Estándar: 0.00781805
Descifrado - Tiempo Medio: 0.09478906
Descifrado - Desvío Estándar: 0.00419962
==========
Clefia ECB 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.02265625
Cifrado - Desvío Estándar: 0.00777035
Descifrado - Tiempo Medio: 0.02352637
Descifrado - Desvío Estándar: 0.00782002
==========
Clefia ECB 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.02751562
Cifrado - Desvío Estándar: 0.00663765
Descifrado - Tiempo Medio: 0.02773535
Descifrado - Desvío Estándar: 0.00653207
==========
Clefia ECB 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.03195605
Cifrado - Desvío Estándar: 0.00334451
Descifrado - Tiempo Medio: 0.03187207
Descifrado - Desvío Estándar: 0.00321656
==========
Clefia CBC 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.02349805
Cifrado - Desvío Estándar: 0.00780104
Descifrado - Tiempo Medio: 0.02358398
Descifrado - Desvío Estándar: 0.00781261
==========
Clefia CBC 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.02779492
Cifrado - Desvío Estándar: 0.00645663
Descifrado - Tiempo Medio: 0.02763965
Descifrado - Desvío Estándar: 0.00653586
==========

Clefia CBC 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.03199121
Cifrado - Desvío Estándar: 0.00340475
Descifrado - Tiempo Medio: 0.03195508
Descifrado - Desvío Estándar: 0.00335777

==========
Peso del archivo de prueba (en bits): 524288
==========
Present 80 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.19923047
Cifrado - Desvío Estándar: 0.00656046
Descifrado - Tiempo Medio: 0.18679395
Descifrado - Desvío Estándar: 0.00277989
==========
Present 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.19987305
Cifrado - Desvío Estándar: 0.00611470
Descifrado - Tiempo Medio: 0.18671289
Descifrado - Desvío Estándar: 0.00317702
==========
Clefia ECB 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.04361816
Cifrado - Desvío Estándar: 0.00629448
Descifrado - Tiempo Medio: 0.04383008
Descifrado - Desvío Estándar: 0.00613440
==========
Clefia ECB 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.05265430
Cifrado - Desvío Estándar: 0.00755143
Descifrado - Tiempo Medio: 0.05250391
Descifrado - Desvío Estándar: 0.00751720
==========
Clefia ECB 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.06123145
Cifrado - Desvío Estándar: 0.00414782

Descifrado - Tiempo Medio: 0.06113574
Descifrado - Desvío Estándar: 0.00437872
==========
Clefia CBC 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.04399512
Cifrado - Desvío Estándar: 0.00601407
Descifrado - Tiempo Medio: 0.04367480
Descifrado - Desvío Estándar: 0.00625483
==========
Clefia CBC 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.05250977
Cifrado - Desvío Estándar: 0.00752915
Descifrado - Tiempo Medio: 0.05226758
Descifrado - Desvío Estándar: 0.00743566
==========
Clefia CBC 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.06143262
Cifrado - Desvío Estándar: 0.00392119
Descifrado - Tiempo Medio: 0.06137012
Descifrado - Desvío Estándar: 0.00397115

==========
Peso del archivo de prueba (en bits): 1048576
==========
Present 80 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.39532617
Cifrado - Desvío Estándar: 0.00740037
Descifrado - Tiempo Medio: 0.37095508
Descifrado - Desvío Estándar: 0.00650287
==========
Present 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.39631250
Cifrado - Desvío Estándar: 0.00767242
Descifrado - Tiempo Medio: 0.37107520
Descifrado - Desvío Estándar: 0.00642493
==========
Clefia ECB 128 bits

Cantidad de cifrados: 1024

Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.08473242
Cifrado - Desvío Estándar: 0.00773591
Descifrado - Tiempo Medio: 0.08378516
Descifrado - Desvío Estándar: 0.00755642
==========
Clefia ECB 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.10180762
Cifrado - Desvío Estándar: 0.00782301
Descifrado - Tiempo Medio: 0.10214062
Descifrado - Desvío Estándar: 0.00777165
==========
Clefia ECB 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.11973828
Cifrado - Desvío Estándar: 0.00731687
Descifrado - Tiempo Medio: 0.11944434
Descifrado - Desvío Estándar: 0.00742330
==========
Clefia CBC 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.08489355
Cifrado - Desvío Estándar: 0.00776534
Descifrado - Tiempo Medio: 0.08570703
Descifrado - Desvío Estándar: 0.00780932
==========
Clefia CBC 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.10221777
Cifrado - Desvío Estándar: 0.00777798
Descifrado - Tiempo Medio: 0.10235742
Descifrado - Desvío Estándar: 0.00775878
==========
Clefia CBC 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.12026465
Cifrado - Desvío Estándar: 0.00710880
Descifrado - Tiempo Medio: 0.11965820
Descifrado - Desvío Estándar: 0.00737426

==========
Peso del archivo de prueba (en bits): 2097152
==========

Present 80 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.78690918
Cifrado - Desvío Estándar: 0.00784469
Descifrado - Tiempo Medio: 0.73732031
Descifrado - Desvío Estándar: 0.00688695
==========
Present 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.78679004
Cifrado - Desvío Estándar: 0.00774040
Descifrado - Tiempo Medio: 0.73743848
Descifrado - Desvío Estándar: 0.00693301
==========
Clefia ECB 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.16635840
Cifrado - Desvío Estándar: 0.00738188
Descifrado - Tiempo Medio: 0.16576367
Descifrado - Desvío Estándar: 0.00756082
==========
Clefia ECB 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.20158984
Cifrado - Desvío Estándar: 0.00425802
Descifrado - Tiempo Medio: 0.20087988
Descifrado - Desvío Estándar: 0.00512867
==========
Clefia ECB 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.23628125
Cifrado - Desvío Estándar: 0.00551254
Descifrado - Tiempo Medio: 0.23598926
Descifrado - Desvío Estándar: 0.00519934
==========
Clefia CBC 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.16782422
Cifrado - Desvío Estándar: 0.00671879
Descifrado - Tiempo Medio: 0.16736719
Descifrado - Desvío Estándar: 0.00695510
==========

Clefia CBC 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.20300586
Cifrado - Desvío Estándar: 0.00379071
Descifrado - Tiempo Medio: 0.20258398
Descifrado - Desvío Estándar: 0.00412426
==========
Clefia CBC 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.23813672
Cifrado - Desvío Estándar: 0.00688637
Descifrado - Tiempo Medio: 0.23717480
Descifrado - Desvío Estándar: 0.00629510

==========
Peso del archivo de prueba (en bits): 4194304
==========
Present 80 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 1.57391699
Cifrado - Desvío Estándar: 0.00599201
Descifrado - Tiempo Medio: 1.47422070
Descifrado - Desvío Estándar: 0.00782108
==========
Present 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 1.57383691
Cifrado - Desvío Estándar: 0.00559707
Descifrado - Tiempo Medio: 1.47432520
Descifrado - Desvío Estándar: 0.00783709
==========
Clefia ECB 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.32975879
Cifrado - Desvío Estándar: 0.00542607
Descifrado - Tiempo Medio: 0.32954590
Descifrado - Desvío Estándar: 0.00518682
==========
Clefia ECB 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.40084180
Cifrado - Desvío Estándar: 0.00720842

Descifrado - Tiempo Medio: 0.39935840
Descifrado - Desvío Estándar: 0.00764508
==========
Clefia ECB 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.47077734
Cifrado - Desvío Estándar: 0.00598019
Descifrado - Tiempo Medio: 0.46918945
Descifrado - Desvío Estándar: 0.00420646
==========
Clefia CBC 128 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.33318262
Cifrado - Desvío Estándar: 0.00751856
Descifrado - Tiempo Medio: 0.33187109
Descifrado - Desvío Estándar: 0.00696556
==========
Clefia CBC 192 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.40199707
Cifrado - Desvío Estándar: 0.00658288
Descifrado - Tiempo Medio: 0.40143945
Descifrado - Desvío Estándar: 0.00693020
==========
Clefia CBC 256 bits

Cantidad de cifrados: 1024
Cantidad de descifrados: 1024
Cifrado - Tiempo Medio: 0.47229004
Cifrado - Desvío Estándar: 0.00698391
Descifrado - Tiempo Medio: 0.47061230
Descifrado - Desvío Estándar: 0.00584059
==========

A los efectos de mostrar gráficamente el comportamiento de los algoritmos y sus variantes para los distintos tamaños de archivos de cifrado/descifrado, se incluyen a continuación algunos gráficos que nos ayudarán en el desarrollo de las conclusiones.

1. Comparación de cifrados del Algoritmo Present de 80 y 128 bits

Archivo	Algoritmo Present		Diferencia
	80 bits	128 bits	
16 kb	51,90723000	51,93652000	100,05642759
32 kb	101,18164000	101,32227000	100,13898767
64 kb	199,23047000	199,87305000	100,32253099

128 kb	395,32617000	396,31250000	100,24949778
256 kb	786,90918000	786,79004000	99,98485975
512 kb	1573,91699000	1573,83691000	99,99491206

Tabla 3: tiempo de cifrado Present

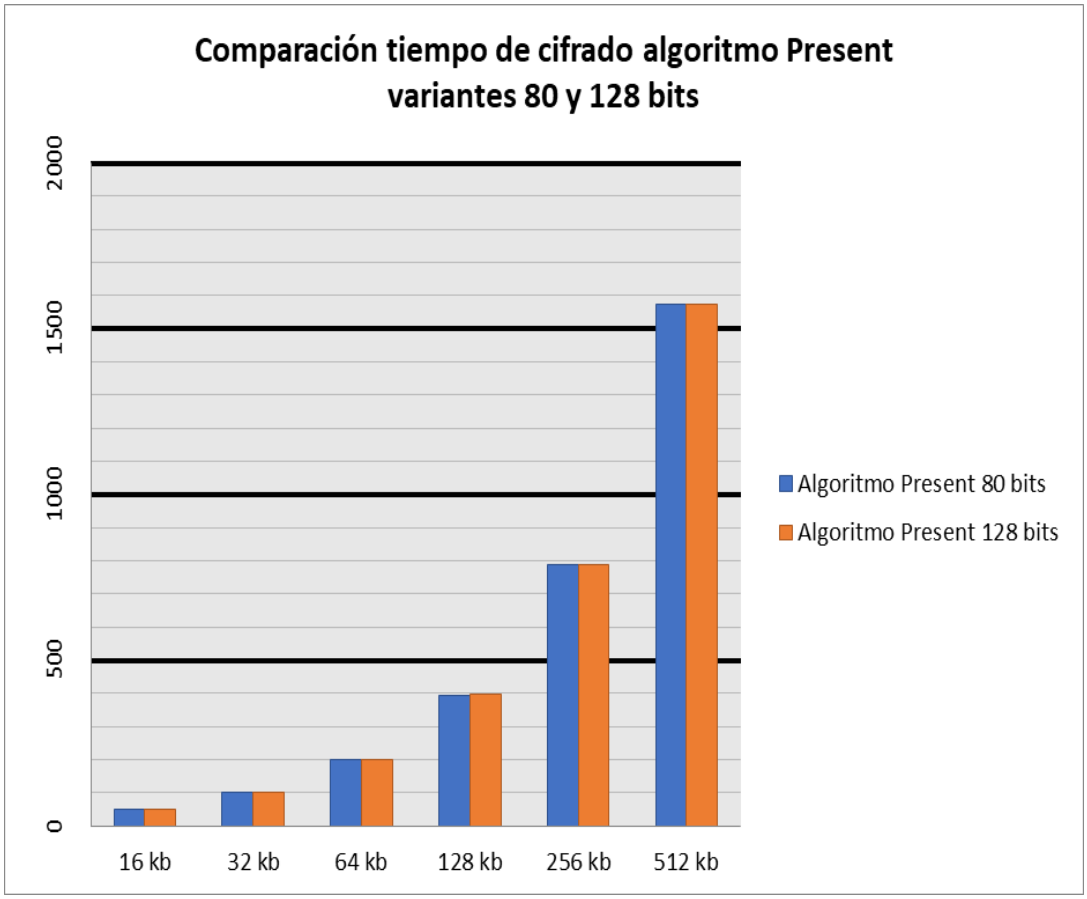


Gráfico 1: tiempo de cifrado de Present 80 y 128 bits

2. Comparación de descifrados del Algoritmo Present de 80 y 128 bits

Archivo	Algoritmo Present		Diferencia
	80 bits	128 bits	
16 kb	48,51465000	48,54004000	100,05233471
32 kb	94,74609000	94,78906000	100,04535280
64 kb	186,79395000	186,71289000	99,95660459
128 kb	370,95508000	371,07520000	100,03238128
256 kb	737,32031000	737,43848000	100,01602696
512 kb	1474,22070000	1474,32520000	100,00708849

Tabla 4: tiempo de descifrado Present

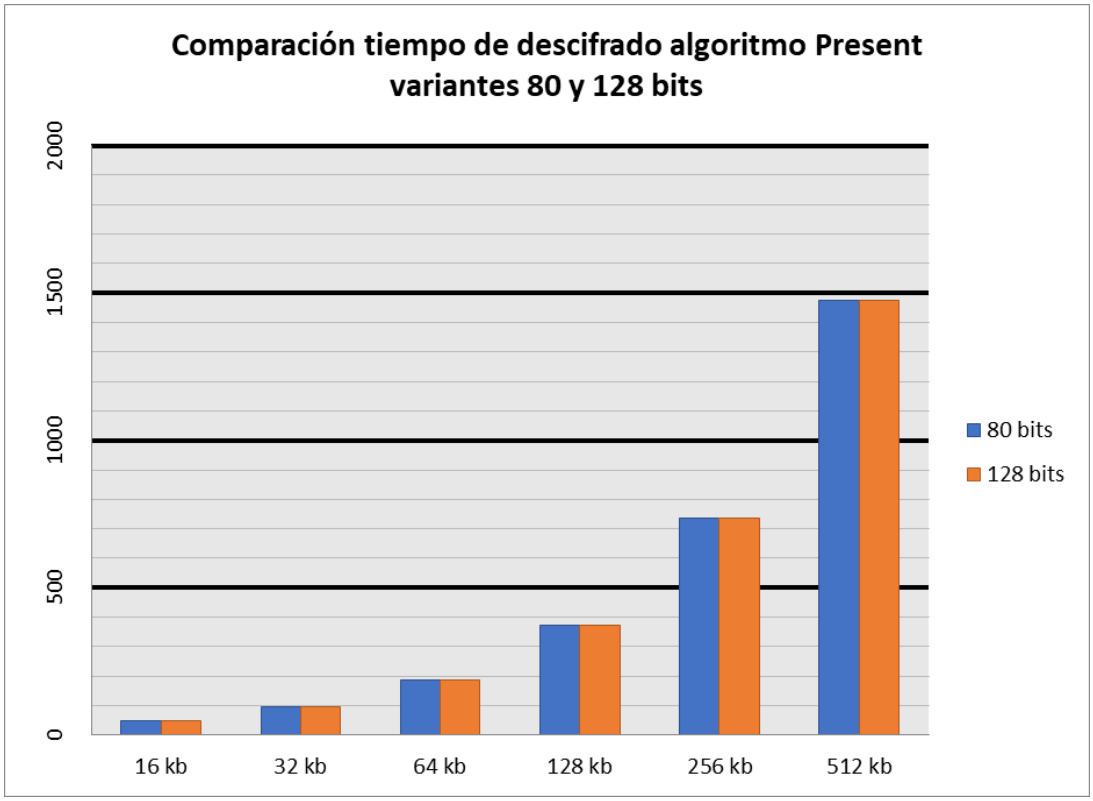


Gráfico 2: Comparación tiempo descifrado de Present 80 y 128 bits

3. Comparación de cifrado y descifrado del algoritmo Present de 80 bits

Archivo	Present 80 bits		Diferencia
	Cifra	Descifra	
16 kb	51,90723000	48,51465000	93,46414748
32 kb	101,18164000	94,74609000	93,63960695
64 kb	199,23047000	186,79395000	93,75772190
128 kb	395,32617000	370,95508000	93,83519437
256 kb	786,90918000	737,32031000	93,69827278
512 kb	1573,91699000	1474,22070000	93,66572121

Tabla 5: comparación de tiempos de cifrado y descifrado Present 80 bits

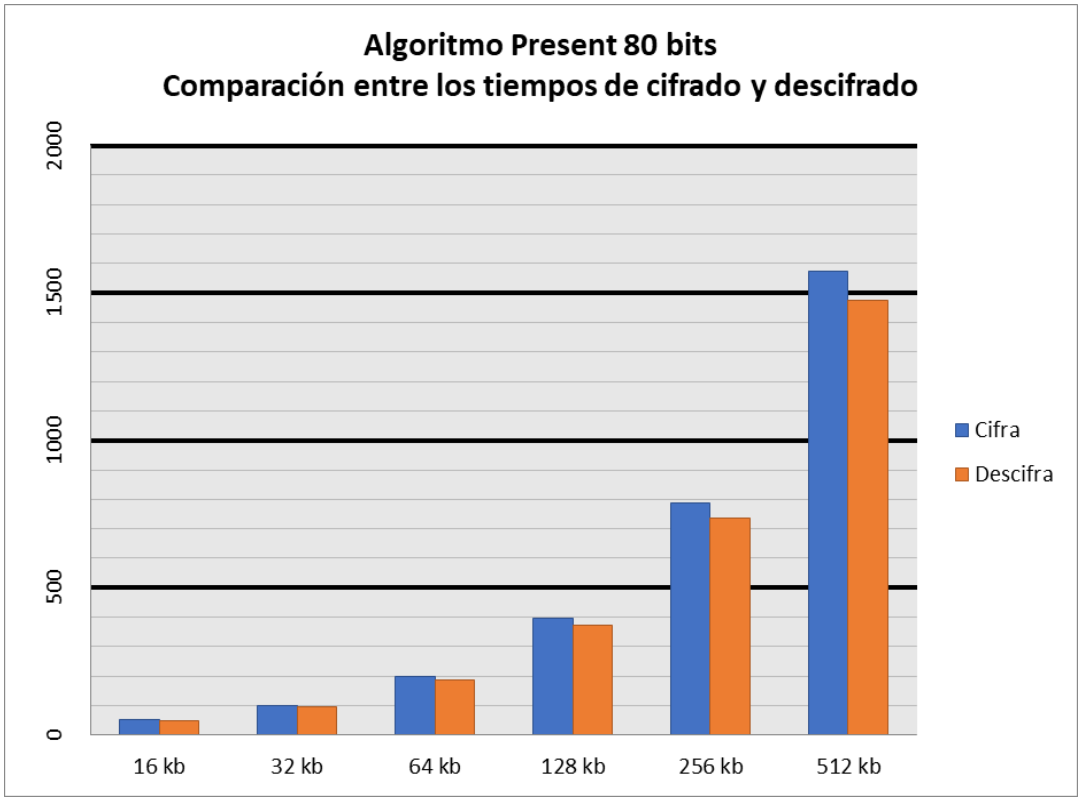


Gráfico 3: tiempos de cifrado y descifrado Present 80 bits.

4. Comparación de cifrado y descifrado del algoritmo Present de 128 bits

Archivo	Present 128		Diferencia
	Cifra	Descifra	
16 kb	51,93652000	48,54004000	93,46032426
32 kb	101,32227000	94,78906000	93,55204932
64 kb	199,87305000	186,71289000	93,41574064
128 kb	396,31250000	371,07520000	93,63196972
256 kb	786,79004000	737,43848000	93,72748033
512 kb	1573,83691000	1474,32520000	93,67712694

Tabla 6: comparación de tiempos Present 128 bits

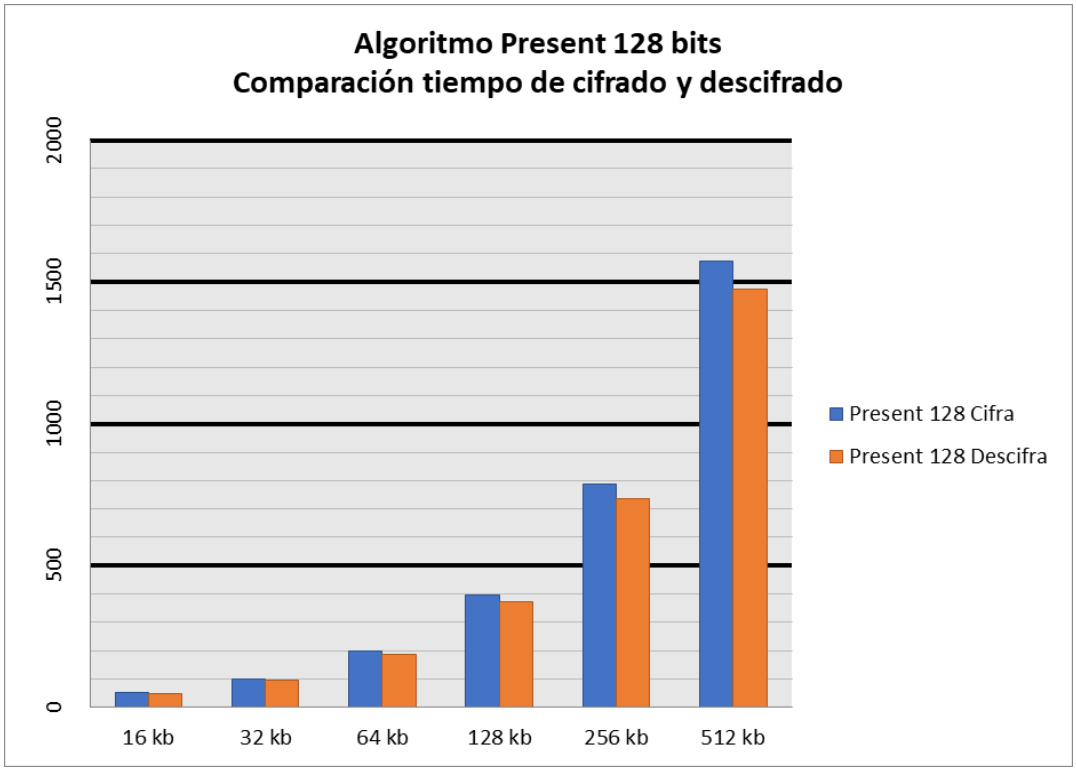


Gráfico 4: tiempo de cifrado y descifrado de Present 128 bits

En cuanto a la Formación de Recursos Humanos, este proyecto surge de la materia electiva Criptografía de la carrera de Ingeniería en Informática de la UNLaM, de donde se habían sumado un par de alumnos luego de haberla cursado y de descubrir su interés por profundizar en esta área del conocimiento.

Pero lamentablemente, debido a que estos alumnos ya se han recibido de Ingenieros y se encuentran trabajando en empresas de primera línea, no dispusieron del tiempo necesario para continuar participando del proyecto de investigación durante el 2do año del mismo.

Con respecto a la difusión de resultados preliminares en Jornadas y Congresos y en publicaciones en revistas especializadas, se ha enviado un paper el XVIII Workshop de Investigadores en Ciencias de la Computación – WICC 2016, el cuál fue aprobado para su presentación en la Universidad Nacional de Entre Ríos, el 14 y 15 de abril de 2016.

También se envió un trabajo a la Décima Sexta Conferencia Iberoamericana en Sistemas, Cibernética e Informática - CISCI 2017, desarrollado del 08 al 11 de julio de 2017 en Orlando, Florida, USA. Este trabajo también fue aceptado para su presentación en la modalidad “virtual”.

Otro trabajo fue enviado y aceptado para su presentación en el CoNaIISI 2017, 5to Congreso Nacional de Ingeniería Informática – Sistemas de Información, desarrollado en la Universidad Tecnológica Nacional – Facultad Regional Santa Fe, los días 02 y 03 de noviembre de 2017.

Los trabajos se encuentran en el Anexo IV.

- **Discusión**

  Finalizadas las pruebas de carga y luego de realizados los estudios estadísticos de los resultados, se procedió al análisis y discusión de los datos obtenidos, determinando así al algoritmo que mejor performance y rendimiento ha tenido en todas las pruebas, si es que alguno se destaca.

  Dado que el proyecto no buscaba elegir al mejor algoritmo sino diseñar, desarrollar y presentar un entorno de trabajo que sí realice tal tarea y ofrecerla a la comunidad científica, el análisis del comportamiento de los algoritmos en los diversos test a los que fueron sometidos queda fuera del alcance de este trabajo. Pudiendo hacerse efectivo este estudio en futuros Trabajos y Proyectos de Investigación.

- **Conclusiones**

  Como este trabajo presenta una innovación en la manera de evaluar aplicaciones criptográficas ligeras usando pruebas de carga en entornos virtualizados, consideramos que podríamos estar haciendo un aporte significativo a la comunidad criptológica al poner a disposición un framework para medir el rendimiento de las mismas.

  Asimismo, se seleccionaron como ejemplos para evaluar a los algoritmos Clefia y Present, pertenecientes a la Criptografía Ligera, rama de la Criptografía que estudia cómo segurizar sistemas que dispongan de recursos reducidos, como ser la cantidad de memoria, los requerimientos de energía y el poder de cómputo. Como ejemplo de este tipo de dispositivos, se pueden mencionar los Smart Devices, las tarjetas y los sistemas RFID, entre otros.

  El método requiere del diseño y desarrollo de una Plataforma de Pruebas (Testing Framework) para realizar las Pruebas de Carga tendientes a obtener resultados comparativos que permitan hacer la selección del mejor algoritmo a implementar.

  Se sometieron a los algoritmos Clefia y Present a las Pruebas de Carga aquí detalladas y se describen los parámetros estadísticos obtenidos.

  De igual manera se podría desarrollar un método equivalente para evaluar los algoritmos que se orientan al Hardware.

- **Bibliografía**

[1] Shirai, T. et. al. 2007. The 128-Bit Blockcipher CLEFIA (Extended Abstract). Fast Software Encryption

[2] 14th International Workshop, FSE 2007, Vol. 4593. ISBN 978-3-540-74617-1. Luxembourg, 2007.

[3] http://www.sony.net/Products/cryptography/clefia/ (consultada el 20/7/2016).

[4] http://www.sony.net/Products/cryptography/clefia/about/appendix_01.html (consultada el 20/7/2016).

[5] Bogdanov, A. et al. 2007. PRESENT: An Ultra-Lightweight Block Cipher. Cryptographic Hardware and Embedded Systems - CHES 2007. Springer. Berlin.

[6] http://link.springer.com/book/10.1007/978-3-540-74735-2 (consultada el 21-7-2016).

[7] FIPS PUB 197: Advanced Encryption Standard (AES).

[8] ISO/IEC 18033-3: Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers.

[9] http://www.ecrypt.eu.org/stream/ (consultada el 21-7-16).

[10] Vajda I. and Buttyán L. Lightweight authentication protocols for low-cost RFID tags. In Proc. of UBICOMP'03, 2003.

[11] Juels A. Minimalist cryptography for low-cost RFID tags. In Proc. Of SCN'04, volume 3352 of LNCS, pages 149–164. Springer-Verlag, 2004.

[12] A Juels. RFID security and privacy: A research survey. Manuscript, 2005.

[13] H.-Y. Chien. SASI: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity. IEEE Transactions on Dependable and Secure Computing, 4(4):337–340, 2007.

[14] ISO/IEC 29192-2:2012. Information technology - Security techniques - Lightweight cryptography - Part 2: Block ciphers. 2012.

Otras referencias bibliográficas:

- Instituto Nacional de Tecnologías de la Comunicaciòn y Agencia Española de Protecciòn de Datos.: Guìa sobre seguridad y privacidad de la tecnologìa RFID. Spain. 2010. www.inteco.es.

- Román R., Nájera P., López J.; "Securing the Internet of Things". University of Malaga, Spain. 2011.

- Román R., Nájera P., López J. "Los Desafíos De Seguridad En La Internet De Los Objetos" University of Malaga, España. 2010.

- Bhattasali Tapalina. “LICRYPT: Lightweight Cryptography Technique for Securing Smart Objects in Internet of Things Environment”. Research Scholar, University of Calcutta. 2013.

- Heer, T.; Garcia-Morchon, O.; Hummen, R.; Keoh, S.L.; Kumar, S.S.; Wehrle, K. “Security challenges in the IP-based internet of things”. Wirel. Pers. Commun. 61, 527–542. 2011.

- Garcia-Morchon, O.; Keoh, S.; Kumar, S.; Hummen, R.; Struik, R. “Security Considerations in the IP-based Internet of Things”. IETF Internet Draft draft-garcia-core-security-04; The Internet Engineering Task Force (IETF): Fremont, CA, USA, 2012.

- Cirani S., Ferrari G., Veltri L.. “Enforcing Security Mechanisms in the IP-Based Internet of Things: An Algorithmic Overview”. Algorithms 2013, 6, 197-226;

- International Association for Cryptologic Research. 2015. http://www.iacr.org/events/

- GSI- Information Security Group. RFID Security & Privacy Lounge http://www.avoine.net/rfid/index.php

- Suzaki,T. Minematsu K., Morioka S., Kobayashi E. “TWINE: A Lightweight, Versatile Block Cipher”. NEC Corporation, Japan. 2014.

- Bogdanov A., Knudsen L., Leander G. et al. “PRESENT: An Ultra Weight Block Cipher”. Springer-Verlag Berlin Heidelberg 2007 HES 2007, LNCS 4727, pp. 450–466. 2007.

- Wentao Z., Zhenzhen B., Dongdai L., Rijmen V., Yang B., Verbauwhede I.; “RECTANGLE: A Bit-slice Ultra-Lightweight Block Cipher Suitable for Multiple Platforms”. China, Bélgica. 2015.

- Beaulieu R., Shors D y otros. “The SIMON and SPECK Families of Lightweight Block Ciphers”. Cryptology ePrint Archive: Report 2013/404. 2013.

- Mouha N., Mennink B., y otros. “Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers”. Department of Electrical Engineering, Leuven and iMinds, Bélgica. 2013.

- Hongjun W., Tao H. “JAMBU Lightweight Authenticated Encryption Mode and AES-JAMBU (v1)” Division of Mathematical Sciences Nanyang Technological University, Singapur. 2014.

- Engels D., Fan X., Gong G., Hu, H, Smith M. “Hummingbird: Ultra Lightweight Cryptography for Resource-Constrained Devices.” 1st International Workshop on Lightweight Cryptography for Resource-Constrained Devices (WLC’2010). Tenerife, Canary Islands, Spain, 2010.

3. Anexos

Anexo I: Formulario FPI-015: Rendición de gastos del proyecto de investigación acompañado de las hojas foliadas con los comprobantes de gastos.

Anexo II: Documentación de alta de integrante del equipo de investigación.

No corresponde

Anexo III: Copias de certificados de participación de integrantes en eventos científicos.

- Certificado de Autores WICC 2016
- Certificado de Participación de artículo CISCI 2017
- Certificado de Expositor CoNaIISI 2017

Anexo IV: Copia de artículos presentados en publicaciones periódicas, y ponencias presentadas en eventos científicos.

a) Presentaciones a congresos con referato:

- Análisis comparativo de Algoritmos Criptográficos Livianos para dispositivos RFID de bajo costo.
  Eterovic, Jorge; Donadello, Domingo; Cipriano, Marcelo; Capuya, Mara; Pomar, Pablo.
  WICC 2016 - XVIII Workshop de Investigadores en Ciencias de la Computación.
  Universidad Nacional de Entre Ríos.
  Concordia, Provincia de Entre Ríos, 14 y 15 de abril de 2016.

- Propuesta de una Metodología Innovadora para la Evaluación del Rendimiento de Aplicaciones Criptográficas mediante Pruebas de Carga en un Entorno Virtualizado.
  Eterovic, Jorge; Cipriano, Marcelo; Jordi, Brian.
  CISCI 2017 - Décima Sexta Conferencia Iberoamericana en Sistemas, Cibernética e Informática.
  International Institute of Informatics and Systemics.
  Orlando, Florida, USA, 08 al 11 de julio de 2017.

- Propuesta de software evaluador del rendimiento de algoritmos criptográficos para dotar de confidencialidad a dispositivos de IoT en un entorno virtualizado.
  Eterovic, Jorge; Cipriano, Marcelo; Jordi, Brian
  CoNaIISI 2017 – 5to. Congreso Nacional de Ingeniería Informática - Sistemas de Información
  Universidad Tecnológica Nacional – Facultad Regional Santa Fe
  Santa Fe, 02 y 03 de noviembre de 2017

b) Publicaciones con referato:

- Eterovic, Jorge; Donadello, Domingo; Cipriano, Marcelo; Capuya, Mara; Pomar, Pablo; “Análisis comparativo de Algoritmos Criptográficos Livianos para

dispositivos RFID de bajo costo"; WICC 2016 - XVIII Workshop de Investigadores en Ciencias de la Computación; Concordia, Provincia de Entre Ríos, 14 y 15 de abril de 2016; Universidad Nacional de Entre Ríos. RedUNCI. Páginas: 850-853; ISBN: 978-950-698-377-2.
Link: http://sedici.unlp.edu.ar/handle/10915/52766

- Eterovic, Jorge; Cipriano, Marcelo; Brian Jordi; "Propuesta de una Metodología Innovadora para la Evaluación del Rendimiento de Aplicaciones Criptográficas mediante Pruebas de Carga en un Entorno Virtualizado"; CISCI 2017 – 16ta Conferencia Iberoamericana en Sistemas. Cibernética e Informática; Orlando, Florida, EE. UU.; 08 al 11 de julio de 2017; International Institute of Informatics and Systemics; Páginas 310-314.
  ISBN: 978-1-941763-62-9

- Eterovic, Jorge; Cipriano, Marcelo; Brian Jordi; "Propuesta de una Metodología Innovadora para la Evaluación del Rendimiento de Aplicaciones Criptográficas mediante Pruebas de Carga en un Entorno Virtualizado"; CoNaIISI 2017 - 5to Congreso Nacional de Ingeniería Informática / Sistemas de Información; Santa Fe; 02 y 03 de noviembre de 2017; Universidad Tecnológica Nacional – Facultad Regional Santa Fe; Páginas 52-57.
  ISSN: 2347-0372
  Link: http://conaiisi2017.frsf.utn.edu.ar/index.php/memorias/

Anexo V: Alta patrimonial de los bienes adquiridos con presupuesto del proyecto.

No corresponde.

Mag. Jorge E. Eterovic

Anexo I: Conteniendo el formulario FPI-015: Rendición de gastos del proyecto de investigación acompañado de las hojas foliadas con los comprobantes de gastos.

Anexo II: Documentación de alta de integrante del equipo de investigación.

Anexo III: Copias de certificados de participación de integrantes del proyecto de investigación en eventos científicos:

- Certificado de Autores WICC 2016
- Certificado de Participación de artículo CISCI 2017
- Certificado de Expositor CoNaIISI 2017

[FPI-009-Guía de elaboración de Informe de avance y final de proyecto. SECyT- UNLaM. Versión 2.1 13/10/2015]

Facultad de Ciencias de la Administración
UNER

WICC 2016
XVIII Workshop de Investigadores en Ciencias de la Computación

Se certifica que **Jorge E. Eterovic (UNLaM), Domingo F. Donadello (UNLaM), Marcelo Cipriano (UNLaM), Mara Capuya (UNLaM), Pablo Pomar (UNLaM),** han participado en calidad de AUTORES del artículo ***"Análisis comparativo de Algoritmos Criptográficos Livianos para dispositivos RFID de bajo costo",*** aceptado en el XVIII Workshop de Investigadores en Ciencias de la Computación, llevado a cabo los días 14 y 15 de abril de 2016 en la ciudad de Concordia, Entre Ríos.

Concordia, 15 de abril de 2016.

LIC. GUILLERMO E. FEIERHERD
COORDINADOR
RED DE UNIVERSIDADES
CON CARRERAS EN INFORMÁTICA

CR. HIPÓLITO B. FINK
DECANO
FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
UNIVERSIDAD NACIONAL DE ENTRE RÍOS

[FPI-009-Guía de elaboración de Informe de avance y final de proyecto. SECyT- UNLaM. Versión 2.1 13/10/2015]

Certificado de Participación

Se certifica que

Jorge Eterovic

participó virtualmente en la

Décima Sexta Conferencia Iberoamericana en
Sistemas, Cibernética e Informática

celebrada del 8 al 11 de julio de 2017, en Orlando, Florida, EE.UU.

con la ponencia titulada

Propuesta de una Metodología Innovadora para la Evaluación del Rendimiento de Aplicaciones Criptográficas mediante Pruebas de Carga en un Entorno Virtualizado

International Institute of
Informatics and Systemics

Nagib C. Callaos
Presidente del IIIS

[FPI-009-Guía de elaboración de Informe de avance y final de proyecto. SECyT- UNLaM. Versión 2.1 13/10/2015]

REPÚBLICA ARGENTINA

UTN SANTA FE

CONAIISI 2017

CERTIFICADO DE ASISTENCIA

Se certifica que **ETEROVIC, JORGE** DNI Nº **11917135**

ha participado en carácter de **EXPOSITOR**

en el 5º Congreso Nacional de Ingeniería Informática / Sistemas de Información (CONAIISI 2017) realizado los días 2 y 3 de Noviembre del corriente año en la UTN Santa Fe, según Resolución de Consejo Directivo Nº 246/17,

se le otorga el presente certificado.

Santa Fe, Noviembre de 2017.-

Dr. Aldo Vecchietti
Director Dpto. Ing. en Sistemas de Información
UTN - FRSF

Ing. Eduardo Donnet
Decano
UTN - FRSF

Anexo IV: Copia de artículos presentados en publicaciones periódicas, y ponencias presentadas en eventos científicos:

- Trabajo publicado en WICC 2016: “Análisis comparativo de Algoritmos Criptográficos Livianos para dispositivos RFID de bajo costo”
- Trabajo publicado en CISCI 2017: “Propuesta de una Metodología Innovadora para la Evaluación del Rendimiento de Aplicaciones Criptográficas mediante Pruebas de Carga en un Entorno Virtualizado”
- Trabajo publicado en CoNaIISI 2017: “Propuesta de una Metodología Innovadora para la Evaluación del Rendimiento de Aplicaciones Criptográficas mediante Pruebas de Carga en un Entorno Virtualizado”

Análisis comparativo de Algoritmos Criptográficos Livianos para dispositivos RFID de bajo costo

Mg. Jorge Eterovic; Mg. Domingo Donadello; Esp. Marcelo Cipriano; Lic. Mara Capuya; Esp. Pablo Pomar

Programa CyTMA2 / Departamento de Ingeniería e Investigaciones Tecnológicas
Universidad Nacional de La Matanza
Florencio Varela 1903 (B1754JEC), San Justo, (5411) 4480-8900

eterovic@unlam.edu.ar; ddonadel@ing.unlam.edu.ar; cipriano1.618@gmail.com; mcapuya@gmail.com; pablo_pomar@yahoo.com.ar

1. Resumen.

El desarrollo de la Internet de los objetos dará lugar a un inmenso despliegue de millones de objetos inteligentes que interactuarán entre sí y con Internet. El papel de la tecnología RFID será primordial en este escenario.

El ritmo de adopción del RFID es vertiginoso y ya se ha convertido en una realidad. Las etiquetas de bajo costo representan el mayor desafío en términos de seguridad y privacidad, pero sus escasos recursos influyen sobre los métodos criptográficos existentes.

El objetivo de este proyecto de investigación es realizar un análisis comparativo del comportamiento de los Algoritmos Criptográficos Livianos existentes para ser usados en dispositivos RFID de bajo costo.

Palabras Clave:

Criptografía Ligera, Algoritmos Cripto-gráficos Livianos, RFID

2. Contexto.

La Universidad Nacional de La Matanza mantiene una política en la que se fomenta y promueve la investigación académica y la inclusión en ella de alumnos de grado, posgrado y maestría.

Es por ello que esta línea de investigación se en marca en el siguiente programa:

- Programa CyTMA2 (Programa de Investigación Científica, Desarrollo y Transferencia de Tecnologías e Innovaciones. UNLaM).

3. Introducción.

Cuando se habla sobre Internet de las Cosas, Internet of Things (IoT) por sus siglas en inglés), en realidad de lo que se está hablando es de la conectividad a través de Internet entre objetos. Pero Internet de las cosas va mucho más allá. Estas cosas van desde electrodomésticos controlados por un Smartphone hasta niveles más profesionales.

Con la Internet de las Cosas, todo lo real se convierte en virtual, lo que significa que cada persona y las cosas tienen una ubicación en Internet. Estas entidades virtuales pueden producir y consumir servicios y colaborar entre sí con un objetivo en común.

La manera en que estos objetos pueden comunicar o recibir información es a través de sensores que, en algunos casos, pueden visualizarse. Pero no siempre es posible notar su presencia. Dentro de la conexión de los objetos con los sistemas de información, dos son las tecnologías clave que ya se están insertando en diversos sectores de la industria para acercar la Internet de las Cosas a la realidad. Estas tecnologías son la identificación por radiofrecuencia (RFID) y las redes de sensores inalámbricas.

La International Telecommunication Union (ITU), en su "Informe sobre la Internet de las Cosas" califica a la tecnología RFID como un "pivote que habilitará el Internet de las Cosas", permitiendo la conversión de los "objetos cotidianos" en "inteligentes" [1]. Sin embargo, sin bases sólidas de seguridad, es posible que estos objetos sean pasibles de ataques. Estas amenazas podrían llegar a ser cada vez más perjudiciales que cualquiera de sus beneficios [2,3].

Las investigaciones sobre algoritmos criptográficos están avanzadas y cada día se generan nuevos algoritmos para las claves de autenticación. La investigación académica y la Asociación Internacional de Investigaciones en Criptografía (IACR-International Association for Cryptologic Research) [4], en particular, impulsaron la definición de distintos mecanismos que proporcionaron un nivel de seguridad y privacidad adecuados a las limitaciones del hardware de las etiquetas RFID.

Se trabaja continuamente sobre el área que dio en llamarse “Criptografía Ligera” [5] y se abordan los temas de privacidad, protección de datos personales y seguir-dad en las comunicaciones electrónicas sobre las amenazas específicas para las aplicaciones RFID.

La Criptografía Ligera o Liviana (LICRYPT - Lightweight Cryptography) es un nuevo campo de investigación que apunta a estudiar métodos criptográficos con el fin que puedan utilizarse en objetos inteligentes.

La LICRYPT está orientada a Hardware o Software, determinándose parámetros para evaluar y medir las implementacio-nes que apliquen a este tipo de criptografía. Por ejemplo, para hardware se estudian el tamaño de los chips y el consumo de energía que se requiere. Para software, en cambio, se analizan la longitud del código, el uso y consumo de memoria Ram.

En LICRIPT se podrán encontrar: algoritmos de clave pública, clave privada, block ciphers y stream ciphers. Además, funciones hash y mecanismos de autenticación, como pueden hallarse en la criptografía tradicional.

La comunidad científica, a la actualidad, aún no tiene un criterio determinado para clasificar a un algoritmo criptográfico como ligero. Lo que sí está claro es que las técnicas criptográficas involucradas tienen que usar la mínima cantidad de recursos posibles de los objetos en los que se las aplicará.

4. Líneas de Investigación, Desarrollo e Innovación.

Muchos son los avances a nivel de criptografía que se están realizando, pero no todos los algoritmos livianos son eficientes a la hora de implementarlos en la seguridad de los RFID de bajo costo.

La presunción que es posible evaluar el desempeño general de un algoritmo perteneciente a LICRIPT funcionando sobre una determinada plataforma móvil [6,7,8].

El mismo podrá ser un teléfono celular en particular, una tablet, un dron, lentes como el google glass o equivalentes con capacidad de comunicaciones, o cualquier otro tipo de dispositivo inteligente y portable, para el cual querrá hacer que ejecute un algoritmo determinado. Tal equipo será simulado a través de una virtualización.

De esa manera obtener las métricas que permitan determinar la performance del mismo y emitir un juicio acerca de su comportamiento.

Esta línea de investigación propone evaluar la posibilidad de determinar el funcionamiento performático de un algoritmo criptográfico [9-16] ejecutado en distintos perfiles de HW/SW. Luego, poder evaluar su comportamiento para ser aplicados en dispositivos RFID de bajo costo.

5. Resultados y Objetivos.

El objetivo de este proyecto es realizar un análisis comparativo, de acuerdo a criterios de aplicabilidad y seguridad, de 3 Algoritmos Criptográficos Livianos para dispositivos RFID de bajo costo.

Se realizará un relevamiento exhaustivo de los principales algoritmos criptográ-ficos ligeros existes y determinará cuáles se podrían utilizar para dispositivos RFID de bajo costo.

Se definirán indicadores utilizando otras experiencias internacionales para avaluar comportamientos y permitir comparacio-nes.

Se simulará el funcionamiento de los algoritmos seleccionados y se realizará una tabla comparativa sobre el comportamiento de los algoritmos estudiados.

Finalmente se redactará un informe final y se presentarán en diferentes congresos los resultados obtenidos de esta investigación, para difusión y conoci-miento de la comunidad científica.

Se desarrollará un capítulo específico de “Criptografía Ligera” en la materia electiva Criptografía y “Aplicaciones de Criptografía Ligera” en la materia Auditoria y Seguridad informática de la carrera de Ingeniería en Informática del DIIT.

6. Formación de Recursos Humanos.

La Lic. Mara Capuya se suma al equipo de investigadores como alumna de la Maestría en Informática de la UNLaM. Tanto la Lic. Capuya como el Esp. Pablo Pomar se encuentran desarrollando su trabajo de tesis de posgrado de la Maestría en Informática. Ambos están siendo tutoreados por el Mag. Jorge Eterovic, director del proyecto de investigación y por el Esp. Marcelo Cipriano.

Asimismo, parte del equipo de investigación dictan la asignatura electiva Criptografía en el 5to. Año de la carrera de Ingeniería Informática de la UNLaM, invitarán a sus alumnos a participar de la investigación. Dado que es un proyecto nuevo, aún no se ha logrado la incorporación de ningún alumno.

7. Referencias

[1] Instituto Nacional de Tecnologías de la Comunicaciòn y Agencia Española de Protecciòn de Datos.: Guìa sobre seguridad y privacidad de la tecnologìa RFID. Spain. 2010. www.inteco.es

[2] Román R., Nájera P., López J.; ”Securing the Internet of Things”. University of Malaga, Spain. 2011.

[3] Román R., Nájera P., López J. “Los Desafíos De Seguridad En La Internet De Los Objetos” University of Malaga, España. 2010.

[4] International Association for Cryptologic Research. 2015. http://www.iacr.org/events/

[5] Bhattasali Tapalina. “LICRYPT: Lightweight Cryptography Technique for Securing Smart Objects in Internet of Things Environment”. Research Scholar, University of Calcutta. 2013.

[6] Heer, T.; Garcia-Morchon, O.; Hummen, R.; Keoh, S.L.; Kumar, S.S.; Wehrle, K. “Security challenges in the IP-based internet of things”. Wirel. Pers. Commun. 61, 527–542. 2011.

[7] Garcia-Morchon, O.; Keoh, S.; Kumar, S.; Hummen, R.; Struik, R. “Security Considerations in the IP-based Internet of Things”. IETF Internet Draft draft-garcia-core-security-04; The Internet Engineering Task Force (IETF): Fremont, CA, USA, 2012.

[8] Cirani S., Ferrari G., Veltri L. “Enforcing Security Mechanisms in the IP-Based Internet of Things: An Algorithmic Overview”. Algorithms 2013, 6, 197-226;

[9] Suzaki,T. Minematsu K., Morioka S., Kobayashi E. “TWINE: A Lightweight, Versatile Block Cipher”. NEC Corporation, Japan. 2014.

[10] Bogdanov A., Knudsen L., Leander G. et al. “PRESENT: An Ultra Weight Block Cipher”. Springer-Verlag Berlin Heidelberg 2007 HES 2007, LNCS 4727, pp. 450–466. 2007.

[11] Wentao Z., Zhenzhen B., Dongdai L., Rijmen V., Yang B., Verbauwhede I.; “RECTANGLE: A Bit-slice Ultra-Lightweight Block Cipher Suitable for Multiple Platforms”. China, Bélgica. 2015.

[12] Beaulieu R., Shors D y otros. “The SIMON and SPECK Families of Lightweight Block Ciphers”. Cryptology ePrint Archive: Report 2013/404. 2013.

[13] Mouha N., Mennink B., y otros. “Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers”. Department of Electrical Engineering, Leuven and iMinds, Bélgica. 2013.

[14] Hongjun W., Tao H. “JAMBU Lightweight Authenticated Encryption Mode and AES-JAMBU (v1)” Division of Mathematical Sciences Nanyang Technological University, Singapur. 2014.

[15] Engels D., Fan X., Gong G., Hu, H , Smith M. “Hummingbird: Ultra Lightweight Cryptography for Resource-Constrained Devices.” 1st International Workshop on Lightweight Cryptography for Resource-Constrained Devices (WLC’2010). Tenerife, Canary Islands, Spain, 2010.

[16] ISO/IEC 29192-2:2012. Information technology - Security techniques - Lightweight cryptography - Part 2: Block ciphers. 2012.

Propuesta de una Metodología Innovadora para la Evaluación del Rendimiento de Aplicaciones Criptográficas mediante Pruebas de Carga en un Entorno Virtualizado

Jorge ETEROVIC, Marcelo CIPRIANO y Brian JORDI
Departamento de Ingeniería e Investigaciones Tecnológicas.
Universidad Nacional de La Matanza,
San Justo, B1754JEC, Provincia de Buenos Aires
Argentina.

ABSTRACT

Este trabajo propone una metodología innovadora que mediante la realización de Pruebas de Carga permitan la evaluación de aplicaciones criptográficas. Esto permite la selección de la aplicación más performante entre algoritmos similares.
Esta propuesta permite cubrir la ausencia de una métrica unificada adoptada por la Comunidad Científica Internacional acerca de la medición del desempeño y rendimiento de algoritmos de cifrado. De manera que la decisión de implementar una aplicación de cifrado (que dote de Confidencialidad a un conjunto de datos o canal de comunicación) pueda ser tomada considerando su desempeño y rendimiento sumado a la seguridad y robustez de los mismos.
En particular esta propuesta puede orientarse para algoritmos implementados en Smart Devices. Estos emplean la menor cantidad de recursos posibles pues responden a los principios de la Criptografía Liviana.
Se proponen a los algoritmos Clefia y Present como ejemplos de aplicaciones a evaluar. Pero bien podrían ser otros de similares características.

Keywords: Criptografía Liviana. Clefia. Present. Pruebas de Carga. Evaluación del Rendimiento.

1. INTRODUCCION

Hasta el presente la comunidad científica no ha determinado una única métrica que permita medir y dar un ranking de algoritmos criptográficos en función de su velocidad, desempeño y performance. Los indicadores existentes, como se verá más adelante, no son totalmente determinantes.
En el caso de los algoritmos de cifrado que se ejecutan por software, variables como el SO[3], el lenguaje de programación, su Carga de Trabajo, el Ecosistema Informático[4], frecuencia de uso, entre otras, hacen que el rendimiento del mismo se mantenga incógnita.
Una mala elección de algoritmo podría determinar un bajo rendimiento, que sólo podrá comprobarse luego de largo tiempo en funciones. A esta pérdida de tiempo y recursos se deben sumar los costos de su reemplazo.
En especial interesan los esquemas de Criptografía Liviana o Ligera presentada en el parágrafo 2. En el parágrafo 3 se exponen los conceptos generales de las Pruebas de Rendimiento de Software. En el parágrafo 4 se describe el tipo de Prueba de Carga a realizar y una descripción de la misma. Y finalmente en el parágrafo 5 se describen las generalidades de los algoritmos Clefia y Present, elegidos para comenzar a probar su performance en un ecosistema informático.

2. CRIPTOGRAFÍA LIGERA O LIVIANA

Los tecnólogos avizoran una nueva era en la informática conocida como Computación Ubicua o Inteligencia Ambiental[5]. Un trabajo fundacional de este campo puede encontrarse en [1]. Siguiendo esa idea, aparece el concepto de Internet de las Cosas[6].
Dotar de Confidencialidad y Autenticación a la información tanto en la Computación Ubicua como en la Internet de las Cosas es un desafío importante[7].
La Criptografía Ligera o Liviana[8] es un subcampo de la Criptografía que permite ser utilizada en Objetos Inteligentes y Plataformas Móviles, conocidos como Smart Devices. Estos dispositivos tienden a reducir sus tamaños y costos a medida que evolucionan y ofrecen menor cantidad de recursos como espacio, memoria, poder de cómputo y energía.

[3] Sistema Operativo donde se correrá la aplicación.
[4] contexto de las aplicaciones que se ejecutan junto a una determinada aplicación.
[5] En inglés se conoce con los términos “Pervasive computing”, “Calm technology”, “Things That Think” y “Everywhere”.
[6] IOT: Internet of Things, en inglés.
[7] Las técnicas y sistemas RFID pueden ser empleados tanto en la Computación Ubicua en general como en la Internet de las Cosas en particular.
[8] LICRYPT: Lightweight Cryptography por sus siglas en inglés.

Las técnicas criptográficas involucradas tienen que usar la mínima cantidad de recursos de los objetos donde se aplicarán. La norma [2] ISO/IEC 29192-1:2012 propone algunos indicadores: área del chip medido en GE[9], consumo de energía, cantidad de líneas de código, tamaño de RAM, ancho de banda de la comunicación y tiempo de ejecución.
LICRYPT [3] puede estar orientada al Hardware (HW) o al Software (SW) pues hay tareas que ejecutadas en un contexto puede tener una performance muy diferente en el otro. Sin embargo, algunos criptosistemas pueden ser implementados en ambos contextos indistintamente.
La Criptografía Liviana ofrece todos los servicios que los de la criptografía tradicional: Algoritmos de Clave Pública, Algoritmos de Clave Privada, Algoritmos que trabajan en Bloque (Block Ciphers), Algoritmos que trabajan en Cadena (Stream Ciphers), Funciones Resumen (Hash) y Mecanismos de Autenticación (Firma Digital).

3. PRUEBAS DE RENDIMIENTO DE SOFTWARE

Las Pruebas de Rendimiento de Software[10], también conocidas como Pruebas de Performance, son pruebas que se llevan a cabo para determinar el desempeño de una aplicación. Los aspectos a evaluar, entre otros, son el tiempo de respuesta y la estabilidad cuando se lo somete a una determinada carga de trabajo.
Las pruebas también pueden ser utilizadas para medir, verificar y validar diferentes atributos de calidad, como lo son la escalabilidad, la confiabilidad y el uso de recursos, entre otros.
Estas pruebas se dividen en 6 grupos: Pruebas de Carga, Pruebas de Estrés, Pruebas de Estabilidad, Pruebas de Picos, Pruebas de Configuración y Pruebas de Aislamiento.
Si se somete a dos o más aplicaciones a Pruebas de Rendimiento, se puede decidir cuál de ellas realiza las tareas con mejores prestaciones al poder evaluar su desempeño.

3.1. Pruebas de Carga

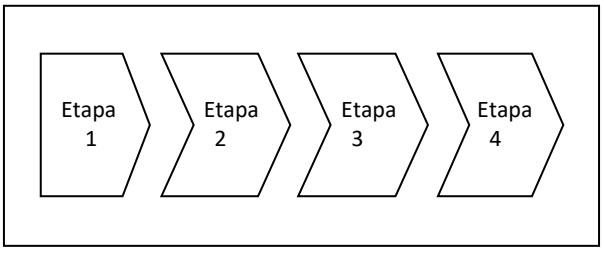
Para evaluar el desempeño de los algoritmos se han elegido Pruebas de Rendimiento[11] de tipo “Test de Carga” y poder determinar al que mejor performance demuestre para un determinado contexto.
Se someterá a los algoritmos al procesamiento de archivos de distinto tamaño para medir los tiempos que tardarán en cifrar y descifrarlos.
Con los resultados obtenidos se generará un informe estadístico acerca del rendimiento de cada uno, para luego realizar una comparación entre los mismos.

3.2. Ecosistema Informático Controlado: Virtualización

Para poder obtener resultados coherentes y confiables se debe tener en cuenta que los algoritmos deben ser ejecutados bajo idénticas condiciones, es decir que deben correrse en los mismos entornos de hardware y software. Esto implica:
Un mismo tipo de procesador e igual cantidad de ellos, idéntico SO, misma cantidad de memoria RAM, compilador y archivos de entrada. Para ello podría usarse siempre la misma computadora donde se realizarán los test. Sin embargo, no se puede tener un control y registros exactos sobre ella.
No siempre los algoritmos se ejecutarán en idénticas condiciones. Por ejemplo, aplicaciones latentes que se activen durante la prueba podrían introducir errores de medición, entre otras posibilidades.
Por ello se propone que los test se ejecuten en “entornos virtualizados” que permiten tener un mayor control de las condiciones del equipo, ecosistema informático y demás variables.

4. MÉTODO PROPUESTO PARA EVALUAR APLICACIONES CRIPTOGRÁFICAS EN ENTORNOS VIRTUALIZADOS

El método para determinar el rendimiento de aplicaciones de Criptosistemas consta de 4 etapas:



[9] Para medir la “complejidad” de un circuito digital se emplea la unidad llamada “Gate Equivalent” (GE): cantidad de compuertas lógicas individuales que se deben interconectar para realizar la misma tarea que realiza el circuito que se está midiendo.
[10] Pruebas de Software (Software Testing en inglés) son una herramienta de la Ingeniería de Software. En particular las Pruebas de Rendimiento son un componente de la Ingeniería de Pruebas.
[11] Pruebas de Rendimiento: conjunto de pruebas que permiten evaluar la velocidad con la que un sistema o aplicación cumple con su tarea, sometido a diferentes condiciones de trabajo.

Figura 1: esquema propuesto para evaluar aplicaciones criptográficas

1. Creación del Entorno de Virtualización.
2. Selección de los criptosistemas candidatos.
3. Ejecución de las Pruebas de Carga.
4. Análisis de los resultados de las pruebas.

4.1 Creación del Entorno de Virtualización

Tal como se expresará en los parágrafos 4.2 a 4.4, se creará en una computadora personal el entorno de virtualización adecuado al Smart Device que se pretende simular.
De los diferentes sistemas que permiten realizar VM se elegirá el que más control del entorno permita, ya que los resultados serán mejores cuanto más fiel sea la virtualización y mayor el control que de la misma se tenga.

4.2 Selección de los Criptosistemas a Evaluar

Dado el perfil y tipo de dispositivo en el que se busca implementar el criptosistema, se puede seleccionar un conjunto de los mismos que mejor se adapten a tal dispositivo.
Esto se logra mediante el estudio de las especificaciones técnicas de los algoritmos. Para los algoritmos orientados al SW, deben programarse respetando las características que los autores indicaron.
Incluso algunos de ellos ofrecen a la comunidad los códigos de sus criptosistemas, por ejemplo, los que se publicaron en el Proyecto e-Stream [12].

4.3 Ejecución de las Pruebas

Como primer paso se diseñará y programará una Plataforma de Pruebas (Testing Framework) que permita realizar las Pruebas de Carga.
La misma deberá ser capaz de: Entregar los "archivos de prueba" a las aplicaciones criptográficas, tomar el tiempo en el que los algoritmos a testear tardan en cifrar/descifrar dicho archivo y almacenar los tiempos de ejecución de los mismos.
En este caso en particular, la Plataforma de Pruebas hará que Clefia y Present, en este caso, logren cifrar y descifrar 4 "archivos de prueba" cuyos tamaños serán de 512 B, 1 KB, 512 KB y 1 MB.

4.4 Análisis de los Resultados de las Pruebas

Se escogieron estos tamaños de archivos debido a las características propias del diseño de los algoritmos, ya que los mismos no fueron creados para cifrar grandes volúmenes de datos.
Los archivos de prueba contendrán secuencias pseudoaleatorias binarias generadas por un Stream Cipher, como por ejemplo Salsa-20, Trivium u otros, a los efectos de evitar que cualquier sesgo en la distribución de los bits pueda afectar las pruebas.
Para minimizar los efectos y perturbaciones que los algoritmos puedan sufrir al ejecutarse, las pruebas deben realizarse un gran número de veces. Cantidad a determinar por el responsable de la selección. El reducido tamaño de los archivos de prueba permite que los test se ejecuten un número elevado de veces: podrían ejecutarse 1000 cifrados y 1000 descifrados.
Además, la Plataforma de Pruebas podría disponer de manera aleatoria tanto el tamaño del archivo a tratar como así también el turno del algoritmo. Así asegurar la mayor aleatoriedad posible en la realización de las pruebas y por consiguiente la minimización de sesgos.
Cabe aclarar que hay algoritmos que antes de estar listos para el cifrado o descifrado, realizan tareas de puesta a punto, como por ejemplo ejecutar procesos para la obtención de las subclaves, whitening u otras actividades.
Este tiempo previo puede o no considerarse en la prueba. Queda a criterio del evaluador la inclusión o no de los mismos en la medición de los tiempos.
Se sugiere que el proceso se ajuste lo más posible al que finalmente se realizará en el dispositivo de destino.

5. ALGORITMOS CLEFIA Y PRESENT

5.1 Algoritmos de Cifrado en Bloque

Se llaman Algoritmos de Clave Privada o Simétricos a aquellos algoritmos que requieren que el emisor de un mensaje o comunicación cifrada, y el receptor, compartan la misma clave. Tal clave debe ser ingresada al algoritmo para que, con ella, se pueda cifrar/descifrar.
A su vez este tipo de algoritmos se clasifican de acuerdo a la forma con la que procesan los bits del mensaje:
Cifrando bit a bit: Cifradores en Cadena o Cifradores en Flujo (Stream Ciphers).
Cifrando un grupo de bits de longitud fija, por vez: Cifradores en Bloque (Block Ciphers).
El tamaño de los bloques es un aspecto muy importante a tener en cuenta. Un valor pequeño puede disminuir la seguridad del algoritmo e incluso hacerlo vulnerable. En la actualidad los tamaños de bloques están acotados entre los 64 y 128 bits.
Los algoritmos Clefia y Present son Cifradores en Bloque públicos y de uso libre. A su vez la Norma ISO/IEC 29192 los reconoce e incluye en su listado de algoritmos criptográficos ligeros.

5.2 Clefia

El algoritmo Clefia[12] fue desarrollado por investigadores de Sony Corporation en el año 2007 [4-6].
El mismo es un algoritmo de Cifrado en Bloques de 128 bits y con la posibilidad de usar claves de 128, 198 y 256 bits de longitud.
En principio fue creado para dotar de Autenticación y Protección de Derechos de Autor en sistemas DRM[13] de la empresa. Sus autores también proponen otras aplicaciones [7] de este algoritmo, como ser: la generación de números pseudoaleatorios, ser parte componente de funciones hash o la generación de claves en cadena o flujo.
De manera esquemática el algoritmo se divide en dos partes diferentes que procesan la clave y la información a cifrar, llamadas Key Scheduling Part y Data Processing Part respectivamente.
Emplea una Estructura o Red de Feistel[14] que divide la información a cifrar en bloques de 128 bits. A su vez cada bloque es dividido en 4 sub-bloques de 32 bits y los hace transitar por 4 caminos o "ramas".
Esto permite el uso de funciones f de menor tamaño respecto de una Estructura de Feistel de 2 "ramas". Al ser estas más pequeñas el efecto de Difusión[15] que se persigue no es tan bueno, por lo que se aumenta el número de vueltas para compensar y mantener alta la seguridad.
Además, Clefia tiene otras propiedades que hacen que pueda ser implementado tanto en HW como en SW.

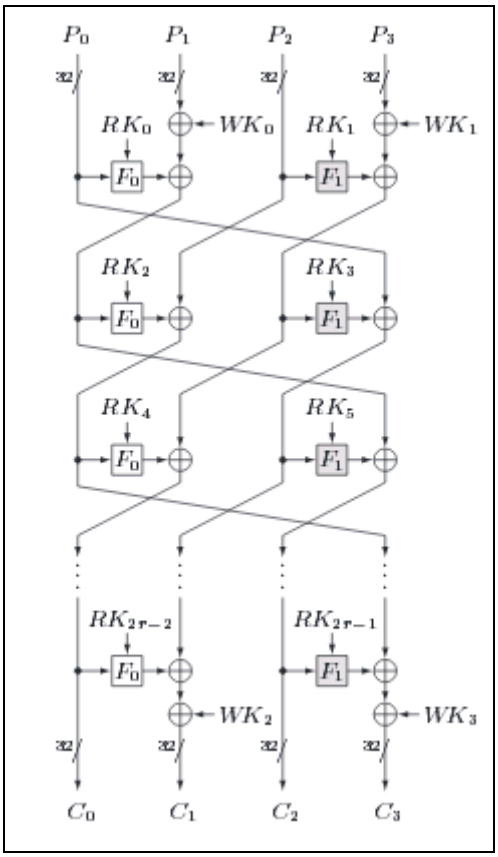


Figura 2: detalle del mecanismo de cifrado del algoritmo Clefia.

5.2 Present

Present [8] fue desarrollado, en colaboración, por científicos de Horst-Görtz-Institute for IT-Security, Ruhr-University Bochum (Alemania), Technical University Denmark, DK-2800 Kgs. Lyngby (Dinamarca) y France Telecom R&D, Issy les

[12] Clef en francés significa "clave".
[11] Gestión de Derechos Digitales (DRM: Digital Rights Management en inglés). Es un conjunto de tecnologías que permitirían proteger los Derechos de Autor.

[14] Procedimiento empleado en el cifrado por bloques creado por Horst Feistel. Su trabajo dio nacimiento al Data Encryption Standard (DES).
15 Difusión: concepto básico de la Teoría de la Información y la Comunicación, el cambio de un bit del texto claro haga que se cambie la mayor cantidad posible de bits en el texto cifrado.

Moulineaux (Francia) en el año 2007 y presentado en el congreso Cryptographic Hardware and Embedded Systems - CHES 2007 [9].
Present es un algoritmo de Cifrado en Bloques de 64 bits, con dos tamaños de claves posibles: 80 bits y 128 bits de longitud, diseñado para ser implementado en Hardware.
El algoritmo se destaca por su tamaño compacto, alrededor de 2,5 veces más pequeño que el algoritmo AES[16] [10-11], ya que fue específicamente diseñado para ser implementado en hardware y así optimizar su desempeño.
Present está emparentado de manera lejana con AES, pues como él emplea una una red SPN (Substitution Permutation Network).
Por cada bloque de texto claro se aplican 31 rondas consistentes en una etapa o capa de Mezclado de Clave, una etapa de Confusión (S-Box[17]) y una etapa de Difusión) (P-Box[18]) tal como propone Claude Shannon[19,] padre de la Teoría Matemática de la Información. Al final se realiza una última ronda que sólo mezcla la clave.
Aunque ha sido diseñado para HW el algoritmo es muy veloz si se lo implementaría en SW.

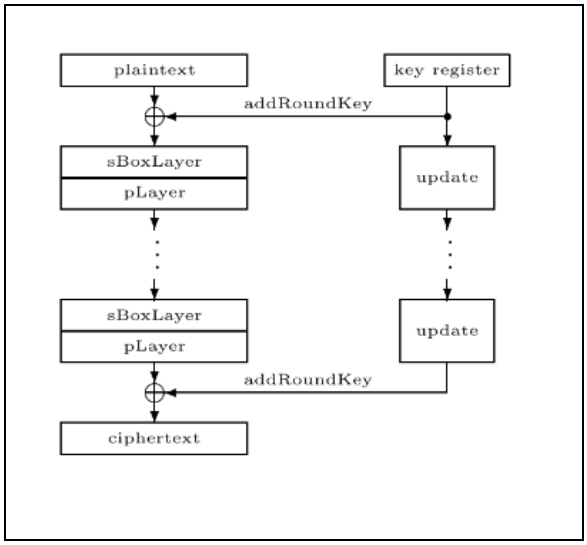


Figura 3: diagrama de bloques del algoritmo Present.

6. CONCLUSIONES Y FUTUROS TRABAJOS

Este trabajo presenta una innovación en la manera de evaluar aplicaciones criptográficas usando pruebas de carga en entornos virtualizados. Hasta ahora la comunidad criptológica aún no adopta, de manera única, una métrica acerca del rendimiento de tales aplicaciones.
Asimismo, se presentó como ejemplo para evaluar a Clefia y Present, algoritmos pertenecientes a la Criptografía Ligera, rama de la Criptografía que estudia segurizar sistemas que dispongan de recursos reducidos, como ser uso y cantidad de memoria, energía, poder de cómputo. Como ejemplo de este tipo de aplicaciones se pueden mencionar los Smart Devices, tarjetas y sistemas RFID, entre otros.
El método requiere del diseño y desarrollo de una Plataforma de Pruebas (Testing Framework) para realizar las Pruebas de Carga tendientes a obtener resultados comparativos que permitan hacer la selección del mejor algoritmo a implementar.
En futuros trabajos se presentará su diseño y los resultados que con ella se obtengan para un determinado dispositivo Smart.
Se someterá a los algoritmos Clefia y Present a las Pruebas de Carga aquí detalladas. Y se describirán los parámetros estadísticos obtenidos. Se espera poder evaluar las pruebas y responder cuál de los dos algoritmos es el más performante.
Asimismo, se podrá desarrollar un método equivalente para evaluar los algoritmos que se orienten al Hardware.

7. AGRADECIMIENTO

[16] AES: Acrónimo de Advanced Encryption Standard (Estándar de Encripción Avanzado). Es un algoritmo de cifrado por bloques adoptado por el NIST (National Institute of Standards and Technology) en 2001como estándar. por medio de los documentos FIPS PUB 197: Advanced Encryption Standard (AES) y ISO/IEC 18033-3: Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers.
[17] S-Box (Substitution Box por sus siglas en inglés) es una operación que permite que la relación entre el texto claro y su cifrado sea lo más compleja de determinar posible.
[18] P-Box (Permutation Box por sus siglas en inglés) es una operación que aplica el concepto de Difusión al cifrado.
[19] Claude Elwood Shannon (1916 – 2001). Criptólogo e Ingeniero Eléctrico estadounidense.

Este trabajo de investigación se desarrolla en el marco del Proyecto de Investigación C2-ING-031 - "Análisis comparativo de Algoritmos Criptográficos Livianos para dispositivos RFID de bajo costo", financiado por la Secretaría de Ciencia y Tecnología de la Universidad Nacional de La Matanza, Argentina.

8. REFERENCIAS

[1] Weiser, Mark 1991. **The computer for the 21st century,** Scientific American, vol. 265, no 3 pp 94-104.

[2]ISO/IEC 29192-1. **Information technology -- Security techniques -- Lightweight cryptography** -- Part 1: General

[3] Eisenbarth, T. et al. L.A 2007 **Survey of Lightweight-Cryptography Implementations.** IEEE Design & Test of Computers, vol. 24, Issue: 6, , pp 522 – 533.

[4] Shirai, T. et. al. 2007. **The 128-Bit Blockcipher CLEFIA** (Extended Abstract). Fast Software Encryption

[5]14th International Workshop, **FSE 2007**, Vol. 4593. ISBN 978-3-540-74617-1. Luxembourg, 2007.

[6]http://www.sony.net/Products/cryptography/clefia/ (consultada el 20/7/2016).

[7]http://www.sony.net/Products/cryptography/clefia/about/appendix_01.html (consultada el 20/7/2016).

[8] Bogdanov, A. et al. 2007. **PRESENT: An Ultra-Lightweight Block Cipher**. Cryptographic Hardware and Embedded Systems - CHES 2007. Springer. Berlin.

[9] http://link.springer.com/book/10.1007/978-3-540-74735-2 (consultada el 21-7-2016).

[10] FIPS PUB 197: **Advanced Encryption Standard** (AES).

[11] ISO/IEC 18033-3: **Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers**.

[12] http://www.ecrypt.eu.org/stream/ (consultada el 21-7-16).

Propuesta de software evaluador del rendimiento de algoritmos criptográficos para dotar de confidencialidad a dispositivos de IoT en un entorno virtualizado

Mag. Jorge Eterovic[1], Esp. Marcelo Cipriano[1,2], Brian Jordi[1]

[1]*Universidad Nacional de La Matanza. Departamento de Ingeniería e Investigaciones Tecnológicas.*
Florencio Varela 1903, (B1754JEC) San Justo, Buenos Aires, Argentina.
Tel: (54 11) 4480-8900.

[2]*Universidad Nacional de Quilmes. Departamento de Ciencia y Tecnología.*
Roque Sáenz Peña 352, (B1876BXD) Bernal, Buenos Aires, Argentina.
Tel: (54 11) 4365-7100/1.

eterovic@unlam.edu.ar; cipriano1.618@gmail.com; brianejordi@yahoo.com

Abstract

Abstract- This developing piece of work presents the details of a modular software design to evaluate the performance of lightweight cryptographic algorithms that belong to "Lightweight Cryptography", which will be able to be used in devices called "Internet of the Things". The evaluation is made through load tests executed in a virtualized environment that presents the hardware and software's limits of the final context where the algorithm is implemented: size of the chip, energy consumption, length of the code and memory consumption among others.

Introducción

El presente trabajo presenta la propuesta para la creación de un software que permita evaluar el desempeño de algoritmos criptográficos. No desde un punto de vista respecto a la seguridad criptológica de los mismos, sino acerca de su rendimiento cuando trabajan en entornos restringidos en Hardware y/o Software como pueden ser los dispositivos que se usan en Internet de las Cosas (IoT), Sistemas en Chip (SoC: Systems on a Chip) usados en módulos de comunicaciones inalámbricas como WiFi o Bluetooth, RFID, entre otros.
Aunque la Criptografía Liviana ofrece métricas e indicadores para medir la performance de algoritmos y su comparación podría orientar acerca de su rendimiento, se carece de datos acerca de la interacción entre ellos y el resto de las aplicaciones que conforman el ecosistema informático en el cual han de funcionar. Esto puede llegar a presentar una importante falla al momento de obtener el producto final por los costos inherentes a tal situación.
El software que se propone responde a una metodología de trabajo la cual mediante pruebas de cifrado y descifrado, posibilita la elección del algoritmo que mejor desempeño presente, ejecutados en entornos virtuales lo más cercanos a los contextos reales donde finalmente trabajarán.
En la primera parte del trabajo se presentan las necesidades particulares de los dispositivos de Internet de las Cosas y la necesidad de proteger los enlaces y la información que estos equipos almacenan, procesan y transmiten.
Seguidamente se presenta a la Criptografía Ligera como campo de investigación, del cual proceden los nuevos algoritmos para emplear en entornos reducidos de Hardware y Software. Luego se presenta el problema de la falta de un algoritmo confiable y adecuado para operar en estos contextos.

Finalmente se presenta la solución propuesta para la búsqueda del algoritmo que mejores prestaciones ofrezca, luego de ser sometido a pruebas de carga de cifrado/descifrado ejecutándose en un entorno virtual.

Marco Teórico

IoT: Internet de las Cosas

Cuando se habla sobre la Internet de las Cosas (Iot: Internet of Things) en realidad de lo que se está hablando es de la conectividad a través de Internet entre dispositivos de muy variada naturaleza y características tecnológicas. Son ya conocidos los electrodomésticos que se conectan a la red. Menos difundidos son otro tipo de dispositivos como las zapatillas "361 Smart Kid Shoe". O más espectaculares aún los marcapasos ANTHEM(tm) RF y ACCENT SR RF. Sin embargo, es posible que estos ejemplos queden prontamente eclipsados con los dispositivos actualmente en desarrollo o aún por venir.

Con Internet de las Cosas, todo lo real tendrá su imagen especular en el mundo virtual: cada persona y los dispositivos que posea tendrán una "ubicación" en Internet. Estas entidades pueden producir y consumir servicios e información, como así también, colaborar entre sí con un objetivo en común.

IoT se sustenta fundamentalmente en 2 tecnologías que han demostrado un importante crecimiento en los últimos años. Ellas permiten alcanzar el nivel de interconectividad requerido:

- Redes Inalámbricas de Sensores.
- Identificación por Radiofrecuencia.

La Unión Internacional de Telecomunicaciones en su "Informe sobre la Internet de las Cosas "califica a la tecnología RFID como un "pivote para la IoT permitiendo la conversión de los objetos cotidianos en inteligentes" [1].

Seguridad en Internet de las Cosas

Dada la cantidad y naturaleza de la información que muchos de estos dispositivos colectan y transmiten, sumado a las limitaciones de hardware propias de estos aparatos; es posible que sean blanco de ataques y posean vulnerabilidades que esperan ser explotadas. Estas amenazas podrían llegar a ser tanto o más perjudiciales que cualquiera de los beneficios que ofrecen su uso. [5,9].

Muchos pacientes de renombre, entre ellas políticos y líderes de estado, llevan marcapasos en sus cuerpos. Uno de los más conocidos fue el Papa Benedicto XVI, el cual lo ha usado, incluso antes de ser elegido [10]. ¿Bajo qué riesgos hubiera estado sometido él o cualquier otra persona, suponiendo que usa un marcapasos como los descriptos anteriormente?

Criptografía Liviana

El advenimiento en años recientes de este nuevo campo de investigación y aplicación, llamado Criptografía Ligera o Liviana persigue el estudio de nuevos métodos criptográficos con el fin que puedan utilizarse en objetos inteligentes, particularmente adecuados a las limitaciones de los dispositivos que se emplean en IoT, pues los algoritmos tradicionales no pueden funcionar adecuadamente en dichos entornos.

Los algoritmos livianos pueden estar optimizados para entornos de Hardware, Software e incluso pueden tener buenos rendimientos en ambos. Se pueden encontrar algoritmos de: Clave Pública, Clave Privada, Block Ciphers, Stream Ciphers, Hash y mecanismos de Autenticación. Tal es el caso de los algoritmos que se encuentran en la norma ISO/IEC 29192:2012 [6].

Además, la norma mencionada propone algunos criterios e indicadores para que un algoritmo sea considerado “liviano” o “ligero”:

- área del chip medido en GE (gate equivalent)
- consumo de energía,
- cantidad de líneas de código,
- tamaño de RAM,
- ancho de banda de la comunicación
- tiempo de ejecución

Dado que cada dispositivo tiene sus propias limitantes y aun conociendo en detalle estos indicadores para cada algoritmo, no es fácil elegir el más adecuado para ser implementado. Entre otras razones se puede mencionar la falta de consideración del funcionamiento en conjunto del algoritmo y su ecosistema. La realización de estas pruebas directamente sobre el sistema físico tiene costo económico y temporal, que podría ser importante, de acuerdo a los limitantes del problema.

Se puede reducir estos costos mediante aplicación de una nueva metodología que se ha está llevando adelante. En trabajos anteriores [3] se ha presentado en forma general una solución. En este artículo se expondrán los detalles y fundamentos de la misma.

Planteamiento del Problema

No existe en la actualidad y probablemente no existirá tampoco a futuro (tal como la historia de la criptografía enseña, aunque es una conjetura) un algoritmo criptográfico que pueda ser adaptable y funcione en cualquier entorno de Hardware/Software. Que mantenga las mismas prestaciones y rendimiento y lo más importante aún, que conserve intactas sus propiedades de seguridad, por siempre.

Los avances en criptoanálisis hacen que gobiernos y empresas estén en la búsqueda de nuevos y mejores algoritmos de cifrado. Esto puede apreciarse observando el aumento en cantidad y frecuencia de los concursos y llamados a presentar nuevos algoritmos. Por ejemplo, NESSIE [2,4] y e-STREAM [10] I y II, (ambos proyectos de investigación europeo), CRYPTREC [7,8,9] (proyecto de investigación japonés), entre otros.

Es por ello que se presenta la dificultad de decidir cuál es el mejor algoritmo de cifrado/descifrado que permita segurizar el intercambio y almacenamiento de la información en dispositivos con recursos limitados.

Solución Propuesta

4.1 Software para Evaluación del Rendimiento de Algoritmos Criptográficos Entornos Virtualizados

La propuesta de solución al problema planteado es llevar adelante pruebas a los algoritmos criptográficos por medio de un software. Este permitirá la evaluación del rendimiento virtualizando el ecosistema donde el algoritmo se ejecutará.

Al cabo de las mismas, se procederá a un análisis estadístico del rendimiento de cada algoritmo y que permita la elección del más adecuado por sus prestaciones.

Para poder llevar adelante la metodología propuesta se deben realizar los siguientes pasos:

1. Creación del Entorno de Virtualización y del lenguaje del Programa Evaluador.
2. Selección de los criptosistemas candidatos.
3. Ejecución de las Pruebas de Carga por medio de un Algoritmo Marco o Framework.
4. Análisis de los resultados de las pruebas.

4.2 Creación del Entorno de Virtualización y del Programa Evaluador

Es de suma relevancia la definición de ciertos conceptos acerca de la tecnología a tener en cuenta para llevar adelante esta etapa. Como son: el software para virtualizar, el perfil del hardware que se desea simular, el sistema operativo que se virtualizará, el lenguaje en el que el algoritmo ejecutará en la VM20 y demás aspectos muy específicos que exceden el alcance de este trabajo.

Existen diferentes productos de software para virtualizar. Algunos de los más conocidos son VMWare Workstation Player y Oracle VirtualBox. Algunas de las características destacables y compartidas por ambos son: gratuidad, rendimiento y la posibilidad de poder ser ejecutados sobre un entorno Windows, MacOS o Linux.

Finalmente se decidió optar por VMWare Workstation Player, y esta elección se debió a que posee un mejor rendimiento, de acuerdo a diferentes pruebas realizadas por entidades relevantes y una interfaz más amigable y de mayor facilidad de uso. Asimismo, es el software del que se dispone en los laboratorios.

Otro aspecto a tener en cuenta y que es decisión de los desarrolladores del sistema final donde la Metodología propuesta finalmente será implementada, es la elección del lenguaje del Programa Evaluador. En particular para continuar con este estudio y proseguir en futuros trabajos avanzando en ello se decidió por realizar el Programa Marco o Framework en el lenguaje C. Este lenguaje es uno de los más populares y difundidos de la actualidad por su ductilidad, expresividad y eficacia para llevar adelante todo tipo de algoritmos. Es capaz de producir programas reducidos en extensión y notablemente veloces en su ejecución. Por consiguiente, se logran mediciones de tiempo más exactas. Permite un manejo de código a más bajo nivel que otros lenguajes. Y simultáneamente una sintaxis sencilla, facilidad en la lectura e interpretación del código por ser un lenguaje estructurado.

4.3 Selección de los Criptosistemas Candidatos.

Esta decisión debe responder a la lectura de las características descriptas en el acápite II anteriormente expuesto, de acuerdo a la norma ISO/IEC 29192-1:2012 u otras. Es importante que se elijan los algoritmos más rendidores teniendo en cuenta las métricas presentadas en la norma y de acuerdo con el software/hardware donde se ejecutarán.

Esta Metodología se presenta para llevar adelante con 2 algoritmos y así lograr la elección del mejor de ellos. En caso de que ambos tengan un rendimiento estadísticamente semejante y ninguno se distinga en comportamiento del otro, cualquiera de ellos podría seleccionarse.

4.4. Descripción del Programa Evaluador

Este programa debe encargarse de hacer que se ejecuten las Pruebas de Carga, tomar el tiempo en que cada algoritmo tarda en hacerlas y registrarlo en una tabla para su posterior análisis estadístico.

Para tal fin se necesita que el usuario tenga previamente:

- Codificación de los algoritmos candidatos en el lenguaje que se determine para la prueba de los mismos.
- Tamaño de las pruebas (en bits) para llevar adelante.
- Cantidad de pruebas que se desean realizar a cada algoritmo.

Se propone que estos últimos 2 ítems sean valores que correspondan a potencias de 2, dado que mediante secuencias binarias equilibradas en 1`s y 0`s se persigue que las pruebas se lleven adelante con muestras de bits con las mejores propiedades de equiprobabilidad y distribución.

[20] VM (Virtual Machine): Software que permite simular el funcionamiento de un equipo real, el cual hasta puede ejecutar programas.

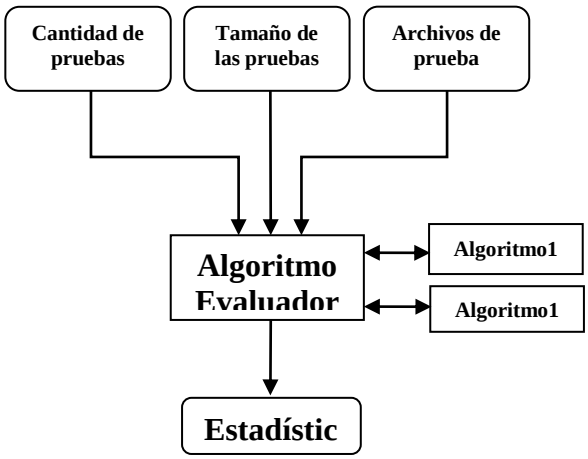


Figura 1: diseño modular del algoritmo evaluador

4.5 Los Archivos de Prueba

Los archivos de prueba servirán para ser cifrados y descifrados la cantidad de veces que se especifique en "Cantidad de Pruebas".

Dichos archivos estarán formados por secuencias seudoaleatorias de bits producidos por LFSRs[21] basados en Polinomios Primitivos[22] en GF(2)[23] y de tamaños variables, acordes a los tamaños de los archivos de prueba posibles.

Se propone el uso de LFSR´s y no NLFSR´s pues estos últimos no garantizan las propiedades de ciclos máximos y la distribución equiprobable de 1´s y 0´s que se persigue.

Si el LFSR tuviera una longitud *L*, entonces la cantidad de bits de la secuencia máxima que produciría sin ciclar sería:

$$|S| = 2^L - 1 \qquad (1)$$

Donde *S* es la secuencia obtenida, |S| la cantidad de bits de la secuencia, L es la longitud del registro.

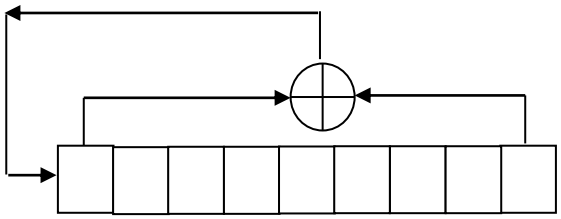


Figura 2: **esquema de un LFRS de 9 estados internos**

Por propiedad de los LFSR basados en polinomios primitivos, la cantidad de 1´s en la secuencia es mayor por un valor, que la cantidad de 0´s. Para compensar y que tengan la misma cantidad, se debe agregar un 0 en la parte de la secuencia con *L-1* ceros consecutivos. La secuencia ahora obtenida recibe el nombre de Secuencia de De Bruijn. Esta modificación corrige la fórmula (1) así:

[21] LFSR: Linear Feedback Shift Register. Registro de Desplazamiento Realimentado Linealmente.

[22] Polinomios Primitivos: aquellos polinomios irreducibles y generadores del campo en el que se está trabajando.

[23] GF: Galois Fields. Estructura algebraica que consta de un conjunto y 2 operaciones, usualmente de suma y producto; tales que satisfacen ciertas propiedades. En particular GF(2) es el conjunto binario con la suma y el producto usuales binarios.

$$|S| = 2^L \qquad (2)$$

Entonces se tiene

$$L = \log_2(T) \qquad (3)$$

Siendo *L* la longitud de los LFSR a usar para construir los archivos de prueba y *T* el tamaño de los archivos de prueba.

4.6 La Ejecución de los algoritmos candidatos

Además de los Archivos de Prueba y los tamaños de dichos archivos, también el Programa Evaluador podrá ejecutar los algoritmos candidatos de acuerdo a un orden que responda a una secuencia seudoaleatoria. Este aspecto del proceso permite reducir al mínimo cualquier factor incontrolable que pueda interferir con la obtención de los tiempos de ejecución.

Nuevamente el uso de LFSR´s permite acercar al máximo la ejecución de los algoritmos lo más cercana al azar. Por ejemplo, haciendo que el "candidato 1" se ejecute, por ejemplo, cuando el bit de salida del LFSR elegido sea 0 y el "candidato 2" cuando el bit sea 1.

Se tiene entonces que:

$$L = 1 + \mathrm{Log}_2 P \qquad (4)$$

Donde L es la longitud del LFSR "selector" y P la cantidad de las pruebas que se desean realizar.

Por ejemplo, si P=1024 entonces la longitud del registro será de 11. De acuerdo a la ecuación 2 (respetando la corrección sugerida anteriormente) se tiene que la longitud de la secuencia será de 2048 bits. La mitad de ellos, serán 1024 ceros y 1024 unos. Dado que al salir 0 se ejecuta un algoritmo y el otro al salir 1, entonces cada uno de ellos se ejecutó la cantidad deseada de veces.

4.7 Toma de Tiempo de la Ejecución de los algoritmos candidatos

A efectos de evaluar el rendimiento de cada algoritmo, el Programa Evaluador toma el tiempo en que cada algoritmo tarda en ejecutar el cifrado/descifrado de los archivos de carga.

Una vez que se han terminado las pruebas, el programa efectuará un estudio estadístico asumiendo que estos experimentos pertenecen a una variable de distribución discreta. Tal estudio busca hallar los tiempos medios de cifrado y descifrado para cada tamaño de archivo de prueba.

$$t_m = \frac{1}{p}\sum_{i=1}^{m} t_i \qquad (5)$$

Donde t_m es el tiempo medio (de cifrado o descifrado), *p* la cantidad de pruebas ejecutadas para cada algoritmo y cada t_i es el tiempo de ejecución de la prueba *i*.

$$s = \sqrt{\frac{1}{p-1}\sum_{i=1}^{m}(t_i - t_m)^2} \qquad (6)$$

Donde el argumento de la raíz recibe el nombre de Varianza de la Distribución de Probabilidad Discreta y *s* el Desvío Estándar, t_m es el tiempo medio, *p* la cantidad de pruebas ejecutadas para cada algoritmo y cada t_i es el tiempo de ejecución de la prueba i.

El programa calcula los valores de las ecuaciones (5) y (6) para cada algoritmo y para cada proceso de cifrado y descifrado. Finalmente, el programa emite los valores de t_m y *s*. El programa seleccionado será aquel con los menores valores t_m y *s*.

4.8 Ejemplo del Funcionamiento

Se tienen 2 algoritmos candidatos para asegurar las comunicaciones en determinado sistema que utilizará tarjetas tipo "MIFare Classic 1K" con transferencia de archivos de 1Kb (1024 bits) de longitud.

Para ello se ha decidido que cada uno de ellos se ejecute 1.048.576 veces, (es decir 2^{20}) para minimizar cualquier factor que pueda interferir con la toma de las mediciones de tiempo. Sólo habrá un tamaño de Archivo de Prueba de 1024 bits.

Tabla1: descripción de las tarjetas RIDF más difundidas.

Tipo de Tarjeta	Frecuencia	Protocolo	Tamaño EEPROM
Estándar UHF	860-960MHz	ISO18000-6C EPC Gen2	EPC 128bits User 32-512bits
Estándar UHF/HF	860-960MHz y 13.56MHz	ISO18000-6C /EPC Gen2 ISO14443A Mifare Classic	UHF (EPC 128bits, User 32-512bits), HF (1K Bytes)
MIFare UltraLight	13.56MHz	ISO14443A	512bits
MIFare Classic 1K	13.56MHz	ISO14443A	1 Kb.

Se alimentará al Programa Evaluador con el valor de la cantidad de pruebas (p=1024), la cantidad de archivos de prueba según la fórmula (1) y la longitud del tamaño de las pruebas (1024 bits).

Con la información cargada, entonces el programa calculará la fórmula (3) y seleccionará el LFSR de tamaño adecuado para generar el contenido del archivo de prueba y lo generará. En este caso elegirá un LFSR de longitud 10 para generar los 1024 bits del archivo. Debe programarse la corrección mencionada anteriormente en la fórmula (2).

Luego el programa usará la fórmula (4) y seleccionará el LFSR acorde a la misma. Con él generará el "selector". En este caso el valor de *L* del mismo será de 21. Por lo que la longitud de dicha secuencia asciende a 2.097.152, la mitad de ella (1.048.576) corresponderá a la ejecución de cada algoritmo.

El programa llamará a cada algoritmo asignándole el archivo de carga para cifrado y luego para descifrado, tomando el tiempo en cada proceso y almacenándolo en una lista. Repetirá el proceso hasta que no queden bits en el selector. Al finalizar el proceso, computará los valores de las fórmulas (5) y (6) para cada algoritmo y para cada proceso. Al cabo de lo cual emitirá un informe con los valores calculados.

Conclusiones

Internet de las Cosas provoca desafíos de seguridad. Las limitaciones de los dispositivos y equipos hacen que no todos los algoritmos de cifrado puedan funcionar adecuadamente. No es suficiente saber que los algoritmos son seguros, sino que además es muy importante conocer el rendimiento que tendrán en un contexto de ejecución con recursos reducidos compartiendo los recursos con otros programas, conformando un ecosistema informático.

Para evaluar tal situación se ha presentado una Metodología de Trabajo que permita elegir al algoritmo más adecuado para un determinado contexto de hardware y software, mediante pruebas de carga. Las mismas se realizarán en un entorno virtual lo más parecido posible al contexto final donde el algoritmo realizará su tarea.

Y en particular se ha descripto al Programa que lleva adelante esta metodología, discutido sus particularidades y propuesto soluciones para los problemas planteados. A su vez se da un ejemplo global con detalles para mayor comprensión.

Trabajo Futuro

Luego de presentar la Metodología propuesta y los detalles de diseño del Programa Marco, resta por llevar adelante su codificación, pruebas de implementación y finalmente someter a un par de algoritmos a prueba.

Se espera que pronto se tenga disponible esta herramienta para iniciar la depuración de errores y comenzar a realizar las pruebas de comparación del rendimiento entre algoritmos.

Agradecimientos

Este trabajo de investigación se desarrolla en el marco del Proyecto de Investigación *"Análisis comparativo de Algoritmos Criptográficos Livianos para dispositivos RFID de bajo costo*" del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza.

La Universidad Nacional de Quilmes, promoviendo la colaboración y cooperación científico-tecnológica, brinda su apoyo a este proyecto y su equipo de investigación.

Referencias

[1] Biggs, P. (ITU); Garrity, J. (Cisco); LaSalle, C. (Cisco); Polomska, A. (ITU). International Telecommunication Union. Place des Nations CH-1211 Geneva 20. Switzerland.ISBN 9789261164010. Disponible en https://www.itu.int/en/ac tion/broadband/Documents/Harnessing-IoT-Global-Develop ment.pdf (consultada el 1-6-17).

[2] Biryukov, A.; De Canniere, C.; et all. "Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity, and Encryption". Springer-Verlag Berlín. 2004.

[3] Eterovic, J.; Cipriano, M.: Jordi, B.; "Propuesta de una Metodología Innovadora para la Evaluación del Rendimiento de Aplicaciones Criptográficas mediante Pruebas de Carga en un Entorno Virtualizado*".* Memorias de la Décima Sexta Con-ferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI 2017). IIIS. 8 al 11 de Julio, Orlando, Florida, Estados Unidos. 2017. Pág. 310-314.

[4] https://competitions.cr.yp.to/nessie.html (consultada el 12-6-2017).

[5] Instituto Nacional de Tecnologías de la Comunicación y Agencia Española de Protección de Datos.: Guía sobre seguridad y privacidad de la tecnología RFID. Spain. 2010. www.inteco.es (consultada el 1-6-17).

[6] ISO/IEC 29192-1. Information technology -- Security techniques -- Lightweight cryptography -- Part 1: General.

[7] http://www.cryptrec.go.jp/english/images/cryptrec_01en.pdf (consultada el 12/6/2017).

[8] http://www.cryptrec.go.jp/english/cryptrec_03_spec_cypher list_ files /PDF/1002espec.pdf. (consultada el 12/6/2017).

[9] http://www.cryptrec.go.jp/english/method.html (consultada el 12/6/2017).

[10] http://www.ecrypt.eu.org/stream/portfolio_revision1.pdf. (consultada el 12/6/2017).

[11] http://www.elmundo.es/elmundo/2013/02/12/internacional /1360673510. html

[12] Román R., Nájera P., López J. "Securing the Internet of Things". University of Malaga, Spain. 2011.