FACULTAD DE INFORMATICA

UNIVERSIDAD NACIONAL DE LA PLATA

Trabajo Final Integrador

Alumno
Marcelo José Cipriano

Carrera
Posgrado en Redes y Seguridad

Directora
Paula Venosa

Criptografía Liviana e Internet de las Cosas: Confidencialidad de la Información mediante Stream Ciphers estandarizados en las normas ISO/IEC 18033 y 29192

Marzo - 2021

INDICE

PREFACIO

Este trabajo es el fruto de la investigación de temas que me resultan de sumo interés. Por un lado la llamada *Internet de las Cosas* (IoT por sus siglas en inglés) y su amplio espectro de aplicaciones: casas, ciudades inteligentes, *Internet de las* Cosas *Industriales* - también llamada *Industria 4.0*-, agricultura inteligente, dispositivos "vestibles" o "wearables" y los dispositivos IoT aplicados a la Salud o e-Health.

Por otro lado, la *Criptología*. Es decir el campo del conocimiento convergente de la Criptografía y el Criptoanálisis en particular. Como así también la Teoría de Números, Campos de Galois, Complejidad Computacional y Algorítmica entre otros, en general.

El encuentro de ambas áreas temáticas presenta un enorme e interesante desafío: alcanzar el mayor nivel de confidencialidad, integridad y autenticación posible, en el contexto limitado en recursos de los dispositivos IoT.

Sin embargo, el interés no se reduce a lo académico. Sino que resulta de suma importancia para el usuario en general, empresas y fábricas, organismos gubernamentales y países. No hay opción: las consecuencias de no hacerlo, serían inimaginablemente desastrosas. Algunas ya se han comenzado a experimentar, en años recientes, tal como se verá más adelante.

Este documento se ha basado fundamentalmente en los contenidos, publicaciones y conclusiones que se llevaron adelante en los proyectos de investigación de la Facultad de Ingeniería de la Universidad del Salvador VRID 1735, 1737, 1739 a lo largo de los años 2017 al 2020. En los dos primeros, me desempeñé como su Director y en todos ellos, he podido compartir la enriquecedora experiencia de indagar, explorar y aprender, junto al equipo de investigadores que los conformaron.

La estructura que organiza el material y contenidos, consiste en 2 partes de acuerdo las áreas temáticas presentadas con anterioridad.

En la *Parte I*, se describe el nacimiento y desarrollo de IoT. Se indaga sobre las posibles causas de la notable ausencia de seguridad en sus inicios y cómo aún en la actualidad el problema persiste, También se expone cómo dispositivos de variada naturaleza, diseños y fabricantes, presentan vulnerabilidades que son explotadas por personas inescrupulosas.

En la *Parte II,* se exponen las características de la llamada *Criptografía Liviana.* También se presenta una descripción técnica de cada uno de los algoritmos de cifrado en cadena estandarizados bajo las normas *ISO/IEC 18033* y *ISO/IEC 29192.* Sus propiedades criptográficas, usos, aplicaciones, sus autores y en el contexto en el que fueron concebidos.

Es mi deseo que este material, además de permitir lograr su finalidad primordial de satisfacer un requerimiento académico para alcanzar el grado de Especialista en Redes de Datos de la Facultad de Informática, de nuestra querida Universidad Nacional de La Plata, sea capaz de encender la inquietud y curiosidad al que recién se inicia en estos temas y pueda, además, colaborar en la profundización de los conocimientos, para el lector más avanzado.

Marcelo Cipriano
Ciudad Autónoma de Buenos Aires, 16 de Marzo de 2021.

RESUMEN

El presente Trabajo Final Integrador tiene por finalidad indagar acerca del origen y naturaleza de la llamada Internet de las Cosas o IoT por sus siglas en inglés; abordar la necesidad de contar con mecanismos de seguridad en dichos dispositivos, considerando los peligros que conlleva no disponer de ellos. Y finalmente presentar algunos de los algoritmos criprográficos pertenecientes a la llamada Criptografia Liviana o Ligera (Lightweight Cryptography) que pueden ser ejecutados en tales dispositivos.

Los algoritmos de cifrado de información que aquí se abordarán serán aquellos Stream Ciphers descriptos en las normas internacionales ISO/IEC 18033 Y 29192 de Criptografía Liviana. Se indagará acerca de su origen, autores, año de creación, su proceso de estandarización. Por último se expondrá una breve reseña acerca de su funcionamiento y contexto de uso.

Palabras Claves:
IoT, DoS, DDoS, RoT, Lightweight Cryptography, Stream Ciphers, ISO/IEC 18033, ISO/IEC 29192.

1. Introducción

En forma simple, *Internet de las Cosas*, también conocida por sus siglas en inglés *IoT*, es el fruto de la convergencia de la microelectrónica, informática, los sistemas de comunicaciones y las redes de datos. Gracias a ello se han recreado dispositivos de uso extendido por la sociedad, como así también nuevos equipos, los que aprovechando sus avances, ofrecen a los usuarios prestaciones ampliadas sin precedentes.

Hace un tiempo ya se pueden adquirir televisores inteligentes o *Smart-TV* que pueden conectarse y navegar por Internet, recibir comandos por voz y algunos hasta de movimiento, entre otras prestaciones. También son conocidos los lavarropas que pueden comunicarse inalámbricamente con el usuario para recibir comandos, informar el estado de avance del lavado, los parámetros de funcionamiento y el diagnóstico del dispositivo. O tal vez cafeteras que permiten configurar el café a gusto del usuario, heladeras con pantallas que pueden registrar la lista de compras y realizar el pedido a la tienda o supermercado online. Tostadoras que se conectan a Internet mediante el *WiFi* hogareño para configurar el "dibujo" que se desea "imprimir" en el pan: el estado del tiempo, la temperatura, emoticones, caracteres y hasta pequeños mensajes. También cámaras web de vigilancia que transmiten por Internet, luminarias que pueden cambiar el tono, color y brillo de su luz, incluso programar el encendido y apagado. Existen además dispositivos *IoMT* llamados *Internet de las Cosas Médicas* (*Internet of Medical Things*) tal vez no tan conocidos para todos, pero no por ello menos importantes. Por ejemplo marcapasos inteligentes que pueden ser controlados inalámbricamente o bombas de insulina que analizan, dosifican y liberan la medicación adecuada en el torrente sanguíneo del paciente, de acuerdo a un control en tiempo real.

Tales aplicaciones verdaderamente están avanzando en su promesa de revolucionar la sociedad…y esto aún está comenzando. Pero por otro lado la mayoría de los fabricantes y diseñadores de estos dispositivos no incluyen mecanismos de seguridad robustos en los mismos. Eso significa que la información que estos recaban del usuario o su entorno, procesan, almacenan y transmiten, se encuentra expuesta a diferentes riesgos. Se pueden comprobar la existencia de un sinnúmero de ataques que afectan la confidencialidad, integridad, disponibilidad y autenticidad en el universo IoT.

¿Cómo es posible que siendo el alba de un negocio multimillonario, la mayoría de los dispositivos exponen así la información de sus usuarios? ¿No existe nada que se pueda hacer al respecto? ¿Será cuestión de aceptar que así serán las cosas, hasta que surja alguna luz en la oscura problemática?

Según un reporte publicado a fines de Noviembre del presente año en el portal *IoT Analytics,* para el año 2019 los dispositivos IoT ya igualaban en cantidad a todos lo demás dispositivos juntos (tablets, teléfonos celulares, notebooks, netbooks, laptops, desktops, etc.). Y se proyecta un crecimiento casi exponencial para los próximos 5 años de estos aparatos. Y como se ha expuesto, la gran mayoría de ellos, carente casi completamente

de mecanismos de seguridad robustos. Este crecimiento en cantidad y sin recaudos de seguridad, está provocando que tecnologías *IoT* se vayan tornando en un peligro, no sólo para su propietario, sino para todo el mundo: ofrece miles de millones de dispositivo conectados a la red, que procesan y transmiten información, al alcance de cualquier personan o grupos inescrupulosos con habilidades mínimas para la intrusión y el hacking.

Es necesaria la difusión y sensibilización acerca de los peligros y amenazas de no contar con mecanismos robustos de seguridad para ser implementados en *IoT,* dada extensa y variada superficie de ataque que estos dispositivos presentan. Sólo por mencionar algunos ejemplos, se pueden encontrar: ataques a la privacidad de niños mediante vulnerabilidades de los llamados *IoT Toys*, el secuestro de *Smart Cars* mediante *Jackware* o dispositivos *IoT* Industriales mediante *Ransomware of Thing*. *Ataques de Denegación de Servicios Distribuidos*, llevados adelante por redes *zombies* conformadas por cientos de miles de dispositivos *IoT* contra los servidores *DNS*. Este ataque impidió que varios millones de personas en todo el mundo puedan acceder a recursos y servidores en Estados Unidos, en septiembre de 2016.

Ante este panorama, la comunidad científica ha podido ofrecer algunas soluciones, basadas en el uso de algoritmos de cifrado pertenecientes a la llamada *Criptrografía Liviana o Ligera,* estandarizados bajo normas *ISO/IEC* internacionales y que se encuentran disponibles para todo el mundo.

PARTE 1

Internet de las Cosas y sus Riesgos de Seguridad

2. Internet de las Cosas: Surgimiento y Características

2.1 Pre-historia de IoT.

A mediados de la década del 70 del siglo pasado, un grupo de personas que se desempeñaban en el *Departamento de las Ciencias de la Computación* de la *Universidad Carnegie Mellon* (*Pensilvania, Estados Unidos*) tenían un pequeño problema con su máquina expendedora de gaseosas [1]. Y decidieron hacer algo al respecto: era el momento de hacerle un par de mejoras. Tantas veces se habían tomado el trabajo de ir hasta ella y descubrir, decepcionados, que ya no quedaban botellitas[1] para vender. O lo que era peor aún, que no estaban tan frías como las querían.

Figura 1: Escudo de la Universidad Carnegie Mellon (EE.UU).

Esas botellitas eran muy demandadas pues costaban algunos centavos menos que sus hermanas en otras máquinas del campus. Se consumían a tal velocidad que cuando llegaban, no quedaba ninguna. Y si por suerte aún quedaba alguna, tal vez no estaban frías porque las habían repuesto hacía tan poco tiempo.

Fue entonces que decidieron aprovechar la sinergia electrónica-informática. Instalarían micro-interruptores que les permitirían detectar y contar la cantidad de botellitas que había en la máquina. También se las ingeniaron para conocer su temperatura, al menos conceptualmente hablando. Ya que aquellas botellitas que estuvieran más tiempo en la máquina, estarían más frías. Lograron un avance, pero ¿para qué les servía dicha información? Igualmente tendrían que ir hasta la máquina. Y en el peor de los casos, volver con las manos vacías.

Entonces decidieron no ir hasta la información, sino que la información vaya hasta ellos. Conectaron la máquina a la red *Ethernet.* Y de allí hasta el servidor local del Departamento. Asunto resuelto. Sólo debían conectarse al servidor para que este les entregue, programa mediante, la información actualizada de la cantidad de botellitas y el tiempo transcurrido de la última reposición. Por consiguiente, si ya habían o no alcanzado la temperatura adecuada.

Es este evento donde algunos autores a través de libros, artículos de periódicos y demás, consideran el nacimiento de la llamada *Internet de las Cosas*. Sin embargo y estrictamente hablando, no sería correcto. Ocurre que la máquina así intervenida, no es un dispositivo verdaderamente *IoT*.

Son 2 razones, a falta de una, para justificar tal afirmación. La primera es que solamente el personal que tenía acceso al servidor podía obtener a la información de la máquina. Y

[1] Coca-Cola en 1915 creó un concurso para obtener un diseño único. Así nació la botella “Contour”, tan representativa que aún se sigue usando, junto a las botellas PET y las latas.

la segunda y tal vez de mayor peso: la red a la que se hace referencia la historia era la red local y no Internet, porque aún no existía.

En 1982 se decidió conectar la máquina a la red *Arpanet*. Cabe recordar que en aquellos tempranos tiempos, sólo 300 computadoras alrededor del mundo, se conectaban a ella. El logro de este avance es enorme, ya que la conectividad de la máquina aumentó vertiginosamente. Pero aunque mayor que el anterior, no tiene ni por asomo la masividad y el alcance que tiene la "Internet" de hoy.

Los años fueron pasando y el formato de las botellitas cambió. La vieja expendedora cayó en desuso y fue sustituida por otra que soportaba el nuevo diseño. También se retiró el viejo servidor de la red. Por un tiempo no hubo novedades.

Foto 1: maquinitas pioneras de IoT
(principios de los años 90´s del siglo pasado).

Pero aquella sinergia electrónico-informática que diera aquellos frutos no fue olvidada. El siguiente paso se dio 10 años después.

Fue en 1992 que se repitió la historia pero con tecnología mejorada. Se emplearon sensores ópticos y conectaron la interfaz de la máquina a una *PC* modelo *IBM-XT*. La que además llevaba el nombre en la red de ***COKE.LABS.CS.CMU.EDU.***

Se recuerda aún su vieja dirección IP: ***128.2.209.43.***[2]

Todo evoluciona y el área informática no es la excepción, fue en 2007 que la máquina tuvo su propia página web: https://www.cs.cmu.edu/~coke/.

Esta página aún sigue online. Y aunque ya no ofrece información sobre las botellitas, es un recordatorio de la gente que participó a lo largo de todos estos años del proyecto. Contiene una breve reseña histórica entre otros aspectos por demás interesantes. Fuente parcial de la información contenida en este trabajo.

La idea se propagó hacia otras universidades, las que también intervinieron sus propias máquinas, dotándolas de la inteligencia suficiente para que se comuniquen con los usuarios interesados, enviándoles así la información. Y no solamente a través de la red interna y de acceso local, sino que en la mismísima Internet. En gran medida, esa es la idea detrás de los dispositivos IoT en la actualidad, próximos a cumplir los 40 años desde aquel verano del ´82.

[2] A la fecha dela realización de esta trabajo de tesis, esta IP pública sigue perteneciendo a la Carnegie Mellon University.

Curiosamente tampoco esta familia de máquinas intervenidas, alojadas en varias universidades, son consideradas los primeros casos de dispositivos IoT.

2.2 Finalmente nace IoT.

Fue durante 1989 que *John Romkey*[3] es desafiado a conectar una tostadora a Internet [2-3]. *Romkey* es un reconocido investigador y desarrollador cuyas contribuciones lo posicionan como uno de los padres de la Internet. Junto a *Donald W. Gillies* desarrolló en 1983 el *MIT PC/IP*, la primera pila *TCP/IP* en la industria para *MS-DOS* en la *PC* de *IBM* en 1983, mientras trabajaba en el *Instituto de Tecnología de Massachusetts*[4].

En sus propias palabras:

> *"…allá en 1990 sólo 3 millones de personas tenían acceso a Internet y 313.000 computadoras (no dispositivos) estaban conectados a ella. Para realizar búsquedas utilizábamos Wide Area Information Network (WAIS), Gopher, y ARCHIE. Para tranferir o compartir archivos, usábamos File Transfer Protocol (FTP). Aunque la World Wide Web había sido inventada en 1989, aún no era común su uso*." (John Romkey, 2016).

Foto 2: Simon Hacket mostrando la tostadora, centro de atención en la Interop de 1990.

El mismo año que *ARPANET* dejó de existir, cediendo su lugar a una incipiente Internet, *John* junto a su amigo *Simon Hackett* lograron conectar la *Sunbeam Deluxe Automatic Radiant Control Toaster* a Internet. Tras una serie de pruebas, la presentan en la convención y feria anual *Intertop 1990*. Aunque la operatoria relacionada con el pan tenía que ser realizada por un ser humano, el encendido y el nivel de tostado se llevaba adelante gracias a Internet. Para tal fin se conectó el aparato a la red *TCP/IP*. Y su control se llevó adelante por medio de *Simple Networking Management Protocol-Management Information Base (SNMP-MIB)* [5].

Al año siguiente, en *Intertop 1991* los fanáticos pudieron ver como una pequeña grúa, también conectada a Internet, permitió depositar el pan en el artefacto y automatizar así todo el proceso.

Resulta sorprendente que aún con un recorrido de 2 décadas y haber sido creadas varias máquinas con conexiones a la red, el concepto siguió sin tener un nombre. Fue a finales de la década de los '90 que finalmente se bautizó esta conjunción de tecnologías y dispositivos. Es decir que contrariamente a lo que el gran público puede suponer*, Internet de las Cosas* es completamente un invento del siglo XX.

[3] En su propio website personal John se autodefine "inversionista privado y vagabundo jubilado".

2.3 La primera cámara IoT

Es curioso descubrir que la búsqueda de una solución a un problema extremadamente similar tuvo la gente del *Carnegie Mellon*, condujo al personal de la *Universidad de Cambridge* a la creación de la primera cámara *IoT*. Del otro lado del océano no fue una gaseosa fría la motivación, sino un café caliente.

Como en muchos lugares de trabajo, la cafetera es uno de los artefactos que favorecen la ejecución de varios procesos. Y siendo el café un recurso limitado, hay un acuerdo implícito entre todos sus consumidores: cuando el recurso deja de estar disponible, el último que accedió a él debe procurar nuevamente su disponibilidad. Dicho en otras palabras, el último prepara más café. Lamentablemente y como en todos lados, había personas que no cumplían su parte del trato.

El lugar donde estaba la cafetera era el pasillo, afuera de la sala llamada *"Trojan Room*" [6]. Para aquellos investigadores y alumnos que se encontraban a varias escaleras de distancia, era una gran decepción comprobar que no quedaba café disponible. Al parecer el café sólo era bebible en caso de estar recientemente preparado. En cualquier otro estado, era una poción bastante difícil de digerir. Por lo que capturar un café relativamente recién hecho era como perseguir el grial [7].

Fue así que en 1991 *Quentin Stafford-Fraser* y *Paul Jardetzky*, que trabajaban dentro de la "*Trojan Room"* lograron resolver 2 problemas con una simple solución. Conectando una cámara frente a la cafetera y transmitiendo la imagen por la red, lograron que nadie tuviera que llegar hasta la cafetera para enterarse que ya no tenía más café y además descubrir a los compañeros holgazanes y poco solidarios.

Lo que hicieron fue montar una cámara (tenían una en el proyecto en el que se encontraban trabajando) frente a la cafetera en el pasillo. Para ello tendieron los cables y los conectaron a una computadora. En aquel tiempo no era tan sencillo hacer tales conexiones, pero lo lograron. El trabajo les llevó toda una tarde. Una vez que tuvieron el hardware listo, pasaron al software. *Quentin* se encargó del programa cliente y *Paul* se encargaría del programa servidor. Llamaron a su sistema *XCoffee*. Eso porque usaban el protocolo *X Window* de *UNIX.* Este protocolo permite construir Interfaces Gráficas de Usuario mediante un modelo cliente-servidor.

Foto 3: Imagen de la ventana del servicio XCoffee.

Finalmente, integraron todo y lo encendieron. Funcionó perfectamente. Aunque la cámara era blanco-negro y las imágenes que transmitía era en escala de grises, no trajo problemas porque no se precisaba una paleta cromática para ver el estado de la cafetera. Además, aunque la cámara capturaba 3 imágenes por minuto, tampoco se precisaba de mayores frecuencias de refresco: la cafetera se tomaba su tiempo para hacer su trabajo y la idea era ver el progreso del mismo.

Ahora cada computador conectado a la red del Departamento de Informática de la *Universidad de Cambridge* podía tener en pantalla una pequeña ventana cuadrada de 129 pixeles de lado, con la imagen de la cafetera en blanco y negro actualizada cada 20 segundos. La idea fue un éxito entre los usuarios. Ya no hubo viajes en vano ni tampoco usuarios holgazanes.

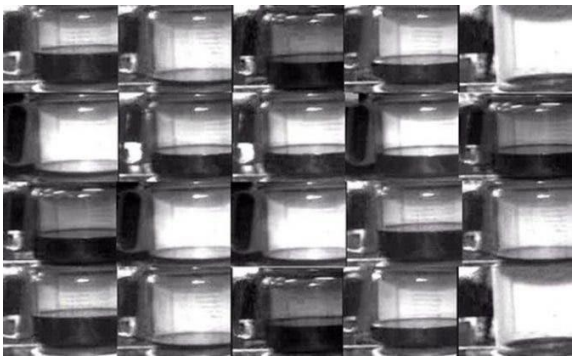


Foto 4: imágenes de la cafetera en webcam.

A fines de enero de 1992 fue nada menos que *Bob Metcalfe*[4] quien escribió sobre el *XCoffee* y dio a conocer la cafetera al mundo, en ocasión de visitar el laboratorio.

Pero la cámara aún no es un dispositivo netamente *IoT*. Al igual que la primera versión de la máquina de gaseosas, la imagen sólo estaba disponible para los clientes de la red interna de la Universidad. Pero en apenas 2 años lograron que la cámara pudiera transmitir imágenes por Internet.

Gracias al advenimiento de la web y usando uno de los primeros navegadores de la época, el *Mosaic* que *Daniel Gordon* y *Martyn Johnson*, extendieron la tecnología *LAN* a Internet. Montaron las imágenes dentro de los protocolos *HTTP* creando así la primera *WebCam* de la historia y así lograron crear un nuevo dispositivo *IoT*. En el proceso convirtieron a la cafetera en la más famosa del mundo. Se sabe que cientos de miles de personas tal vez millones, consultaron las imágenes capturadas por la webcam pionera.

Foto 5: última imagen transmitida por la webcam: su desconexión.

Fue el 22 de Agosto de 2001 que la cámara web transmitió su última imagen. Se puede ver cómo *Daniel Gordon*, *Martyn Johnson* y *Quentin Stafford-Fraser* presionan el botón de off del equipo capturador de imágenes. Así se apagó el dispositivo, dando inicio a su historia.

Ese año el Departamento de Informática se mudaría a nuevas locaciones dentro de la universidad y el antiguo aparataje que le daba soporte a la cafetera, era difícil de mantener. Fue así que para conseguir recursos para el nuevo sistema de café, se decidiera jubilar al equipo y que la cafetera fuera subastada en *e-Bay*. La misma fue vendida por *£ 3350*, alrededor de *u$s 4500*. "*La vendimos por el suficiente dinero como para comprar café para el*

[4] Robet Melcalfe. Ingeniero electrónico estadounidense, co-creador de Ethernet, WiFi, fundador de 3COM y creador de la Ley de Metcalfe.

laboratorio de informática durante bastante tiempo", comentó *Stafford-Frase*. Fue adquirida por la revista alemana *Der Spiegel*. Desde mediados de 2015 la misma fue otorgada en préstamo permanente al *Museo Alemán de la Tecnología*, en *Berlín*.

2.4 Finalmente, el nombre.

Foto 6: Kevin Ashton, creador del Término "Internet de las Cosas".

A casi 30 años del nacimiento y evolución de un concepto innovador y que dio a luz dispositivos tan disímiles y simultáneamente emparentados entre sí, aún no tenía nombre. Nacido y criado en el seno de los claustros más tecnológicos del mundo, de manos de los hacedores de la Internet. Fue visto nacer y crecer por miles de personas, y siendo además presentado y expuesto en conferencias de enorme alcance y convocatoria, aún nadie sabía qué era ni cómo nombrarlo. Parece que tampoco había interés en hacerlo.

Tal vez porque aún no se le veía una aplicación empresarial, un uso masivo. Tal vez fuera visto y entendido como "cosa de nerds" de la informática y la electrónica. Tal vez porque no había una industria detrás y era considerado un juego, lo cierto es que la sinergia cosas-Internet no fue reconocida masivamente. Al menos no que se conozca.

A finales del siglo XX fue por fín la "cosa" bautizada. Sin embargo no fue en un ambiente académico de investigación. En esta oportunidad ni gaseosas, confites, tostadas o café estuvieron involucrados ni tampoco universidades o convenciones de fanáticos de la tecnología. Fue en el corazón de una empresa y a causa de un lápiz labial. *Gran Bretaña* se lleva el mérito de ser el lugar donde el término fue acuñado por un informático empleado en *Procter & Gamble*, en *Surrey*.

Una simple pregunta sobre un cosmético inquietaba a *Kevin Ashton*, quería conocer las razones por las que cierto tono de lápiz labial parecía siempre agotado en una de las tiendas de ventas. Entonces se le ocurrió que agregando *chips RFID*[5] podría realizarse un seguimiento de los productos. Y junto con otros sensores, seguir el derrotero de los mismos en la cadena de suministros. Si los productos ya estaban en un depósito, expuestos en el local o fueron vendidos. Durante 6 meses sus ideas se encontraron con diferentes dificultades en P&G.

[5] RFID (Radio-frequency identification): Identificación por frecuencias de radio. Utiliza campos electromagnéticos generados por automáticamente por etiquetas y demás dispositivos, los que al pasar por un sensor, se induce dicho campo y de acuerdo a la frecuencia del mismo, el objeto que lo posee es "identificado". Su uso se ha hecho extendido también a otros formatos: se puede usar dentro de tarjetas plásticas, como ser tarjetas de crédito, tarjetas de identificación o tarjetas de pago de transportes, como por ejemplo, la tarjeta SUBE o similares.

Estaba escribiendo una presentación en 1999 para la empresa. Sabía que tenía que poner un nombre a su idea. Algo que llame la atención y tal vez por ello, provoque el interés de sus jefes, logrando que acepten sus ideas y autoricen las pruebas. Fue así que Kevin escribió y habló por primera vez de "*Internet de las Cosas*".

En sus propias palabras:

> *"Estaba hablando de que la cadena de suministro es una 'red de cosas' y que Internet es una 'red de bits', y de cómo la tecnología de sensores fusionaría los dos. Entonces pensé en un "Internet de las Cosas" y pensé: "Eso servirá, o tal vez incluso mejor". Tenía un timbre. Se convirtió en el título de la presentación.(Ashont, "Cómo volar un Caballo").*

Así motivado presenta su propuesta ante los ejecutivos de *Gillette*, que estaban interesados en asociarse con *P&G* en el tema de los sensores. Lo dijo en público… y no pasó nada. Aunque aceptaron sus ideas, el término quedó ahí confinado.

Sin embargo la gente de *Gillette* lo contactó con un alto ejecutivo de la compañía en Boston. Le propusieron financiar sus investigaciones nada menos que al mítico *MIT*, donde 17 años antes *Ronkie* y *Gillies* crearon parte del *TCP/IP* (tal como se ha descrito en parágrafos precedentes). De alguna manera, la idea se las había ingeniado para volver a su lugar de nacimiento.

Una vez instalado allí, fue co-fundador del *Centro de Investigación Automática: Auto-Id Center*, por sus siglas en inglés. El cual aún se encuentra en funciones. Sorprende saber que aún el gran público tendrá que esperar otra década hasta enterarse de la existencia de *IoT*. Es que todavía queda el concepto relegado a los ámbitos académicos y tecnológicos en el corazón de universidades y empresas.

Recién en un artículo periodístico a fines de la primera década del siglo XXI que el propio *Kevin* da a conocer masivamente el concepto "Internet de las Cosas". En un artículo del año 2009 titulado *That ´Internet of Thing´ Thing"* haciendo un juego de palabras gracioso (algo así como "*Esa cosa ´Internet de las Cosas*¨), publicado en *RFID Journal* del mes de Junio de ese año[8]. Allí Kevin utiliza este nombre por primera vez y el impacto que tuvo en los lectores fue tal que a partir de allí, quedó acuñado. Internet de las Cosas ya tenía existencia propia.

2.5 El problema de la definición

Al principio los dispositivos IoT podrían ser vistos como curiosidades intrascendentes, incluso graciosas o simpáticas: máquinas que mostraban o informaban sobre el contenido de sus productos a efectos de no ir hacia ellas en vano, el seguimiento de los productos en una cadena de suministros, etc. Sorprende que junto con largo tiempo que se tardó en darle un nombre a este concepto, también es llamativa la falta de reconocimiento de su trascendencia, los usos potenciales que hoy ya son una realidad. Lo mismo le ocurrió a la

Internet, es llamativo que escritores de reconocimiento mundial en la literatura de ciencia ficción, no haya podido vislumbrar un concepto, siquiera parecido, a Internet.

Una posible explicación podría encontrarse en que *IoT* no es un invento en particular, es decir un "tal" o "cual" dispositivo, identificable en sí mismo. La idea fue evolucionando en paralelo al paradigma de la "*Computación Ubicua*" o *Pervasive Computing*. Término acuñado por el informático estadounidense *Mark Weiser*, en 1991. Propone que la evolución de la informática y la electrónica permitirán que el ser humano conviva con un ecosistema interconectado con ciertas características: tales objetos interactuarán con las personas mediante interfaces amigables, estarán encendidos y conectados permanentemente esperando ofrecer su servicio y pasarán desapercibidos repartidos en el entorno, entre otros.

La *Computación Ubicua* y la *IoT* se parecen demasiado, aunque no son el mismo concepto. *Iot* es un paradigma que tardó su tiempo en ser visibilizado y reconocido como tal. Además convergen en él otros conceptos que lo constituyen, lo que pudo haber contribuido a su ocultamiento, a la vista de todos.

Sea como fuere, *IoT* permaneció oculto y a las sombra de otros logros que avanzaban y lo eclipsaban. IoT avanzó en la sombra de Internet.

Excede el alcance de este trabajo profundizar sobre el origen de *IoT*, su desarrollo, estado actual y lo que vendrá. De hecho aún los tecnólogos y científicos no se ponen de acuerdo en una definición generalizada y completa del concepto. Existen diferentes definiciones, que si bien comparten algunos conceptos y criterios, de alguna manera divergen. Y no existe aún una definición aceptada por la comunidad científica y tecnológica.

2.6 La definición de la ITU

La *Unión Internacional de Telecomunicaciones*[6] es un organismo internacional que depende de las *Naciones Unidas*. Tiene la misión de regular y ordenar las telecomunicaciones. A mediados de la primera década de este siglo, ya abordaba la temática *Internet de las Cosas*. Dio su mirada sobre este concepto emergente y que se adopta en el presente trabajo, por ser la que mejor define, a criterio del autor, a IoT. Incluye diferentes aspectos de las tecnologías involucradas e incluye también una apreciación de la seguridad y privacidad de la información, siendo en ese sentido una de las primeras en reconocer las implicaciones negativas y riesgos subyacentes.

En su recomendación *ITU.Y.6060* define:

"...IoT puede ser considerada una infraestructura global para la sociedad de la información, permitiendo servicios avanzados para interconectar (física y virtualmente) de cosas, basadas en tecnologías de la información y las comunicaciones interoperables. A través de la identificación, captura de datos, capacidades de comunicaciones y procesamiento, IoT hace un uso integral de las cosas para ofrecer servicios para todo tipo

[6] International Telecomunications Union (ITU) es un organismo de las Naciones Unidas con sede en Ginebra. Tiene por finalidad regular la relación entre empresas operadoras y organismos estatales.

de aplicaciones mientras asegura que los requisitos de seguridad y privacidad sean cumplimentados".

Resulta interesante observar que esta definición abarca múltiples aspectos de *IoT*:

- Identificación y captura de datos
- capacidades de interconexión, comunicación y procesamiento
- ofrecer servicios y aplicaciones
- requerimientos de seguridad y privacidad.

Este último punto es, salvo algunas excepciones, mayoritariamente ignorado por muchos diseñadores y fabricantes. Esta ausencia genera vulnerabilidades, susceptibles de ser explotadas por personas maliciosas, por manipulación directa o a través de software malicioso o malware.

3. Seguridad en IoT

3.1 La gran ausencia presente en la cuna de IoT

Indagar acerca de las causas y motivaciones que llevaron a los creadores de los primeros dispositivos *IoT* de la historia para no destinar recursos a la seguridad y privacidad, excede el presente trabajo. No obstante, sería un interesante tópico de investigación porque al día de hoy la inseguridad de estos dispositivos es notable y los riesgos de tal situación, son enormes. De haber considerado mecanismos de seguridad, aunque sean elementales, tal vez hubiera iniciado una práctica saludable y la *IoT* tal como la conocemos actualmente, hubiera evolucionado considerando este aspecto tan crucial.

A modo de hipótesis y analizando el origen y evolución temprana de los primeros dispositivos *IoT*, tal como se puede apreciar en la exposición del capítulo precedente, se podrían presentar 2 posibles causas:

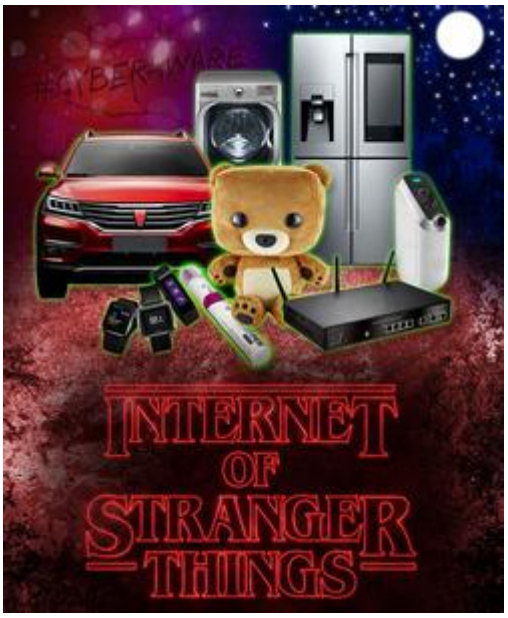


Figura 2: "Internet de las Cosas Extrañas" Survey de la empresa ESET (Fuente: Blog ESET 2016).

A) La información era considerada pública.

El servicio que aquellos primeros dispositivos ofrecían era considerado una curiosidad y ofrecían un servicio superfluo y poco convencional. Las imágenes que transmitía una cámara enfocada hacia una cafetera o los números que enviaba una máquina de gaseosas indicando la cantidad de botellitas que tenía en su interior, no constituían un servicio esencial, como tampoco procesaban y transmitían información sensible o confidencial. No violaba la privacidad de las personas, departamento o universidad involucrados. Por otro lado, tampoco exponía vulnerabilidades explotables.

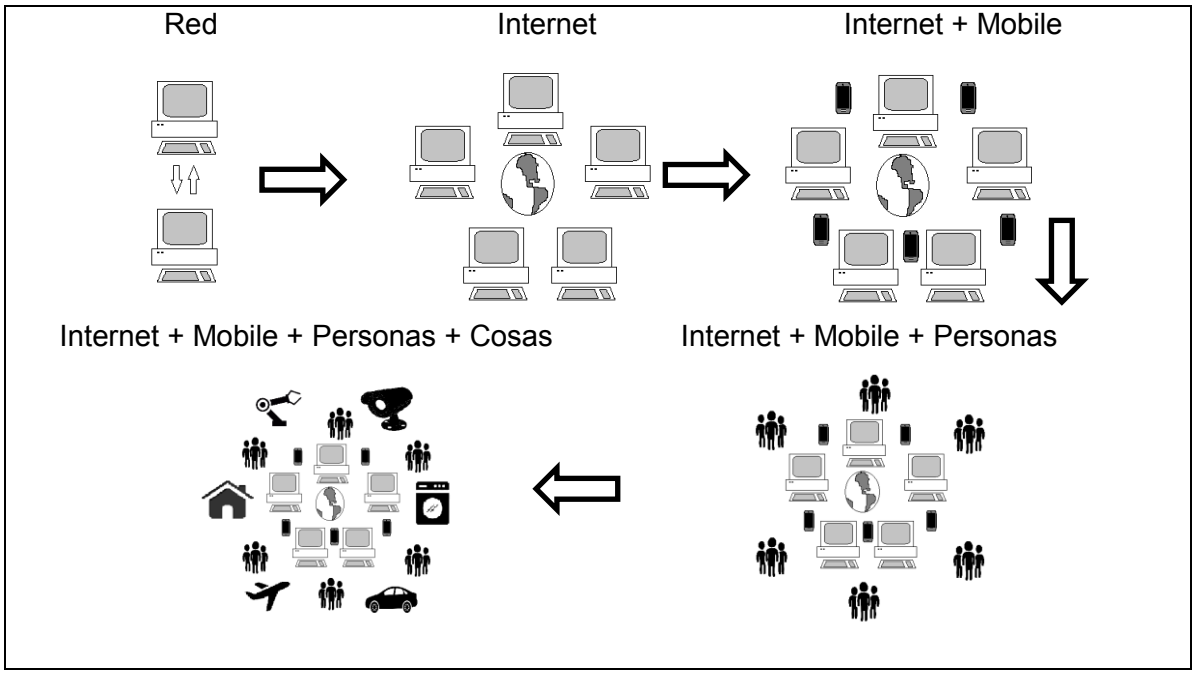
Nada hacía pensar que aquella información no fuera otra cosa que pública. Al fin y al cabo esa era la razón de ser de esos aparatos: ofrecer al público su servicio.

B) Comunidad reducida o la "pequeña comarca":

En aquellos tiempos la Internet estaba confinada a unos cientos de computadoras y como mucho, al alcance de un par de miles de personas a lo largo del mundo. Todas ellas de alguna manera relacionadas con la tecnología en general y la informática en particular. Podrían incluso ver con cierta simpatía a aquellos curiosos dispositivos, en el "edificio de al lado" o "al otro lado del océano", siempre protegidos entre las paredes de una universidad. No había razón para intentar afectar, si es que eso era posible en aquellas épocas, el funcionamiento de estos dispositivos o la información que ellos transmitían.

Tal como aquí se ha inferido, se puede entender que la seguridad de la información o de los equipos no era necesaria. Como tampoco que aquellos dispositivos presentaran un riesgo tangible. Pero la dinámica de la informática y el resto de las tecnologías que la acompañan, van evolucionando. *Hoy conocidas como Tecnologías de la Información y Comunicación*, familiarmente llamadas *TIC´s*.

Desde una *Red de Área Local* (*LAN: Local Area Network*) hasta la masiva *Red de Redes* (Internet), la cantidad de dispositivos y personas con acceso a ella, creció exponencialmente. Tal como puede observarse en el Esquema 1 el crecimiento de las *Tecnologías de la Información y la Comunicación* ha incorporado más y más personas y



Esquema 1: evolución de la Internet desde la simple red local hasta el estado actual.

dispositivos, sobre todo en los últimos años. Y con este crecimiento, aparecen los riesgos, tal como se estudiará en el próximo acápite.

3.2 Ataques a los dispositivos IoT

La situación de inseguridad que se vivía en la cuna de *IoT* y su primera infancia, no es extrapolable al día de hoy, de forma que es insostenible. El crecimiento exponencial de los dispositivos *IoT*, tal como puede observarse en el Gráfico 1 [9] como así también en el sinnúmero de aplicaciones y nichos tecnológicos en los que se encuentran operando, provoca que no se pueda ignorar las falencias de seguridad. Por las implicaciones y consecuencias que ellas acarrean en el presente y que se proyectan hacia el futuro.

También es parte del problema la *Tasa de Reemplazo* de los actuales dispositivos inseguros, por sus versiones mejoradas. Tal reemplazo se podría extender por un largo tiempo. Por lo que el problema de la inseguridad prevalecerá.

Y no se resolvería aún con la adopción de nuevos estándares y procedimientos de seguridad porque podría ser difícil y hasta imposible instalar tales mecanismos a los dispositivos existentes, producto de su propia naturaleza restringida en recursos.

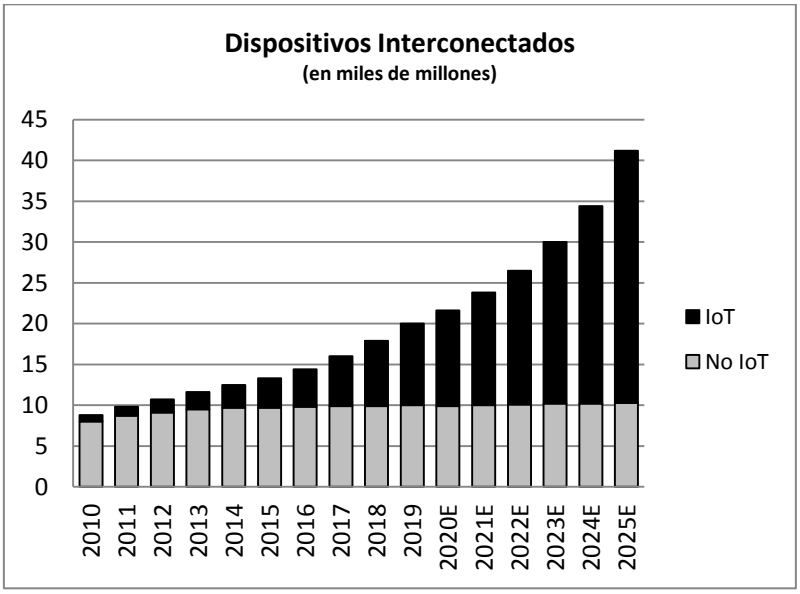


Gráfico 1: comparación de la evolución y crecimiento de los dispositivos Iot y no IoT para el decenio 2010-2025. (E: estimado). Fuente: IoT Analytics.

De acuerdo a los ataques conocidos, las tendencias y análisis de seguridad de los dispositivos, se pueden observar 3 finalidades u objetivos del ataque:

a) El dispositivo en sí mismo.

Este ataque recibe el nombre de *Ransonware de las Cosas* (*RoT: Ransomware of Things*), aunque el conocido investigador *Stephen Cobb*, experto en seguridad y privacidad de *ESET*, también ha dado en llamar "*Jackware*" [10].

Si bien expone un importante flanco para los ataques y muestra la vulnerabilidad de los equipos, aún no se han detectado ataques masivos. O al menos los mismos no se han difundido y dado a conocer públicamente. Lo que sí se han publicado son ataques de concepto y demostraciones [11].

Foto 7: Los Smart Car ya son una realidad cotidiana para muchísimas personas. Pero la débil seguridad de sus componentes IoT los hacen vulnerables a los ataques RoT o Jackware.

El atacante puede acceder al equipo y cambiar la configuración del mismo, tomando el control sobre él. Generalmente esta acción pasa desapercibida por el

auténtico usuario. Muchas veces estos dispositivos no requieren que se ingrese seguido a la configuración y usualmente una vez que inicia su servicio, no se vuelve a tocar.

Cuando el atacante lo crea oportuno, dará a conocer el “secuestro” del dispositivo y solicitará un rescate. Por supuesto, existe la posibilidad que el usuario pueda recuperar el control del equipo. Por ejemplo restableciendo el dispositivo a sus valores de fábrica (*RESET*) y tal vez el ataque no tenga mayores consecuencias. Esto sería sencillo de hacer cuando se tiene fácil acceso al dispositivo. Sin embargo podrían existir situaciones en las que tal contramedida podría tardar un tiempo en poder realizarse. Tal vez pudiera ser muy difícil de llevar a cabo o prácticamente imposible de hacerlo.

Tal es el caso de la siguiente lista de dispositivos y situaciones, que a modo de ejemplo, permiten comprender la trascendencia de este ataque:

- Dispositivos de *Internet de las Cosas Médicas* (*IoMT*) destinados a la medicina (marcapasos, bombas de Insulina o cualquier otro dispositivo que se implante en el cuerpo humano): por la situación del dispositivo, acceder al él no sería fácil ni rápido. Y ni mencionar si las intenciones del ataque no fueran solicitar un rescate, sino perpetuar un daño aún mayor. Habida cuenta, además, que muchos líderes políticos y grandes empresarios disponen de este tipo de dispositivos a su alcance.
- *Smart-Car* [12] Por la propia seguridad física del vehículo, acceder al dispositivo *IoT* es prácticamente imposible, sin provocar roturas de relevancia. Aun así, el problema es que tal ataque podría dejar varado en un lugar distante al usuario. Tal vez podría provocar una demora importante hasta su solución, elevados costos de traslado del vehículo, recuperación del mismo, etc.
- *Drones*: el atacante toma control de los mismos durante el vuelo, impidiéndole al usuario la recuperación física de los mismos. El usuario recibe el pedido de rescate a costa que el dispositivo se quede sin energía y caiga.
- *Internet de las Cosas Industriales* (*IIoT*): sensores, dispositivos de control industrial y demás aparatos que podrían afectar la producción de la planta o fábrica.
- *Internet de las Cosas Militares* (*IoTM*): Dispositivos como drones, sistemas de armas, sensores y demás dispositivos *IoT* cuyas aplicaciones en el ámbito militar como de las fuerzas civiles de seguridad, podrían ser intrusados, con los problemas que tal situación podría conllevar para la Ciberdefensa. Por ejemplo denegándose su servicio ante un ciberataque o un ataque cinético convencional.

b) La información que los dispositivos recolectan, almacenan, procesan y transmiten.

La información que estos dispositivos recolectan, procesan y transmiten tiene un alto riesgo de ser afectada. La intimidad y privacidad de las personas, como así también la

confidencialidad, integridad y autenticación de la información que dichos dispositivos recolectan y transmiten, puede ser vulnerada fácilmente. [13-15]

Existen casos documentados de ataques a los dispositivos asistentes personales como *Alexia, Siri, Google Home*, en los que el intruso pudo oír conversaciones, detectar la presencia de personas en la casa. Incluso existe un antecedente en el que uno de estos dispositivos grabó el audio de un asesinato ocurrido en su presencia y la controversia posterior, de índole legal, acerca de la validez o invalidez de tal prueba en el juicio [16].

Lo mismo ocurre con los *Smart-TV* que puede recibir órdenes de voz y hasta incluso pueden portar una cámara para recibir comandos gestuales [17].

Aquí se podría hacer una aclaración sobre la naturaleza del "intruso". Existen evidencias que, en ciertos casos, personal de las empresas que están detrás de los equipos tienen acceso a oír y capturar el audio que estos aparatos "oyen" [18]. Tanto sea a sabiendas de la empresa con la finalidad de "mejorar" los dispositivos, como así también con otros fines maliciosos. A la larga, el resultado es el mismo: la invasión a la privacidad de sus usuarios.

Peor aún son aquellos dispositivos como cámaras de vigilancia hogareñas, web cams, monitores de niños y bebés, juguetes inteligentes, llamados "*Smart Toys*". Ellos exponen más aún a los usuarios en su intimidad al abrir un canal directo a la escucha y observación de los mismos, en su propio hogar. Acceder a las imágenes [19] de un menor en la intimidad de su cuarto u hogar, podría ser un incentivo para que las redes de pedofilia y pornografía infantil encontrar en este tipo de ataques una importante fuente de imágenes para sus enfermizas actividades. Como así también podrían contribuir al *grooming* en el que el atacante acosa a un menor.

Otras víctimas susceptibles de ser afectadas por esta vulneración de la privacidad serían otra vez las personas implantadas con dispositivos *Internet de las Cosas Médicas* (*IoMT*). En ese caso interesa la información que podrían entregar sobre la salud de los pacientes-usuarios a los atacantes [20]. Y si además se tratarse de una persona con responsabilidades gubernamentales o empresariales, tal información podría ser usada en su contra o contra la organización que dirige o gobierna. O tal vez un deportista de elite a punto de realizar una importante y multimillonaria firma de un contrato podría afectar los montos del mismo si se conociesen datos médicos privados. Sólo por mostrar hipotéticos ataques.

c) El control del dispositivo

Tal como se ha indicado en párrafos anteriores, la cantidad de dispositivos IoT es tan enorme y muchos de ellos tan vulnerables, que resultan ser un blanco apetecible de recursos informáticos. Así, poder llevar adelante actividades maliciosas en las que se precise disponer de cientos, tal vez miles de estos dispositivos.

El atacante procura afectar al dispositivo víctima y "secuestrarlo" al igual que el ataque tipo *RoT*. Sin embargo aquí el objetivo no es impedirle el uso a su legítimo usuario. Por el contrario, mientras más tiempo tarde el usuario en detectar del ataque, mejor. Ya que no realizará contramedidas para recuperar el control del dispositivo y más tiempo estará bajo el comando del atacante.

Este ataque tiene 2 etapas. La primera consiste en infectar el dispositivo víctima con un malware, diseñado para que el dispositivo pueda reciba instrucciones. La segunda etapa consiste en asignarle al equipo una tarea como por ejemplo consultar una determinada página web, ejecutar una aplicación como minar *bitcoins*, enviar correo spam, al igual que un sinnúmero de otras acciones. Por supuesto, de acuerdo a la naturaleza y capacidades del equipo infectado.

El atacante procura tener bajo su control a cientos, miles y tal vez más dispositivos. Los que conformarán una red, conocida con el nombre de *BOTNET* [21] (por el acrónimo de *roBOT y NETwork*). A cada equipo así controlado se lo conoce con el nombre de *"zombie*" o "*esclavo*".

El atacante llevará el comandado a través de un dispositivo intermedio entre su equipo y sus zombies, llamado "maestro" (master). Incluso podría haber 2 equipos entre él y su red, para mejorar aún más su anonimato y su "distancia" con la red así controlada.

Abordar más detalles acerca de las botnets, excede el alcance del presente trabajo, aunque es un tema de sumo interés técnico para la *Ciberseguridad* y *Ciberdefensa* [22]

El mayor impacto del uso de *BOTNETS* es en el área de los ataques por *DDoS* (*Distributed Denial of Service*). Mediante este tipo de redes se pueden sacar de línea a servidores y servicios en Internet, dejando fuera de alcance los recursos de los mismos, por parte de los usuarios. Existen varios ataques documentados de este tipo, como por ejemplo *MIRAI* en 2016 [23].

PARTE 2

Criptografía Liviana y Stream Ciphers livianos estandarizados por la norma ISO/IEC 18033 y 29192.

4. La Criptografía Liviana, Ligera o Lightweght Cryptography

4.1 Más fuertes, más pesados.

Bruce Schneier, físico, autor de múltiples algoritmos criptográficos y experto en seguridad informática, define en forma sencilla el significado de "romper" un criptosistema:

> *"… simplemente significa encontrar una debilidad en el cifrado que puede ser explotada con una complejidad inferior a la de la fuerza bruta. No importa que la fuerza bruta pudiera requerir 2^{128} cifrados; un ataque que requiera 2^{110} cifrados se consideraría una ruptura... puesto de una manera simple, una ruptura puede ser tan sólo una debilidad certificacional: una evidencia de que el código no es tan bueno como se publicita". (Schneier. 2012).*

Fuerza Bruta es el nombre que se le ha dado a un ataque que, dado un mensaje cifrado *c*, siempre es posible hallar el texto claro *m* al que corresponde probando todas las claves *k* posibles, una por una. Dichas claves conforman un conjunto llamado "*Espacio de Claves*" *K* del sistema en cuestión. Y se llama *Cardinal(K)* a la cantidad de elementos de dicho conjunto.

Dado que cualquier mensaje *m* produce un mensaje cifrado *c* a través de la función de cifrado *f* y la clave *k*

$$f(m;k) = c \qquad (1)$$

Entonces el *Ataque por Fuerza Bruta* trata de descifrar *m* usando la función de descifrado *d* y probando todas y cada una de las claves k_i del espacio de claves:

$$d(c, k_i) = m \text{ (donde } k_i \in K) \qquad (2)$$

A la corta o a la larga, se logrará alcanzar la clave *k* con la que un cierto mensaje fue cifrado, porque se están probando todas y cada una de las claves del universo de claves posibles. Este método puede utilizarse tanto para sistemas de clave pública como para los de clave privada, aclarando en ese caso, que las k_i a probar no pertenecen al conjunto de claves públicas, sino al conjunto de posibles claves privadas.

Fuerza Bruta también introduce un parámetro de comparación, que permite evaluar por un lado cierta robustez de un criptosistema y por otro, la potencia de un ataque.

¿Todos los Criptosistemas son vulnerables a él? La respuesta es "todos, excepto el *Cifrado de Vernam*". Este mecanismo también es conocido por el nombre de "*Cuaderno de Uso Único"* o *One Time Pad.* Es el único que bajo las condiciones de "*Secreto Perfecto de Shannon"*[7] se puede demostrar matemáticamente que es irrompible, aún por *Fuerza Bruta*.

[7] Resulta paradógico conocer la existencia de un algoritmo irrompible ante cualquier ataque y que no se lo pueda utilizar en la realidad. Satisfacer las condiciones del llamado "secreto perfecto" de Shannon, es prácticamente imposible. Se requiere que la cadena cifrante sea totalmente aleatoria y de uso único. Que tal secuencia pueda estar disponible en

¿Por qué no se usa *OTP* si es irrompible matemáticamente? Por las mismas condiciones que le dan su robustez, impiden cualquier uso real del mismo: es prácticamente imposible que la secuencia aleatoria *k* generada en el emisor y que sirve para cifrar, sea generada en el receptor y sea usada para descifrar. Claro, siempre alguien puede decir… "que el emisor use la clave k y luego se la mande al receptor" y ahí se impone la repregunta ¿con qué clave se cifra la clave *k*? y si no es necesario cifrarla porque el canal de transmisión es seguro ¿por qué no transmitir el mensaje primeramente por ese mecanismo y listo? Como se ve, hay una enorme brecha entre los conceptos teóricos y el mundo de la realidad física.

Analizando ahora el tamaño o longitud de las claves *l*, el alfabeto o conjunto de caracteres *c* que se emplea para escribir las mismas y el tamaño del conjunto de claves *K*, se puede demostrar la existencia de una conexión exponencial entre dichas cantidades:

$$Card(K) = c^l \qquad (3)$$

Esta relación exponencial provoca que pequeños aumentos en la longitud de las claves usadas, conlleva enormes aumentos en su espacio de claves. Así por ejemplo, para claves binarias de *7* bits, se tendrán *128* claves. Mientras que para claves de apenas *3* bits más de longitud, es decir *10*, su espacio de claves asciende a *1024*, es decir *8* veces mayor.

Si es tan efectivo el ataque, ¿por qué los sistemas no son rotos a diario y siguen ofreciendo robustez? Porque recorrer el espacio de claves *K* no es tan sencillo de hacer, como de decir.

Cifra	Longitud de la Clave (bits)	Espacio de Claves K
César	5	$2^5=32$
Vigèner	25	$2^{25} \approx 3{,}2*10^7$
DES	56	$2^{56} \approx 3{,}2*10^{16}$
AES	128	$2^{128} \approx 2{,}56*10^{38}$
	192	$2^{192} \approx 4*10^{57}$
	256	$2^{256} \approx 6{,}4*10^{75}$

Tabla 1: Longitud de la Clave de una cifra y el espacio de claves correspondiente.

En la Tabla 1 se puede ver cómo a lo largo de los años, el tamaño de las claves y por supuesto, el consiguiente crecimiento del *Espacio de Claves* fue en aumento. Por supuesto, acompañado por el cambio de algoritmo de cifrado. Desde el *Cifrado de César* cuya clave era la cantidad de caracteres que se desplazaba el alfabeto cifrado respecto al claro, hasta el actual *AES: Advanced Encryption Standard*. Se observa que en apenas poco más de dos milenios, el tamaño de los espacios de claves aumentó varios cientos de órdenes de magnitud. El crecimiento más notorio se puede observar en los últimos 50 años, coincidente obviamente con el desarrollo de la informática.

---

ambos extremos de la comunicación es la primera de varias complicaciones que se deberían superar para obtener el cifrado "irrompible".

El problema se puede pensar mediante la siguiente analogía: alguien desea guardar un mensaje secreto, lo escribe en un papel y lo introduce dentro de un cofre. Para que nadie pueda alcanzar el papel, cubre el cofre con una piedra. Por supuesto, siempre es posible mover la piedra. Para evitarlo se usan piedras que nadie pueda mover. Cuando la fuerza humana no alcanza, los atacantes recurren a mecanismos como aparejos y grúas en procura de mover la piedra. Para evitar eso, los defensores usan piedras más grandes y pesadas aún, que pueda resistir el empuje de una grúa. Los atacantes usarán grúas cada vez más poderosas y los defensores usarán piedras más grandes. Y así sucesivamente, entrando en un bucle sin fin. Eso es lo que se observa entre la longitud y el espacio de claves versus la potencia de cómputo y las técnicas de computación paralela y sistemas distribuidos.

Volviendo al *AES*, es imposible de atacar por fuerza bruta en la práctica. Un cálculo sencillo permite conocer que si un atacante tuviera una potencia de cómputo que le permita probar 1 billón de claves por segundo (10^{12} claves/seg) recorrer todo el espacio de claves para atacar *AES*-256 le insumiría 10^{56} años. Basta comparar esta cantidad de años con la edad del universo, según los astrofísicos: *14 mil millones* de años, es decir $1,4*10^{10}$ años. Por lo que romper este algoritmo, requeriría disponer de muchísimos universos uno detrás del otro.

4.2 Clasificación de Seguridad de los Algoritmos

Además del ataque por Fuerza Bruta, en Criptoanálisis no todo es encontrar la clave y recuperar el texto claro a partir del cifrado. También se deben considerar otras técnicas de ataque, de las que hay muchas de ellas que nutren al Criptoanálisis y el número va en aumento.

Además de vulnerar la clave hallándola por fuerza bruta, también se puede perseguir la generación de mensajes sin necesidad de haber hallado la clave.

Nivel de Seguridad	Criterio
Aprobado (Approved)	El algoritmo se encuentra especificado en los documentos del NIST o hace parte de los algoritmos acreditados por el FIPS (Federal Information Processing Standard: Estándares Federales de Procesamiento de la Información). Puede ser empleado sin restricciones.
Aceptable (Acceptable)	El algoritmo y sus longitudes de clave son seguros para el uso y no se conocen riesgos de seguridad en ese momento.
Obsoleto (Deprecated):	El uso del algoritmo y de la longitud de clave es permitido pero el usuario debe aceptar algunos riesgos.
Restringido (Restricted)	El uso del algoritmo o de la longitud de la clave está obsoleto y hay restricciones adicionales requeridas para procesos de protección criptográfica de datos.
Uso heredado (Legacy-use)	El algoritmo o la longitud de clave puede ser usado para procesar datos previamente protegidos (descifrar datos o verificar una firma digital) pero pueden existir riesgos en este proceso.
No permitidos (Disallowed)	El algoritmo o la longitud de clave no son aceptados debido a los riesgos asociados.

Tabla 2: Nivel de Seguridad de los Algoritmos Criptográficos NIST-FIPS.

Afectar la clave de un sistema significa haber afectado su secreto, mientras que lograr generar mensajes aún sin conocimiento de la clave, afecta su autenticidad.

A medida que se logran avances en criptoanálisis y en potencia de cómputo, los algoritmos deben ser capaces de obtener mejoras… o sucumbir. Es por ello que los algoritmos reciben una clasificación que permite conocer el nivel de seguridad en el que trabajan, asumiendo que un algoritmo totalmente robusto en un momento, puede ser totalmente vulnerable, al siguiente. Esta clasificación es la que utiliza *el Instituto Nacional de Estándares y Tecnología* de *Estados Unidos* (*NIST* por sus siglas en inglés), tal como se observa en la tabla 2.

4.3 La criptografía convencional no aplica en IoT

Cuando la Criptografía se ejecutaba sobre computadoras, servidores y equipos informáticos tradicionales, no se tenía demasiada conciencia sobre los recursos que se requerían para ello. Si bien nunca fueron ilimitados, bastaban y sobraban para llevar adelante su cometido. Los algoritmos no se medían en los recursos que usaban o cómo lo hacían. Los criterios que más se observaban en cuanto a los algoritmos de cifrado, estaban más orientados a la seguridad, robustez y fortaleza de su función, que en tanto los consumos de recursos y eficiencia.

Pero al momento que la informática trasciende las barreras de las plataformas tradicionales y comienza a avanzar sobre las “cosas”, tal como se ha expuesto en parágrafos anteriores, los recursos reducidos con los que cuentan estos dispositivos comienzan a tomar un rol limitante. Ya no se puede recurrir a recursos abundantes, sino restringidos... y bastante restringidos.

Por su propia naturaleza, los dispositivos *IoT* tienen una característica en común: son aparatos limitados en tamaño, consumo de energía, poder de cómputo, almacenamiento y alcance de la señal. Razón por la cual asegurar los enlaces de comunicaciones desde y hacia estos equipos es una tarea muy difícil.

La criptografía “convencional”, es decir el tipo de técnicas criptográficas y primitivas matemáticas sobre las cuales los algoritmos descansan, no pueden ejecutarse en dichos entornos porque requieren para ello recursos que los dispositivos no pueden dar. Eso los convierte en inaplicable en esos contextos. Son algoritmos de gran “tamaño” medidos en cantidad de instrucciones, iteraciones y tamaño del bloque (el caso de los *block ciphers*), que requieren para su ejecución grandes cantidades de memoria *RAM*, potencia de cómputo (operaciones matemáticas extensas o con números muy grandes) entre otros factores. Esto hace que un algoritmo convencional no pueda ser implementado sobre estos dispositivos.

¿Significa entonces que los dispositivos *IoT* están condenados por su naturaleza, a la inseguridad y estado de vulnerabilidad permanente?

4.4 Criptografía Liviana o Ligera

Parecía que la vertiginosa competencia entre potencia de cómputo y tamaño de los algoritmos y sus claves, tuvo de alguna manera un final. Es decir que para tener mayor seguridad no era necesaria y exclusivamente, tener que ganarle a la potencia de cómputo y a los ataques por fuerza bruta.

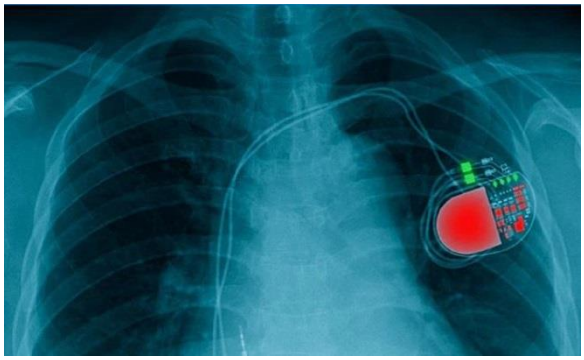


Foto 8 Implante de un marcapasos inalámbrico. (Fuente Blog Técnico Security Grarage, Entelgy Innotec.)

Fue así que nace toda una rama de la criptografía, que contradice ese "aparente" axioma. La llamada *Criptografía Liviana o Ligera* (*Lightweight Cryptography* en inglés) es la respuesta que la comunidad científica ha encontrado para proteger la información sin necesidad de algoritmos de mayor tamaño y potencia de cómputo. Ideales para ser adoptados en sistemas y dispositivos con recursos restringidos, limitados o que requieran bajo consumo de energía.

La norma I*SO/IEC 29192* del año 2012[24] define esta clase de criptografía como la adecuada para ejecutarse en entornos restringidos. Aunque se entiende que no necesariamente y exclusivamente en dichos entornos. Nada impide que se puedan implementar en sistemas y plataformas "convencionales".

Cabe señalar que en el seno de la *Criptografía Liviana* se pueden encontrar algoritmos livianos de *Cifrado en Flujo, Cifrado en Cadena* o también llamados *Stream Ciphers*. También *Cifrado en Bloque* o *Block Ciphers*. *Funciones Resumen* o *Hash* y también mecanismos de firma digital. Es decir el mismo espectro de funcionalidades y mecanismos criptográficos de la *Criptografía Convencional*, se podrán encontrar también en su hermana menor, la *Criptografía Liviana*.

Queda por delante explicitar el concepto de "*entorno restringido*". Se llamará así a las limitaciones o restricciones de recursos que un dispositivo pudiera tener. Según la norma, algunas de ellas podrían ser:

- área del chip
- consumo de energía
- tamaño del código del programa
- consumo o uso de memoria RAM
- ancho de banda del cifrado
- tiempo de ejecución

Para una mejor comprensión del significado y alcance de lo que se ha dado en llamar "*contextos restringidos"* basta observar la Foto 8. En ella se puede apreciar la radiografía

de un paciente implantado con un marcapasos *IoMT (Internet de las Cosas Médicas).* Estos dispositivos mantienen al corazón latiendo a un ritmo constante y adecuado, mediante la emisión de impulsos eléctricos. Y disponen de un canal inalámbrico para control, lectura de datos y cambios de configuración. Son obvias las limitaciones o restricciones de tal equipo:

- no pueden ser demasiado grandes en tamaño ya que se implantan en el interior del cuerpo humano y ello requiere de un cierto tamaño.
- no deben consumir mucha energía porque reduciría su vida útil y la renovación frecuente del dispositivo, con las complicaciones, riesgos médicos, molestias para el paciente, costos, etc. que ello representa.
- no pueden llevar adelante largos y complicados cálculos y procedimientos matemáticos, propios de muchos algoritmos de autenticación, firma digital, funciones hash, entre otros.

Esto provoca la reducción al mínimo de los recursos destinados a la confidencialidad, integridad y autenticación. Pero por otro lado, si la seguridad estuviera ausente o fuera débil, los riesgos serían inaceptables, tal como ya se ha visto anteriormente. Para ello, *Criptografía Liviana o Ligera*.

También se pueden encontrar algoritmos criptográficos livianos y estandarizados en la norma *ISO/IEC 18033* del año 2015.

5. Stream Ciphers Livianos Estandarizados Mediante Normas ISO/IEC

5.1 ¿Qué son los Stream Ciphers según la norma ISO/IEC 18033?

Se llaman así también a los algoritmos de cifrado en flujo o cadena. De alguna manera, son los herederos de muchas de las técnicas y algoritmos pertenecientes a la criptografía clásica, es decir desde el principio de la historia hasta el fin de la Segunda Guerra Mundial.

Los Stream Ciphers son definidos en la norma *ISO/IEC 18033-1:2015 "Information technology. Security techniques. Encryption algorithms. Part 1: General"* como sistemas de cifrado cuyo algoritmo tenga la propiedad de combinar una secuencia de símbolos de texto plano con una secuencia de símbolos de la clave (o secuencia cifrante), un símbolo a la vez, utilizando una función invertible. Así:

$$E_k(m_i) = c_i \quad (3)$$

$$E_{(k,\, vi)}(m_i) = m_i \oplus k_i \quad (4)$$

$$D_{(k,\, vi)}(c_i) = m_i \quad (5)$$

$$D_{(k,\, vi)}(c_i) = c_i \oplus k_i \quad (6)$$

Siendo $E_{(k,\, vi)}$ la *Función de Cifrado* para la clave *k* y vector de inicialización *vi;* $\oplus$ la función *XOR*; m_i el símbolo *i* del mensaje; k_i el símbolo *i* de la secuencia cifrante y c_i el símbolo *i* del texto cifrado. A su vez $D_{(k,\, vi)}$ es la *Función de Descifrado*.

Un *vi* o *Vector de Inicialización* es una secuencia de bits que se introducen en el algoritmo, junto a la clave y el texto en claro. Su finalidad es impedir que mensajes iguales sean cifrados idénticamente. Así, si un atacante estuviese "mirando" el canal de comunicaciones no advertiría una retransmisión. El *vi* es un número aleatorio de uso único, también conocido como nonce[8] y lo genera el emisor. Dado que le resulta imposible al receptor generar el mismo número, el *vi* debe transmitirse en texto claro, junto al texto cifrado.

5.2 Beneficios de la Estandarización

Los primeros algoritmos criptográficos en ser estandarizados y tener una norma internacional fueron los del tipo *Cifradores en Bloque* o *Block Ciphers*, a los que con posterioridad les siguieron los *Stream Ciphers*.

Es el caso del algoritmo *AES* (Advanced Encryption Standard) que fue reconocido en el 2005 en la norma *ISO/IEC 18033-3:2005 "Information technology. Security techniques. Encryption algorithms. Part 3: Block ciphers"*, junto a los algoritmos *TDEA*, *MISTY1*, *CAST-128*, *Camellia* y *SEED*. Uno de ellos es una variante del *algoritmo DES* (Data

[8] Nonce: Acrónimo en inglés de Number Once.

Encryption Standard: algoritmo que fue sustituido por el *AES*) llamado *TDEA*[9]. Y a partir de allí, otros *Block Ciphers* han sido normalizados.

Las ventajas que se obtienen de implementar algoritmos reconocidos en estándares internacionales son, entre otras:

- Libre disponibilidad del algoritmo para su uso.
- Descripción detallada de las funciones que lo conforman, como así también del diseño en general.
- Verificación del funcionamiento y conformidad de un grupo independiente de expertos.
- Existencia de "*Test Vectors*"[10] para la corroboración del buen funcionamiento de los mismos.

5.3 Normas ISO/IEC 18033 Y 29192

La norma *ISO/IEC 18033-4:2011 "Information technology. Security techniques. Encryption algorithms. Part 4: Stream ciphers"* presenta 5 algoritmos de *Cifrado de Flujo*. Cada uno definido detalladamente, su forma de trabajo, la carga de la clave *k* y el vector de inicialización *iv,* entre otra información relevante.

Partes de esta norma presentan soluciones criptográficas livianas, como por ejemplo *Block Ciphers* (parte 2), mecanismos asimétricos para el *Intercambio de Claves* (parte 4), *Algoritmos de Hash* (parte 5), *Códigos para Autenticar Mensajes* (parte6), entre otras técnicas. Las que serán motivo de otras investigaciones dadas sus importantes aplicaciones.

La norma *ISO/IEC 29192-3:2012* "*Information technology. Security techniques. Lightweight cryptography. Part 3: Stream ciphers"* además de la definición de Stream Ciphers, presenta dos algoritmos de cifrado en flujo que cumplen con todas las características para ser "livianos". Ambos están orientados a hardware, esto significa que fueron pensados y optimizados para que sean implementados directamente en ese entorno de trabajo, por sus características y propiedades de diseño.

5.4 Descripción de los Algoritmos

Primeramente resulta sorprendente cómo los creadores y diseñadores de algoritmos han podido y sabido construir con recursos muy limitados. Algunos algoritmos sorprenden por la simpleza y elegancia de sus diseños, contrariamente a muchos de sus hermanos "mayores" que a veces recurrían a confusas mezclas y complejos circuitos. En segunda medida, parecería que la idea de "ahorrar recursos", o tal vez mejor sea decir "aprovechar mejor los recursos", se está por convertir en un nueva tendencia de diseño de

[9] TDEA: Triple Data Encryption Algorithm. Es también conocido por el nombre de Triple DES.
[10] Test Vectors o Vectores de Prueba: dadas ciertos estados iniciales de los algoritmos, se listan los primeros *n* bits de su salida.

los algoritmos criptográficos, funcionen o no sobre plataformas o sistemas con recursos limitados.

A continuación se presentarán los algoritmos livianos estandarizados en ambas normas.

- **Decim-V2.**

Es un algoritmo orientado a Hardware [26]. Fue creado en 2005 por *Come Berbain, Olivier Billet, Anne Canteaut, Nicolas Courtois, Blandine Debraize, Henri Gilbert, Louis Goubin, Aline Gouget, Louis Granboulan, Cédric Lauradoux, Marine Minier, Thomas Pornin* y *Hervé Sibert*. El gobierno, empresas y centros de estudios franceses patrocinaron la creación de este algoritmo a través del *INRIA (Instituto Nacional de Investigación en Informática y Automatización.), Axalto Smart Cards, Cryptolog International, France Telecom, Departamento de Informática* de la *Ecole Normale Superieur* y el *Laboratoire PriSM* de la *Universidad de Versailles. DECIM* fue presentado en el *eSTREAM*[11] [27] (patrocinado por el *E-CRYPT*[12]).

Stream Ciphers Livianos Estandarizados	
ISO/IEC 18033	***Decim-v2***
	KCipher-2 (K2).
	MUGI.
	Rabbit
	SNOW 2.0.
ISO/IEC 29192	***Enocoro***
	Trivium

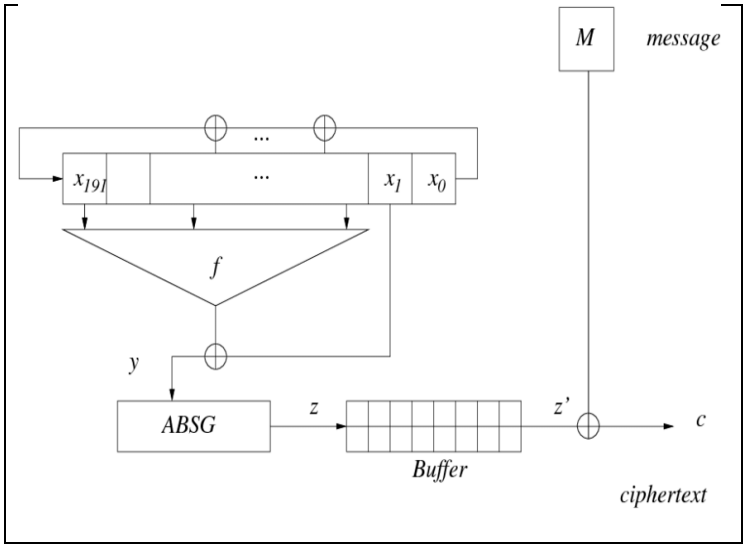
Tabla 3: algoritmos contenidos en las normas ISO/IEC.

DECIM-V2 avanzó hasta la ronda final del concurso, aunque no fue incluido en el portfolio, el cual estuvo conformado por 3 algoritmos de tipo Stream Ciphers orientados a Hardware:

- *Grain*
- *Mickey*
- *Trivium*

Y 4 algoritmos orientados a Software:

- *HC-128,*
- *Rabbit,*
- *Salsa20/12*
- *Sosemanuk*



Esquema 2: algoritmo DECIM-V2.

[11] eSTREAM: Proyecto de investigación europeo (2004-2008) con el propósito de "Identificar nuevos cifrados de flujo adecuados para una adopción generalizada".

[12] ECRYPT: *European Network of Excellence in Cryptology*. Iniciativa europea con el objetivo de promocionar la colaboración principalmente de investigadores europeos en el campo de la Seguridad de la Información, con énfasis en la Criptología, entre otros.

DECIM-V2 es la versión mejorada a la que fue presentada en el concurso. Utiliza *80* bits de clave y *64* bits de vector de inicialización. Existe además una versión presentada en 2007, llamada *DECIM-128* que trabaja con una clave y un vector de inicialización de *128* bits cada uno.

Su diseño consta de 2 mecanismos de filtrado: un filtro no lineal basado en una función booleana actuando sobre un *Registro de Desplazamiento Realimentado Linealmente* (*LFSR* por sus siglas en inglés) y un mecanismo de *Decimación* irregular, cuyos autores dieron en nombrar como *ABSG*, tal como puede observarse en el Esquema 2.

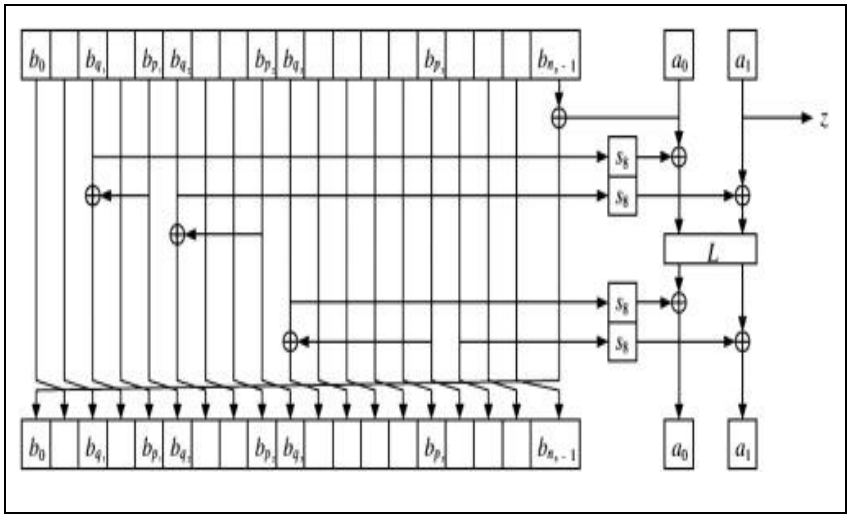
Demás está decir que tanto este *LSFR* como cualquiera de los que aquí se presentan, están basados en polinomios asociados primitivos, por lo que el ciclo de la secuencia binaria seudoaleatoria, será máxima.

También se observa un componente llamado "*Buffer".* Los autores indican que como la función *ABSG* no genera siempre la misma cantidad de bits por cada ciclo de reloj, entonces se van acumulando en dicho espacio los bits para que de esa forma puedan alimentar una salida uniforme y conformar el *Keystream* para cifrar.

- **Enocoro.**

En realidad no es un único algoritmo, sino una versión de *80* bits y tres de *128* bits las que son conocidas por los nombres de *Enocoro-80, Enocoro-128v1, Enocoro-128v1.1 y Enocoro-128v2*. Fueron creados en 2007 para la empresa japonesa *HITACHI* por *Dai Watanabe, T. Kaneko* [28-29].

Es interesante destacar que originalmente estos algoritmos fueron diseñados para ser *Generadores de Números Seudoaleatorios* basados en la filosofía *Stream Cipher,* sus autores han sabido convertirlos en algoritmos para el cifrado de información, conservando la misma filosofía.



Esquema 3: algoritmo Enocoro.

La *Generación de Números Seudoaleatorios* (*PRNG* por sus siglas en inglés) conforma un subcampo de estudio dentro de la Criptografía. Algunos de los usos que se les puede dar a los *Números Generados Seudoaleatoriamente* y que requieran robustez y seguridad criptográfica, son:

- *Generación de Vectores de Inicialización* para *Block Ciphers o Stream Ciphers*.
- *Generación de Salt* (sal) para dar seguridad al proceso de *Autenticación de Usuarios*, combinado con una función hash.
- En las *blockchain* basadas en *Proof of Work*, también se usa el hash combinado con un *nonce* en procura de evitar que la información de los bloques sea maliciosamente manipulada.

Enocoro es una variante del algoritmo *Panamá*[13] propuesto en el *5th International Workshop Fast Software Encryption* del año 1998.

El proyecto *CRYP-TREC*[14] [30] lo incluyó en la lista de "candidatos" del año 2013.

Si bien se dio a conocer un ataque sobre la versión *Enocoro-80*, no fue suficiente para considerar al algoritmo como significativamente vulnerable al mismo ni tampoco se ha reclasificado como "obsoleto", según la Tabla 2.

- **KCipher-2**

También conocido por el nombre *K2*, es un algoritmo creado por *Shinshaku Kiyomoto*, *Toshiaki Tanaka* y *Kouichi Sakurai* en conjunto para *KDDI Research Inc*[15] y la *Universidad de Kyushu, Japón*. Fue presentado en el año 2007 en el *3th The State of the Art of Stream Ciphers (SASC-07)* [31].

KCIPHER-2 es un algoritmo orientado a Software y muy veloz dada su sencillez. Puede ser implementado en Hardware y posee propiedades que le permiten llevar adelante procesos de paralelización. Emplea *128* bits de clave y *128* bits de vector de inicialización.

El algoritmo trabaja con 6 registros:

- *2 Registros de Desplazamiento Realimentados (Feedback Shift Registers)* llamados *Register(A)* y *Register(B)* (los que se pueden observar en el esquema se puede ver en el Esquema 4.
- *4 Registros de Almacenamiento Internos*, llamados *L1, R1, L2* y *R2*.

K2 tiene una curiosidad, uno de sus estados internos se actualiza a través de una función llamada *NEXT().* En su interior se puede encontrar una "Sustitución" definida en forma de caja, es decir es una *S-BOX*[16] tan reconocidas y usadas en los Block Ciphers. En este

[13] PANAMA es un algoritmo creado por *Joan Daemen y Craig* Clapp en 1998. Puede ser usado como Stream Cipher y como Hash. Ha manifestado algunas vulnerabilidades como hash. Una de sus variantes que resuelve las debilidades, llamada *RadioGatún* ha inspirado al algorimo *Keccak,* el que recientemente ha sido elegido como el *SHA-3 (Secure Hash Algorithm)*.

[14] CRYPTREC: Comité de Investigación y Evaluación de Criptografía creado por el Gobierno japonés para evaluar y recomendar técnicas criptográficas para uso gubernamental e industrial. Se inició en el año 2000 y la primera "Lista de Algoritmos Recomendados" fue publicada en 2003. La revisión de la misma se publicó en 2013.

[15] KDDI Research Inc: empresa destinada a la investigación, cuyos accionistas son las corporaciones KDDI, KYOCERA y TOYOTA MOTOR.

[16] SBox (Sustitution Box) es una función de sustitución, determinada por extensión y no por comprensión o analíticamente. Generalmente son capaces de aportar no linealidad, por lo que aumenta la complejidad matemática para llevar adelante el criptoanálisis. Es una de las estrategias usadas para ofrecer "Confusión", según Shannon.

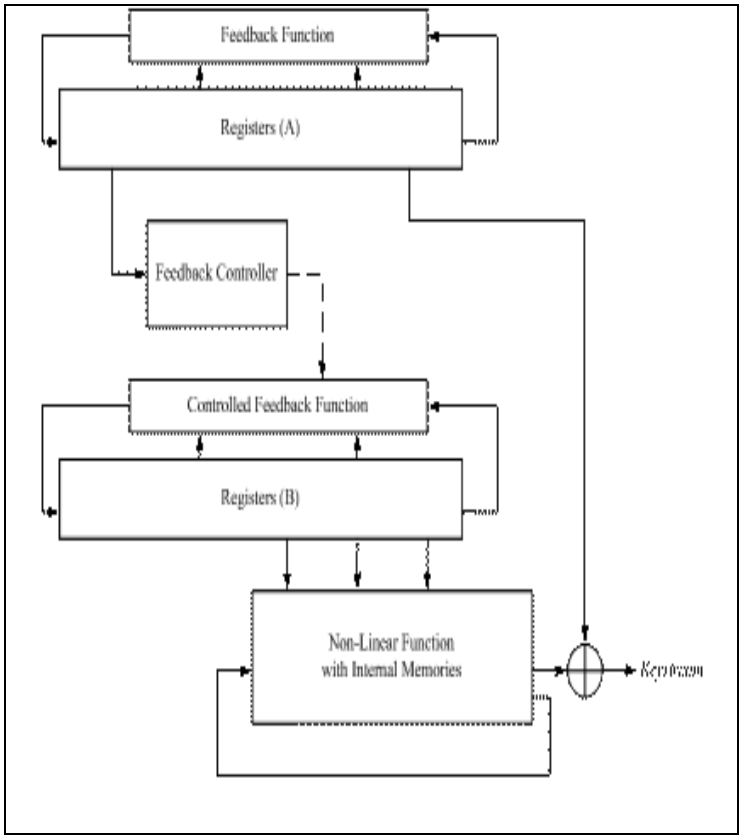
caso aplicada a un Stream Cipher. Interesante propuesta aunque no es el único *Stream Cipher* que tiene *SBox* en su constitución: *Beam, Mugi* (tal como se verá más adelante), *Self Synchronous Sober (SSS), ZUC, Sfinks, NLSv2,* y la familia *Snow,* entre otros.

Sus autores ubican a este algoritmo en la categoría "*Criptografía Liviana*" dado que, entre otras características, se pueden obtener altas prestaciones en entornos reducidos en recursos. Se pueden conseguir rendimientos entre *7* y *10* veces más veloces que el algoritmo *AES*.

Entre otras pruebas que realizaron, llevaron adelante una competencia de rendimiento en la que aseguran que un video de *4.7* GB[17] puede descifrarse en *8* segundos con *KCipher2*, mientras que con *AES* lograron hacerlo en *90* segundos [32-33]. Sostienen que el algoritmo tiene una cadencia de cifrado de *5*Gbps[18].

Hasta el momento no se conocen vulnerabilidades ni técnicas de criptoanálisis efectivas.

El proyecto *CRYPTREC* del gobierno japonés lo colocó en la *Lista de Algoritmos Recomendados para el Gobierno Electrónico* [34-35] y recomienda su uso.



Esquema 4: algoritmo KCIPHER-2

Además de estar normalizado en la ISO/IEC 18033, también lo está en la *RFC 7008*.

- **Mugi.**

Es un algoritmo creado por *D. Watanabe, S. Furuya, H. Yoshida, K. Takaragi* para la empresa japonesa *HITACHI* en el año 2001. Su nombre es el acrónimo de *Multi Giga Cipher.*

Fue presentado en el *9th International Workshop Fast Software Encryption* del año 2002[36]. *Mugi* y *Enocoro* comparten la misma empresa madre: *HITACHI* y un autor en común: *Dai Watanabe.* Y las comparaciones no terminan allí, al igual que *Enocoro*, este

[17] GB: giga byte.
[18] Gbps: giga bits por segundo, por sus siglas en ingles.

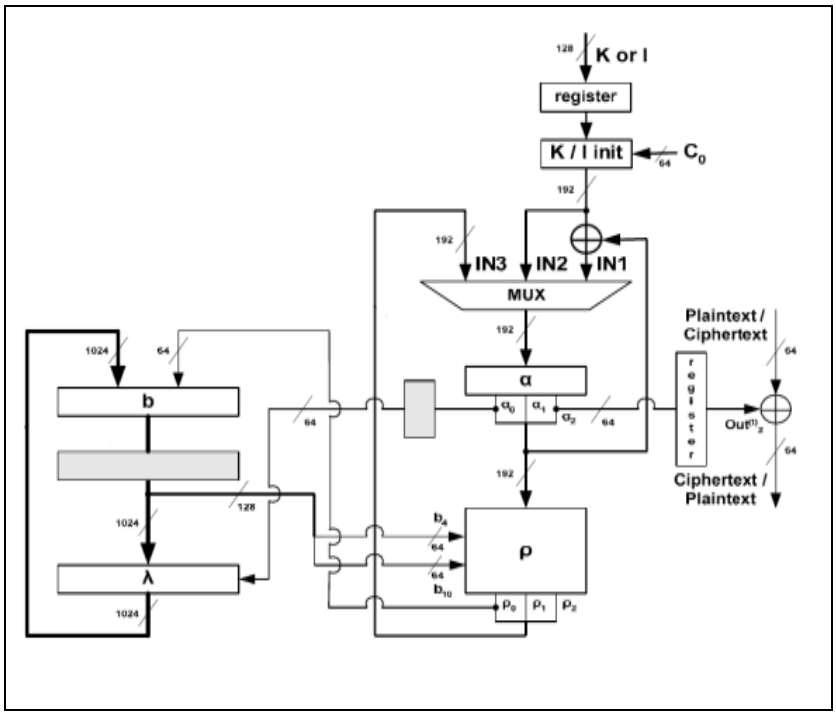
algoritmo también es un *Generador de Números Seudoaleatorios (PRNG*) devenido en stream cipher.

Es una variante del algoritmo *Panamá* propuesto en el *5th FSE 1998*. Panamá fue un algoritmo que podía utilizarse en modo hash o modo stream cipher. En 2007 fue publicada una vulnerabilidad, lo que ha provocado su recomendación para retirarlo de los usos criptográficos.

Mugi emplea una clave secreta y un vector de inicialización de *128* bits cada uno. Originalmente orientado a Hardware, también mostró muy buen rendimiento en Software.

Otra característica del algoritmo es que en él también se puede encontrar una *SBox*, para aportar no linealidad y confusión. Pero esta función de sustitución no es cualquiera, sino la mismísima empleada en el algoritmo *AES*, la cual ha demostrado ampliamente sus buenas características de seguridad a lo largo de todos estos años en los que el algoritmo ha sido atacado y aún ofrece resistencia.

Pero aquí no terminan las coincidencias con *AES*. Además de su *SBox*, *Mugi* comparte otra componente llamada *MDS Matrix* (*Maximun Distance Separable).* Su función es aportar "difusión" tal como la ha definido Shannon. Es decir que se procurará que pequeños cambios en los bits de entrada, provoquen grandes cambios en los bits de salida, de allí proviene el nombre de esta matriz.



Esquema 5: algoritmo MUGI.

El algoritmo está formado, a rasgos generales, por *3* registros de *64* bits cada uno, lo que los autores han dado en llamar "*Estados".* Y otros *16* registros, también de *64* bits cada uno, los que conforman el llamado *"Buffer".*

Es interesante hacer notar que aun habiendo sido utilizadas *2* componentes clásicas de las ideas de *Claude Shannon* como son la "C*onfusión*" (expresada en la *SBox*) y la "*Difusión*" (expresada mediante la *MDS Matrix*). Y siendo además las mismas funciones que se pueden hallar en el *AES,* que es un block cipher, los autores de *Mugi* han sabido combinarlas con otros componentes, de manera que han dado en crear un stream cipher.

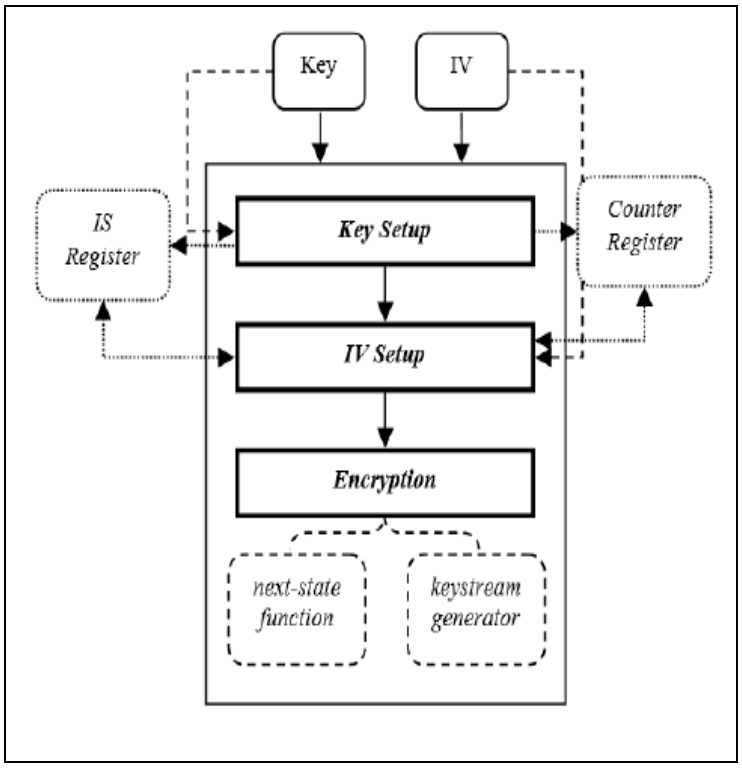
El proyecto *CRYPTREC* lo colocó en la lista de *Lista de Algoritmos Recomendados para el Gobierno Electrónico* [37] en el año 2003. Aunque en la revisión de algoritmos del año 2013, *Mugi* cambió de categoría y pasó a la de "candidato".

Cabe aclarar que este cambio no fue por haberse detectado debilidades en su seguridad, sino porque esos algoritmos no son de tan amplia difusión como otros.

- **Rabbit**

Este algoritmo fue creado por *Martin Boesgaard*, *Mette Vesterager, Thomas Christensen* y *Erik Zenner* pertenecientes a la empresa dinamarquesa *CRYPTICO A/S*.

En su diseño se pueden observar ciertas características de optimización orientado a software. Fue presentado en 2005 en el concurso internacional *eSTREAM* [38] y resultó ser uno de los algoritmos seleccionados. Integra el portfolio de los algoritmos finalistas en software, por su rendimiento y prestaciones.



Esquema 6: algoritmo RABBIT.

El algoritmo trabaja con una clave de *128* bits de longitud y un vector de inicialización de *64* bits. Utiliza además *8* registros de *32* bits de *Estado Interno* y otros *8* registros de *32* bits llamados "*Contadores*", los cuales pueden observarse en el esquema 6.

Mediante elementales operaciones aritméticas e ingeniosas rotaciones, los autores lograron un algoritmo muy veloz, que consume mínimos recursos de memoria y requiere poca potencia de cómputo.

Aunque se conoce un sesgo en algunos casos en los bits de salida del algoritmo, esto no reduce la seguridad del mismo pues la complejidad del ataque para explotar dicha vulnerabilidad fue calculada en el año 2006 en 2^{247}. Una mejora en el ataque, redujo la complejidad en 2008 a 2^{158}. Cabe aclarar que aun así, es mayor que la del ataque por "Fuerza Bruta", ya que recorrer el espacio de claves, tendría una complejidad de 2^{128}.

Es interesante destacar, cómo según se ha explicado anteriormente, como el "*Ataque por Fuerza Bruta*" ofrece aquí un indicador o "métrica" de la eficacia de un determinado ataque.

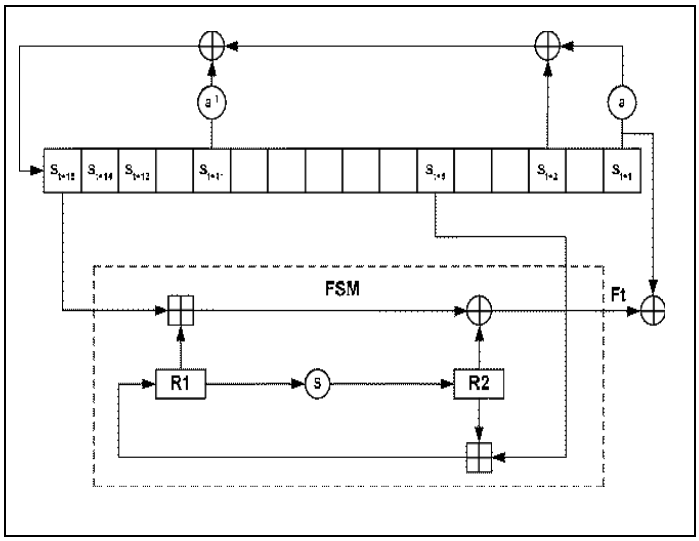
La empresa *Cryptico A/S* inicialmente tenía planeado patentar el algoritmo y así poder comercializarlo. Sin embargo, al resultar finalista en el *e-Stream*, desistieron de ello.

Además de ser uno de los algoritmos de la norma *ISO/IEC 18033,* también fue descripto en la *RFC 4503* del año 2006.

Además de ser un algoritmo de cifrado que también podría ser usado en sensores inalámbricos[39], también fue incluido en la implementación integrada, de código abierto y ligera, del protocolo *SSL/TLS*, *WolfSS*L [40].

- **Snow 2.0.**

Se conoce como *Snow* o *Snow 1.0* a la primera versión de este algoritmo, publicada en 2003[41]. Fue creado por *Thomas Johansson y Patrik Ekdahl* en la *Universidad Lund*, *Suecia*. Fue presentado originariamente en el *NESSIE*[19] *(New European Schemes for Signatures, Integrity and Encryption)* [42].



Esquema 6: algoritmo SNOW-2.0

Durante el proceso de evaluación de *NESSIE* fue descubierta una vulnerabilidad y retirado de la competencia. Por lo que, una vez resuelta, sus autores presentaron la versión mejorada bajo el nombre *SNOW 2.0.*en el *e-Stream.* Y esta es la versión normalizada en *ISO/IEC 18033.*

Con el correr del tiempo, *Snow* ya ha dejado de ser un algoritmo, sino que ya es una familia. En 2006 es presentada y aceptada la versión *Snow 3G* para ser utilizado en telefonía móvil para las redes de *3era Generación (3G)*. Y por su fortaleza y prestaciones, también se hecho extensivo su uso en las redes de *4ta. Generación (4G).* En el año 2018 se ha publicado un nuevo algoritmo integrante de la familia, llamado *Snow V*, cuyas aplicaciones podrían extenderse a redes de *5ta. Generación (5G*). El este trabajo se describirá al algoritmo normalizado, es decir *Snow 2.0.*

Según los autores, la versión 2.0 resuelve la vulnerabilidad descubierta en la versión 1.0, mediante la introducción de pequeños cambios. El algoritmo permite usar claves de 128 y 256 bits e internamente trabaja con palabras de 32 bits, las que componen un *Registro de Desplazamiento con Realimentación Lineal (LFSR),* al igual que en la versión anterior. Lo que sí han cambiado fue el *Polinomio de Realimentación*.

[19] NESSIE: Proyecto de investigación europeo (2000-2003) para la búsqueda de nuevos estándares para Europa de firmas, integridad y cifrado. Es equivalente al concurso del *AES* patrocinado por NIST (Estados Unidos) y su par japonés, el CRIPTREC.

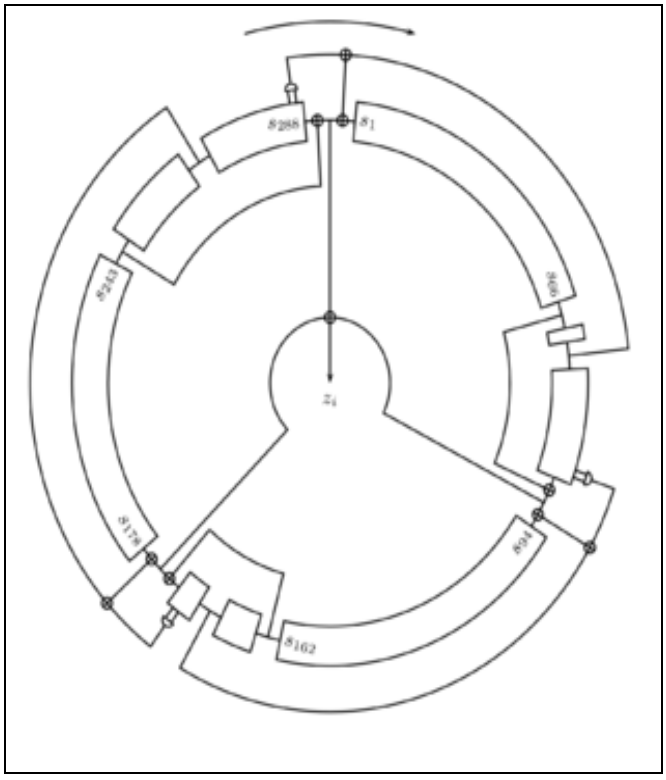
El otro componente destacado del algoritmo es una *Máquina de Estado Finito* (*FSM* por sus siglas en inglés) que tampoco ha cambiado de una versión a la otra. Dicha *FSM* está formada por *2* registros de *32* bits cada uno, identificados con los nombres *R1* y *R2* tal como puede observarse en el Esquema 6.

Una particularidad, tal como se ha indicado en acápites anteriores, *Snow* contiene una *SBox*, pero en este caso, se ubica en el interior de la *Máquina de Estado Finito*. La mencionada *Sbox* es la misma que se usa en el algoritmo *Rijndael*, es decir, *AES.*

- **Trivium.**

Este algoritmo fue creado por *Christophe De Cannière y Bart Preneel* de la *Graz University of Technology* en *Austria* y la *Katholieke Universiteit Leuven* en Bélgica.

Trivum se presentó en el 2005 en el *eSTREAM* en el área de Hardware [43-44]. Al finalizar este proyecto fue uno de los finalistas e incluido en el portfolio. Al finalizar este proyecto fue uno de los finalistas e incluido en el portfolio.



Esquema 7: algoritmo Trivium.

Es un algoritmo compuesto por un estado interno de 288 bits. Tiene una longitud de clave y de vector de inicialización de 80 bits cada uno. Los autores aseguran al menos 2^{64} bits de salida de su secuencia binaria seudoaleatoria.

El algoritmo está formado por 3 Registros de Desplazamiento Realimentados No Linealmente (NLFSR por sus siglas en inglés). Además de sus propios bits de salida usados para su realimentación, recibe bits de los otros registros, lo que se asemeja a un proceso de "acarreo". Tal mecanismo puede observarse en el Esquema 7.

Los bits *s286, s287* y *s288* de su estado interno, deben inicializarse siempre en 1. Los 80 bits de la clave se ingresan entre los bits *s1* a *s80* de su estado interno. Y los 80 bits del iv se deben ingresar desde los bits *s94* a *s174.* El resto de los bits pueden inicializarse con 0. Aunque si se desea, podrían usarse longitudes mayores a los bits estipulados de clave e iv, sin que ello necesariamente aumente la seguridad del algoritmo.

Antes de comenzar a cifrar, los autores recomiendan desechar 4 rondas de su estado interno: es decir 4 rondas de 288 bits cada una. Con lo cual los primeros 1152 bits de

salida no deben usarse para cifrar información. Esto se debe a que dichos bits conservan información del estado interno del algoritmo y pueden usarse en su contra mediante criptoanálisis. Este no es el único algoritmo que especifica este tipo de procesos preparatorios para el cifrado. Algunos autores a estos procesos lo llaman "*blanqueo*" o "*whitening*".

Aunque el algoritmo fue pensado para funcionar en hardware y para lo cual algunas decisiones de diseño se tomaron para optimizarlo en ese contexto, su implementación en software también es posible, obteniéndose buenos rendimientos y prestaciones.

Aunque se conocen varios ataques exitosos contra *Trivium*, ellos fueron llevados adelante en implementaciones reducidas o que no respetaran las recomendaciones de sus autores. Es por ello que estos ataques, al menos hasta ahora, no han resultado válidos como para afirmar su presunta vulnerabilidad.

Varios autores han podido demostrar que una reducción en cuanto a la longitud de sus registros es posible, sin perder por ello la robustez y seguridad que el algoritmo posee.

En particular el autor de este trabajo, junto a otros integrantes del equipo de investigación, han publicado una mejora en el ya veloz rendimiento del mismo. Tal variante se ha dado en llamar Trivium-Toy[45]. La misma consta de 3 registros, con una longitud total de 92 bits y se obtendría una velocidad de trabajo de alrededor de 3 veces mayores a la del algoritmo original con un tercio del uso de memoria.

6. Conclusiones

A lo largo de este trabajo se ha mostrado como la *Internet de las Cosas*, nacida y criada en sus primeros tiempos de vida sin ningún tipo de mecanismo de seguridad y privacidad de la información, ha evolucionado hasta la fecha mostrando que tal situación es insostenible.

Se han presentado muchas vulnerabilidades y ataques. Los que podrían evitarse si sobre tales dispositivos pudiesen incorporarse algunos mecanismos criptográficos que ofrezcan confidencialidad, integridad y autenticación. Se ha mostrado fundamentalmente 3 tipos de riesgos: hacia la información del usuario, contra el dispositivo en sí mismo y el dispositivo como plataforma de ataque o uso malicioso hacia terceros.

El problema surge cuando los servicios criptográficos que podrían mitigar esos riesgos son ofrecidos por algoritmos pertenecientes a la Criptografía convencional, la cual hace uso de una cantidad de recursos que los dispositivos *IoT* no tienen para dar.

Año de Publicación	Nombre	Norma ISO/IEC	Autores
2001	Mugi	**18033-4**	**Watanabe, Furuya, Yoshida, Takaragi, Preneel.**
2003	Rabbit.	**18033-4**	**Boesgaard, Vesterager, Christensen, Zenner.**
	Snow 2.0.	**18033-4**	**Ekdahl, Johansson.**
2005	Decim-v2.	**18033-4**	**Berbain, Billet, Canteaut, Courtois, Debraize, Gilbert, Goubin, Gouget, Granboulan, Lauradoux, Minier, Pornin, Sibert.**
	Trivium	**29192-1**	**De Cannière, Preneel.**
2007	Enocoro	**29192-1**	**Watanabe, Kaneko.**
	KCipher-2	**18033-4**	**Kiyomoto, Tanaka, Sakurai**

Tabla 4: algoritmos Stream Ciphers bajo normas ISO/IEC.

Se ha pesentado a la *Criptografía Liviana* o *Ligera* surge como un sub-campo de la *Criptografía* "convencional". La cual permite la creación de algoritmos criptográficos robustos, con menor consumo de recursos.

Estos algoritmos "livianos" pueden trabajar en entornos reducidos tales como los dispositivos *IoT*, logrando así integrar una necesidad con su solución.

Existen algoritmos livianos en casi todas las modalidades de la criptografía, ya que parecería que es la filosofía adoptada en los últimos años por los diseñadores de algoritmos: *Cifrado en Flujo, Cadena o Stream Ciphers*, *Block Ciphers o Cifrado en Bloque* y *Funciones Resumen* o *Hash*, entre otros.

Muchos de ellos se encuentran estandarizados bajo normas *ISO/IEC*. En el trabajo se han mostrado algunas de las ventajas que se obtienen al momento de estandarizar un algoritmo. También presenta los stream ciphers estandarizados bajo las normas *ISO/IEC 18033* y *ISO/IEC 29192*, pertenecientes a la Criptografía Liviana y que pueden implementarse en dispositivos *IoT*.

La Tabla 4 presenta, a modo de resumen, algunos datos de dichos algoritmos. La descripción más técnica, características criptográficas, diseño, debilidades y ataques conocidos, se expone con mayor detalle en el trabajo.

En resumen, se ha podido presentar a la comunidad científica los peligros de la *IoT* no segura y cómo se pueden mitigar algunos de esos riesgos mediante algoritmos criptográficos de tipo stream cipher, livianos y estandarizados bajo normas internacionales.

7. Agradecimientos

Un sinnúmero de situaciones me han traído hasta este momento. Una cantidad aún mayor de personas me han acompañado y contribuido a lo largo del camino, para llegar hasta aquí. A todas esas personas, que aprecio y valoro, muchas gracias.

En especial va mi reconocimiento y agradecimiento a:

- Mi familia, que ha sufrido mi ausencia cuando me tocó cursar, estudiar, rendir, investigar y finalmente, escribir. Largas horas de espera que pacientemente llevaron adelante por mí. Su esfuerzo está plasmado en este documento.
- Mi tutora y amiga la Licenciada Paula Venosa. Sin su apoyo, conocimientos y aportes no hubiera podido enfrentar esta tarea.
- A los profesores, administrativos y autoridades de la Especialización y de la Maestría en Redes, de la Facultad de Ingeniería de la Universidad Nacional de La Plata por sus enseñanzas, sabiduría, dedicación, acompañamiento y hospitalidad. Me sentí en familia y como en mi casa.
- Al Ingeniero Marcelo Zanitti, Decano de la Facultad de Ingeniería de la Universidad del Salvador, por su apoyo invaluable y su determinación para sortear los obstáculos.
- A mis colegas y amigos de los diferentes grupos de investigación de los que formo parte. Gracias a ellos, transitar por el arduo camino de la investigación se hace más liviano.
- Y por último a mi s padres, a los que ya no tengo a mi lado, pero que siempre me acompañan. Vaya mi recuerdo en el día que finalizo este trabajo y que coincide también con el de mi cumpleaños.

8. Referencias

[1] Ramirez, F. Troncoso, R. htttps://www.elladodelmal.com/2018/05/el-primer-dispositivo-iot-de-la.html

[2] Elder, J. “The Internet’s first thing – John Romkey’s ‘smart’ toaster”. Portal de la empresa de seguridad ESET, Septiembre de 2019. https://blog.avast.com/the-Internets-first-smart-device

[3] Martínez, M. “El primer objeto conectado a Internet fue una tostadora”. Portal La Piedra de Sísifo, Diciembre de 2016. https://lapiedradesisifo.com/2016/12/19/iot-tostadora/

[4] Autor no indicado, portal Wikipedia. https://en.wikipedia.org/wiki/John_Romkey

[5] Autor no indicado. “The Internet Toaster”. Portal Broadbandnow. Sección Legends & Miths. https://broadbandnow.com/Internet/i/ia_myths_toast.htm

[6] Stafford-Fraser, Q. “The Trojan Room Coffee Machine “. Página Web con la historia de la primera cámara web de la historia. https://www.cl.cam.ac.uk/coffee/qsf/

[7] Blanco, J.M. “Veinticinco años de la primera 'webcam': su creador solo quería “café recién hecho”. Portal del periódico El Confidencial, sección Tecnología. Abril de 2016. https://www.elconfidencial.com/tecnologia/2016-04-17/veinticinco-anos-de-la-primera-webcam-su-creador-solo-queria-cafe-recien-hecho_1184716/

[8] Ashton, K. “That ‘Internet of Things’ Thing”. RFID Journal. June 2009. (2009). https://www.rfidjournal.com/that-Internet-of-things-thing

[9] Lueth, K. “State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time”. Portal IoT Analytics. Noviembre de 2020. https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/

[10] Cobb, S. RoT: Ransomware of Things “ Portal de la empresa de seguridad ESET, Marzo de 2017. “https://www.eset.com/us/business/resources/tech-briefs/rot-ransomware-of-things/

[11] Autor no indicado. “Jackware, what is it?”. Portal de la empresa de seguridad Tech Sentries. https://www.techsentries.com/jackware-what-is-it/

[12] “What your car knows about you and the good and bad of that available data”. Portal del periódico The Dallas Morning News. Sección Automóviles. Septiembre de 2016. https://www.dallasnews.com/business/autos/2016/09/30/what-your-car-knows-about-you-and-the-good-and-bad-of-that-available-data/

[13] Larson, S. “Why buying used cars could put your safety at risk “. Portal CNN News, Sección Business. Febrero de 2017. https://money.cnn.com/2017/02/17/technology/used-car-hack-safety-location.

[14] Wang, A. “‘I’m in your baby’s room’: A hacker took over a baby monitor and broadcast threats, parents say” Portal de periódico The Washington Post. Diciembre de 2018. https://www.washingtonpost.com/technology/2018/12/20/nest-cam-baby-monitor-hacked-kidnap-threat-came-device-parents-say/?noredirect=on

[15] Instituto Nacional de Ciberseguridad, Gobierno de España. “Ciberseguridad en Smart Toys. Protección de Menores y su entorno desde la fabricación”. https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_smarttoys_final.pdf

[16] Masnick, M. “Amazon Refuses To Comply With Police Request For Amazon Echo Recordings In Murder Case”. 2016. https://www.techdirt.com/articles/20161227/12042636351/amazon-refuses-to-comply-with-police-request-amazon-echo-recordings-murder-case.shtml

[17] Laughlin, A. “Smart TV spying – are you watching TV, or is it watching you?” Portal Which? Agosto de 2014. https://www.which.co.uk/news/2014 /08/smart-tv-spying-are-you-watching-tv-or-is-it-watching-you/ - Which?

[18] Palomino, M. “Alexa, Siri y Google Home, ¿un riesgo para nuestra privacidad?” Portal 415 legal. Agosto 2014. https://451legal.com/alexa-siri-google-home-riesgo-para-privacidad/

[19] Cobb, S. “Cuando la realidad supera a la ficción: seguridad y privacidad en juguetes IoT. Marzo 2017. Portal We Live Security de la empresa de seguridad ESET. https://www.welivesecurity.com/la-es/2017/03/06/seguridad-privacidad-juguetes-iot/

[20] Cobb, S. “Pacientes ocultan su información por miedo a brechas en instituciones de salud”. Febrero 2016. Portal We Live Security de la empresa de seguridad ESET. https://www.welivesecurity.com/la-es/2016/02/22/pacientes-ocultan-informacion-brechas-salud/.

[21] Putman, C. G. J.; Abhishta; Nieuwenhuis, L. J. M. (March 2018). "Business Model of a Botnet". 2018 26th Euromicro International Conference on Parallel, Distributed and Net-work-based Processing (PDP): 441–445. arXiv:1804.10848.Bibcode: 2018arXiv 180410848P. doi:10.1109/PDP2018.2018.00077. ISBN 978-1-5386-4975-6.

[22] United States. Congress. Senate. Committee on the Judiciary. Subcommittee on Crime and Terrorism (2018). Taking Down Botnets: Public and Private Efforts to Disrupt and Dismantle Cybercriminal Networks: Hearing before the Subcommittee on Crime and Terrorism of the Committee on the Judiciary, United States Senate, One Hundred Thirteenth Congress, Second Session, July 15, 2014. Washington, DC: U.S. Government Publishing Office. Retrieved 18 November 2018.

[23] Cobb, S. “10 cosas que debes saber sobre los ataques DDoS a Dyn del 21 de octubre” Octubre 2016. Portal We Live Security de la empresa de seguridad ESET. https://www.welivesecurity.com/la-es/2016/10/26/ataques-ddos-a-la-iot-octubre/

[24] Portal de la International Organization for Standardization: Online Browsing Platform https://www.iso.org/obp/ui/#iso:std:iso-iec:29192:-1:ed-1:v1:en

[25] Portal de la International Organization for Standardization: Online Browsing Platform https://www.iso.org/standard/37970.html

[26] O. Billet, A. Canteaut, N. Courtois, B. Debraize, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin, and H. Sibert. Decim – A new Stream Cipher for Hardware applications. En ECRYPT Stream Cipher Workshop SKEW 2005.

[27] http://www.ecrypt.eu.org/stream/portfolio_revision1.pdf.

[28] D. Watanabe and T. Kaneko, A construction of light weight Panamalike keystream generator. Information Security and Cryptology: 7th International Conference, Inscrypt 2011. Revised Selected Papers. Springer. Beinjing.

[29] D.Watanabe, K. Okamoto and T. Kaneko, A Hardware-Oriented Light Weight Pseudorandom Number Generator Enocoro-128v2. Symposium on Cryptography and Information Security, SCIS2010, 3D1-3, 2010 (in Japanese).

[30] Página web del Concurso Criptográfico Cryptrec http://www.cryptrec.go.jp /english/images/cryptrec_01en.pdf.

[31] Página web del Concurso Criptográfico Cryptrec http://www.cryptrec.go.jp/ english/cryptrec_03_spec_cypherlist_files/ PDF /1002espec.pdf.

[32] Kiyomoto, S., Tanaka, T., and K. Sakurai, A Word-Oriented Stream Cipher Using Clock Control, Proc.SASC 2007, pp. 260-274.

[33] Página Web de los laboratorios KDDI Research, https://www.kddi-research.jp/sites/default/files/products/kcipher2/specification.pdf

[34] Página web del Concurso Criptográfico Cryptrec http://www.ecrypt.eu.org/stream/papersdir/2007/029.pdf

[35] Watanabe D., Furuya S., Yoshida H., Takaragi K., Preneel B. A New Keystream Generator MUGI. En Daemen J., Rijmen V. (eds) Fast Software Encryption. FSE 2002. Lecture Notes in Computer Science, vol 2365. Springer, Berlin, Heidelberg.

[36] Página web del Concurso Criptográfico Cryptrec http://www.cryptrec.go.jp/english/method.html

[37] Boesgaard, M.; Vesterager, M.; Pedersen, T. Christiansen, J. and Scavenius, O. Rabbit: A New Hagh-Performance Stream Cipher. Fast Software Encryption. 10th International Workshop, FSE 2003, LUND, Sweden. 2003.

[38] Portal web del Concurso Internacional e-CRYPT. https://www.ecrypt.eu.org/stream/index.html

[39] Tahir, R. Younas Javed, M. & Tahir, M. LRSA: Lightweight Rabbit Based Security Architecture for Wireless Sensor Networks. IITA '08: Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application - Volume 03, December 2008. Pages 679–683. https://doi.org/10.1109/IITA.2008.523.

[40] Portal de la implementación abierta y liviana WolfSSL https://www.wolfssl.com/

[41] Ekdahl, P. Johansson, T.A New Version of the Stream Cipher SNOW. Springer-Verlag Berlin Heidelberg. 2003.

[42] Portal web del Concurso Internacional Nessie. https://competitions.cr.yp.to/nessie.html

[43] 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 Berlin. 2004.

[44] Portal web del Concurso Internacional e-CRYPT http://www.ecrypt.eu.org/stream/papersdir/2006/021.

[45] Castro Lechtaler, A. Cipriano, M. García, E. Liporace, J. Maiorano, A. Malvacio, E.Model design for a reduced variant of a Trivium Type Stream Cipher. Journal of Computer Science & Technology; vol. 14, no. 1. ISSN: 1666-6038, Pg. 55-58. 2014