Institución Universitaria

MÉTODO PARA LA PRESERVACIÓN DE LA PRIVACIDAD EN DISPOSITIVOS IOT VESTIBLES EXTENDIENDO LA SEGURIDAD USANDO FOG COMPUTING

RAFAEL JOSÉ MONTOYA PONCE

Instituto Tecnológico Metropolitano
Facultad de ingenierías
Medellín, Colombia
2018

MÉTODO PARA LA PRESERVACIÓN DE LA PRIVACIDAD EN DISPOSITIVOS IOT VESTIBLES EXTENDIENDO LA SEGURIDAD USANDO FOG COMPUTING

RAFAEL JOSÉ MONTOYA PONCE

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:
Magister en Seguridad Informática

Director (a):
Magister HÉCTOR FERNANDO VARGAS MONTOYA

Línea de Investigación:
Ciencias Computacionales
Grupo de Investigación:
Automática, Electrónica y Ciencias Computacionales

Instituto Tecnológico Metropolitano
Facultad de ingenierías
Medellín, Colombia
2018

Dedicatoria

Agradezco el apoyo metodológico y anímico recibido en los dos primeros semestres a los docentes Gloria Mercedes Díaz y Mauricio Arias Correa.

A mi asesor Héctor Fernando Vargas por el tiempo y la entrega para codesarrollar este trabajo.

Al Instituto Tecnológico Metropolitano por brindarme parte de los recursos necesarios para mi formación y desarrollo de esta tesis.

A mi familia por apoyarme emocionalmente en la etapa académica.

Resumen

Los dispositivos Wearables o vestibles son elementos IoT personales que permiten la recolección de datos de una persona y enviarlas a un sistema informático para su procesamiento, dichos dispositivos hacen uso de conexiones locales (área de cobertura) y normalmente están dentro de dicha área en dónde se puede tener control de la privacidad, integridad o confidencialidad, por lo cual, cuando es necesario una movilidad de la persona, el IoT debe salir de su área de cobertura, , perdiendo dicho proceso de seguridad. El objetivo es proponer un método que ayude a mitigar el riesgo de pérdida de privacidad sobre los datos en los dispositivos IoT Wearables, cuando estos salen de un área protegida (en la cual normalmente se conectan) a través de la computación en la niebla o Fog Computing (en adelante FC) que viaje con el Wearable. Para esto se han evaluado tres tecnologías de comunicación inalámbrica, las cuales son Bluetooth, RFID y NFC; determinando así cuál es la que mejor se adapta para transportar la información desde IoT hasta FC. También se evalúan los algoritmos de cifrado RSA, Diffie-Hellman Elliptic Curve - DHEC, Homomórfico JPAILLIER - HJP y AES, como mecanismo de protección de la información que se envía desde FC hacia la nube, por lo cual, para la selección de la tecnología y algoritmo de cifrado más adecuado se hace una serie de pruebas técnicas, entregando a través de un sistema de puntos, la valoración de cada prueba.

Por último, los resultados de las pruebas son positivos y demuestran que la tecnología NFC es la que mejor se adapta a las limitantes de IoT y que DHEC es un algoritmo de cifrado que proporciona mejor escalabilidad al método planteado. Más tarde se realiza una evaluación de la propuesta en la cual se demuestra que soluciona el problema de pérdida de protección por salir de una cobertura protegida.

Palabras clave:
Wearable, Seguridad, Internet de las cosas, ataque informático, computación en la niebla.

Abstract

Wearables devices are personal IoT elements that allow the collection of data from a person and send them to a computer system for processing, said devices make use of local connections (coverage area) and are usually within that area in where you can have control of privacy, integrity or confidentiality, whereby, when a mobility of the person is necessary, the IoT must leave its coverage area, losing this security process. The objective is to propose a method that helps mitigate the risk of loss of privacy on the data in the IoT Wearables devices, when they leave a protected area (in which they normally connect) through fog computing or Fog. Computing (hereinafter FC) that travels with the Wearable. For this, three wireless communication technologies have been evaluated, which are Bluetooth, RFID and NFC; determining which is the best adapted to transport information from IoT to FC. The RSA encryption algorithms, Diffie-Hellman Elliptic Curve - DHEC, Homomorphic JPAILLIER - HJP and AES, are also evaluated as a protection mechanism for the information sent from FC to the cloud, for which, for the selection of The most suitable technology and encryption algorithm is a series of technical tests, delivering through a points system, the valuation of each test.

Finally, the results of the tests are positive and show that the NFC technology is the one that best adapts to the limitations of IoT and that DHEC is an encryption algorithm that provides better scalability to the proposed method. Later an evaluation of the proposal is made in which it is demonstrated that it solves the problem of loss of protection by leaving a protected coverage.

Keywords:
Security, Internet of things, computer attack, fog computing

Contenido

Contenido

Introducción....1

1. Marco Teórico y Estado del Arte....5
1.1 Marco teórico....5
1.1.1 Dispositivos y tecnologías IoT....5
1.1.2 Tecnologías de comunicación involucradas en el método....6
1.1.3 Criptografía....8
1.1.4 Fog Computing o FC....8
1.2 Estado del Arte....9
1.2.1 Propuestas de arquitecturas, diseño y desarrollo del IOT....10
1.2.2 Uso de algoritmos de cifrado....10
1.2.3 Propuestas con uso de FC....11

2. Metodología....14
2.1 Determinación del método de transporte....15
2.1.1 Definición Compatibilidad....16
2.1.2 Consumo de energía....17
2.1.3 Evaluación del procesamiento....17
2.1.4 Evaluación de la seguridad....18
2.1.5 Resultados finales....18
2.2 Determinación del algoritmo de cifrado....19
2.3 Diseño del método y desarrollo del prototipo....20
2.4 Evaluación de la capacidad de proteger la privacidad al extender FC con el método planteado....22

3. Resultados....23
3.1 Tabulación de la información y selección de la tecnología de comunicación....23
3.1.1 Evaluación de la compatibilidad de la tecnología de comunicación....23
3.1.2 Evaluación del consumo de energía de la tecnología de comunicación....24
3.1.3 Evaluación del consumo de procesamiento tecnología de comunicación....24
3.1.4 Evaluación de la seguridad de la tecnología de comunicación....29
3.1.5 Resultados finales consolidados....29
3.2 Tabulación de la información y determinación del algoritmo de cifrado....30
3.2.1 Condiciones de la prueba....31
3.2.2 Procedimiento....32
3.2.3 Resultados consolidados:....41
3.3 Resultados del método propuesto....42
3.4 Evaluación de la capacidad de proteger la privacidad al extender FC con el método planteado....45
3.4.1 Ejecución de la prueba de evaluación del método....47

4. Conclusiones y recomendaciones....50
4.1 Conclusiones....50
4.2 Recomendaciones, lecciones aprendidas y trabajo futuro....51

5. Anexo: Códigos de los desarrollos mencionados.....52

6. Bibliografía....53

Lista de figuras

Pág.

Ilustración 1, Modelo de conexión de los dispositivos IoT por zonas de cobertura. 2
Ilustración 2, Niebla entre el IoT y la nube. 5
Ilustración 3, Metodología de trabajo. 14
Ilustración 4, Arquitectura del IoT y conexión hacia FC. 15
Ilustración 5, Funcionamiento del prototipo. 21
Ilustración 6, API version distribution, tomado de IDE Android Studio Rama Canarian ...23
Ilustración 7, Arduino UNO 25
Ilustración 8,Módulo PN532 25
Ilustración 9, Comunicación entre PN532 y ARDUINO. 26
Ilustración 10, Módulo RC522 26
Ilustración 11,Comunicación entre RC522 y ARDUINO 27
Ilustración 12,Módulo HC05 27
Ilustración 13, Vinculación de HC05 con App PruebaBluetooth. 28
Ilustración 14, Comunicación entre HC05 y ARDUINO. 28
Ilustración 15, Capturas de App para las PruebasAlgoritmosCifrado. 30
Ilustración 16, Consideraciones técnicas. 32
Ilustración 17, Android Profiler – AES. 33
Ilustración 18, Captura comando TOP – AES. 33
Ilustración 19. Lectura comando TOP – AES. 34
Ilustración 20, Envío de información AWS – AES. 34
Ilustración 21, Android Profiler – RSA. 35
Ilustración 22, Captura comando TOP – RSA. 35
Ilustración 23, Lectura comando TOP – RSA. 36
Ilustración 24, Envío de información AWS – RSA. 36
Ilustración 25, Android Profiler – DHEC. 37
Ilustración 26, Captura comando TOP – DHEC. 37
Ilustración 27, Lectura comando TOP – DHEC. 38
Ilustración 28, Envío de información AWS – DHEC.. 38
Ilustración 29, Android Profiler – HJP. 39
Ilustración 30, Captura comando TOP – HPJ.. 39
Ilustración 31, Lectura comando TOP – HPJ. 40
Ilustración 32, Envío de información AWS – HPJ.. 40
Ilustración 33, Capturas Etiquetas NFC. 42
Ilustración 34, Capturas APP MyNFC. 43
Ilustración 35, Captura envío de información sin protección. 43
Ilustración 36, Captura envío de información protegida 44
Ilustración 37, Capturas APP ServerNFC. 44
Ilustración 38, Configuración de la red de la maquina Ubuntu en la nube de AWS. 45
Ilustración 39, Configuración del MFC. 46
Ilustración 40, Tabla DHCP del servidor de Gateway de la red LNV-PC. 46
Ilustración 41, Configuración de Interfaz de salida del servidor de Gateway.. 47

Ilustración 42, Evidencia de no protección del tráfico. 47
Ilustración 43, Filtro de captura en Wireshark. 48
Ilustración 44, Captura de tráfico con wireshark.. 48
Ilustración 45, TCPFollow con wireshark.. 49
Ilustración 46, Información en reposo.. 49

Lista de tablas

Tabla 1 Características de seguridad, ...7
Tabla 2, Tecnología en las redes PAN. 16
Tabla 3, Puntuación de compatibilidad definida por los autores. 16
Tabla 4, Puntuación consumo de energía. 17
Tabla 5, Puntuación de procesamiento.....18
Tabla 6, Rangos de comunicación por tecnología. 18
Tabla 7, Puntuación por rango de comunicación. 18
Tabla 8, Posibles nodos FC móviles.....21
Tabla 9, Consumo de energía por tecnología. 24
Tabla 10, Descripción de la asignación de puntos en la evaluación de seguridad. 29
Tabla 11, Tabulación de pruebas tecnología de comunicación.. 30
Tabla 12, Resultados de las mediciones de algoritmos de cifrado. 41
Tabla 13, Tabulación de pruebas algoritmo de cifrado. 41

Introducción

El concepto de IoT (por sus siglas en inglés –Internet of Things) ya hace parte de nuestras vidas, dispositivos como televisores, sensores en la agricultura, cámaras, neveras, relojes, entre otros se encuentran conectados a una red de computadores o a la nube. Dado el nivel de datos que se transporta o que residen en los dispositivos IoT conectados, se visualizan varios riesgos ante ataques informáticos como la pérdida de privacidad, disponibilidad o fuga de información, en particular, los dispositivos IoT vestibles que son los que acompañan a las personas, pueden almacenar información personal, del ambiente, de salud, entre otros. Estos dispositivos no fueron diseñados pensando en la seguridad de los usuarios dado la poca capacidad de procesamiento que poseen, por lo cual, las implementaciones de medidas de seguridad los hacen muy lentos en su nivel de respuesta.

La privacidad de la información tiene su fuerza en no develar (por omisión, accidental, por error o por conocimiento) aquellos aspectos que puedan identificar a una persona, pero el reto de proteger dicha privacidad cada día crece en la medida que el almacenamiento de la información personal va quedando cada vez en las tecnologías emergentes, considerando que nuevos mecanismos de comunicación o procesamiento de información se masifican (como IoT), dicha información tiene la posibilidad de quedar en manos de personas individuales (no necesariamente empresariales). En los dispositivos IoT vestibles si se quiere tener interacción con la nube u otro sistema externo, es necesario realizar conexiones directas hacia estas zonas, por lo cual, el concepto de tener un sistema de procesamiento más cercado (y superior al de IoT) que haga las veces de puente, viene tomando gran fuerza y es conocido como computación en la niebla o Fog Computing – FC. Delegar tareas de procesamiento a la FC entrega diversas ventajas a IoT como menor latencia, procesamiento, capacidad de almacenamiento, descentralización de administración, soporte de múltiples protocolos, reduce el consumo de banda ancha, entre otros [1]. El reto de evitar la pérdida de la privacidad en IoT ha prevalecido dado el aumento de las diferentes tecnologías y tendencias, considerando su forma de crecimiento [2][3][4]; Aún más cuando al día de hoy evoluciona el mercado y se desarrollan una gran cantidad de dispositivos (tales como vestibles o Wearables (traducción del Ingles)), técnicas, servicios y procesos que recolectan datos privilegiados de las personas; por ejemplo en el caso de la salud, se hace uso de elementos IoT para capturar y transportar información del cuerpo, que acorde a la ilustración 1, muchos de los dispositivos recolectan datos propios de las personas (que son implantados o van con la persona) y estos están conectados en un área de cobertura específica, (dada las limitaciones de potencia y cobertura propia de los dispositivos IoT), dichas áreas son únicas y su conexión no se hace de manera simultánea, por lo cual, cada que el dispositivo se traslada de área, es posible que se pierda la conexión y se generen inconsistencias en la información que se envía cuando se pierde la cobertura, dado que se debe iniciar el proceso de envió de datos una vez se alcanza una nueva conexión.

Los elementos IoT recolectan y envían esta información hacia un procesamiento central o a la nube de forma no segura, esto es, la mayoría de los dispositivos por sus capacidades de procesamiento no se les es configurada opciones de seguridad (esto consume potencia y procesamiento), por lo cual, la información viaja en texto plano sin controles suficientes o necesarios que eviten su develación por personal no autorizado, y esto sucede por las limitantes en recursos de computo que poseen los dispositivos IoT, ya que el tener limitantes, se dificulta la implementación de medidas computacionales de control que reduzcan los riesgos (dado que se debe pensar en la funcionalidad)

[4][5][6]. El cuerpo humano se está convirtiendo en una rica fuente de información y ésta es capturada a través de dispositivos electrónicos que a menudo incluye metadatos, como la ubicación, el tiempo, signos vitales y el contexto, lo que permite fácilmente Inferir hábitos personales, comportamientos y preferencias de los individuos[5]. En conclusión, muchos datos personales deben ser protegidos, reduciendo los niveles de exposición frente a posibles ataques informáticos.

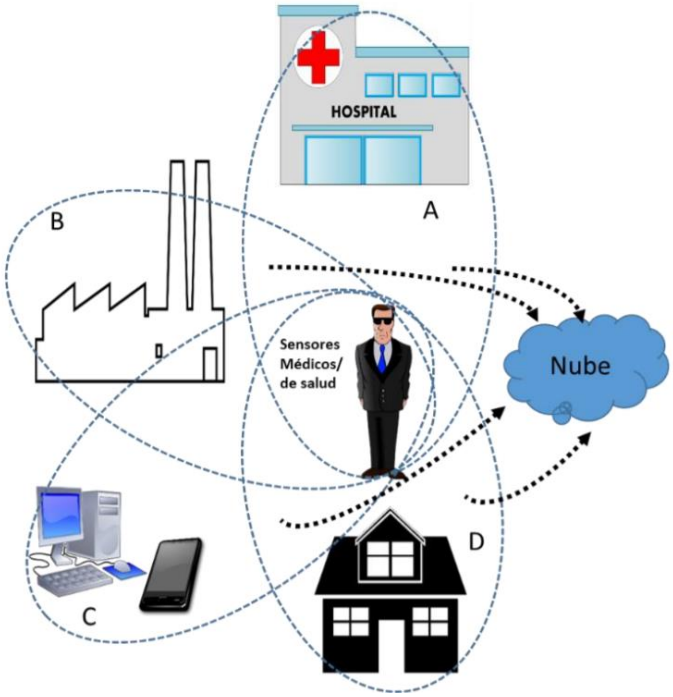

Ilustración 1, Modelo de conexión de los dispositivos IoT por zonas de cobertura. Fuente: Elaboración propia.

La privacidad en IoT es cada vez más relevante ya que el crecimiento del uso de los dispositivos aumentará, según la empresa consultor Gartner, se espera un crecimiento pronosticado del IOT para el año 2020 de 2.8 billones de dispositivos[7], dichos dispositivos podrían ser conectados a la red local y/o a Internet, así mismo, la organización GSMA (que representa los intereses de los operadores móviles de todo el mundo) estima que para el 2022 habrán 18 billones de dispositivos IoT conectados[8], por lo anterior, veremos que a medida que el tiempo transcurra el IOT estará más impregnado a nuestras vidas.

FC ofrece protección a IoT, pero esta se limita a la cobertura de un área protegida, si los datos capturados por los IoT wearables salen de esta área o son enviados directamente a la nube, se podría perder la privacidad ya adquirida.

Por lo cual se derivaron las siguientes preguntas de investigación:

- ¿Cómo lograr transportar de información entre los IoT y el FC conservando los niveles de procesamiento, seguridad y compatibilidad?
- ¿Cómo se podría proteger la información cuando es transportada desde y hacia el dispositivo IoT?
- ¿Cómo lograr integrar los mecanismos de transporte y de protección de información en un método que permita extender la FC para reducir los riesgos de seguridad?

- ¿Cómo revisar que la integración de transporte y seguridad sean funcionales?

Hipótesis

Si se logra extender FC para que ésta acompañe a todos lados a las personas que usan dispositivos IoT wearables, usando una tecnología inalámbrica, que se adapte a dichas necesidades para transportar la información generada por los wearables hasta la FC, y allí según se requiera se cifra o descifra con un algoritmo eficiente (rápido y de consumo bajo de recursos), entonces se agregaría una capa de seguridad a la comunicación, solucionando el problema de perdida de protección por salir de una área protegida.

Objetivo general

Proponer un método de extensión de FC que proporcione transporte y cifrado de la información generada por dispositivos IOT wearables, que mitigue el riesgo asociado con la perdida de la privacidad al salir de un área protegida.

Objetivos específicos

- Proponer el método de transporte que se adapte a las limitantes de IOT como consumo de energía, procesamiento, seguridad y compatibilidad. esto es para transportar la información desde los wearables hasta FC y viceversa.
- Proponer el algoritmo de cifrado que permita proteger la información entre FC y la nube, permitiendo al método planteado ser más eficiente y escalable.
- Diseñar un método que permita extender FC usando la tecnología de transporte y algoritmo de cifrado determinados anteriormente, sobre dispositivos móviles con mayores capacidades de procesamiento; como tabletas y celulares de alta gama, que funcionen como nodos FC móviles para disminuir los riesgos al salir del área de cobertura.
- Evaluar la capacidad de proteger la privacidad al extender FC en un conjunto de wearables con el método planteado.

Alcance:

El método propuesto busca proteger los datos que han sido entregados por el IoT wearable a la FC, está información es protegida y enviada a la nube, por lo cual, solo se protege la información en tránsito desde FC a nube y no en los elementos finales. La FC está compuesta por equipos Smartphone con Android, ya que es el sistema operativo más usado, es funcional, es de código abierto, cuenta con una comunidad libre más amplia y permite múltiples entornos de desarrollo [44], así mismo, en más probable que una persona lleve consigo dicho dispositivo, haciéndolo un elemento de FC.

En éste documento final se ha organizado iniciando con el marco teórico, en éste se desglosan los diferentes componentes de la solución (dispositivos IoT, tecnologías de comunicación, Fog computing y criptografía), luego va el estado del arte sobre la problemática ya descrita, seguidamente se hace una descripción de la metodología usada para el desarrollo de los objetivos; en el capítulo 3 se muestran los resultados obtenidos acorde a la metodología usada, finalmente, se entregan las conclusiones y recomendaciones del desarrollo del proyecto.

1. Marco Teórico y Estado del Arte

1.1 Marco teórico

A continuación, la ilustración 2 muestra como los dispositivos IoT se comunican con Fog Computing o FC a través de diferentes tecnologías de comunicación inalámbricas, luego se apoyan en FC para realizar tareas como el cifrado de la información a través de la criptografía, luego tener comunicación con la nube.

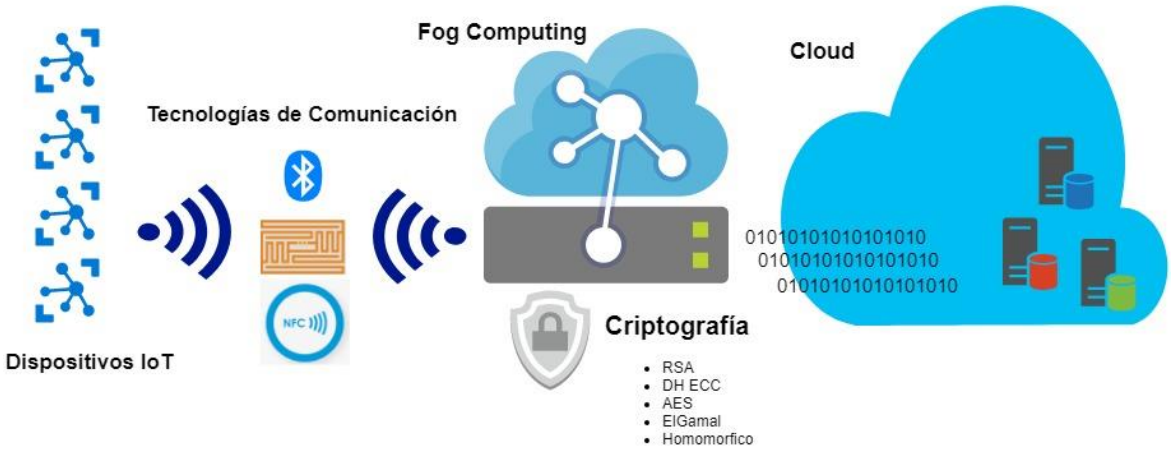

Ilustración 2, Niebla entre el IoT y la nube. Fuente autores

En ese sentido, el marco teórico se suscribe a los diferentes componentes que apoyan el FC y éste proyecto como tal (los dispositivos, las tecnologías de comunicación, el FC y la criptografía), para lo cual, a continuación, se da una descripción de los éstos componentes.

1.1.1 Dispositivos y tecnologías IoT

Según Yi, S., Li, C., & Li, Q [13], los dispositivos IoT despliegan una alta gama de equipos y sensores capaces de comunicarse entre sí a través de diferentes tecnologías; dentro de los elementos IoT podemos distinguir aquellos que apoyan las temáticas de ciudades inteligentes, vehículos, hogares inteligentes y elementos que las personas llevan consigo y que poseen la capacidad de recolectar, procesar y transmitir información, tal es el caso de los relojes inteligentes, medidores de temperatura y pulsaciones, marca-pasos, entre otros, éste tipo de dispositivos conocidos como IoT vestibles o Wearables en Ingles, poseen una gran funcionalidad y cubrimiento de necesidades inmediatas para las personas. Sin embargo, estas tecnologías hacen uso de redes de computadores para potencializar sus funcionalidades [37] [38], con ello lograr diferentes trabajos colaborativos y de procesamiento de información, tal es el caso del uso de la computación en la nube para procesar la

información de IoT, que es un modelo de servicio a demanda de capacidades de computo, almacenamiento, seguridad entre otros recursos a través de internet u otra red.

Es así como la FC [13], que se considerada una extensión del paradigma de computación en la nube, posee un dispositivo en el borde de la red que proporciona computación, almacenamiento, entre otros recursos, con la capacidad de estar en medio de la nube y el dispositivo final, ofreciendo realizar tareas que otros dispositivos no pueden realizar por sus limitantes. Según Cisco System [39] FC es un estándar que define cómo debería funcionar Edge Computing o computación perimetral, facilita el funcionamiento de los servicios computacionales, de almacenamiento y redes, entre los dispositivos finales y los centros de datos de computación en la nube.

Para Jiang [50], uno de los retos más importantes es la creación de dispositivos IoT wearables es la baja potencia, en dónde la eficiencia energética es el nuevo campo de batalla en IoT, así mismo, Jiang establece la necesidad que el uso de CPU sea lo más eficiente posible, dado el procesamiento de los dispositivos IoT deben tener a la hora de realizar alguna medición, si el procesamiento es bajo, la respuesta ante el monitoreo no será la adecuada.

1.1.2 Tecnologías de comunicación involucradas en el método

Los dispositivos IoT presentan diversos mecanismos de conexión [54], algunos dan un rango de distancia más amplio que otros, los dispositivos IoT hacen conexiones punto a punto, lo cual se logra a través de tecnologías de comunicación y enrutamiento, una vez establecida la conexión entre dos dispositivos, surge el proceso de intercambio de información. Dentro de dichas tecnologías se pueden considerar al menos los siguientes: NFC, Bluetoohh y RFID.

Según Kowalevicz, Pirrone, & Huerta [9], una de las primeras conexiones en IoT a que se hace referencia es el NFC por sus siglas en ingles Near Field Comunication, NFC utiliza el campo electromagnético inductivo a la comunicación, la investigación de esta tecnología fue iniciada en 2002 por PHILLIPS y SONY, en 2004 se crea el foro NFC, que se dedica a la creación de estándares para esta tecnología. Las características de esta tecnología se definen en la norma ISO 14443 [9] e ISO 180922, este último es el estándar actual para esta tecnología [33], opera en la banda de frecuencia de 13,56 MHz con niveles de potencia muy bajos, lo que significa que los dispositivos deben estar muy cerca, a una distancia de no más de 10 cm alcanzando velocidades de comunicaciones de hasta 464 Kbps.

Similar a NFC, existe otra tecnología basada en la proximidad, es el caso de RFID por sus siglas en ingles *Radio Frequency Identification*, Según Perera et al [1], esta tecnología utiliza campos electromagnéticos para transferir datos. Los sistemas RFID pueden clasificarse según los tipos de etiquetas y lectores. La etiqueta activa de lector pasivo (PRAT) comprende un lector activo y etiquetas activas donde el lector recibe señales de radio de etiquetas activas. Active Reader Passive Tag (ARPT) comprende un lector activo y etiquetas pasivas. El lector activo transmite las señales del interrogador y lee los datos almacenados en la etiqueta pasiva. Active Reader Active Tag (ARAT) tiene un lector

activo y etiquetas activas. El RFID activo puede realizar comunicaciones a lo largo de 100 metros y el RFID pasivo normalmente se limita en unos 100 cm. Las etiquetas RFID vienen en diferentes tamaños y parecen pegatinas.

Por otro lado, Según Perera et al [1], otra tecnología que se ha difundido desde su creación y hoy día todos los dispositivos tecnológicos lo traen para su uso es el Bluetooth (BLE) por sus siglas en inglés Bluetooth Low Energy, éste es una variación de bluetooth tradicional diseñado específicamente para funcionar con un bajo consumo de energía. Puede comunicarse en cortas distancias a 1 Mbps usando menos de 10mA. Además, a diferencia de Bluetooth, no tiene limitaciones en el número de dispositivos que se van a conectar y el tiempo necesario para conectar dos dispositivos es 0,006 segundos.

Según Ndjiongue et al & Grabovica et al [35] [36], así como el NFC Forum [54], cada tecnología de comunicación viene con sus ventajas y desventajas, tales como mayor o menor rango de comunicación, menor consumo de energético, mayor ancho de banda, mayor soporte en el mercado, las cuales deben ser evaluadas acorde a las necesidades de proximidad, uso y seguridad. Como se ha descrito, tanto RFID como NFC son tecnologías de contacto, esto es, que la distancia debe ser muy corta para poder establecer una comunicación entre dispositivos, contrario a Bluetooth, cuyo rango de distancia es mayor, por lo cual, si la distancia en más corta se tendría menor probabilidad de interceptación del tráfico.

Según Padgette et al [34], actualmente estas tecnologías (Bluetooth, NFC y RFID) sufren diferentes ataques como DOS (denial of service), MITM (man-in-the-middle), espionaje, modificación de mensajes y apropiación indebida de recursos; pero también cuentan con características de seguridad.

A continuación, en la tabla 1 se describen las características de seguridad de las tecnologías mencionadas.

Tecnología	Bluetooth	NFC	RFID
Características de seguridad	Cuatro modos de seguridad Autenticación Tres modos de Cifrado Autorización	Autenticación Cifrado	Cuatro modos de seguridad Autenticación Cifrado Firmado

Tabla 1 Características de seguridad, Fuente: [35] [36]

En cuanto al consumo de energía, según Yeon, Ki y Choi [48] es necesario que las tecnologías de comunicación sean eficientes, toda vez que los recursos son limitados en los dispositivos IoT. Así mismo, L. Salaman et al. [49] indica que un alto consumo de potencia daría como resultado una ineficiencia operativa en los dispositivos, generando reducción en la operatividad y cumplimiento de los objetivos como tal.

Para E. Valea et al. [50] Las consideraciones en el uso de CPU para los proceso de seguridad son fundamentales, toda vez que dichos procesos requieren de un consumo de CPU adicional a lo requerido por los dispositivos IoT en condiciones normales, por lo cual, un uso eficiente, rápido y alto de CPU (con la afectación mínima de los demás procesos), pueden fortalecer los temas de seguridad.

1.1.3 Criptografía

Para Granados [45] y Fernandez [46] la criptografía es un proceso o disciplina que permite hacer ininteligible un mensaje a través de algoritmos de cifrado, protegiendo la confidencialidad de los datos, la autenticación, su integridad y su no repudio. Según Focardi & Squarcina [10], la criptografía hace uso de tecnologías dominantes que da seguridad y ésta se presenta a nivel de software o de hardware, siendo la de hardware un elemento en continuo desarrollo por su nivel de fortaleza.

La criptografía [12] es uno de los procesos que puede brindar seguridad en diferentes áreas del procesamiento de información, así como en protocolos, servicios y dispositivos, así como a diferentes tecnologías, por lo cual, no se enmarca en un elemento tecnológico o tecnología en especial.

Según Vollala et al [11], los algoritmos de cifrado tradicionales como el RSA y el AES siguen siendo usados de forma muy frecuente; el algoritmo de cifrado RSA (el cual es clave pública) es el más utilizado y es aplicable para transformaciones criptográficas y firma digital en comunicaciones electrónicas seguras, correo electrónico, autenticidad de documentos electrónicos e implementación de redes privadas virtuales, RSA se basa en la multiplicación modular y la exponenciación modular. Este algoritmo [11] es costoso en términos de procesamiento, pero se han realizado modificaciones para mejorar su desempeño. Los mensajes enviados usando el RSA se representan mediante números y el funcionamiento se basa en el producto de dos números primos grandes (mayores que 10^100) elegidos al azar para conformar la clave de descifrado. Así mismo, Según Ma, Chen & Shi, el algoritmo de cifrado AES (por sus siglas en inglés Advanced Encryption Standard) es un algoritmo simétrico, es decir, que la llave para cifra y descifrar es la misma. El algoritmo AES [12] es capaz de utilizar claves criptográficas de 128, 192 y 256 bits.

Otro mecanismo de cifrado [4] que se han estado explorando son los algoritmos ECC por sus siglas en ingles Elliptic curve cryptography. Según Upadhyay & Patel, dicho algoritmo se basa en el método algebraico estructurado de curvas elípticas sobre campos finitos, éste método matemático utiliza un algoritmo para generar una clave pública y una privada para cifrar y descifrar los mensajes, este tipo de cifrado tiene unas funciones matemáticas más complejas y tiene la posibilidad de generar tamaño de clave más pequeño, algunos algoritmos basados en ECC son el Diffie-Hellman y ElGamal.

1.1.4 Fog Computing (FC) y el Cloud Computing

En consideración con Garcia et al. [52], la computación en la nueva trae nuevas alternativas que permiten la interconexión de los dispositivos finales con la nube, es un estado intermedio de procesamiento que tiende al aprovechamiento de los recursos locales antes de enviarlos a la nube.

Por otro lado, de acuerdo con Yi, S., Li, C., & Li, Q., [53] la FC es considerada una extensión del paradigma de computación en la nube, donde un dispositivo en borde de la red proporciona computación, almacenamiento, entre otros recursos, para realizar tareas que otros dispositivos no

pueden realizar por sus limitantes. El énfasis en FC se traduce en mejoras de rendimiento y procesamiento que IOT no haría y que podría delegar dichas tareas al FC, esto entrega diversas ventajas a IOT como menor latencia, procesamiento, almacenamiento, descentralización de administración, soporte de múltiples protocolos, reduce el consumo de banda ancha, entre otros (Perera et al. [1]).

Yi, S., Li, C., & Li, Q., [53] indican que FC se crea como concepto racional para lograr la conexión y procesamiento de los dispositivos IoT que requieren alguna conexión con la nube, pero por razones de rendimiento les es difícil establecer las conexiones de gran capacidad.

El cloud computing [55] [57] como elemento tecnológico de gran escala que posibilita el ofrecimiento de servicios en Internet, permite a personas y compañías de manera centralizada en almacenamiento y procesamiento y distribuida a nivel del acceso, realizar diferentes operaciones a través de servicios de tipo IaaS, SaaS o PaaS.

La FC [56] hace uso de múltiples nodos en los bordes de las redes para realizar el procesamiento, luego de ello, podría eventualmente enviarlo a la nube (o recibir de ésta la información y procesarla), con el procesamiento más cerca, es posible que las decisiones se tomen más rápido.

1.2 Estado del Arte

Según el Instituto Nacional de Seguridad de España -INCIBE [14], a pesar de algunos avances en temas de seguridad en IoT, aún se tiene muchos problemas que pueden facilitarle a un atacante obtener la información de los usuarios, según estudios realizados por el centro de respuesta de seguridad industrial de España (CERTSI) y la Agencia de la Unión Europea para la seguridad de la red y la información (ENISA – por sus siglas en inglés European Union Agency For Network And Information Security, encontraron al menos 17 amenazas técnicas que afectas a los dispositivos IoT. En dicho estudio encontraron que existen 10 escenarios de ataques con una importancia alta, dado su posible materialización en vulnerar los datos, considerando posible pérdida de integridad, confidencialidad y/o disponibilidad [14].

Así mismo, ENISA [15] indica que dichas vulnerabilidades han sido encontradas por diferentes atacantes y las han explotado formando redes de botnets (Redes de computadoras que fueron secuestrados y usados como robot para atacar a otros equipos y redes, también conocidos como zombies) como la red Mirai, que han realizado ataques de denegación de servicio como el ocurrido sobre el proveedor Dyn (o su evolución Wicked), el cual ofrece el servicio de resolución de nombre o DNS, logrando interrumpir el servicio de sitios tan relevantes como New York Times, Reddit, Twitter, Spotify, eBay, Netflix o CNN con tasas hasta de 620 gb/s [15].

Dado lo anterior, se realizó una búsqueda de los trabajos previos sobre seguridad en IoT, los cuales están divididos en las siguientes categorías:

1.2.1 Propuestas de arquitecturas, diseño y desarrollo del IOT

Según Frustaci, Pace & Aloi [41], las arquitecturas, propuestas de diseño y desarrollo en IoT han sido propuestas a través del tiempo por diferentes autores con el objetivo de construir soluciones basadas en IoT, asegurando las diferentes capas (aplicación, transporte y percepción), y teniendo el consentimiento del usuario, esto, independiente de los tipos y funcionalidades de los dispositivos IoT (sensores, Wearables, SmartHome, SmartCity, etc).

Es así, como Rose, Eldridge, & Lyman [8] proponen que debiesen existir notificaciones de consentimiento o un modelo de privacidad en línea, en la que los usuarios hacen valer sus preferencias de privacidad interactuando directamente con la información presentada en una computadora o pantalla del móvil, por ejemplo, hacer clic en botón de “estoy de acuerdo”, éste estado se rompe cuando los sistemas no proporcionan ningún mecanismo para la interacción del usuario.

Pero los dispositivos IOT con frecuencia no tienen interfaz de usuario para configurar las preferencias de privacidad y hacen que este reto no se solucione por completo y que el usuario no tenga como interactuar con el sistema ni aceptar las condiciones. Sin embargo, esto solo se encuentra en propuestas.

Según Minoli, Sohraby & Occhiogrosso [16], algunas alternativas para el diseño se intentan sobreponer, es así como se propone IoTSec, la cual es una iniciativa de seguridad para el diseño de un sistema IOT, su objetivo es asegurar cada dispositivo en cada capa de la red. Además, surgen varios modelos arquitectónicos de IOT, como Industrial Internet Reference Architecture (IIRA), the Internet of Things Architecture (IoT-A), the Standard for an Architectural Framework for the IOT IEEE, High Level Architecture for M2M, Internet of Things Reference Architecture (IoT RA - ISO/IEC WD 30141.) y OSiRM, Pero estos modelos no abarcan todo el alcance de la IoT y no solucionan el problema de pérdida de privacidad.

1.2.2 Uso de algoritmos de cifrado

Los algoritmos de cifrado han sido utilizados para proteger diferentes tipos de tecnologías, por ejemplo, tecnologías de comunicación que transportan información en texto plano, este tipo de algoritmos también pueden ser usados en IOT como se describe a continuación.

Según R. Lu et al [17], los algoritmos de cifrado tradicional utilizan varias técnicas matemáticas que evitan que personal no autorizado o quién no tenga las claves respectivas vean, modifiquen o alteren la información, por ejemplo Lu y sus colaboradores diseñaron un esquema de agregación eficiente para la preservación de la privacidad a través de cifrado para las comunicaciones de redes inteligentes EPPA (por sus siglas en ingles privacy-preserving aggregation scheme), que consta principalmente de cuatro partes: (i) inicialización del sistema, (ii) generación de informes de usuario, (iii) agregación de informes que preservan la privacidad y (iv) lectura y respuesta de informes seguros en el cual usan el algoritmo de cifrado asimétrico Paillier Cryptosystem, sin embargo, esto solo esta propuesto y no hay implementaciones, además por las limitantes de procesamiento que requiere para el cifrado, no todos los dispositivos IOT podrían adaptarse a este esquema.

Por otro lado, Yao et al [18] logran la autenticación multicast basada en cifrado de claves simétricas, proponen un mecanismo de autenticación de multidifusión de peso ligero que puede autenticar mensajes multicast a bajo costo para aplicaciones de pequeña escala, cumpliendo con los requisitos de autenticación de multidifusión en aplicaciones IOT que son escasos de recursos. Para lograrlo, usaron una función de tipo resumen matemático o hash unidireccional (conocido como el acumulador rápido de una vía de Nyberg), es una alternativa a las firmas digitales, su uso en este trabajo fue para verificar si un valor está en un conjunto especifico o no, sin embargo, esto está enfocado a los mensajes multicast.

Luego, Ning et al [19] diseñaron un esquema de autenticación jerárquica (APHA) basado en la arquitectura U2IoT (por sus siglas en inglés Unit IoT and Ubiquitous IoT) para las redes en capas, esta arquitectura hace uso de funciones criptográficas homomorficas y los mapas caóticos de Chebyshev, para la autentificación mutua entre dispositivos IOT, asignando diferentes autoridades de acceso para lograr un control de acceso jerárquico, pero esto solo aporta a la autenticación y transporte.

Otra alternativa fue expuesta por Upadhyay & Patel [4] para asegurar la comunicación entre dispositivos en la red IOT, proponen el uso de la criptografía de curva elíptica (por sus siglas en ingles Elliptic curve cryptography - ECC), que se basa en el método algebraico estructura de curvas elípticas sobre campos finitos. Utilizan este algoritmo para generar una clave pública y una privada para cifrar y descifrar los mensajes, como este tipo de cifrado tiene unas funciones matemáticas más complejas tiene la posibilidad de generar tamaño de clave más pequeño en comparación con otros algoritmos criptográficos, reduciendo con esto la potencia computacional requerida para cifrar y descifrar. La ventaja que trae es la dificultad de encontrar las ecuaciones de las curvas desde dónde se ha generado las ecuaciones criptográficas, por lo que un atacante podría tardar más tiempo en descifrar o desistir de intentarlo. Esta es la solución más ligera posible en el contexto de cifrar la información y proteger la privacidad en IOT, sin embargo, se ha trabajado solo en proteger la transferencia de información.

Por otra parte, Wang et al [20] propusieron que el cifrado homomórfico a partir de lo creado por J. Domingo-Ferrer en 1996, se puede utilizar para preservación de la privacidad en IOT, esta es una forma de cifrado donde la operación algebraica realizada en el texto sin formato es equivalente a otra operación algebraica realizada en el texto cifrado, pero sólo son seguros contra ataques de texto cifrado. Es decir, el atacante no tiene canal que proporcione acceso al texto plano antes de la encriptación. Pero en el entorno de ataque práctico, el atacante puede obtener algún conocimiento del texto plano.

1.2.3 Propuestas con uso de FC

De acuerdo con Yi et al [13], FC es considerada una extensión del paradigma de computación en la nube, donde un dispositivo en borde de la red proporciona computación, almacenamiento, entre otros recursos, para realizar tareas que otros dispositivos no pueden realizar por sus limitantes. El énfasis en FC se traduce en mejoras de rendimiento y procesamiento que IOT no haría y que podría delegar dichas tareas a FC, Según Perera et al [1] esta capa entrega diversas ventajas a IOT como menor latencia, procesamiento, almacenamiento, descentralización de administración, soporte de múltiples protocolos, reduce el consumo de banda ancha, entre otros.

Ibrahim, M [21] propone un esquema eficiente de autenticación mutua Edge-Fog de tres fases (inicialización, registro y autenticación), para permitir que cualquier usuario y servidor de la niebla se autentiquen mutuamente sin depender de ninguna infraestructura de clave pública PKI (por sus siglas en inglés *Public Key Infrastructure*) que deba generar certificados digitales o proveer cifrado de manera centralizada, para esto se requiere que los usuarios de la niebla almacenen solo una llave secreta de larga vida, el usuario debe ser capaz de autenticar con los nuevos servidores y unirse a la niebla sin la necesidad de volver a registrarse y sin ningún tipo de gastos adicionales. Este esquema utiliza herramientas elementales como funciones hash criptográficas y cifrado simétrico. Sin embargo, esto no es suficiente ya que se encarga solo de la autenticación y no tiene soporte a la movilidad que ciertos sensores IoT pueden requerir, además la llave secreta no varía durante el tiempo e instalar una PKI con sus componentes supone un nivel de soporte y mantenimiento especializado.

Por otra parte, Alrawais, et al [6] proponen que los nodos de FC pueden ser representados como proxies que proporcionan cálculos criptográficos, dado que los dispositivos y sensores IoT que están detrás carecen de los recursos necesarios para hacerlo. Con referencia a lo anterior, la computación en la niebla podría proporcionar no sólo los recursos computacionales adicionales, sino también un nivel de seguridad sin precedentes que ayudará a minimizar los ataques en los entornos de la IoT. También proponen un nuevo esquema de revocación de certificados para IoT basado en FC, que consiste en que un nodo de niebla es responsable de servir a un grupo de dispositivos que usan certificados digitales emitidos por una CA (autoridad de certificación) particular donde usan un filtro de Bloom, que es una estructura de datos de espacio eficiente para almacenar un grupo de elementos en un vector de bits, que se puede utilizar para comprobar si un elemento es un miembro del grupo. Con el uso de este filtro crean una lista corta que puede reducir efectivamente el tamaño de lista de revocación con sobrecarga aceptable, sin embargo, este trabajo solo se enfoca en la revocación de certificados y no tiene en cuenta la movilidad.

Luego Wang et al [20] presentan el concepto de esquema de agregación anónima y segura (ASAS) en cloud computing basado en niebla. En este modelo, un nodo de niebla reúne los datos de los nodos terminales y reenvía la información al servidor de nube público (PCS), por medio del uso del esquema ASAS la FC puede ayudar a los dispositivos finales como IoT a cargar sus datos en el PCS. Mediante el uso de la técnica de agregación de datos, Este esquema puede ahorrar ancho de banda entre FC y PCS, al mismo tiempo protege las identidades de los dispositivos terminales usando seudónimos.

El Modelo ASAS se basa en técnicas criptográficas que incluyen la firma, el cifrado y la agregación en la criptografía de clave pública, específicamente en criptografía de curva elíptica y el criptosistema Castagnos Laguillaumie [20], Aunque sea una buena aproximación está limitado a la cobertura de la niebla.

Por otro lado, el mismo Wang, Q et al proponen un esquema de protección de privacidad para sistemas de publicación/suscripción, en el cual usan nodos FC como brocker (Agentes intermediarios).

Según Motoda et al [42], este esquema está dividido en tres partes: notificación de eventos, Descubrimiento y minería de atributos, haciendo uso del algoritmo UApriori (el cual es una variación del Apriori, utilizado en el ámbito de minería de datos) y la distribución de Laplace, por último Los brokers utilizan los atributos encontrados (de cada usuario) para que coincida con los editores y

suscriptores apropiados, todo esto usando la tecnología Differential privacy, la cual consiste en generar ofuscación de la información de los usuarios y así los atacantes no puedan analizar la información y saber sus preferencias [22]. Sin embargo, este esquema solo busca proteger la privacidad contra ataques de colusión.

Por otra parte, Lu et al [23] plantean un ligero esquema de preservación de privacidad usando FC llamado LPDA, el cual combina el algoritmo homomórfico de Paillier, el Teorema del Resto de China y las técnicas de cadena hash unidireccional para permitir que un dispositivo FC filtre anticipadamente los datos falsos inyectados. En este esquema existe una autoridad de confianza la cual asigna las llaves a los dispositivos IOT, al nodo FC y al servidor de control de la nube. no obstante, en este esquema se considera la entidad de confianza es completamente confiable y no se tiene en cuenta que los dispositivos IOT pueden estar comprometidos.

De otro lado, Rahmani et al. [24] proponen que en el área de la salud la FC puede ser usada para ayudar a IOT con tareas como: toma de decisiones de forma local, seguridad, procesamiento, filtrado, compresión, fusión, análisis y almacenamiento de los datos; así mismo proponen controles periféricos tales como sistemas firewall como el IPtables o de forma local como el cifrado de la información almacenada en la FC, además, se indica que los FC deberían ser reconfigurables e interoperables. Finalmente, desarrollan un sistema inteligente de salud IOT, compuesta por una red de sensores apoyados por un prototipo Fog Smart e-Health Gateway llamado UTGATE, para demostrar interoperabilidad se instalaron tres interfaces que soportan Wifi, bluetooth y 6LoWPAN, Proporcionando procesamiento, almacenamiento y comprensión con el método LZW. Esta propuesta contempla la cobertura cuando los dispositivos IoT de la salud requieren moverse de una ubicación a otra, pero aun así está limitado a áreas protegidas (áreas de cierta cobertura).

2. Metodología

Para el cumplimiento de los objetivos, se inició con la determinación del método de transporte de la información entre los IoT wearables y FC, se evaluaron variables como la compatibilidad, consumo de energía, procesamiento y seguridad de las tecnologías Bluetooth, RFID y NFC; Luego se determinó el algoritmo criptográfico que protege la información que se envía a la nube, evaluando variables como el nivel de riesgo, consumo de energía, tiempo de ejecución, consumo de banda ancha. En la tercera fase se procede con el diseño del método y desarrollo del prototipo. Por último, se realizará una evaluación de la protección que entrega la propuesta. El proceso metodológico se puede ver ilustración 3

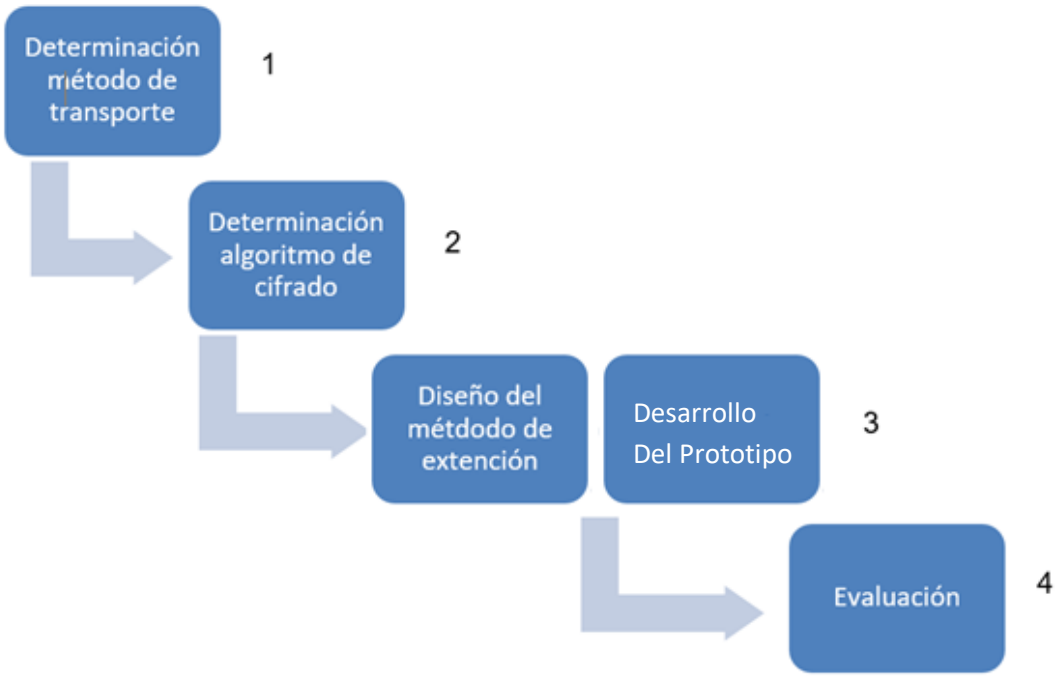

Ilustración 3, Metodología de trabajo. Fuente: Elaboración propia

Para desarrollar cada una de las fases, es importante visualizar la arquitectura del IoT y la conexión hacia FC (ilustración 4).

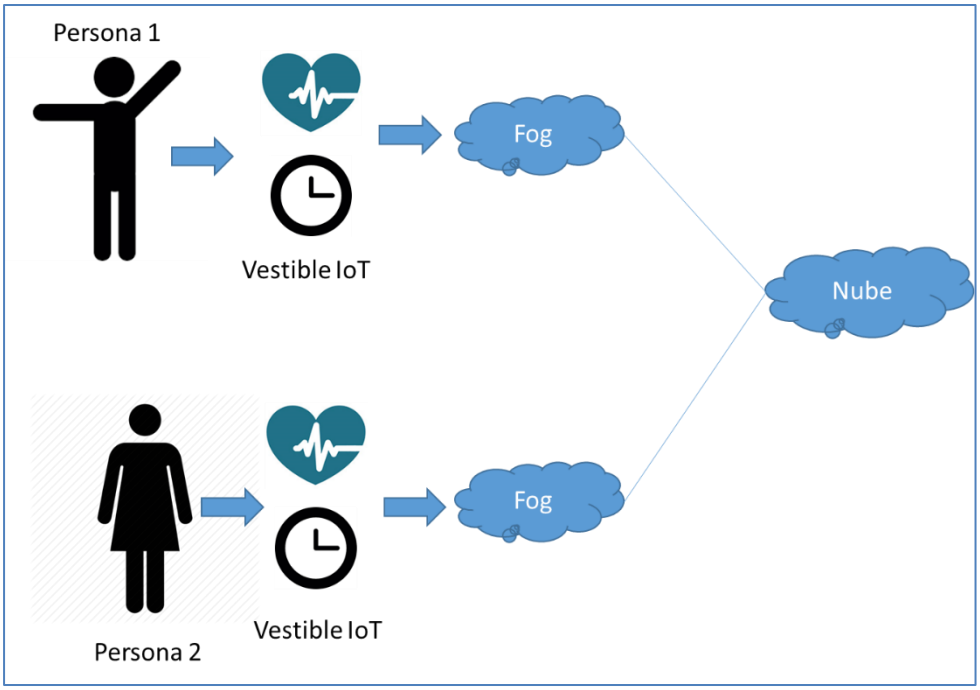

Ilustración 4, Arquitectura del IoT y conexión hacia FC. Fuente: Elaboración propia

Las personas llevan sus dispositivos IoT wearable consigo, pero éstos deben hacer una conexión con FC, para luego hacer la conexión hacia la nube y así transportar datos como temperatura, pulsaciones, estado del azúcar, ritmo cardiaco, entre otros.

2.1 Determinación del método de transporte

En consideración que, acorde a la ISO [43], los dispositivos vestibles hacen parte de las redes personales o PAN (por sus siglas en ingles Personal Area Network), es necesario que la tecnología de comunicación tenga un cubrimiento en distancia no mayor a 10 metros, esto, debido a la posibilidad de interceptación, que puede darse del tráfico como lo indica Yoom y Kim [47], cuando se irradia cierta tecnología, así mismo, sería la distancia para conectarse a FC (evitando enviar la información hasta la nube sin protección).

Acorde a la corta distancia requerida por el método de transporte para el proyecto, entre los métodos más usados de radiofrecuencia para la comunicación en smartphones e IoT se tendrán en cuenta NFC, RFID y Bluetooth, dada las siguientes características en distancia ilustradas en la tabla 2

Tecnología	Distancia máxima
NFC	10 cm
RFID	100 cm
Bluetooth	10 mts – 100 mts

Tabla 2, Tecnología en las redes PAN. Fuente: elaboración propia acorde al marco teórico

Con base en lo anterior, se evaluaron los siguientes aspectos, teniendo en cuenta que son los más relevantes para los temas IoT (dado su uso limitado):

1. Compatibilidad: Hace referencia a que la tecnología sea usada por diferentes dispositivos con Android, lo cual daría escalabilidad considerada en éste proyecto.
2. Consumo de energía: Tasa promedio de consumo cuando se está en una transmisión.
3. Procesamiento: Tasa promedio de uso de CPU cuando se procesa
4. Seguridad: Niveles de seguridad que puede soportar.

En la variable de seguridad solo se tuvo en cuenta la probabilidad de ser interceptada la comunicación entre los dispositivos wearables y FC, en el método propuesto no se tiene en cuenta el aseguramiento entre dichos extremos, es decir, no hace parte alcance del proyecto como se indicó en la introducción, solo los datos que van en tránsito desde FC hasta la nube.

A continuación, se detalla los mecanismos de medición de las 4 variables definidas.

2.1.1 Definición Compatibilidad

Para definir la compatibilidad de las tecnologías de comunicación, se revisó el porcentaje en el que estas pueden ser activados o aprovisionados en los dispositivos con Android. En la tabla 3 se describe la puntuación entregada por el porcentaje de aprovisionamiento en el mercado Android.

En consideración al uso de la versión de este sistema operativo, se definió como parámetro el aprovisionamiento que hay en el mundo (datos arrojados por el proveedor Google, ilustración 6), con ello, si la versión tiene un alto uso y soporta dicha tecnología, quiere decir que existe una gran compatibilidad y posibilidad de utilizarla, por lo cual, tendría el mayor valor en la calificación.

Puntuación de compatibilidad	**Porcentaje de aprovisionamiento en el mercado Android**	
3	Muy aprovisionado	Superior 50% de Android
2	Medianamente aprovisionado	Entre 25%-50%
1	Poco aprovisionado	Menor 25%

Tabla 3, Puntuación de compatibilidad definida por los autores. Fuente autores

2.1.2 Consumo de energía

En consideración con [48] y [49], los puntos son asignados considerando que las tecnologías IoT se caracterizan por el bajo consumo, por lo cual, si la tecnología consume mucha energía, la solución de conectividad en este aspecto se puede ver afectada por el tiempo promedio de uso.

Para evaluar el consumo de energía de una tecnología con respecto a otro, se toma el consumo individual y se ordena de mayor a menor, la tecnología que tenga menor consumo se le asignan 3 puntos y la de menor consumo 1 punto.

En la tabla 4 se representa la puntuación entregada de acuerdo con el consumo de energía por tecnología. En caso tal que el porcentaje de consumo sea igual, se asignará el mismo puntaje

Puntuación por consumo de energía	**Consumo de energía**
3	Tecnología con menor consumo
2	Tecnología con consumo medio
1	Tecnología con mayor en consumo

Tabla 4, Puntuación consumo de energía. Fuente: Autores

2.1.3 Evaluación del procesamiento

Como se indicó en [50] y [51], el uso del procesador es fundamental en IoT, toda vez que esto permite más interacciones y ejecutar más procesos, obteniendo con ello más eficiencia operativa.

Para evaluar el consumo de procesamiento de una tecnología con respecto a otra, se hizo de la misma forma que con la energía, se mide el consumo de procesamiento, se ordena de mayor a menor y quien tenga más uso de procesador se le asigna 1 punto, quién tenga menos uso de procesador se le asigna 3 puntos.

Para la prueba de consumo de procesamiento se usó una placa de Arduino UNO, esto, debido a la practicidad en su uso y configuración, costo, documentación y facilidad para incorporar elementos como los emisores NFC, RFID y Bluetooth. Así mismo, poder homologar las mediciones desde un mismo componente y no hacer uso de varios dispositivos que dificultan la comparación de medidas.

Medición del consumo

En la tabla 5 se describe la puntuación entregada por el porcentaje del consumo de tiempo de procesamiento al transportar un archivo de 1022 byte, para lo cual, se toma la medición en las diferentes tecnologías y se ordena de mayor a menor, quién tenga mayor procesamiento para enviar el mismo archivo se le asigna un punto, quién tenga mejor rendimiento se le asigna tres puntos.

Puntuación	Referido al tiempo de ejecución (milisegundos)

3	Bajo procesamiento
2	Mediano procesamiento
1	Mucho procesamiento

Tabla 5, Puntuación de procesamiento. Fuente: Autores

2.1.4 Evaluación de la seguridad

En consideración a la seguridad, ésta se midió sobre la posibilidad de sufrir una interceptación dado el rango de distancia que cada tecnología opera [34] [47] [54], basado en la protección de la misma tecnología, en la tabla 6 se muestra la definición respectiva de los alcances. Dichos rangos de comunicación (descritos en los respectivas RFC y por los proveedores [1], [9]) posibilitan o no la interceptación de la comunicación, es claro que, si la tecnología irradia con más cobertura, hay más probabilidad de interceptación por parte de terceros.

Tecnología inalámbrica	Rango máximo de comunicación en metros
NFC	0.1
RFID	1
Bluetooth	100

Tabla 6, Rangos de comunicación por tecnología. Fuente autores

En la tabla 7, se indica la puntuación por rango de comunicación. Con la misma estrategia, se ordena de mayor a menor el rango de comunicación, quién tenga menor rango se le asigna un puntaje de 3, dada las posibilidades reducidas que un atacante pueda interceptar el tráfico, así mismo, quién tenga mayor cobertura se le asigna un punto, dada la posibilidad más amplia de que sea interceptado por personas maliciosas

Rango de comunicación en metros	Probabilidad de sufrir intercepta-ción	Puntuación
Menor a 1 metro	Menor	3
De 1 a 5 metros	Media	2
Mayor a 5 Metros	Alta	1

Tabla 7, Puntuación por rango de comunicación. Fuente autores

2.1.5 Resultados finales

Para obtener los resultados finales y así seleccionar la tecnología de transporte más indicada, se hace una sumatoria de todos los puntos obtenidos en cada prueba o evaluación, la tecnología que obtenga mejor puntaje es la seleccionada para ser usada como mecanismos de transporte entre el IoT y la nube. Tener el mejor puntaje indica que dicha tecnología cumple con más características que las demás.

2.2 Determinación del algoritmo de cifrado

Para esta determinación se tuvieron en cuenta las siguientes variables, ya que son limitantes propias de los dispositivos IoT como ya se indicó [5][6]:

- Nivel de riesgo
- Consumo de procesamiento
- Consumo de energía
- Tiempo ejecución
- Consumo de ancho de banda

Con el objetivo de valorar el algoritmo de cifrado sobre la tecnología de comunicación, se desarrolló una aplicación móvil para el sistema operativo Android llamada PruebasAlgoritmosCifrado (ver ilustración 15), donde se implementan dichos algoritmos (El código fuente se encuentra en los anexos); la selección de los algoritmos fue acorde al estado del arte, donde se indica que son populares y actualmente se consideran robustos:

- AES-256
- RSA 2048
- DHEC
- HJP

Cada una de las variables a medir en cada algoritmo de cifrado se les otorga un sistema de puntos igualitario, esto es, cada variable tendrá como máximo 25 puntos (para un total de 100 en la medición total), con la siguiente distribución y asignación, esto a consideración de los autores y a las limitantes propias de IoT:

- 25 puntos: El nivel de riesgo para compartir las llaves, dado que se trata de envió de información hacia redes expuestas. Si las llaves deben compartirse esto genera un punto de riesgo adicional, por lo cual, el algoritmo de cifrado tendrá menos puntaje
    - Quien tenga mejor nivel de seguridad, obtiene 25 puntos.
    - Los otros, obtendrán 10 puntos.
- 25 puntos: Consumo de procesamiento. Se ordenan los resultados de menor a mayor y se asignan los siguientes puntos
    - Quien tenga menor consumo de procesamiento, 25 puntos.
    - En segundo en procesamiento: 20 puntos.
    - Los demás: 10 puntos.
- 25 puntos: Consumo de energía. Se ordena de menor a mayor el consumo de energía y se asignan los siguientes puntos, la definición de alto, medio y bajo es tomada directamente con del módulo profile del IDE Androd Studio (en el cual se hizo la medición):
    - Consumo bajo: 25 puntos.
    - Consumo medio: 15 puntos
    - Consumo alto: 5 puntos
- 25 puntos: Tiempo de ejecución en el envío. Se ordena de menor a mayor el tiempo de envió de los datos, asignando de la siguiente forma:

- Quien tenga menor tiempo de ejecución, 25 puntos.
- El segundo en tiempo de ejecución: 20 puntos.
- Los demás: 10 puntos.

Nota: para cada tecnología se hace 3 mediciones, se toma el tiempo de éstas y se obtiene el promedio de las 3.

- 25 puntos: Consumo de ancho de banda (Cantidad de datos enviados después de cifrado): esto influye en el uso de banda ancha. Se valora el consumo de ancho de banda acorde al envió de datos, si los algoritmos tienen igual consumo, se les asignan los mismos puntos de la siguiente forma:
  - 25 puntos para poco consumo
  - 15 puntos para alto consumo

En consideración que tener el total de los puntos (100) sería el valor ideal de un algoritmo de cifrado, el mejor algoritmo será aquel que obtenga el mayor puntaje sumando todos los puntos individuales de las pruebas.

2.3 Diseño del método y desarrollo del prototipo

Es necesario establecer algunas premisas para el diseño:

- Se da por hecho que la llave privada se encuentra asegurada
- Se da por hecho que cada uno de los componentes no tienen vulnerabilidades reportadas
- Se da por hecho que el servicio web expuesto por el Web server es seguro y se tienen un control de autenticación.

Entendiendo que FC tiene la limitante de cobertura y así brindar protección, se diseñó un método para solucionarlo. Los dispositivos Wearables con limitaciones de cómputo deben enviar la información a través de la tecnología NFC a un Smartphone o tableta de alta gama MFC (Mobile Fog Computing), el cual debe acompañar a los wearables en el momento que se deseé salir de un área protegida. Una vez la información se encuentre en el MFC, esta será asegurada; es decir, debe ser cifrada por un algoritmo criptográfico (el cual se seleccionó acorde al proceso de medición, capítulo 3 en el ítem 3.2), luego de esto será enviada a la nube en la cual se encontraría almacenada con un cifrado en reposo y puede ser consumida desde otro punto y descifrada. A continuación, en la ilustración 5 se representa el flujo del prototipo

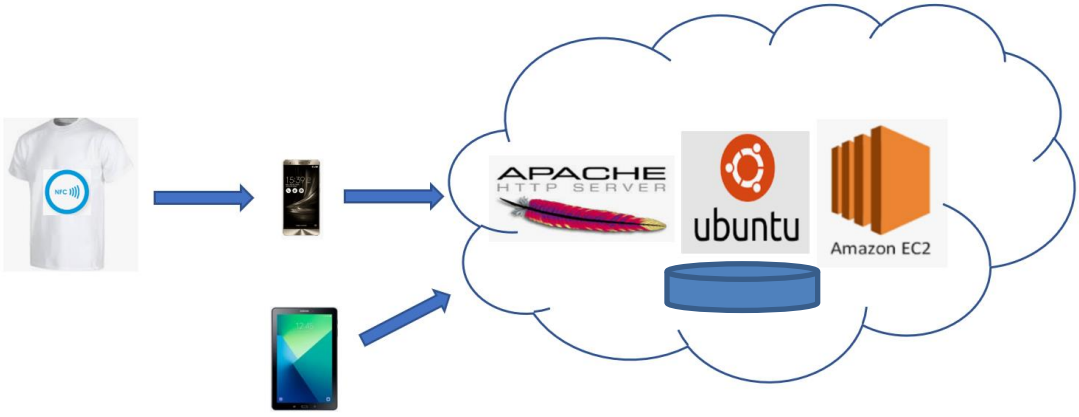

Ilustración 5, Funcionamiento del prototipo. Fuente: Elaboración propia.

A través de dispositivo vestible se enlaza con la FC y ésta establece un mecanismo de seguridad hacia la nube, para poder enviar los datos recolectado por el IoT.

Selección del MFC

En el mercado existen múltiples equipos móviles de alta gama que cuentan con grandes capacidades de procesamiento, para este método es indispensable que soporte la tecnología que ha sido seleccionada, se recomienda el uso de equipos con procesadores mínimo de cuatro núcleos y con velocidades de 2GHz, además el teléfono debe contar con al menos 2GB de RAM. Todo esto para que la experiencia del usuario no se vea afectada y el servicio de protección que se ofrece en el MFC funcione de manera adecuada.

En la tabla 8 se pueden observar el listado de alguno de los dispositivos que se recomiendan para la implementación de este método, dado que cumplen con los requerimientos anteriormente mencionados.

Smartphone	Procesador
Asus Zenfone 3 deluxe	Quad-core 2.2 GHz
Samsumg Galaxy-s9	Octa-Core 2.8GHz
Huawei mate 10 pro	Octa-core 2.4 GHz
Xiaomi Pocophone F1	Octa-core 2.8 GHz

Tabla 8, Posibles nodos FC móviles. Fuente autores

2.4 Evaluación de la capacidad de proteger la privacidad al extender FC con el método planteado

Para la evaluación del modelo, se interceptó tráfico desde el MFC hacia el servidor de AWS (Amazon Web Services), validando que la información que viaja a través de un canal inseguro no pueda ser develada a personas sin autorización una vez aplicado el modelo. Se hace uso de AWS considerando funcionalidad, la facilidad de implementación y la posibilidad de tener el servicio en modo free por un tiempo determinado.

La verificación contó con dos etapas: La primera una interceptación sin el modelo implementado y la segunda, con todo el modelo implementado, verificando así el funcionamiento y resultados.

Para el desarrollo de la prueba se usaron las siguientes herramientas técnicas:

- Servidor de Gateway: Se configuró un servidor Windows para permitir la salida a internet al MFC.
- Wireshark en su última versión: Se utilizó como sniffer permitiendo interceptar los paquetes enviados a la nube.
- MyNCF (Aplicación prototipo desarrollada en el objetivo 3): App desarrollada en Android que protegió la información de la trama.
- Google Chrome: Navegador de internet que permitió la visualización de la protección de la información.
- Putty: Herramienta para la conexión vía ssh hacia las instancias EC2 de AWS y como herramienta visualización de la protección de la información en reposo.
- PingTools: herramienta de visualización de la configuración de red de los dispositivos Android.

De igual forma, en consideración que la prueba implica el envío de información que puede ser sensible, se hizo uso de cadenas simuladas de datos que pudiesen serlo, con ello, evitar cualquier pérdida de confidencialidad en datos reales. Los detalles de la trama y los resultados se encuentran en el numeral 3.3 del presente documento.

3. Resultados

A continuación, se establece por cada fase los resultados objetivos de acuerdo con el diseño metodológico.

3.1 Tabulación de la información y selección de la tecnología de comunicación

3.1.1 Evaluación de la compatibilidad de la tecnología de comunicación

La compatibilidad de la tecnología se determinó con base el porcentaje de aprovisionamiento en el mercado para los dispositivos Android de cada tecnología. Google en su IDE de desarrollo oficial *Android Studio* proporciona una imagen donde se muestra el porcentaje de dispositivos que soportan un nivel de las APIS liberadas, Ver Ilustración 6.

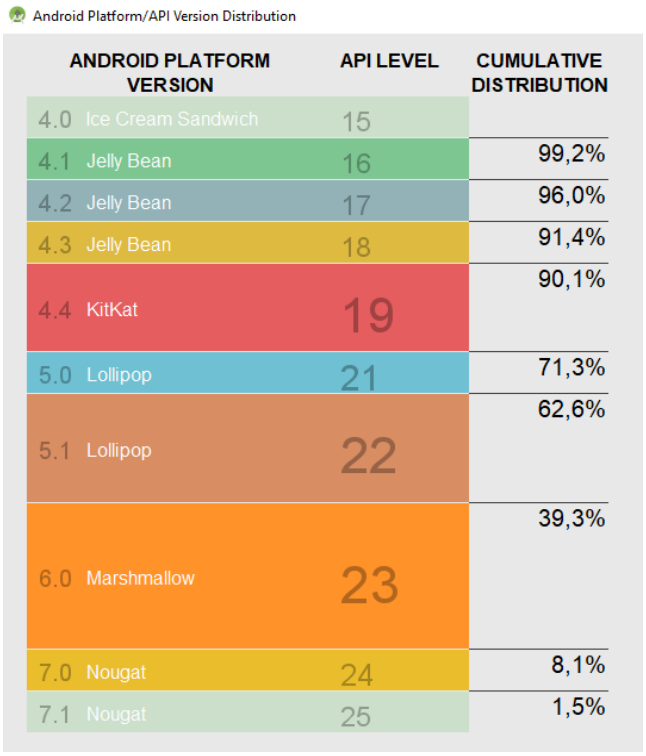

Ilustración 6, API version distribution, tomado de IDE Android Studio Rama Canarian (dato obtenido en agosto 2018)

Medición de la compatibilidad

Se da el máximo valor (3) a la tecnología más aprovisionada (usada) o que lleva más tiempo aprovisionada en los dispositivos Android, y se da menor valor (1) a aquella tecnología menos aprovisionada o que tiene menos tiempo en el mercado, para lo cual:

- NFC está disponible desde el API 9 de Android [25], por lo cual podemos decir que es medianamente aprovisionado. Esta tecnología obtuvo puntuación 2 en esta evaluación
- Mientras que RFID no es aprovisionado en los dispositivos con Android por eso obtuvo una puntuación 1.
- Bluetooth está disponible desde el API 5 de Android [26], con lo cual podemos decir que muy aprovisionado. Esta tecnología obtuvo puntuación 3 en esta evaluación.

3.1.2 Evaluación del consumo de energía de la tecnología de comunicación

Medición del consumo

En la tabla 9 se describe el consumo promedio en miliamperios - mA, dichos datos fueron obtenidos acorde a la información de los diferentes proveedores con respecto [1][9] a la tecnología usada.

Tecnología inalámbrica	Consumo de energía en mA
NFC	10
RFID	10
Bluetooth	100

Tabla 9, Consumo de energía por tecnología. Fuente autores

Al ordenar de menor a mayor consumo, se obtienen los siguientes puntos:

- consumo de Energía NFC = 3 puntos
- consumo de Energía RFID = 3 puntos
- consumo de Energía Bluetooth = 1 punto

3.1.3 Evaluación del consumo de procesamiento tecnología de comunicación

Escenario de pruebas

Para realizar las pruebas de estas tecnologías se hizo uso de Arduino, la cual es una es una plataforma electrónica de código abierto basada en hardware y software de fácil uso [27].

Para las pruebas de las tecnologías Bluetooth, RFID y NFC se utilizó la placa Arduino UNO (ver ilustración 7), debido a que cuenta con la capacidad para incorporar elementos como los emisores NFC, RFID y Bluetooth. Así mismo, poder homologar las mediciones desde un mismo componente y no hacer uso de varios dispositivos que dificultan la comparación de medidas.

Nota: El código Arduino queda como anexo al documento de tesis y los módulos utilizados no son mandatorios, se pueden usar otros del mercado que sean compatibles con las placas Arduino.

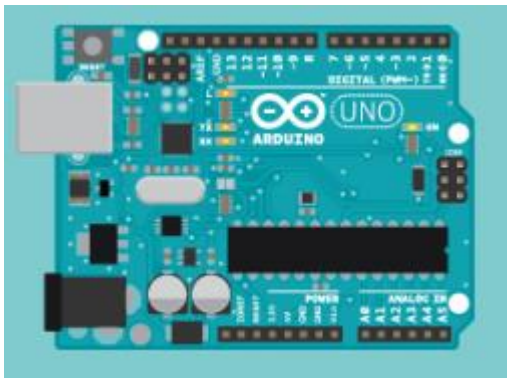

Ilustración 7, Arduino UNO [27]

Evaluación del tiempo de procesamiento en NFC

Para la ejecución de la evaluación de la tecnología NFC se utilizó el módulo PN532 (ver ilustración 8), dado que éste cuenta con comunicación serial, la cual es compatible con la placa Arduino UNO y soporta la comunicación con la tecnología mencionada.

Ilustración 8,Módulo PN532 [28]

Además, se desarrolló un programa en Arduino basado en las librerías publicadas por Elechouse en GitHub [29], el cual leerá un archivo de 1022 bytes almacenado en un TAG NFC NXP MIFARE de 1 KB.

En la ilustración 9 se observa el resultado de la transferencia de la información a través de la tecnología NFC, la cual tardó 4 milisegundos para transferir el archivo de 1022 bytes.

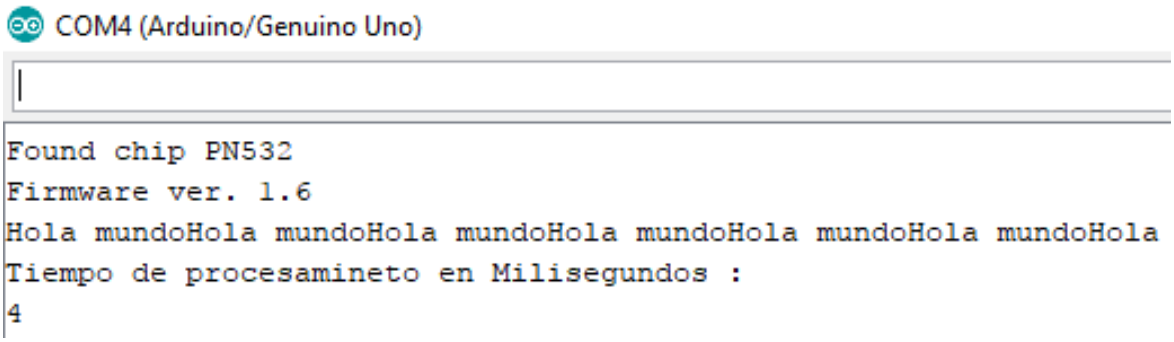

Ilustración 9, Comunicación entre PN532 y ARDUINO. Fuente autores

Evaluación del tiempo de procesamiento en RFID

Para la ejecución de la evaluación de la tecnología RFID se utilizó el módulo RC522 (Ver ilustración 10), dado que este cuenta con comunicación serial, la cual es compatible con la placa Arduino UNO y soporta la comunicación con la tecnología mencionada

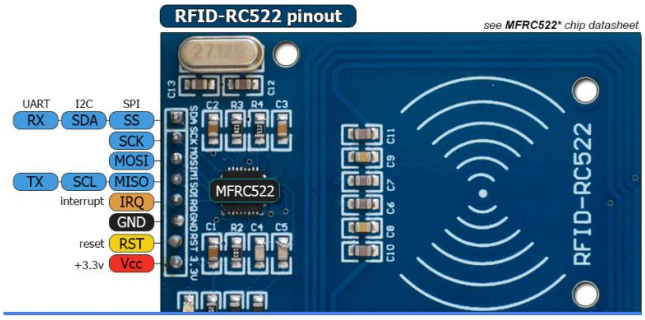

Ilustración 10, Módulo RC522 [30]

Además, se adecuó un programa en Arduino basado en las librerías publicadas por Miguel Balboa en GitHub [31], el cual leerá un archivo de 1022 bytes almacenado en una tarjeta RFID MIFARE classic de 1 KB.

En la ilustración 10 se observa el resultado de la transferencia de la información a través de la tecnología RFID. La cual tardó 456 milisegundos para transferir el archivo de 1022 bytes

COM4 (Arduino/Genuino Uno)

```
Ingrese 1 para Escribir y 2 para Leer
Preparando para Leer acerque la tarjeta
**Card Detected:**
PCD_Authenticate() success:
hola mundohola mundohola mundohola mundohola mundohola mundohola mundc
Milisegundos:
456
```

Ilustración 11,Comunicación entre RC522 y ARDUINO

Evaluación del tiempo de procesamiento en Bluetooth

Para la ejecución de la evaluación de la tecnología Bluetooth se utilizó el módulo HC05 (ver ilustración 12), dado que este cuenta con comunicación serial, la cual es compatible con la placa Arduino UNO, soporta la comunicación con la tecnología mencionada y no tiene restricción para la comunicación con el sistema operativo Android.

Ilustración 12,Módulo HC05 [32]

A demás se creó una app en Android llamada PruebaBluetooth (ver ilustración 13), el cual está hecho usando las APIs del paquete android.bluetooth, desde esta se envía un archivo de 1022 bytes hacia la placa Arduino (el cual es recibido por el módulo HC05), en esta se toma el tiempo de procesamiento de la transferencia de la información.

Nota: El código de la app PruebaBluetooth tiene soporte desde la versión 8.0 de Android, pero puede ser compilada a una versión inferior que soporte las APIs utilizadas, todo el código se encuentra adjunto en los anexos.

En la ilustración 13 se puede observar la vinculación del dispositivo HC05 al teléfono Android

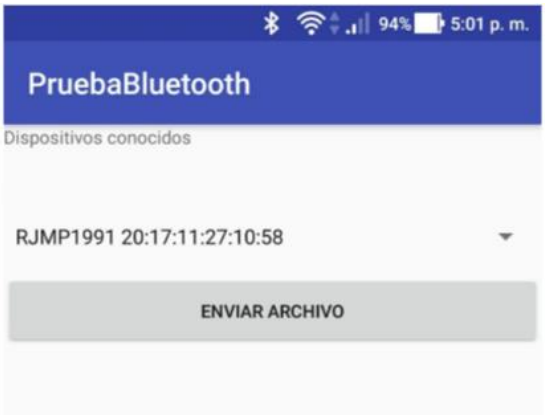

Ilustración 13, Vinculación de HC05 con App PruebaBluetooth. Fuente autores.

En la ilustración 14 se puede observar el resultado de la transferencia de información a través de la tecnología Bluetooth. la cual tardó 561 milisegundos para transferir el archivo de 1022 bytes.

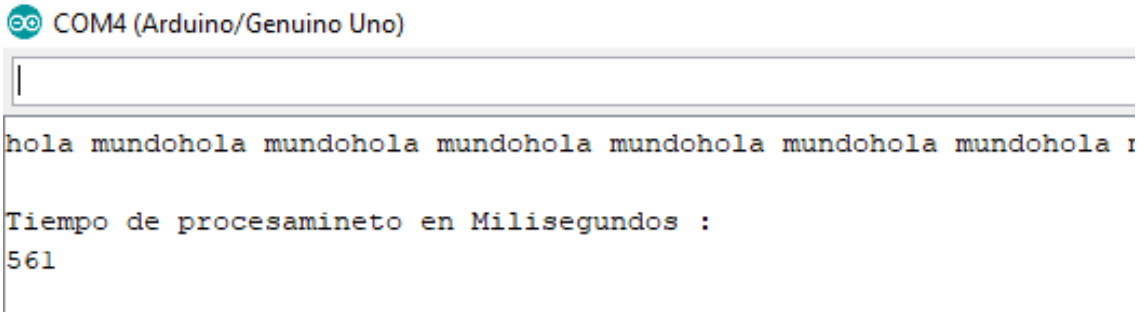

Ilustración 14, Comunicación entre HC05 y ARDUINO. Fuente autores.

Dadas las mediciones anteriores, el tiempo de procesamiento en la transferencia por cada tecnología es el siguiente:

NFC = 4 milisegundos
RFID = 456 milisegundos
Bluetooth = 561 milisegundos

Según la escala de puntuación definida en la tabla 5, la puntuación por tecnología es la siguiente:

NFC = 3 puntos
RFID = 2 puntos
Bluetooth = 1 punto

3.1.4 Evaluación de la seguridad de la tecnología de comunicación

En consideración a la metodología aplicada, la puntuación obtenida por cada una de las tecnologías evaluadas (tabla 10), se dará máximo valor a la tecnología que tenga menor probabilidad de ser interceptada (dada la distancia de irradiación).

Puntuación	Descripción
1	Menor valor, asociado a la probabilidad de que la tecnología de comunicación inalámbrica pueda ser interceptado, esto, con base en la distancia relativa de operación para la comunicación.
2	Valor Medio, asociado a la probabilidad de que tecnología de comunicación inalámbrica pueda ser interceptado, esto, con base en la distancia relativa de operación para la comunicación.
3	Mayor valor, asociado a la probabilidad de que la tecnología de comunicación inalámbrica pueda ser interceptado, esto, con base en la distancia relativa de operación para la comunicación.

Tabla 10, Descripción de la asignación de puntos en la evaluación de seguridad. Fuente autores

De acuerdo con la tabla 6 se NFC tiene menor probabilidad de sufrir interceptación, ya que su rango de comunicación es hasta 0,1 Metros, Mientras RFID tiene probabilidad media debido a que su rango de comunicación es hasta de 1 metro, Por último, Bluetooth tiene una probabilidad alta a causa de que su rango de comunicación es hasta 100 metros.

Por lo tanto, se asigna la siguiente puntuación de acuerdo con la tabla 10:

NFC: 3 puntos
RFID: 2 puntos
Bluetooth: 1 punto

3.1.5 Resultados finales consolidados

Acorde a la metodología, se realizó una suma de todos los puntos obtenidos en las diferentes pruebas y evaluaciones de las tres tecnologías, la tecnología que obtuvo mayor puntaje fue aquella seleccionada a ser implementada. En la tabla 11 se puede observar el consolidado y el resultado final.

Tecnología	Compatibilidad	Consumo de energía	Procesamiento	Seguridad	Resultado
NFC	2	3	3	3	11
RFID	1	3	2	2	8
Bluetooth	3	1	1	1	6

Tabla 11, Tabulación de pruebas tecnología de comunicación. Fuente autores.

Luego de las mediciones obtenidas y recopiladas en la tabla 11, **NFC** es la mejor tecnología calificada para el transporte de la información desde los wearables y FC. Dado que su consumo de energía y procesamiento son bajos, la probabilidad de sufrir interceptación es baja y su compatibilidad en los dispositivos Android es alta.

3.2 Tabulación de la información y determinación del algoritmo de cifrado

Se puede evidenciar en ilustración 15, la interfaz de la app anteriormente mencionada llamada PruebasAlgoritmosCifrado, (La cual fue desarrollado por los autores y su código se encuentra en los anexos), donde se puede seleccionar los algoritmos criptográficos mencionados, especificar la cantidad de iteraciones para cifrar el mensaje. Esto con el objetivo de simular las peticiones entrantes de los wearables.

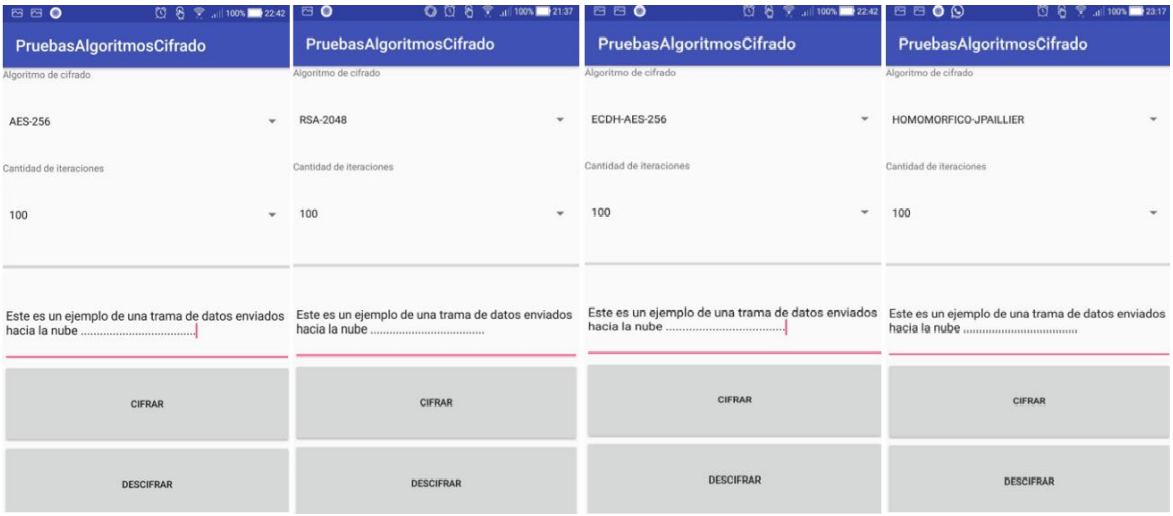

Ilustración 15, Capturas de App para las PruebasAlgoritmosCifrado. Fuente autores

3.2.1 Condiciones de la prueba

En la prueba se ejecutan la app PruebasAlgoritmosCifrado para evaluar cada algoritmo de cifrado de acuerdo con el consumo promedio de procesamiento, energía y tiempo de envío.

Para evaluar el nivel de riesgo se tuvo en cuenta si el algoritmo es simétrico o asimétrico, considerando que el envío de información es a través de canales no asegurados y no es un proceso estacionario, se ha estipulado un menor nivel de exposición para los algoritmos asimétricos, esto debido a que los simétricos requieren hacer uso de otros mecanismos de seguridad para el intercambio de la llave compartida antes de iniciar el proceso de cifrado.

Consideraciones técnicas del análisis de los algoritmos de cifrado:

- Los algoritmos de criptográficos seleccionados son: AES, RSA, DHEC, HJP.
- La app PruebasAlgoritmosCifrado se creó con el IDE de desarrollo Android Studio en la rama Canarian, la cual a la fecha su módulo Profiler cuenta con características no desplegadas a la rama comercial.
- Se enviaron 100 peticiones con cadenas en texto plano.
- La cadena enviada fue de 100 caracteres, a continuación, se muestra la cadena de texto de la prueba:

  "Este es un ejemplo de una trama de datos enviados hacia la nube ....."

- Las pruebas se realizaron en un dispositivo de gama alta, un teléfono ASUS Zenfone 3 deluxe con un procesador 4 Core, 6 GB de memoria RAM, sistema operativo Android 8. Dicho dispositivo fue seleccionado dada la disponibilidad de éste, con ello, se redujo el costo adicional que pudo haber ocasionado dicha adquisición.
- Antes de iniciar el proceso de cifrado la app PruebasAlgoritmosCifrado realiza la creación de llaves criptográficas para cada uno de los algoritmos, para lo cual, en esta medición No se tienen en cuenta ese procesamiento inicial, dado que el sistema las calcula para todo el proceso y no por cada envió de información.
- El consumo de ancho de banda está asociado a la cantidad de caracteres que cada algoritmo genera al cifrar la información. A mayor número de caracteres generados luego del cifrado, más consumo de ancho de banda.
- La red de comunicaciones utilizada en la prueba cuenta con 20 Mbps de ancho de banda.

En la ilustración 16, se visualizan las capacidades y configuración de la red y el teléfono utilizados en la prueba (condición inicial).

Ilustración 16, Consideraciones técnicas. Fuentes autores

3.2.2 Procedimiento

- Se ejecutó la app mencionada para que se habiliten los campos de texto y la función de cifrado, con ello, él envió de datos.
- En campo de texto se ingresó la cadena de texto comentada anteriormente, la cual es cifrada por cada iteración y envío a la nube de AWS, por lo cual, el proceso hace 100 subprocesos de cifrado.
- Se presionó el botón de envío por cada prueba de algoritmo.
- Para cada algoritmo se hace un resumen de los promedios obtenidos
- La medición de consumo de CPU se realizó a través del comando TOP a través de la terminal de Android, llevándolo a un archivo texto y extrayendo de éste los resultados.
- Se ilustra la evidencia del comportamiento de uso de recursos Android.
- Se ilustra el texto cifrado enviado a la nube.

Medición del algoritmo AES

Consumo promedio de energía: Bajo
Tiempo promedio de ejecución: 26.404
Consumo promedio de CPU: 31.8375

En la ilustración 17 se representa por medio del módulo profile del IDE Androd Studio, el consumo de los recursos del smartphone, desde el inicio hasta el final de la ejecución del cifrado con el algoritmo AES.

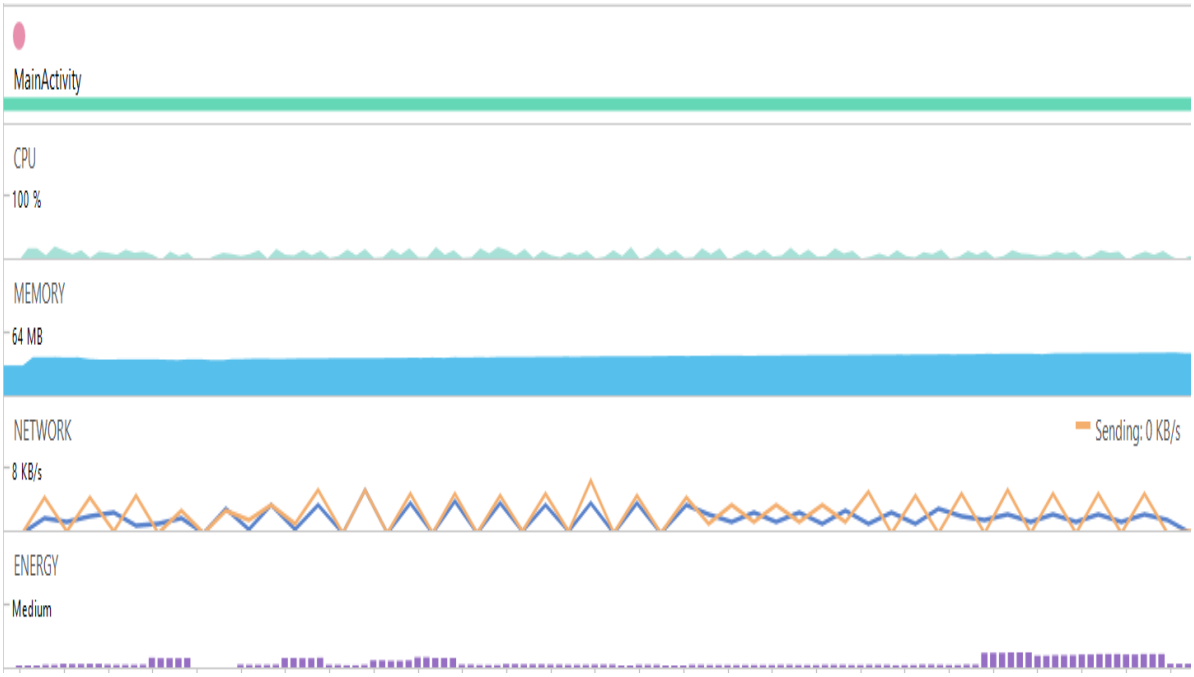

Ilustración 17, Android Profiler – AES. Fuente autores

En la Ilustración 18, se describe la ejecución del comando top y el volcado de la información a un archivo de texto para el algoritmo AES, la opción -u del comando se utilizó para filtrar solo los procesos que corren con los privilegios de un usuario especifico.

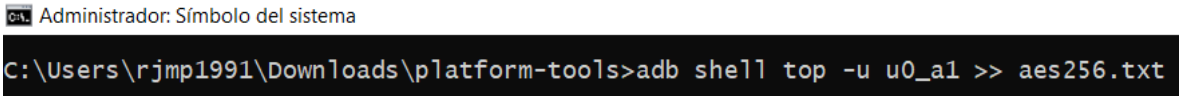

Ilustración 18, Captura comando TOP – AES. Fuente autores

En la ilustración 19, se visualiza la lectura del archivo donde se encuentran los registros de la ejecución del algoritmo AES

PID	USER	PR	NI	VIRT	RES	SHR	S	[%CPU]	%MEM	TIME+	ARGS
3895	u0_a1	10	-10	2.9G	72M	43M	S	35.0	1.2	0:04.19	rjmp1991.prueba+
3895	u0_a1	10	-10	2.9G	73M	45M	S	25.3	1.2	0:05.24	rjmp1991.prueba+
3895	u0_a1	10	-10	2.9G	74M	45M	S	31.6	1.2	0:06.00	rjmp1991.prueba+
3895	u0_a1	10	-10	2.9G	74M	45M	S	35.3	1.2	0:06.95	rjmp1991.prueba+
3895	u0_a1	10	-10	2.9G	75M	45M	R	31.6	1.3	0:08.01	rjmp1991.prueba+
3895	u0_a1	10	-10	2.9G	75M	45M	S	32.6	1.3	0:08.96	rjmp1991.prueba+
3895	u0_a1	10	-10	2.9G	76M	46M	S	29.3	1.3	0:09.94	rjmp1991.prueba+
3895	u0_a1	10	-10	2.9G	76M	46M	S	34.0	1.3	0:10.82	rjmp1991.prueba+

Ilustración 19. Lectura comando TOP – AES. Fuente autores

En la ilustración 20, se visualiza que la información cifrada por el algoritmo AES se almacenó en la nube de aws.

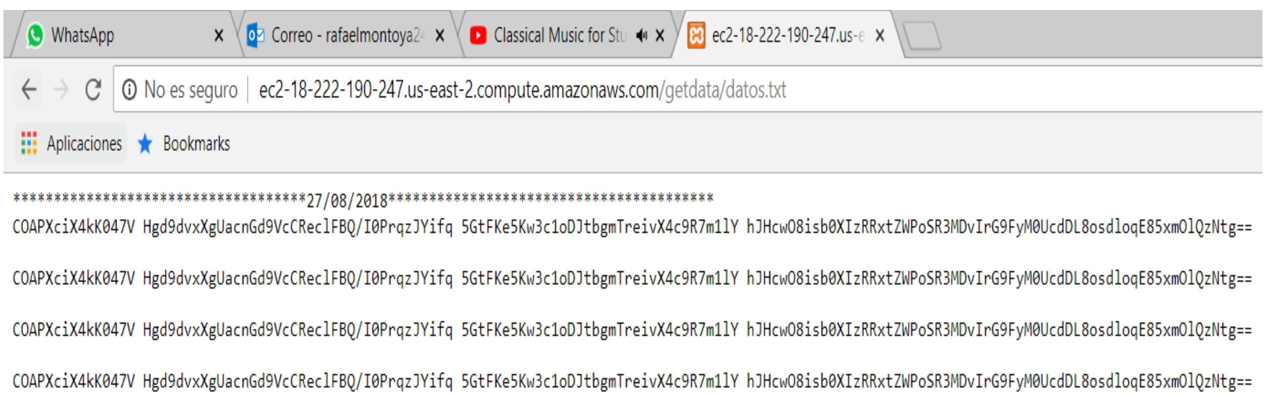

Ilustración 20, Envío de información AWS – AES. Fuente autores

Medición del algoritmo RSA

Consumo promedio de energía: Bajo
Tiempo promedio de ejecución: 27.8
Consumo promedio de CPU: 33.5875

En la ilustración 21, se representa por medio del módulo profile del IDE Androd Studio, el consumo de los recursos del smartphone, desde el inicio hasta el final de la ejecución del cifrado con el algoritmo RSA.

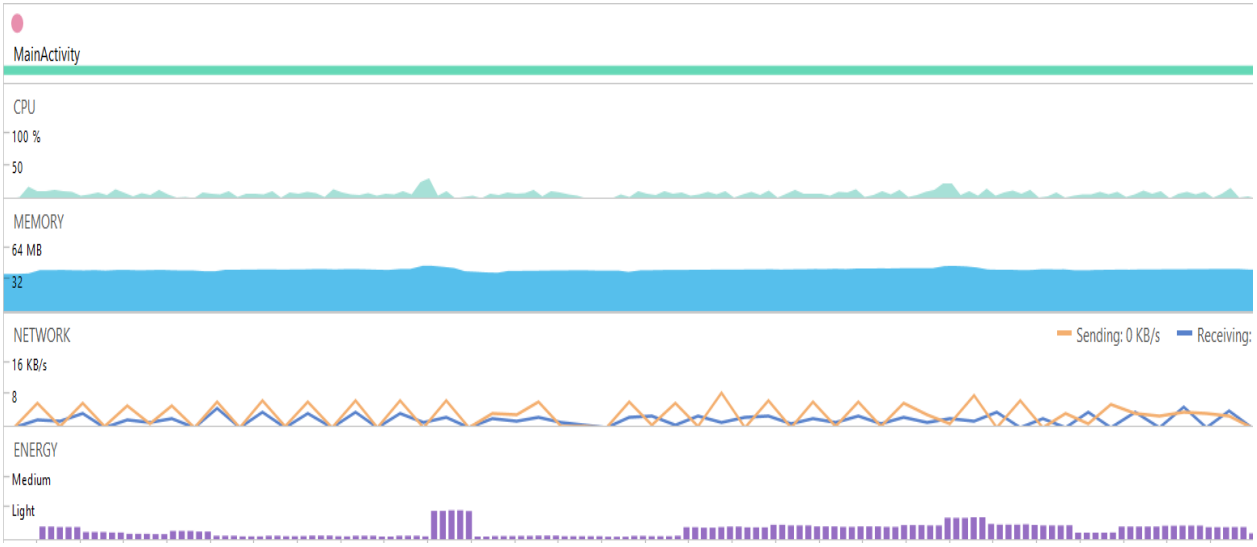

Ilustración 21, Android Profiler – RSA. Fuente autores

En la Ilustración 22, se describe la ejecución del comando top y el volcado de la información a un archivo de texto para el algoritmo RSA, la opción -u del comando se utilizó para filtrar solo los procesos que corren con los privilegios de un usuario especifico.

```
C:\Users\rjmp1991\Downloads\platform-tools>adb shell top -u u0_a1 >> rsa2048.txt
```

Ilustración 22, Captura comando TOP – RSA. Fuente autores

En la ilustración 23, se visualiza la lectura del archivo donde se encuentran los registros de la ejecución del algoritmo RSA, en ella se puede observar el proceso, consumo de CPU y tiempo de ejecución.

```
PID  USER         PR  NI VIRT  RES  SHR S[%CPU] %MEM     TIME+ ARGS
4049 u0_a1        10 -10 2.9G  74M  45M R 43.6   1.2   0:09.77 rjmp1991.prueba+
4049 u0_a1        10 -10 2.9G  75M  46M S 30.3   1.3   0:11.08 rjmp1991.prueba+
4049 u0_a1        10 -10 2.9G  76M  46M S 37.0   1.3   0:11.99 rjmp1991.prueba+
4049 u0_a1        10 -10 2.9G  77M  46M S 35.6   1.3   0:13.10 rjmp1991.prueba+
4049 u0_a1        10 -10 2.9G  76M  46M S 33.3   1.3   0:14.17 rjmp1991.prueba+
4049 u0_a1        10 -10 2.9G  76M  46M S 33.3   1.3   0:15.17 rjmp1991.prueba+
4049 u0_a1        10 -10 2.9G  77M  46M S 22.3   1.3   0:16.17 rjmp1991.prueba+
4049 u0_a1        10 -10 2.9G  77M  46M S 33.3   1.3   0:16.84 rjmp1991.prueba+
```

Ilustración 23, Lectura comando TOP – RSA. Fuente autores.

En la ilustración 24, se visualiza que la información cifrada por el algoritmo RSA se almacenó en la nube de aws.

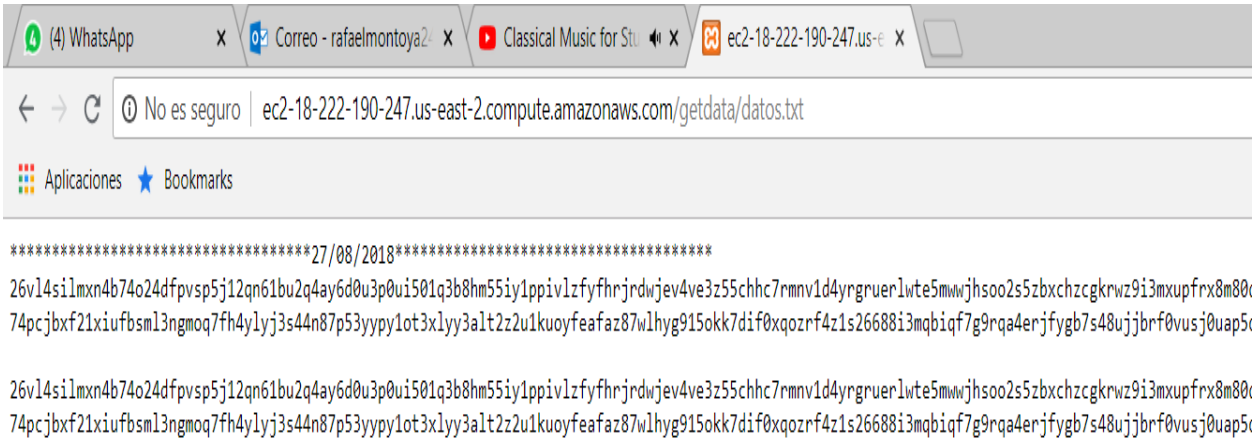

Ilustración 24, Envío de información AWS – RSA. Fuente autores.

Medición del algoritmo DHEC

Consumo promedio de energía: Bajo
Tiempo promedio de ejecución: 26.711
Consumo promedio de CPU: 33.25

En la ilustración 25, se representa por medio del módulo profile del IDE Androd Studio, el consumo de los recursos del smartphone, desde el inicio hasta el final de la ejecución del cifrado con el algoritmo DHCE.

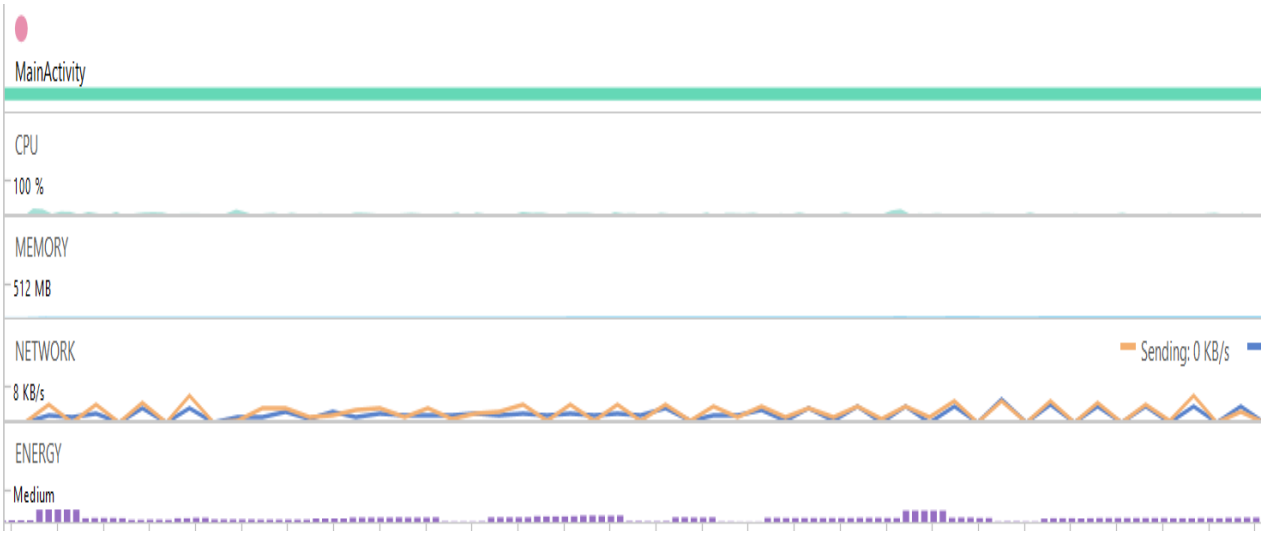

Ilustración 25, Android Profiler – DHEC. Fuente autores.

En la Ilustración 26, se describe la ejecución del comando top y el volcado de la información a un archivo de texto para el algoritmo DHEC, la opción -u del comando se utilizó para filtrar solo los procesos que corren con los privilegios de un usuario especifico

Ilustración 26, Captura comando TOP – DHEC. Fuente autores.

En la ilustración 27, se visualiza la lectura del archivo donde se encuentran los registros de la ejecución del algoritmo DHEC.

```
PID   USER         PR  NI VIRT  RES  SHR S[%CPU] %MEM     TIME+ ARGS            ESC[0m
17539 u0_a1        10 -10 2.9G  71M  43M S 32.6   1.2   0:05.87 rjmp1991.prueba+
17539 u0_a1        10 -10 2.9G  73M  45M S 34.6   1.2   0:06.85 rjmp1991.prueba+
17539 u0_a1        10 -10 2.9G  73M  45M S 34.6   1.2   0:07.89 rjmp1991.prueba+
17539 u0_a1        10 -10 2.9G  74M  45M S 35.3   1.2   0:08.93 rjmp1991.prueba+
17539 u0_a1        10 -10 2.9G  74M  45M S 34.0   1.2   0:09.99 rjmp1991.prueba+
17539 u0_a1        10 -10 2.9G  75M  46M S 35.3   1.3   0:11.01 rjmp1991.prueba+
17539 u0_a1        10 -10 2.9G  76M  46M R 29.3   1.3   0:12.07 rjmp1991.prueba+
17539 u0_a1        10 -10 2.9G  74M  46M S 30.3   1.2   0:12.95 rjmp1991.prueba+
```

Ilustración 27, Lectura comando TOP – DHEC. Fuente autores.

En la ilustración 28, se visualiza que la información cifrada por el algoritmo DHEC se almacenó en la nube de aws

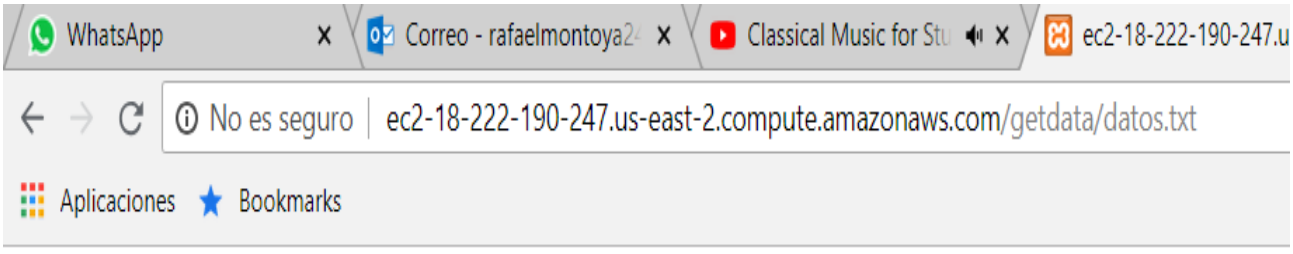

Ilustración 28, Envío de información AWS – DHEC. Fuente autores.

Medición del algoritmo HJP

Consumo promedio de energía: Bajo
Tiempo promedio de procesamiento: 30.163
Consumo promedio de CPU: 32.375

En la ilustración 29, se representa por medio del módulo profile del IDE Androd Studio, el consumo de los recursos del smartphone, desde el inicio hasta el final de la ejecución del cifrado con el algoritmo HPJ.

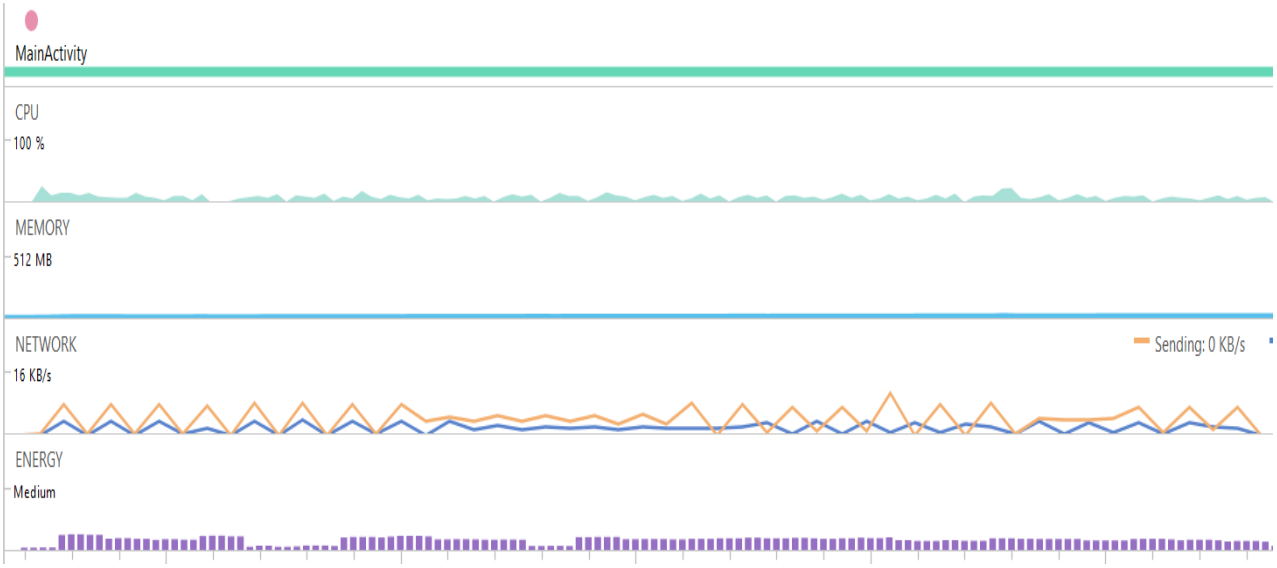

Ilustración 29, Android Profiler – HJP. Fuente autores.

En la Ilustración 30, se describe la ejecución del comando top y el volcado de la información a un archivo de texto para el algoritmo HPJ, la opción -u del comando se utilizó para filtrar solo los procesos que corren con los privilegios de un usuario especifico

```
C:\Users\rjmp1991\Downloads\platform-tools>adb shell top -u u0_a1 >> homomorfico.txt
```

Ilustración 30, Captura comando TOP – HPJ. Fuente autores.

En la ilustración 31, se visualiza la lectura del archivo donde se encuentran los registros de la ejecución del algoritmo HPJ

```
PID   USER         PR  NI VIRT  RES  SHR S[%CPU] %MEM     TIME+ ARGS            ESC[0m
23198 u0_a1        10 -10 2.9G  73M  45M S 33.6   1.2   0:08.14 rjmp1991.prueba+
23198 u0_a1        10 -10 2.9G  74M  45M S 25.6   1.2   0:09.15 rjmp1991.prueba+
23198 u0_a1        10 -10 2.9G  75M  45M S 34.3   1.3   0:09.92 rjmp1991.prueba+
23198 u0_a1        10 -10 2.9G  76M  46M S 37.3   1.3   0:10.95 rjmp1991.prueba+
23198 u0_a1        10 -10 2.9G  74M  46M S 33.3   1.2   0:12.07 rjmp1991.prueba+
23198 u0_a1        10 -10 2.9G  75M  46M S 30.0   1.3   0:13.07 rjmp1991.prueba+
23198 u0_a1        10 -10 2.9G  76M  46M S 32.6   1.3   0:13.97 rjmp1991.prueba+
23198 u0_a1        10 -10 2.9G  76M  46M R 32.3   1.3   0:14.95 rjmp1991.prueba+
```

Ilustración 31, Lectura comando TOP – HPJ. Fuente autores.

En la ilustración 32, se visualiza que la información cifrada por el algoritmo HPJ se almacenó en la nube de aws

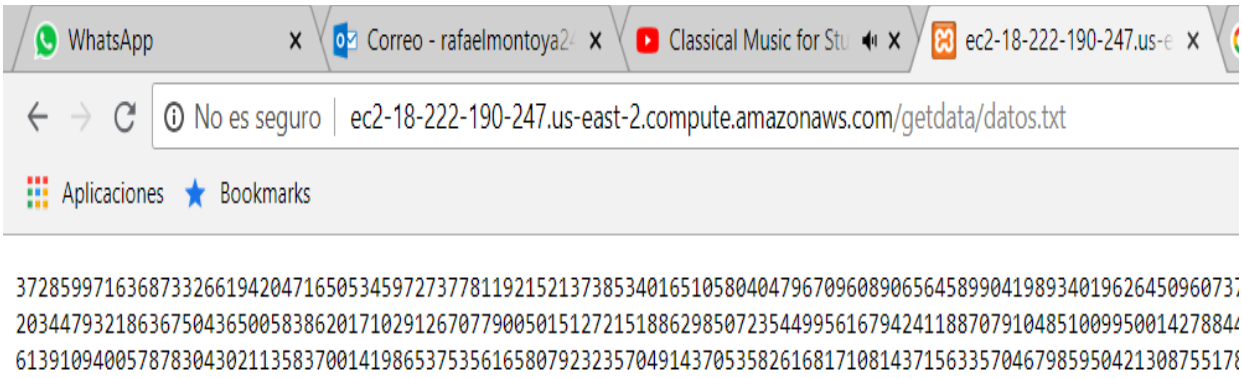

Ilustración 32, Envío de información AWS – HPJ. Fuente autores.

Aplicando la metodología para la selección del algoritmo a implementar, a continuación, en la tabla 12 se presentan los resultados consolidados de las mediciones cuando se ha enviado 100 caracteres:

Algoritmo Cifrado	Uso promedio del CPU	Consumo promedio Energía	T1	t0	Tiempo promedio de ejecución en segundos	Consumo de ancho de banda (KB/s)	Comparte llave para descrifar
AES	31.8375	Bajo	34.349 32.312 33.059	07.977 05.563 06.966	26.372 26.749 26.093 **Promedio**: 26.404	8	Si
RSA	33.5875	Bajo	30.139 37.017 44.471	2.957 8.543 16.725	27.182 28.474 27.746 **Promedio**: 27.8	8	No
DHEC	33.25	Bajo	59.913 50.908 30.610	33.341 24.142 03.814	26,572 26.766 26.796 **Promedio**: 26.711	8	No
HJP	32.375	Bajo	35.389 35.349 34.814	05.309 05.300 04.454	30.080 30.049 30.360 **Promedio**: 30.163	16	No

Tabla 12, Resultados de las mediciones de algoritmos de cifrado. Fuente autores basado en las medidas ejecutadas

3.2.3 Resultados consolidados:

En la tabla 13 se puede observar la asignación de puntos acorde a las características de los algoritmos de cifrado,

Algoritmo	Nivel de riesgo	Procesamiento	Energía	Tiempo ejecución	Consumo BW	Total puntos
AES	10	25	25	25	25	110
RSA	25	10	25	10	25	95
DHEC	25	20	25	20	25	115
HJP	25	15	25	10	15	90

Tabla 13, Tabulación de pruebas algoritmo de cifrado. Fuente autores

Teniendo en cuenta los resultados de las mediciones anteriores, se logra determinar que el algoritmo de cifrado mejor calificado para cifrar la información en FC es el DHEC, permitiendo que el método sea escalable y eficiente.

3.3 Resultados del método propuesto

Para la evaluación del método propuesto, se utilizó como dispositivo IoT wearable una camisa común, donde se adhirieron varias etiquetas NFC con información de prueba (321 bytes), simulando la captura de información del cuerpo de una persona. No se implementó el uso de un dispositivo wearable real dado los tiempos de entrega, además con la simulación se cumple con el alcance.

Las etiquetas usadas fueron tipo ISO 14443-3A de referencia NXP MifareClassic, Ndef, (Ver ilustración 33). Dichas etiquetas fueron seleccionadas ya que pueden almacenar hasta 1KB de información, es decir que cumplen con los requerimientos; pero también pueden ser remplazadas por otras del mercado.

Ilustración 33, Capturas Etiquetas NFC. Fuente autores.

El MFC seleccionado fue el smartphone Asus Zendphone 3 deluxe, dado que se contaba con dicho dispositivo y este cumple con los requerimientos, reduciendo con ellos los costos asociados a posibles compras. En el Smartphone se instaló una app llamada MyNFC desarrollada por los autores (el código fuente se encuentra en los anexos), la cual cuenta con la capacidad de capturar la información de los wearables mediante la tecnología NFC y luego enviarla a la nube. La interface gráfica de dicha app cuenta con un elemento de selección, el cual permite escoger si se desea proteger o no los datos, según sea la selección, se cifra o no la información con la llave pública generada para el algoritmo DHEC, ver ilustración 34.

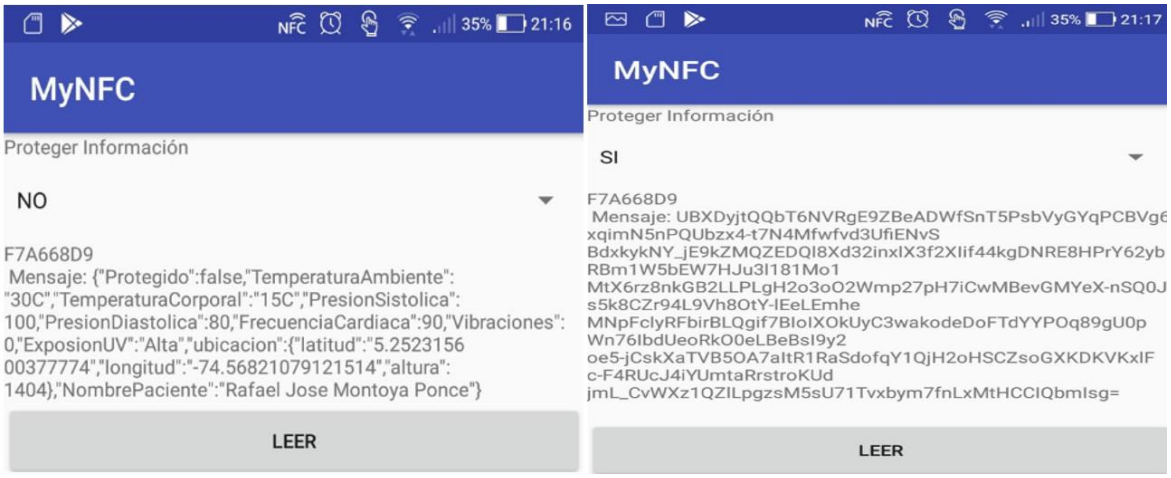

Ilustración 34, Capturas APP MyNFC. Fuente autores.

La nube seleccionada fue AWS, en la cual se instaló una instancia EC2 con el sistema operativo Ubuntu y un servidor web apache en su última versión.

A continuación, en la ilustración 35 se muestra el resultado de la ejecución de la app MyNFC con la selección de no protección de la información, los datos fueron almacenados y estilizados en un archivo HTML.

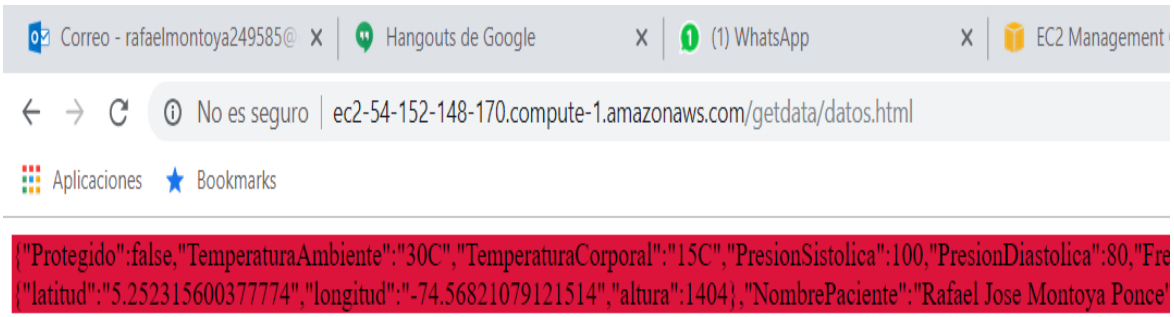

Ilustración 35, Captura envío de información sin protección. Fuente autores.

Del mismo modo, pero seleccionando la opción de proteger la información; en la ilustración 36 se muestra como los datos fueron almacenados cifrados por el algoritmo DHEC.

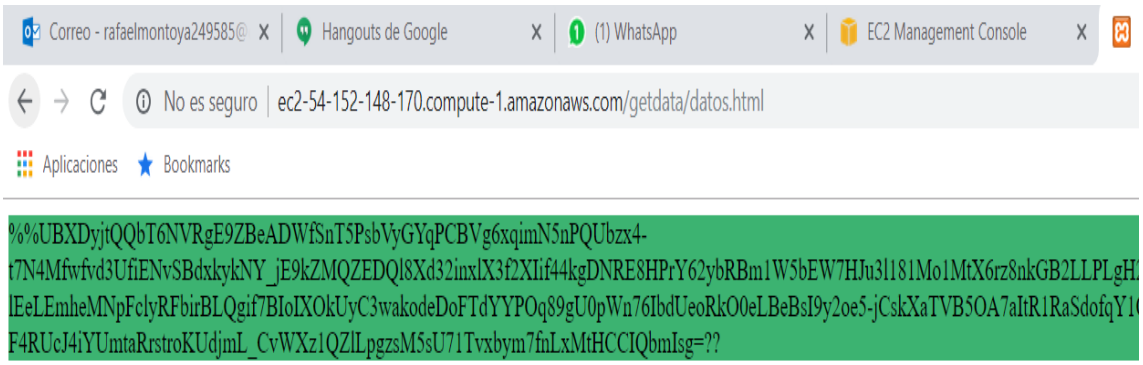

Ilustración 36, Captura envío de información protegida

Para leer la información y validar el funcionamiento del descifrado de la información, se desarrolló una app llamada ServerNFC (el código fuente se encuentra en los anexos), la cual se instala en otro dispositivo, este solicita la información al servidor apache y luego es descifrada usando la llave privada.

En la ilustración 37 se puede visualizar que, desde el emulador de Android Studio se ejecutó la app ServerNFC (esta posee la llave privada del algoritmo DHEC), para descifrar la información que se encontraba almacenada en la nube.

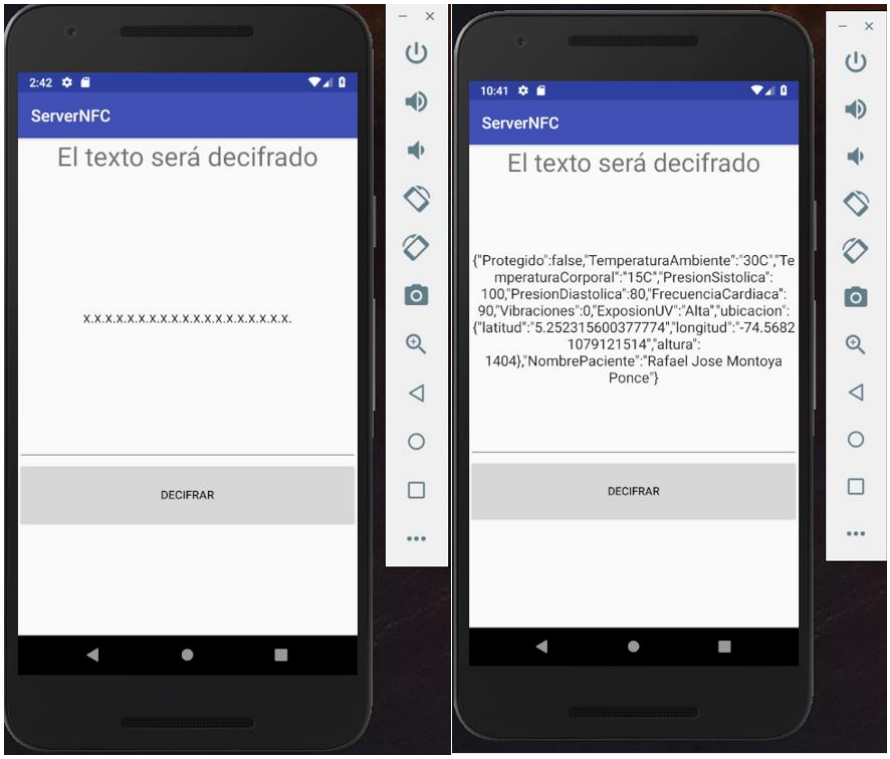

Ilustración 37, Capturas APP ServerNFC. Fuente autores.

3.4 Evaluación de la capacidad de proteger la privacidad al extender FC con el método planteado

Para la evaluación de la capacidad del método, se realizaron diferentes configuraciones de red en el MFC, el servidor en nube y el servidor Gateway.

En la ilustración 38, se describe la configuración de red del servidor que se encuentra alojado en la nube, los datos más importantes son la dirección IPv4 pública y el Registro DNS público; dado que por medio de estos puede ser alcanzado el servidor.

Como se ha indicado, para la evaluación se ejecutó en 2 fases:

1. Fase 1: Transferencia de datos sin cifrados desde el Wearable hasta la nube
2. Fase 2: Transferencia de datos con cifrado.

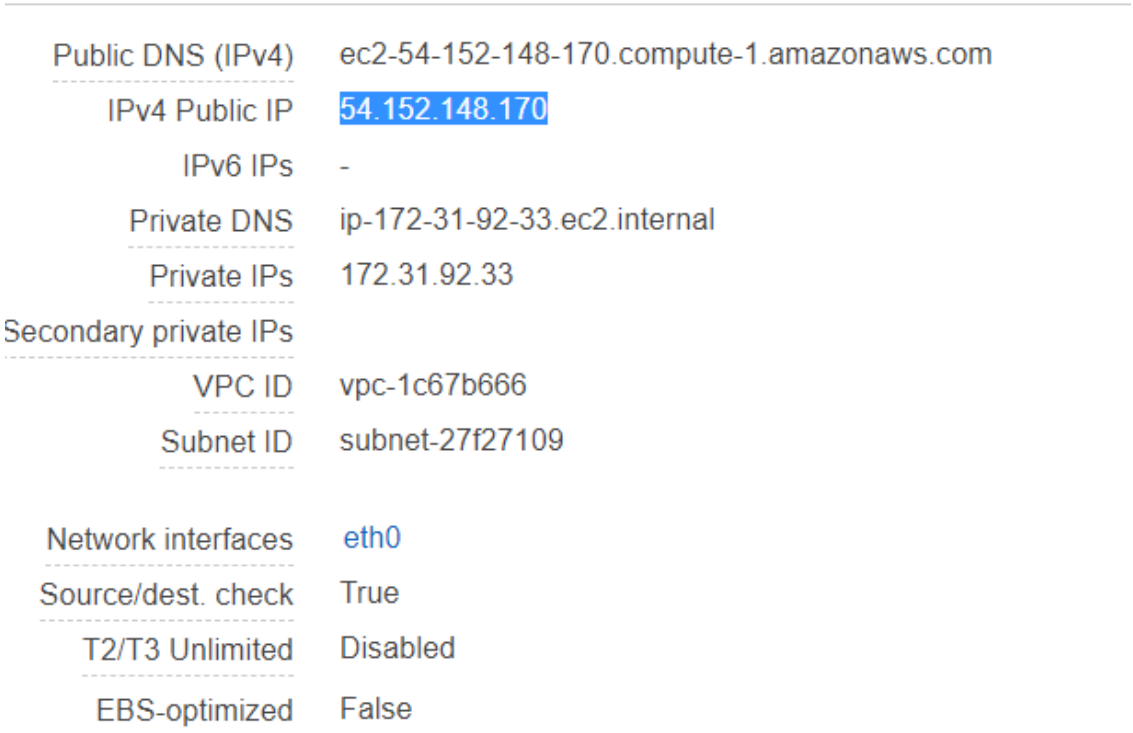

Ilustración 38, Configuración de la red de la maquina Ubuntu en la nube de AWS. Fuente autores

En la ilustración 39, se describe la configuración de red del MFC, los datos más importantes son la dirección IPv4 local, en este caso 192.168.137.204 y la dirección IPv4 del servidor Gateway 192.168.137.1

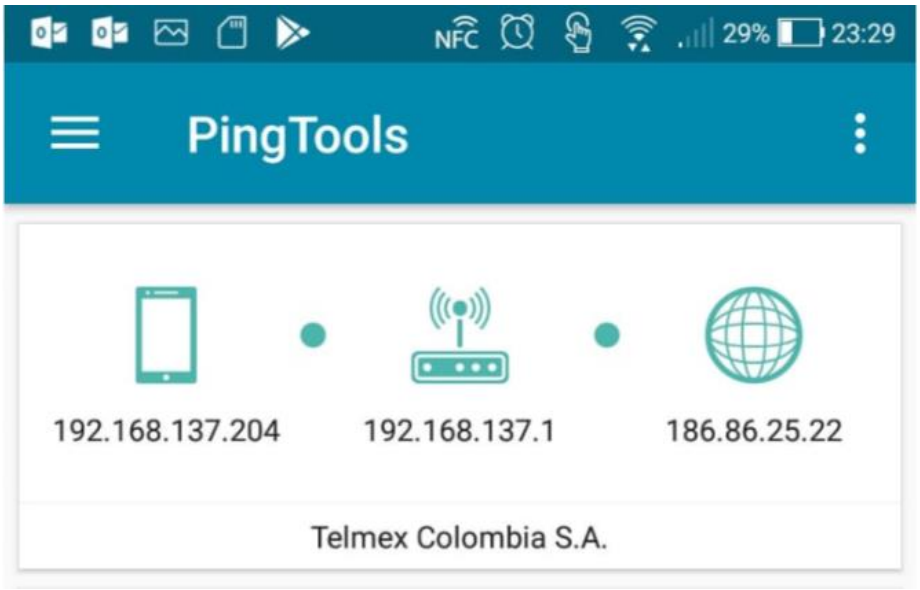

Ilustración 39, Configuración del MFC. Fuente autores.

En la ilustración 40, se puede visualizar la tabla DHCP del servidor Gateway, en la cual se confirma la dirección IPv4 asignada a MFC.

Dispositivos conectados: 1 de 8

Nombre del dispositivo	Dirección IP	Dirección física (MAC)
android-dadf60b...	192.168.137.204	88:d7:f6:9d:c3:0c

Ilustración 40, Tabla DHCP del servidor de Gateway de la red LNV-PC. Fuente autores.

En la ilustración 41, se puede visualizar la dirección IPv4 de la segunda tarjeta de red del servidor Gateway, es importante conocerla dado que por medio de esta sale a internet.

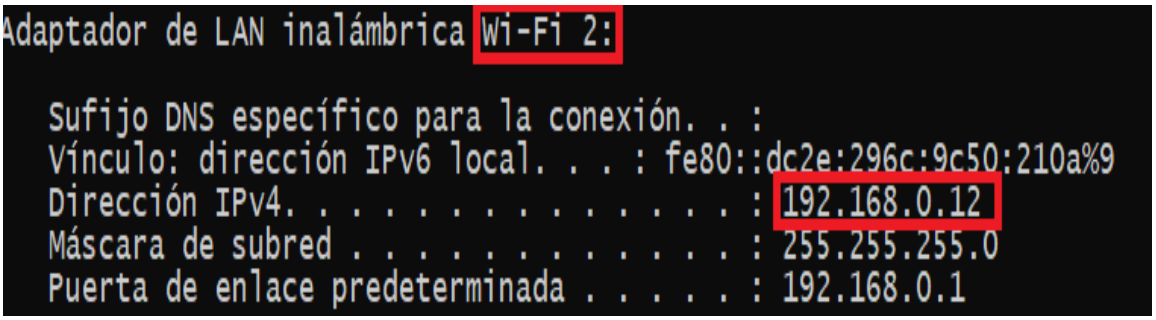

Ilustración 41, Configuración de Interfaz de salida del servidor de Gateway. Fuente autores.

3.4.1 Ejecución de la prueba de evaluación del método

Tal como se aprecia en la ilustración 42, el tráfico que se envía hacia la nube no se protege a nivel de transporte ni aplicación, por lo cual, si no se protege la información esta puede ser capturada por un ente no autorizado.

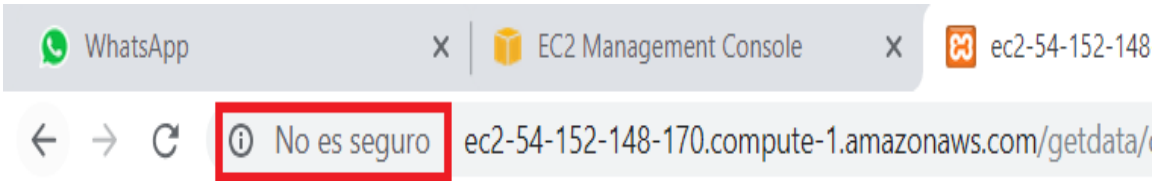

Ilustración 42, Evidencia de no protección del tráfico. Fuente autores

Se aplicó el un filtro en la herramienta Wireshark, ver ilustración 43, el cual sirve para filtrar el tráfico que se dirija a la dirección IPv4 pública asignada al servidor de nube al puerto 80.

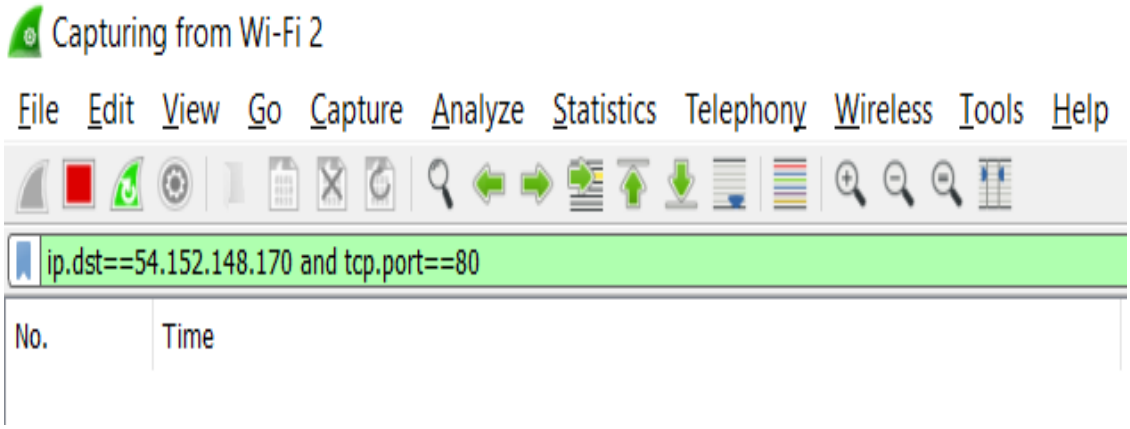

Ilustración 43, Filtro de captura en Wireshark. Fuente autores

Se realizó la prueba de pasar el dispositivo MFC junto al wearable, luego se empieza a observar tráfico hacia el servidor web de AWS por el puerto 80, ver ilustración 44.

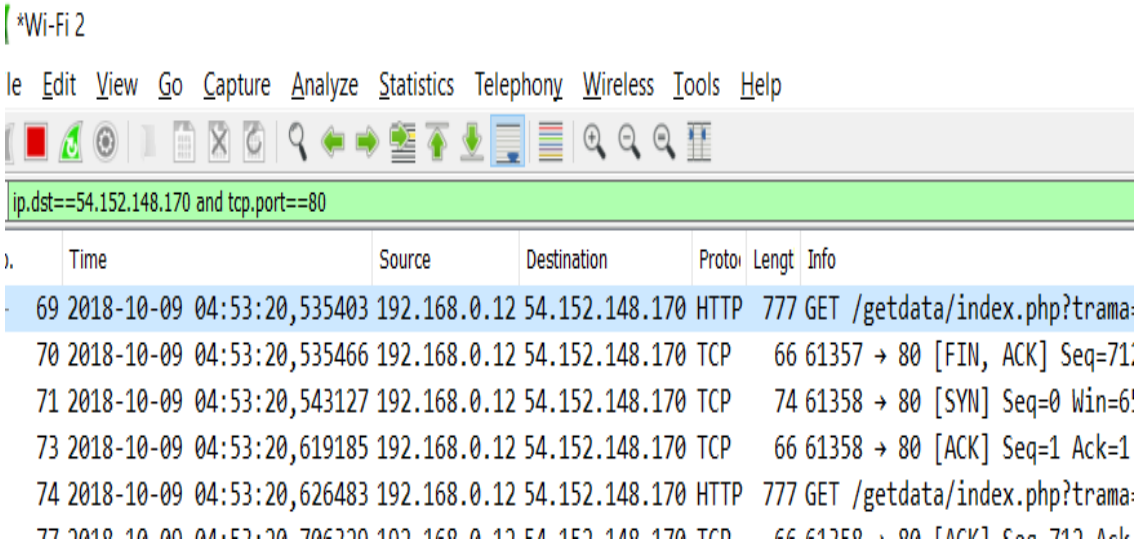

Ilustración 44, Captura de tráfico con wireshark. Fuente autores.

Por medio de un TCPFollow, (funcionalidad de Wireshark) se pueden observar datos relevantes de la captura, ver ilustración 45.

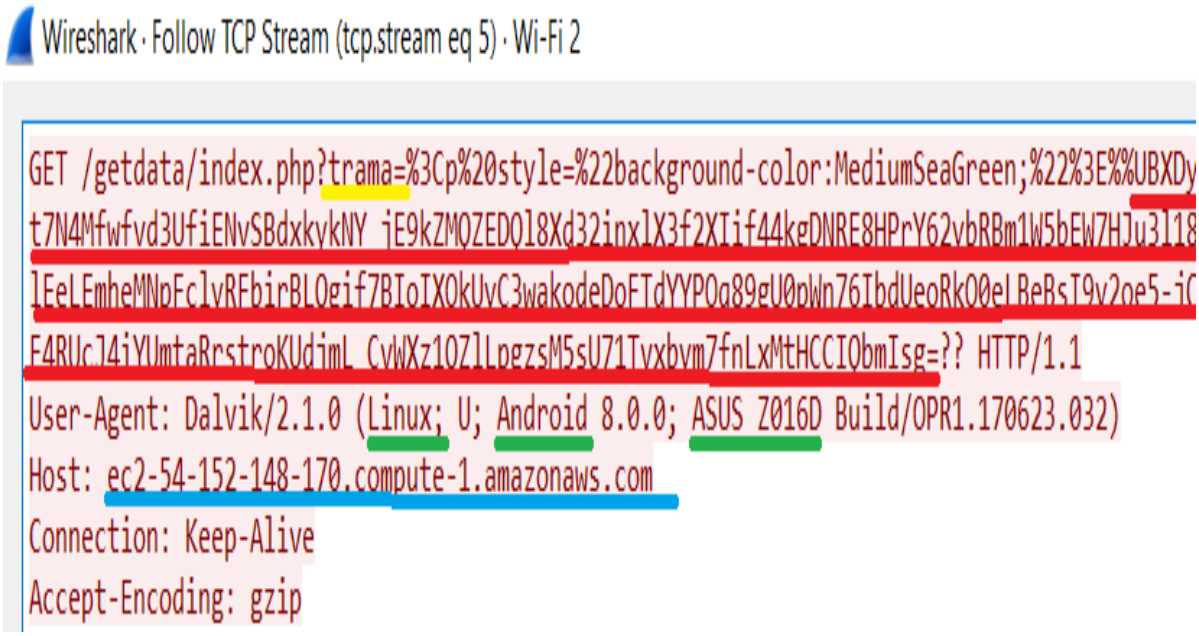

Ilustración 45, TCPFollow con wireshark. Fuente autores.

La variable "*trama*" subrayada en amarillo (ilustración 45), contiene información confidencial subrayada en rojo, el contenido no es legible debido a que viaja cifrada por el algoritmo DHEC. Además, se confirma que la información proviene del MFC ASUS como se puede apreciar en al campo User-Agent (ver subrayado en verde) y por último el destino que es la instancia ec2 de la nube AWS, visualizado en el campo *host (*subrayado en azul).

A continuación, en la ilustración 46, se puede observar que la información es guardada en reposo de forma cifrada.

Ilustración 46, Información en reposo. Fuente autores.

4. Conclusiones y recomendaciones

4.1 Conclusiones

- Se ha logrado el objetivo general al diseñar un método que permita la extensión de FC, proporcionando transporte y cifrado de datos sensibles desde los dispositivos wearables o vestibles, se proporciona protección a los datos en dispositivos con pocas capacidades (procesamiento y cobertura), así mismo, es posible la reducción de riesgos asociados a la posible pérdida de privacidad al salir de un área protegida, esto, al llevar el FC con la persona y aplicando mecanismos de cifrado, lo cual es funcional y factible su implementación, en ese sentido, la cobertura y protección se lleva consigo en todo momento, es decir el MFC.

- Según las pruebas realizadas para cada una de las tecnologías de comunicación, las cuales midieron compatibilidad, consumo de energía, procesamiento y seguridad (hablando de posibilidad de sufrir ataques de interceptación de información), y con la tabulación de la información se determina que NFC es la tecnología que mejor se adapta para el transporte de la información de los wearables hasta la FC.

- La evaluación sobre los algoritmos de cifrado, que midió variables como el nivel de riesgo, consumo de procesamiento, consumo de energía, tiempo ejecución, consumo de ancho de banda, se determina que DHEC es el algoritmo de cifrado, que cuenta con mejores características para proteger la información entre FC y la nuble, permitiendo eficiencia y escalabilidad.

- El método planteado usando la tecnología NFC para el transporte de la información de los wearables hasta FC, protegiendo esta con el algoritmo DHEC sobre un dispositivo móvil de alta gama que siempre acompaña a las personas, salvaguarda la privacidad al salir de un área protegida, por lo cual, la combinación de la tecnología con el cifrado permite el resguardo de datos en la comunicación.

- Las pruebas ejecutadas confirman la capacidad de FC para proteger la información al ser interceptada por un atacante, entregando viabilidad de implementación del método planteado.

4.2 Recomendaciones, lecciones aprendidas y trabajo futuro

Debido al alcance del proyecto, en cada objetivo no fue posible evaluar completamente la confidencialidad, integridad y disponibilidad (CID) de los algoritmos de cifrado y tecnologías de comunicación, lo cual puede variar los resultados a la hora de determinar, se recomienda a futuro ampliar el alcance del método, lo cual puede dar un mejor escenario a la hora de validar su funcionamiento. Con lo realizado en el método propuesto, se observó que FC puede aportar capacidades para proteger CID de los wearables de una forma más completa.

En trabajos futuros se puede revisar una estrategia más completa que permita salvaguardar estos dispositivos u otros IoT disponibles, es decir, tener en cuenta la protección entre del tráfico entre el dispositivo IoT y FC.

5. Anexo: Códigos de los desarrollos mencionados.

La diferente información reposara en el CD final a entregar, a continuación de hace una descripción de los diferentes anexos.

- Códigos Programas ARDUINO
    - Implementación del módulo PN532: Este módulo soporta la tecnología NFC en la tecnología Arduino.
    - Implementación del módulo RC522: Este módulo soporta la tecnología RFID en la tecnología Arduino.
    - Implementación del módulo HC05: Este módulo soporta la tecnología Bluetooth en la tecnología Arduino.
- Códigos Apps Android
    - Prueba Bluetooth: App desarrollada para el envío de información al módulo HC05.
    - Prueba Algoritmos de cifrado: App desarrollada para la ejecución de las pruebas de los algoritmos criptográficos AES, DHEC, HPJ y RSA.
    - MyNFC: App desarrollada para la implementación de la tecnología NFC y el algoritmo criptográfico DHEC, proteger la información y enviarla a la nube.
    - ServerNFC: App desarrollada para consultar y descifrar la información almacenada en la nube.

También puede ser descargados a través del siguiente enlace:

https://correoitmedu-my.sharepoint.com/:f:/g/personal/rafaelmontoya249585_correo_itm_edu_co/EklTiwaVtNdKnhGOM-2YilkB67OKeVF7oI0VH91YgkoAmA?e=EOsq5N

6. Bibliografía

[1] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, “Fog Computing for Sustainable Smart Cities: A Survey,” vol. 0, no. 0, 2017.

[2] G. P. Hancke, K. Markantonakis, and K. E. Mayes, “Security challenges for user-oriented RFID applications within the ‘Internet of things,’” J. Internet Technol., vol. 11, no. 3, pp. 307–314, 2010.

[3] M. Abomhara, “Security and Privacy in the Internet of Things : Current Status and Open Issues,” Priv. Secur. Mob. Syst. (PRISMS), 2014 Int. Conf., pp. 1–8, 2014.

[4] H. Upadhyay and H. Patel, “Mitigation of Privacy issues in IoT by modifying CoAP,” Inven. Comput. Technol. (ICICT), Int. Conf., vol. (Vol. 3, p, pp. 6–9, 2016.

[5] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras, “On the Security and Privacy of Internet of Things Architectures and Systems,” Proc. - 2015 Int. Work. Secur. Internet Things, SIoT 2015, pp. 49–57, 2016.

[6] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, “Fog Computing for the Internet of Things: Security and Privacy Issues,” IEEE Internet Comput., vol. 21, no. 2, pp. 34–42, 2017.

[7] Gartner, “Measuring the Strategic Value of the Internet of Things for Industries,” Measuring the Strategic Value of the Internet of Things for Industries, 2016. [Online]. Available: https://www.gartner.com/doc/3299317?ref=unauthreader.

[8] T. Allegaert, “Internet of Things: capturar valor en un mercado en desarrollo,” in Internet of Things: capturar valor en un mercado en desarrollo, 2017.

[9] C. Kowalevicz, J. Pirrone, and M. Huerta, “Implementation model using a hippocratic protocol in mobile terminals with NFC technology,” 2017 Int. Caribb. Conf. Devices, Circuits Syst. ICCDCS 2017, pp. 113–116, 2017.

[10] R. Focardi and M. Squarcina, “Run-Time Attack Detection in Cryptographic APIs,” Proc. - IEEE Comput. Secur. Found. Symp., pp. 176–188, 2017.

[11] S. Vollala, V. V. Varadhan, K. Geetha, and N. Ramasubramanian, “Design of RSA processor for concurrent cryptographic transformations,” Microelectronics J., vol. 63, no. March, pp. 112–122, 2017.

[12] J. Ma, X. Chen, R. Xu, and J. Shi, “Implementation and Evaluation of Different Parallel Designs of AES Using CUDA,” Proc. - 2017 IEEE 2nd Int. Conf. Data Sci. Cyberspace, DSC 2017, pp. 606–614, 2017.

[13] S. Yi, C. Li, and Q. Li, “A Survey of Fog Computing: Concepts, Applications and Issues,” Proc. 2015 Work. Mob. Big Data - Mobidata ’15, pp. 37–42, 2015.

[14] INCIBE, “Riesgos y retos de ciberseguridad y privacidad en IoT,” Riesgos y retos de ciberseguridad y privacidad en IoT, 2017. [Online]. Available: https://www.incibe-cert.es/blog/riesgos-y-retos-ciberseguridad-y-privacidad-iot.

[15] ENISA, Baseline Security Recommendations for IoT in the context of Critical Information

Infrastructures, no. November. 2017.

[16] D. Minoli, K. Sohraby, and B. Occhiogrosso, “IoT Security (IoTSec) Mechanisms for e-Health and Ambient Assisted Living Applications,” 2017 IEEE/ACM Int. Conf. Connect. Heal. Appl. Syst. Eng. Technol., pp. 13–18, 2017.

[17] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, “EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications,” IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 9, pp. 1621–1632, 2012.

[18] X. Yao, X. Han, X. Du, and X. Zhou, “A lightweight multicast authentication mechanism for small scale IoT applications,” IEEE Sens. J., vol. 13, no. 10, pp. 3693–3701, 2013.

[19] H. Ning, H. Liu, and L. T. Yang, “Aggregated-proof based hierarchical authentication scheme for the internet of things,” IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 3, pp. 657–667, 2015.

[20] H. Wang, Z. Wang, and J. Domingo-Ferrer, “Anonymous and secure aggregation scheme in fog-based public cloud computing,” Futur. Gener. Comput. Syst., 2017.

[21] M. H. Ibrahim, “Octopus: An edge-fog mutual authentication scheme,” Int. J. Netw. Secur., vol. 18, no. 6, pp. 1089–1101, 2016.

[22] Q. Wang, D. Chen, N. Zhang, Z. Ding, and Z. Qin, “PCP: A Privacy-Preserving Content-Based Publish-Subscribe Scheme with Differential Privacy in Fog Computing,” IEEE Access, vol. 5, no. c, pp. 17962–17974, 2017.

[23] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, “A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT,” IEEE Access, vol. 5, pp. 3302–3312, 2017.

[24] A. M. Rahmani et al., “Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach,” Futur. Gener. Comput. Syst., vol. 78, pp. 641–658, 2018.

[25] “Developers” Google Corp, “Android.nfc,” 2018. [Online]. Available: https://developer.android.com/reference/android/nfc/package-summary.

[26] “Developers” Google Corp, “Android.bluetooth,” Android.bluetooth, 2018. [Online]. Available: https://developer.android.com/reference/android/bluetooth/package-summary.

[27] Arduino, “Arduino,” 2018. [Online]. Available: https://www.arduino.cc/en/Guide/Introduction.

[28] Wilson, “PN532 NFC RFID Module User Guide. 2,” 2012.

[29] Elechouse, “PN532,” PN532, 2018. [Online]. Available: https://github.com/elechouse/PN532/tree/PN532_HSU/PN532.

[30] Tinchorton, “RFID-RC522,” 2018. [Online]. Available: http://saber.patagoniatec.com/2016/07/lector-de-tarjetastags-rfid-rc522-13-56mhz-nfc/.

[31] M. Balboa, “Arduino RFID Library for MFRC522,” 2018. [Online]. Available: https://github.com/miguelbalboa/rfid.

[32] HC01, “HC05,” 2018. [Online]. Available: http://www.hc01.com/productdetail?productid=201702040005.

[33] C. H. Chen, I. C. Lin, and C. C. Yang, “NFC attacks analysis and survey,” Proc. - 2014 8th Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput. IMIS 2014, pp. 458–462, 2014.

[34] J. Padgette et al., “NIST Special Publication 800-121 Revision 2 Guide to Bluetooth Security,” p. 67, 2017.

[35] A. R. NDJIONGUE, K. OUAHADA, S. RIMER, and Z. MNGOMEZULU, “A review of Bluetooth and NFC for financial applications,” Sixth Int. Conf. Adv. Comput. Control Netw. - ACCN 2017, no. March, pp. 48–51, 2017.

[36] M. Grabovica, D. Pezer, S. Popić, and V. Knežević, “Provided security measures of enabling technologies in Internet of Things (IoT): A survey,” 2016 Zooming Innov. Consum. Electron. Int. Conf. ZINC 2016, pp. 28–31, 2016.

[37] AWS, “Overview of Amazon Web Services,” vol. 2, no. December, p. 83, 2018.

[38] ENISA, “Cloud Computing Risk Assessment,” 2009.

[39] Cisco Systems, “Edge computing vs. fog computing: Definitions and enterprise uses,” 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/enterprise-networks/edge-computing.html.

[40] I. Stojmenovic and S. Wen, “The Fog Computing Paradigm: Scenarios and Security Issues,” Proc. 2014 Fed. Conf. Comput. Sci. Inf. Syst., vol. 2, pp. 1–8, 2014.

[41] M. Frustaci, P. Pace, and G. Aloi, “Securing the IoT world: Issues and perspectives,” 2017 IEEE Conf. Stand. Commun. Networking, CSCN 2017, no. table I, pp. 246–251, 2017.

[42] H. Motoda et al., Top 10 algorithms in data mining, vol. 14, no. 1. 2007.

[43] I. Xplore, INTERNATIONAL STANDARD ISO / IEC / IEEE Telecommunications and information and metropolitan area networks —, vol. 2018. 2018.

[44] Android, “The world’s most popular mobile OS From phones and watches to cars and TVs,” 2019. [Online]. Available: www.android.com.

[45] Granados Paredes, Gibran. Introducción a la criptografía. Revista digial Universitaria. Universidad Nacional Autónoma de México (UNAM). 10 de julio 2006 • Volumen 7 Número 7 • ISSN: 1067-6079. http://www.revista.unam.mx/vol.7/num7/art55/jul_art55.pdf

[46] Fernandez, Santiago. La criptografía clásica. SIGMA Nº 24 . zk. 24 SIGMA. Abril 2004. https://s3.amazonaws.com/academia.edu.documents/40562076/9_Criptografia_clasica.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1553877933&Signature=%2BFuUOhUkySWBGwDvXUrqFvrGSnk%3D&response-content-disposition=inline%3B%20filename%3DCriptografia_clasica.pdf

[47] Yoon, Seungyong y Kim, Jeongnyeo. Remote Security Management Server for IoT devices. IEEE (2017). Pag 1162-1164.

[48] Tai-Yeon Ku, Wan-Ki Park, Hoon Choi. IoT Energy Management Platform for MicroGrid . 2017 IEEE 7th International Conference on Power and Energy Systems. IEEE 2017. Pag 106-110.

[49] L. Salman et al., "Energy efficient IoT-based smart home," 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, 2016, pp. 526-529. doi: 10.1109/WF-IoT.2016.7845449

[50] S. Jiang, "Internet of things (IoT): Technologies and applications," 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, 2015, pp. 3-3. doi: 10.1109/ICTER.2015.7377657

[51] E. Valea, M. Da Silva, G. Di Natale, M. Flottes, S. Dupuis and B. Rouzeyre, "SI ECCS: SECure context saving for IoT devices," 2018 13th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS), Taormina, 2018, pp. 1-2. doi: 10.1109/DTIS.2018.8368561

[52] J. Garcia, E. Simó, X. Masip-Bruin, E. Marín-Tordera and S. Sànchez-López, "Do We Really Need Cloud? Estimating the Fog Computing Capacities in the City of Barcelona," 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich, 2018, pp. 290-295. doi: 10.1109/UCC-Companion.2018.00070

[53] Shanhe Yi, Cheng Li y Qun Li. A Survey of Fog Computing: Concepts, Applications and Issues. Mobidata '15 Proceedings of the 2015 Workshop on Mobile Big Data. Pages 37-42. Hangzhou, China — June 21 - 21, 2015.

[54] NFC Forum. NFC, Bluetooth and RFID: Unraveling the Wireless Connections. Agosto 2015. https://nfc-forum.org/nfc-bluetooth-and-rfid-unraveling-the-wireless-connections/

[55] K. Monteiro, É. Rocha, É. Silva, G. L. Santos, W. Santos and P. T. Endo, "Developing an e-Health System Based on IoT, Fog and Cloud Computing," 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich, 2018, pp. 17-18. doi: 10.1109/UCC-Companion.2018.00024

[56] N. B.V. and R. M. R. Guddeti, "Heuristic-Based IoT Application Modules Placement in the Fog-Cloud Computing Environment," 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich, 2018, pp. 24-25. doi: 10.1109/UCC-Companion.2018.00027.

[57] F. Fowley, C. Pahl, P. Jamshidi, D. Fang and X. Liu, "A Classification and Comparison Framework for Cloud Service Brokerage Architectures," in IEEE Transactions on Cloud Computing, vol. 6, no. 2, pp. 358-371, 1 April-June 2018. doi: 10.1109/TCC.2016.2537333