PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

Estudio del diseño de un procesador criptográfico de Curvas Elípticas para el dispositivo WISP

TRABAJO DE INVESTIGACIÓN PARA LA OBTENCIÓN DEL GRADO DE BACHILLER EN CIENCIAS CON MENCIÓN EN INGENIERÍA ELECTRÓNICA

AUTOR

Igor Ivan Mendez Cabana

ASESOR:

Carlos Bernardino Silva Cárdenas

Lima, agosto, 2020

Resumen

El rápido avance del internet de las cosas ha supuesto plantear nuevas maneras de implementar las redes de sensores. Es así como la tecnología RFID se ha ido tornando cada vez más atractiva como una alternativa que no requiere el uso de baterías. La plataforma WISP (Wireless Identification Sensing Platform) es uno de los dispositivos que más ha permitido impulsar el desarrollo de sensores RFID. WISP es la primera etiqueta RFID computacional, es decir, que permite programar un algoritmo básico en su memoria. Sin embargo, al igual que con las redes de sensores actuales, estos dispositivos suelen ser blancos fáciles de atacantes cibernéticos ya que son un punto débil en la red debido a sus limitaciones en recursos de hardware y energía que dificultan desarrollar criptografías en software eficientes.

En este trabajo se presenta un estudio sobre el diseño de una arquitectura para un procesador criptográfico de Curvas elípticas (ECC) de bajo consumo energético implementado que cumple con las limitaciones energéticas para ser utilizado con la etiqueta WISP. Este trabajo está basado en las arquitecturas propuestas por Ahmad Salman [1] y Siddika Berna [2].

Keywords: Bajo consumo energético, WISP, RFID, Criptografía de Curva Elíptica, FPGA

Índice General

Introducción

El internet de las cosas (IoT) se refiere a una gran cantidad de dispositivos ubicuos interconectados que pueden recopilar una gran cantidad de datos e interactuar entre ellos. Actualmente, es una de las tecnologías en mayor auge. Acorde con las últimas estadísticas, se estima que la cantidad de dispositivos conectados a internet superará los 31 billones en el 2020 [4]. La velocidad e intensidad con la que esta tecnología avanza depende en gran medida de, en primer lugar, la eficiencia energética y, en segundo, la seguridad de las redes de sensores. Por ello, el uso de la tecnología RFID en estas redes de sensores es una propuesta cada vez más atractiva debido principalmente a que no requiere de baterías para funcionar. Por otro lado, gracias al avance en la microelectrónica, ahora es posible el uso de microcontroladores de propósito general que pueden ser energizados remotamente. Es a partir de esto que nace la etiqueta WISP (Wireless Identification Sensing Platform) la cual es la primera etiqueta RFID computacional, es decir, que incorpora un microcontrolador de propósito general que permite ejecutar diversos algoritmos.

Por otro lado, la seguridad en estas redes está tomando mayor importancia en los últimos años debido al alarmante aumento de ataques a redes IoT. Esto se debe a que los dispositivos que conforman esta red presentan grandes limitaciones de hardware, dificultando de esta manera la implementación de criptografías fuertes. Además, al ser dispositivos que suelen estar en lugares de fácil acceso son más propensos a ataques de canal lateral.

En esta tesis se realizará un estudio del diseño de un procesador criptográfico de curvas elípticas orientado a disminuir la disipación de potencia para poder adecuarse a la los requerimientos de una etiqueta WISP. El desarrollo de este documento será de la siguiente manera. En el primer capítulo se muestra el marco problemático. En el segundo capítulo, se introducen los conceptos teóricos necesarios para comprender acerca de la etiqueta WISP y la Criptografía de Curvas Elípticas. En el capítulo tercero, se muestran las conclusiones y recomendaciones de este estudio.

Capítulo 1

Marco Problemático

1.1. Contexto del problema

En los recientes años, junto con el creciente auge del internet de las cosas, ha surgido un gran interés en las redes de sensores. Las áreas que se pueden beneficiar de estas redes van desde el campo industrial, para el sensado y control de procesos, hasta el de la salud donde promete mejorar la calidad de vida de las personas. Sin embargo, existen problemas que aún no permiten una eficiente implementación. Uno de los inconvenientes más importantes está relacionado al consumo energético debido a que cada cierto periodo de tiempo los dispositivos dentro de la red requieren una recarga o cambio de batería. Para redes de sensores pequeñas el control energético puede ser manejable, pero a medida que se agregan más sensores el problema se vuelve crítico. Además, el uso de baterías incrementa el tamaño, el peso y el costo de cada sensor. Frente a estos problemas, las características de la tecnología RFID surge como una solución atractiva [5].

1.1.1. Redes de sensores RFID

La identificación por radiofrecuencia, o "RFID" por sus siglas en inglés, es una tecnología que se usa para identificar objetos de una manera eficiente y automatizada. En este sistema se utilizan unos dispositivos llamados "etiquetas" los cuales incorporan cada uno un chip con un número de identificación único. De esta manera, cuando la antena lectora realice una lectura sobre un área cada etiqueta responderá con su número único, permitiendo así identificar al objeto. Lo novedoso de esta tecnología es que las etiquetas no requieren de baterías para poder funcionar, sino que energizan su circuito interno aprovechando la energía de las ondas de radiofrecuencia emitidas por la antena del lector al realizar una lectura. Esto permite que las etiquetas sean baratas y pequeñas que pueden incluso implementarse con forma de estíquers que se adhieren a los objetos a identificar. Utilizando este sistema, las empresas de retail, por ejemplo, pueden realizar el inventariado de sus productos de manera eficiente y rápida con solo desplazar el lector a través de su almacén.

En los últimos años, se ha propuesto el uso del RFID en redes de sensores. En estos sistemas cada etiqueta RFID no solo enviaría su número de identificación, sino también el valor medido

por el sensor que tiene incorporado. Este modelo tiene como ventaja que estas etiquetas son resistentes por lo que tienen un tiempo de vida de mayor duración pudiendo llegar incluso a décadas [6]. Además, su reducido peso y tamaño les da el potencial de poder conectar cualquier cosa a una red, reduciendo la brecha entre el mundo físico y el mundo virtual de las cosas. Es por ello que muchos investigadores en Internet de las Cosas (IoT) consideran al RFID como los cimientos de la futura generación IoT [6].

1.1.2. Etiqueta WISP

La etiqueta WISP (Wireless Identification Sensing Platform) nació en los laboratorios de Intel como un proyecto de hardware y software abierto [7]. Este dispositivo es una etiqueta RFID programable. Es la primera etiqueta RFID computacional, es decir, que permite realizar un pequeño procesamiento por un microcontrolador de propósito general, lo cual da la oportunidad de implementar nuevas aplicaciones más allá de la identificación de objetos. Además, ofrece unos pines de expansión que permiten agregar sensores externos u otros dispositivos de bajo consumo energético. La última versión WISP 5 (Fig 1.1) tiene un alcance de seis metros alejado del lector.

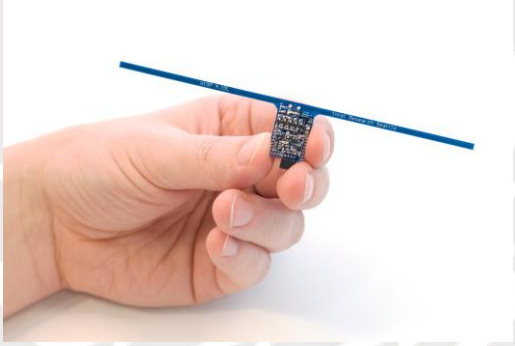

Fig 1.1: Etiqueta RFID UHF: WISP 5 [7]

El inicio de este proyecto dio como resultado que desarrolladores de diversas partes del mundo puedan utilizarla creando nuevos usos para esta tecnología. Un ejemplo es el proyecto WISPCam [8], donde se acopla a WISP una pequeña cámara que toma una fotografía y lo envía al lector cuando detecta movimiento sin requerir uso de baterías; también NeuralWISP [9] en el que detecta pulsos neuronales de insectos aprovechando el bajo peso y tamaño de WISP; además, se han propuesto proyectos en el área de salud para el monitoreo de la actividad de personas mayores así como de parámetros para conocer su estado de salud. Por otro lado, la etiqueta WISP ha creado la oportunidad para que expertos en ciberseguridad puedan investigar y desarrollar nuevos protocolos de seguridad y criptografías en etiquetas RFID, ya que el código del firmware es abierto y modificable. Esto permite probar distintos protocolos seguridad

enfocándose solo en la parte de seguridad y evitar tener que implementar el protocolo de comunicación [10]. Debido a todo esto es que el desarrollo de WISP ha fortalecido el desarrollo de la tecnología de redes de sensores a lo largo de estos años, creando nuevas aplicaciones innovadoras y mejorando la ciberseguridad de estas redes.

1.2. Descripción y Formulación del problema

Según estadísticas proporcionadas por Symantec, corporación internacional en seguridad informática, los ataques a dispositivos IoT aumentaron un 600% entre 2016 y 2017 [11]. Esto se debe a la facilidad con la que estos dispositivos son vulnerados, generando un agujero de seguridad por donde los atacantes se pueden introducir y ganar acceso a redes con información más sensible. La etiqueta WISP es una plataforma de uso libre que es utilizada por gran cantidad de desarrolladores en todo tipo de aplicaciones. Muchas de estas, requieren algún nivel de seguridad para proteger la data y la integridad de la aplicación. Por ejemplo, en el caso de aplicaciones médicas es necesario realizar una autenticación de los dispositivos que solicitan una conexión para evitar que terceros puedan leer o modificar la data, lo cual podría llegar a tener consecuencias críticas [12]. Se han visto casos, por ejemplo, de servidores infectados por malware introducidos por equipos IoT modificados por el atacante. Es por ello que tomar medidas de seguridad se vuelve cada vez más importante con el impacto que el internet de las cosas (IoT) está teniendo actualmente en la sociedad, ya que los ciberataques en este ámbito siguen tomando cada vez más fuerza.

Además, en el caso de dispositivos como las etiquetas WISP que al igual que las redes de sensores suelen ubicarse en ambientes públicos o de fácil acceso, se debe tener en cuenta que son más propensos a ataques de canal lateral. En este tipo de ataque se aprovecha el acceso al dispositivo físico para explotar las vulnerabilidades en la implementación física y así extraer información y ganar accesos.

La manera de mantener la seguridad contra estos ataques es con el uso de funciones criptográficas que impidan que terceros no autorizados puedan ver o modificar la data y el código. Asimismo, siguiendo el paradigma de una plataforma de propósito general como WISP, que enmascara todo el proceso de comunicación para que el desarrollador solo se concentre en la programación a alto nivel también, se debería simplificar el uso de las funciones de seguridad y permitiendo el uso de estándares criptográficos. De esta manera, el desarrollador puede elegir el nivel de seguridad que se desee utilizar teniendo en cuenta que una mayor seguridad exige un mayor costo computacional.

1.2.1. Implementación de criptografía en WISP

Sin embargo, a pesar de los beneficios que significa el uso de criptografía en los dispositivos de red, también implica un gran costo energético debido al intenso cómputo que requiere realizar complicadas operaciones matemáticas. Es por ello que es difícil implementarlas en dispositivos de bajos recursos como lo son las redes de sensores y particularmente complicada en WISP que tiene limitaciones energéticas aún más extremas ya que no incorpora batería.

En [7], C. Pendl, M. Pelnar y M. Hutter diseñaron una librería criptográfica en el microcontrolador de WISP utilizando la Criptografía de Curvas Elípticas (ECC). Esta librería está compuesta de algunas funciones criptográficas que permiten implementar diversos protocolos de seguridad y utilizando un solo nivel de seguridad (llave de 192 bits). Sin embargo, a pesar de la optimización en código para reducir el consumo de energía y mejorar el rendimiento, los resultados muestran que no llega a ser aplicable en una implementación real. El uso de esta criptografía reducía el alcance de la etiqueta a la antena de 6 m a 40 cm. Además, cada operación criptográfica requería alrededor de 58 segundos en completarse a 40 cm.

1.3. Estado del Arte

La implementación de funciones criptográficas en dispositivos de bajos recursos como el RFID es un reto ya que requiere un diseño que equilibre entre rendimiento, seguridad y costo; a estos tipos de criptografía se les conoce como “Criptografía ligera” y se enfocan en utilizar la mínima cantidad de recursos posibles [6]. Las implementaciones en este tipo de criptografía suelen ser en ASIC, ya que permite una mayor optimización de recursos lo que implica un consumo muy bajo de energía y área. Sin embargo, tiene la desventaja de no ser flexible por eso suelen diseñarse para utilizarse solo con algunos protocolos de seguridad específicos. Por otro lado, las implementaciones en software ejecutados en un microcontrolador son muy flexibles ya que es muy fácil modificarlos para adaptarse a nuevos protocolos. Sin embargo, los niveles de optimización de recursos no son tan buenos como una solución en hardware. La implementación en FPGA es un punto medio entre optimización eficiente y flexibilidad, por lo que se puede conseguir un bajo consumo energético y menor tiempo de ejecución que en software.

Las funciones criptográficas se pueden dividir en tres grupos: algoritmo Hash, criptografía simétrica y criptografía asimétrica. Por un lado, los algoritmos Hash se utilizan conjuntamente

con otros tipos de funciones criptográficas por lo que no son convenientes para dispositivos de bajos recursos [6]. Los algoritmos simétricos, por otro lado, son más utilizados como criptografía ligera porque son algoritmos que consumen menos recursos que su contraparte asimétrica. En este tipo de criptografía cada integrante en la red comparte una misma llave para encriptar y desencriptar la información. Esto vuelve más vulnerable a la red, ya que al encontrar la llave de un nodo, todos los otros quedan expuestos. En cambio, en la criptografía asimétrica cada nodo tiene una llave privada que nunca es compartida y otra pública que es compartida al comenzar la comunicación. El principio de funcionamiento consiste en que un paquete encriptado utilizando la llave pública de un nodo sólo puede ser desencriptado con la llave privada del mismo nodo. De esta manera, aunque un nodo sea comprometido, se puede mantener la seguridad de la red. Por este motivo se están proponiendo cada vez más el uso de criptográficas asimétricas para implementar distintos protocolos para dispositivos de bajos recursos como RFID, redes de sensores o dispositivos IoT [13].

Respecto a la criptografía asimétrica, el algoritmo más usado es el RSA. Sin embargo, implementar este algoritmo requiere de muchos recursos de hardware. Desde hace unas décadas, el algoritmo ECC (Elliptic Curve Cryptography) ha llamado la atención, ya que proporciona una seguridad similar al RSA pero utilizando llaves de menor tamaño, lo cual implica un menor uso de recursos de memoria y hardware para implementarlo [6]. Es debido a esta razón que se ha empezado a usar especialmente en dispositivos de bajos recursos. Si bien existen muchas propuestas de criptográficas asimétricas que hacen uso de menos recursos, el algoritmo ECC ha sido largamente estudiado y verificado por varios expertos a lo largo de los años. Es por ello que se está volviendo un estándar en las redes IoT [14].

1.4. Importancia y Justificación

Con la gran relevancia que está tomando la etiqueta WISP, es necesario brindar mejores opciones en seguridad que puedan ser adoptados por distintos desarrolladores en sus proyectos. Por ello es necesario una solución que, en primer lugar, pueda satisfacer todas las limitaciones de recursos que presenta WISP y, en segundo lugar, que brinde una amplia variedad de opciones entre distintos niveles de seguridad (tamaños de llave), diversos estándares y aplicabilidad a diferentes protocolos. Todos estos objetivos se pueden lograr con un dispositivo FPGA que es reconfigurable y permite variar estos parámetros.

Además, un módulo criptográfico en FPGA permitirá que se puedan implementar nuevas aplicaciones con la etiqueta WISP haciendo uso de una criptografía más fuerte. Por otro lado,

al estar implementado en un FPGA facilitará la transición de un prototipo a un integrado ASIC, teniendo en cuenta que ya se dispone de un Core-WISP en Verilog que realiza todo el protocolo de comunicación [15].

Capítulo 2

Marco teórico

En el presente capítulo se explicarán los conceptos básicos respecto a la etiqueta WISP, la criptografía de curvas elípticas (ECC) y el consumo energético en un FPGA. En la primera sección se describe la estructura y funcionamiento de la etiqueta WISP. En la segunda se detallan los fundamentos teóricos de la criptografía de curvas elípticas y los algoritmos que lo componen. Por último, se analizan los distintos tipos de consumo energético en un FPGA y las estrategias para disminuirlas.

2.1. Etiqueta RFID UHF: WISP

La etiqueta WISP 5 (Fig. 2.1) es una etiqueta RFID del tipo UHF pasiva, es decir, no usa batería y opera a una frecuencia aproximada de 900 MHz. En [16] se le acuña el término etiqueta RFID computacional porque incorpora un microcontrolador de propósito general, de ultra baja energía de la familia MSP430. La frecuencia máxima con la que puede trabajar este microcontrolador es de 6 MHz a 3.3 V. Si el voltaje usado es de 1.8 V, la frecuencia máxima será de 4 MHz. Además, incorpora algunos sensores en la placa: un acelerómetro 3D, un sensor de temperatura, dos leds para debugger y brinda un fácil acceso a los puertos digitales del microcontrolador para poder agregar otros sensores externos [16].

Fig. 2.1: Etiqueta RFID UHF: WISP 5 [16]

WISP fue diseñado para que sea compatible con los lectores para etiquetas UHF de uso comercial. Estos utilizan el protocolo Electronic Product Code (EPC) Class 1 Generation 2 en la comunicación lector-etiqueta. En este protocolo se definen los estándares de conexión y protocolos en la capa física y de enlace.

2.1.1. Diagrama de bloques de WISP

La estructura de WISP se compone básicamente de seis partes que se pueden apreciar en la Fig. 2.2. En primer lugar, la antena recibe la señal de radiofrecuencia emitida por el lector y lo transforman a señales eléctricas. El siguiente bloque cumple dos funciones. Primero, se encarga de adaptar las impedancias de la antena para disminuir las pérdidas y, segundo, rectifica la señal recibida. A partir de esta señal, se obtiene la información enviada por el lector, pero también se obtiene la energía para alimentar a todo el circuito. En el bloque de comunicación, se demodula la señal extrayendo la envolvente de la señal ASK recibida y luego es acondicionada para adquirir niveles lógicos que sean interpretados por el microcontrolador. Los bloques de control de energía y microcontrolador se describirán en los siguientes subcapítulos.

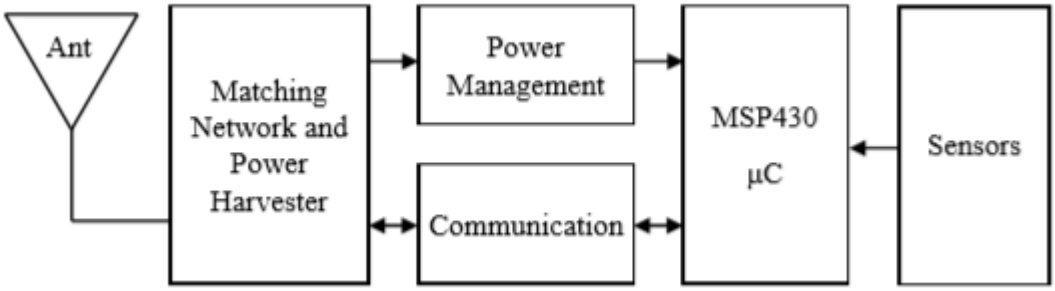

Fig. 2.2: Diagrama de bloques de WISP [16]

2.1.2. Módulo control de energía

Para alimentar a todo el circuito se necesita regular la señal rectificada. Sin embargo, debido a que la cantidad de potencia recibida por la etiqueta disminuye con el cuadrado de la distancia al lector, cuando la etiqueta se aleja más de medio metro la potencia no llega a ser suficiente para obtener un voltaje regulado, lo que significaría que un alcance menor a medio metro la etiqueta ya no funcionaría. Por este motivo, se utiliza un circuito supervisor (Fig. 2.3) que permite almacenar la energía de la señal rectificada en un condensador hasta alcanzar la energía requerida para la tarea programada en el microcontrolador. Este supervisor (SV en la gráfica) envía una señal al microcontrolador cuando la energía es la adecuada. Una vez culminada la tarea, el microcontrolador entra en modo “Sleep” requiriendo menos de 1 uA, y se mantiene a la espera de que el supervisor lo active nuevamente para realizar otra tarea o continuar con alguna inconclusa.

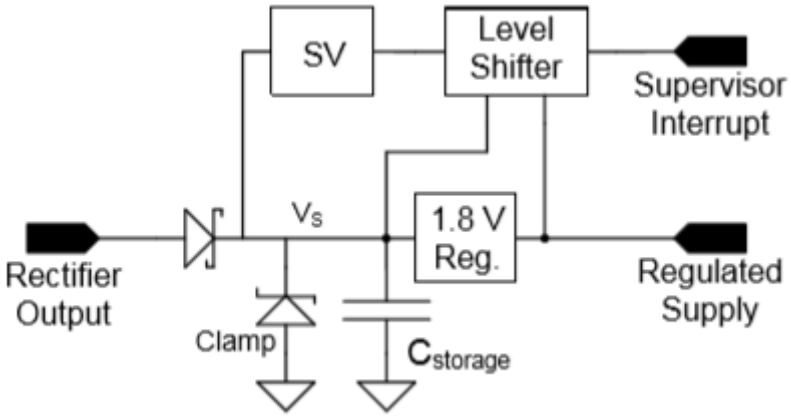

Fig. 2.3: Circuito del módulo de control de energía [12]

De esta forma, se pueden ejecutar algoritmos que requieren más energía del que se recibe del lector. En la Tabla 2.1 se muestran datos experimentales, extraídos de [7], donde se observa como varía la potencia recibida a distintas distancias del lector, cuando su antena emite con una potencia de 1 W.

El protocolo de comunicación requiere 1.2mA a 1.8V, aproximadamente 2mW, por lo que sólo puede ejecutarse de manera continua hasta medio metro de distancia. Superado este límite, el microcontrolador ejecutará el algoritmo en periodos cortos de tiempo cuando el capacitor tenga suficiente energía lo que aumentará el tiempo de ejecución del algoritmo. En general, cualquier algoritmo que se ejecute a una mayor distancia de la antena requerirá más tiempo para concluir.

Tabla 2.1: Variación de la potencia recibida con la distancia

Distancia (m)	Potencia (uW)
0.5	2500
1	271
2	67.6
3	30.1
4	17.0
5	10.8

2.1.3. WISP firmware

El firmware está diseñado en tres niveles como se ve en Fig. 2.4. En el nivel más bajo se encarga del control de energía para asegurarse de que el sistema no caerá en sub-voltajes. Para ello se comunica con el módulo de control de energía quien es el que determina en qué momentos debe estar activo o en estado dormido. En el nivel medio se implementa el protocolo de comunicación EPC class 1 gen 2 para poder recibir e interpretar los comandos enviados por el lector. a partir. Por último, la capa de aplicación incorpora el programa principal. Este esquema permite que el desarrollador solo se enfoque en la capa de aplicación si tener que preocuparse por las capas inferiores de comunicación con el lector y control de energía.

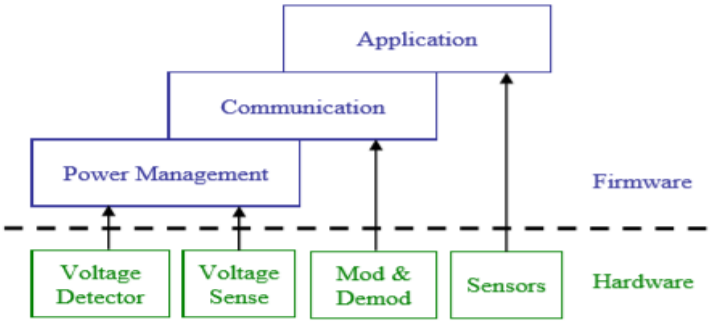

Fig. 2.4: Diagrama de bloques de Firmware [16]

2.2. Criptografía de curva elíptica

En el año 1980, Victor Miller y Neal Koblitz propusieron la criptografía de curva elíptica (ECC). Esta es una criptografía asimétrica o de llave pública que está basada en el problema del logaritmo discreto de las curvas elípticas sobre un campo finito (ECDLP). A partir de este algoritmo, se pueden conseguir varias aplicaciones de seguridad necesarias en una comunicación entre dos dispositivos como intercambio de llaves, firmas digitales, encriptación de datos y autenticación. La mayor ventaja que ofrece ECC respecto a otras criptográficas asimétricas es que posee un tamaño de llave más pequeño y un mejor desempeño con un mismo nivel de seguridad. Esta característica lo vuelve un buen candidato para implementaciones en sistemas con bajos recursos [14].

2.2.1. Curva elíptica sobre campos finitos

Las curvas elípticas pueden ser definidas sobre cualquier tipo de campo, como el real, complejo, etc. Para aplicaciones criptográficas, estas curvas son definidas sobre campos finitos. Existen dos campos finitos importantes: campos binarios (F_{2m}) y campos primos (F_p). Entre

estos dos, los campos primos son más simples y por tanto más utilizados en los estándares de curvas elípticas.

2.2.2. Curvas elípticas sobre F(p)

Existen distintos tipos de curvas elípticas sobre campos primos que son utilizadas en criptografía, tales como la curva Montgomery, la curva Gallant-Lambert-Vanstone o la llamada Curve25519 que se ha introducido recientemente en los últimos años [14]. Sin embargo, la más utilizada y estudiada es la curva Weierstrass que define a la curva elíptica a partir de la ecuación corta de Weierstrass (ecuación 2.1).

Si p es un número primo mayor a 3, una curva elíptica E sobre F(p) es definida de la siguiente manera:

$$E: \quad y^2 = x^3 + ax + b \bmod p, \tag{2.1}$$

donde $a,b,x,y \in F(p)$ y su discriminante $4a + 27b \neq 0$. Los parámetros a, b y p son importantes para determinar la seguridad de la curva, es por ello que diversas instituciones han creado estándares que definen los valores de estos parámetros y aseguran una criptografía confiable. Por ejemplo, el NIST, Instituto Nacional de Estándares y Tecnología de Estados Unidos, recomienda en FIPS 186-2 los estándares p-192, p-224, p-256, p-384 y p-521 [15]. Cabe recalcar que a un valor de campo 'p' mayor, la criptografía será más fuerte de romper, pero también requerirá un mayor costo computacional. Otros estándares de curvas elípticas populares son definidos por Brainpool y SECP. Cada uno de estos es un grupo de curvas de distintos tamaños de campos.

2.2.3. Aritmética de Curva elíptica sobre F(p)

Se tienen dos puntos P = (x1, y1) y Q = (x2, y2) que pertenecen a la curva elíptica E de la ecuación (2.1). A partir de estos se definen las siguientes operaciones:

- Inversa del punto P:

  Se define como el punto simétrico respecto al eje x.

  -P = (x1,-y1)

- Suma de dos puntos:

  La suma de los puntos P y Q da como resultado la inversa del punto generado por la intersección de la curva E con la línea que une P y Q. El nuevo punto R = P + Q = (x3, y3) se puede calcular a partir de las siguientes fórmulas:

$$x_3 = \lambda^2 - x_1 - x_2, \quad y_3 = \lambda(x_1 - x_3) - y_1 \tag{2.2}$$

$$\lambda = \frac{(y_2 - y_1)}{(x_2 - x_1)} \tag{2.3}$$

- Duplicación de punto:

  La duplicación de P es la inversa del punto generado por la intersección de la curva E con la línea tangente a la curva E en el punto P. El punto R = 2P = (x3, y3) se puede calcular a partir de las siguientes ecuaciones:

$$x_3 = \lambda^2 - 2x_1, \quad y_3 = \lambda(x_1 - x_3) - y_1 \tag{2.4}$$

$$\lambda = \frac{(3x_1^2 + a)}{(2y_1)} \tag{2.5}$$

- Multiplicación de punto o multiplicación escalar:

  La multiplicación Q = kP de un punto de la curva P con un escalar k es la suma del punto P consigo mismo k veces. Esta operación es la que permite implementar las diversas funciones de seguridad. Por ello, muchos procesadores criptográficos se diseñan para implementar solo esta operación.

2.2.4. Algoritmo de multiplicación de punto

Al realizar las operaciones aritméticas de curva, las coordenadas de los puntos se pueden representar de dos formas: affine o projective. El primero es la manera normal de representar a la coordenada, es decir, con un valor binario (X,Y). En la segunda forma, el punto se representa por un valor ternario (X,Y,Z) donde $Z \neq 0$ y tiene una correspondencia con las coordenadas affine $\left(\frac{X}{Z^c}, \frac{X}{Z^d}\right)$. Los métodos projective más comunes son el estándar (c=1, d=1), Jacobeans (c=2, d=3) y Lopez-Dahab (c=1, d=2) [14].

La mayoría de algoritmos de multiplicación de punto no solo hacen uso de la suma de punto (SP) sino también de la duplicación de punto (DP) para disminuir la cantidad de operaciones de campo finito. En el caso de dispositivos de bajos recursos los algoritmos binarios, donde se opera de manera iterativa con cada bit de k, ofrecen una mejor optimización de recursos. La manera más utilizada de calcular la multiplicación escalar, debido a su simplicidad y eficiencia,

es el algoritmo Double-and-Add (algoritmo 1). En este método se recorre cada bit de k, realizando una operación SP si el bit es 1 o DP en caso contrario.

Algorithm 1 Calculating the Scalar Multiplication Operation Using Double-and-Add Algorithm

Input: Prime $p \in E(\mathbb{F}_q)$, $P = (x, y)$, where $x, y \in GF(p)$
$k \in \mathbb{Z}, 0 < k < \#E$, $k = (k_{l-1}, k_{l-2}, \ldots, k_0)_2$, $k_{l-1} = 1$
Output: $Q = (x', y')$
$Q = P$
for $i = l - 2$ **downto** 0 **do**
$Q = 2Q$
if $k_i = 1$ **then**
$Q = Q + P$
return Q

Sin embargo, este algoritmo es vulnerable a ataques simples de análisis de consumo energía, un tipo de ataque de canal lateral. Un atacante puede predecir el valor de k analizando las variaciones en la potencia disipada por el dispositivo, ya que la potencia al realizar la operación SP será diferente que al realizar DP. De esta manera se puede inferir el valor de cada bit de 'k', lo cual es crítico ya que 'k' suele ser la llave.

Un algoritmo de multiplicación de punto que se está volviendo popular debido a que no es vulnerable a este tipo de ataques es el Montgomery Ladder (algoritmo 2). En este método también se recorre cada bit de k, pero independientemente de su valor siempre se realiza una operación SP y DP consiguiendo así un trazo de potencia regular y uniforme.

Algorithm 2 Calculating the Scalar Multiplication Operation Using Montgomery Ladder Algorithm

Input: Prime $p \in E(\mathbb{F}_q)$, $P = (x, y)$, where $x, y \in GF(p)$
$k \in \mathbb{Z}, 0 < k < \#E$, $k = (k_{l-1}, k_{l-2}, \ldots, k_0)_2$, $k_{l-1} = 1$
Output: $Q = (x', y')$
$Q = P$
$P = 2Q$
for $i = l - 2$ **downto** 0 **do**
if $k_i = 1$ **then**
$Q = Q + P$
$P = 2P$
else
$P = P + Q$
$Q = 2Q$
return Q

Algorithm 3 Calculating the Montgomery Ladder Algorithm with co-Z addition formulas

Input: Prime $p \in E(\mathbb{F}_q)$, $P = (X, Y, Z)$,
where $X, Y, Z \in GF(p)$, $k \in \mathbb{Z}, 0 < k < \#E$,
$k = (k_{l-1}, k_{l-2}, \ldots, k_0)_2$, $k_{l-1} = 1$
Output: $Q = (X', Y', Z)$

```
P, Q = DBLU(P)
for i = l − 2 downto 0 do
    if k_i = 1 then
        Q, P = ZADDC(P, Q)
        P, Q = ZADDU(Q, P)
    else
        P, Q = ZADDC(Q, P)
        Q, P = ZADDU(P, Q)
return Q
```

2.2.5. Algoritmo de suma y duplicación de punto

Dependiendo del tipo de coordenadas, affine o projective, que se utilicen se adaptarán las ecuaciones (2.2) y (2.3) para la operación SP y (2.4) y (2.5) para la DP. En un trabajo realizado por Melodi [17], se demuestra que la suma de dos puntos puede ser acelerada si los puntos P y Q sumados comparten la misma coordenada Z. De esta manera introduce una nueva fórmula a la que llama Suma de punto con actualización (ZADDU). Esta operación tiene la ventaja adicional de que sin un costo extra también calcula la diferencia de punto. Entonces, la duplicación de punto se puede obtener como una suma seguido de una sustracción: Q = P+Q, R = P-Q, P= R+Q=2P. El primer paso se calcula con la fórmula ZADDU, mientras que los dos siguientes con la fórmula conjugada (ZADDC). Luego, el algoritmo de Montgomery ladder debe ser ligeramente modificado para adaptarse a estas nuevas fórmulas (algoritmo 3) Una explicación más detallada se puede encontrar en [17].

Require: $\boldsymbol{P} = (X_1 : Y_1 : Z)$ and $\boldsymbol{Q} = (X_2 : Y_2 : Z)$
Ensure: $(\boldsymbol{R}, \boldsymbol{P}) \leftarrow \text{ZADDU}(\boldsymbol{P}, \boldsymbol{Q})$ where $\boldsymbol{R} \leftarrow \boldsymbol{P}+\boldsymbol{Q} = (X_3 : Y_3 : Z_3)$ and $\boldsymbol{P} \leftarrow (\lambda^2 X_1 : \lambda^3 Y_1 : Z_3)$ with $Z_3 = \lambda Z_1$ for some $\lambda \neq 0$

```
function ZADDU(P, Q)
    C ← (X1 − X2)^2
    W1 ← X1C; W2 ← X2C
    D ← (Y1 − Y2)^2; A1 ← Y1(W1 − W2)
    X3 ← D − W1 − W2; Y3 ← (Y1 − Y2)(W1 − X3) − A1; Z3 ← Z(X1 − X2)
    X1 ← W1; Y1 ← A1; Z1 ← Z3
end function
```

Fig. 2.5: Algoritmo ZADDU [17]

Require: $\boldsymbol{P} = (X_1 : Y_1 : Z)$ and $\boldsymbol{Q} = (X_2 : Y_2 : Z)$
Ensure: $(\boldsymbol{R}, \boldsymbol{S}) \leftarrow \text{ZADDC}(\boldsymbol{P}, \boldsymbol{Q})$ where $\boldsymbol{R} \leftarrow \boldsymbol{P}+\boldsymbol{Q} = (X_3 : Y_3 : Z_3)$ and $\boldsymbol{S} \leftarrow \boldsymbol{P}-\boldsymbol{Q} = (\overline{X_3} : \overline{Y_3} : Z_3)$

function ZADDC($\boldsymbol{P}, \boldsymbol{Q}$)
$C \leftarrow (X_1 - X_2)^2$
$W_1 \leftarrow X_1 C$; $W_2 \leftarrow X_2 C$
$D \leftarrow (Y_1 - Y_2)^2$; $A_1 \leftarrow Y_1(W_1 - W_2)$
$X_3 \leftarrow D - W_1 - W_2$; $Y_3 \leftarrow (Y_1 - Y_2)(W_1 - X_3) - A_1$; $Z_3 \leftarrow Z(X_1 - X_2)$
$\overline{D} \leftarrow (Y_1 + Y_2)^2$
$\overline{X_3} \leftarrow \overline{D} - W_1 - W_2$; $\overline{Y_3} \leftarrow (Y_1 + Y_2)(W_1 - \overline{X_3}) - A_1$
end function

Fig. 2.6: Algoritmo ZADDC [17]

2.3. Análisis del consumo energético en un FPGA

En circuitos electrónicos se suelen analizar dos tipos de consumo energético: potencia estática y potencia dinámica. Sin embargo, en los dispositivos FPGA volátiles se deben tener en cuenta además otros dos tipos de consumo: potencia de inicio y potencia de arranque.

2.3.1. Potencia inicio y de arranque

Los FPGA volátiles se componen de bloques que están basados en celdas SRAM. Suelen tener una memoria no volátil externa para almacenar la configuración del diseño. Cada vez que el FPGA se energiza, se debe cargar la configuración de esta memoria. Esto requiere cierta cantidad de energía que genera un pico de corriente mucho mayor a la corriente de operación del FPGA por algunos milisegundos.

Los FPGA no volátiles, por otra parte, están basados en tecnología Flash por lo que tienen una significativa ventaja, ya que la configuración no se debe cargar cada vez que se energiza el dispositivo evitando así los altos picos de corriente.

2.3.2. Potencia estática

Esta potencia se refiere al que se observa cuando el FPGA está energizado, pero no está realizando ninguna operación. Este consumo se debe a la corriente de polarización y es resultado de las fugas de corriente en los transistores. Además, está fuertemente influenciado por la tecnología del dispositivo, como la tecnología del transistor o el proceso de fabricación. Los factores que se pueden controlar para disminuirlo son el voltaje de alimentación y la temperatura del ambiente. Los FPGA basados en SRAM están compuestos de celdas de SRAM

que se componen de seis transistores por celda que resulta en una sustancial fuga de corriente. En comparación, los FPGA basados en Flash consisten en solo un transistor por celda con hasta mil veces menor corriente de fuga. Por ello, este tipo de dispositivo se vuelve perfecto para aplicaciones de bajo consumo energético.

2.3.3. Consumo dinámico

Este consumo se debe principalmente, a que en las transiciones de los estados del transistor se genera un pequeño corto que produce un pico de corriente. El consumo dinámico se puede modelar utilizando la fórmula $P = CV^2F$, donde V es el voltaje de operación, F la frecuencia y C la capacitancia de carga. De estos tres factores, el voltaje de operación y la capacitancia dependen del dispositivo que se utilice, mientras que la frecuencia equivalente depende del diseño y sobre el cual se enfocan las técnicas para reducir el consumo dinámico. El propósito principal es reducir las transiciones de estados del transistor.

a) Gating clock

Esta técnica consiste en deshabilitar la señal de reloj de los módulos que están siendo usados. De esta manera la frecuencia en el módulo se vuelve cero y el consumo dinámico también. Para deshabilitar la señal de reloj se suele utilizar una compuerta AND a la entrada de la señal de reloj del bloque. Sin embargo, es necesario que esta compuerta genere un retraso muy bajo porque puede ocasionar problemas de sincronización (Clock skew) y es por ello que se utilizan compuertas especiales que incorporan los FPGA.

2.4. FPGA Igloo Nano

La familia de FPGA IGLOO ofrece los dispositivos de menor consumo energético. Estos FPGA están basados en FLASH (no volátil) y operan con voltajes entre 1.2 y 1.5 V. La principal ventaja de este dispositivo es que permite el uso de modos energía en los que consume menos energía a costa de deshabilitar ciertos módulos y funciones. En el modo ultra-low-power que es el de menor consumo de energía puede llegar a consumir hasta 5 uW mientras mantiene los datos de los registros y SRAM. Este modo se activa a través de un pin especial. De esta manera puede ser controlado por un dispositivo externo.

Capítulo 3

Conclusiones

En este capítulo se presentarán algunos aspectos que se deberían considerar para el diseño de una arquitectura de un procesador criptográfico para la plataforma WISP. Estos resultados se han obtenido a partir de lo establecido en los capítulos anteriores.

3.1. Esquema general de un Procesador de Curvas Elípticas

Se muestra en Fig. 3.1 la arquitectura general de un procesador de Curvas Elípticas. Este esquema se divide en tres partes. Por un lado, se tiene la interfaz de entrada-salida que se encarga de la comunicación con el microcontrolador. Luego, está el datapath que realiza las operaciones aritméticas modulares como (suma, resta, multiplicación e inversión modulares) y las operaciones de puntos (suma de punto y duplicación de punto). Por último, la unidad de control que ejecuta el algoritmo de multiplicación de punto descrito en el capítulo anterior.

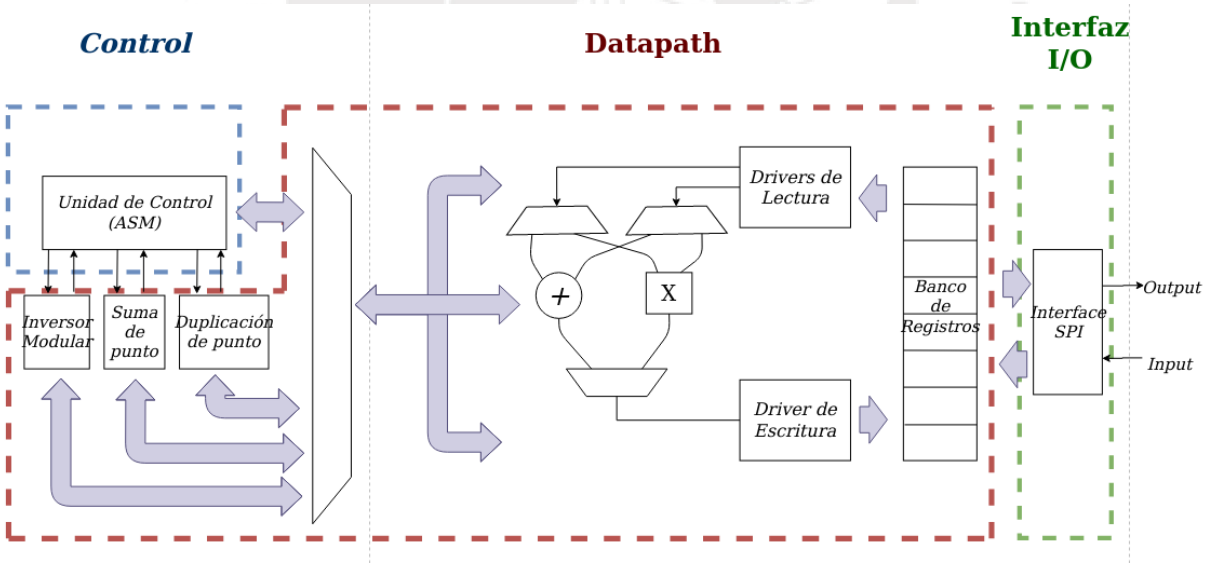

Fig. 3.1: Diagrama general de un procesador ECC

3.3. La interfaz SPI

Se plantea el uso del protocolo SPI para la transferencia de datos con el microcontrolador MSP430 ya que consume menos energía que otros protocolos seriales. De esta manera, el microcontrolador envía los valores de ‘k’, ‘x’ e ‘y’ concatenados, donde (x,y) son las

coordenadas del punto y k la constante a multiplicar. Además, se propone el uso de una señal "DONE" que indique la finalización de la operación del procesador.

En la Fig. 3.2, se observa cómo sería la comunicación con una configuración de SPI en modo 2: CPOL = 1 y CPHA = 0. Esto significa que el estado de la señal del reloj SCLK permanecerá en estado lógico alto cuando está inactivo y la información se enviará en cada transición de bajo a alto.

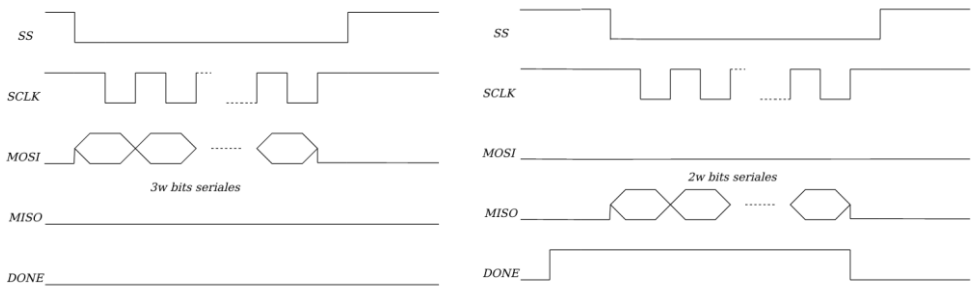

Fig. 3.2: a) iniciando el procesador b) devolviendo los datos procesados

3.4. El datapath

El datapath debe realizar operaciones aritméticas que se requieren en el algoritmo criptográfico. Estas operaciones son de dos tipos: operaciones modulares como suma modular, resta modular, multiplicación modular e inversión modular, y operaciones de punto como suma de punto y duplicación de punto que son una secuencia de operaciones modulares.

Algunos aspectos a considerar en el diseño del datapath se muestran a continuación.

a) Circuito inversor modular

Aprovechando que el módulo M siempre es un número primo en las criptografías de curva elíptica y que por ello no tendrá ningún divisor común con otro número, se puede utilizar el teorema de Fermat para calcular la inversa:

$$a^{-1} \bmod M = a^{M-2} \bmod M, \;\; si \;\; mcd(a, M) = 1$$

De esta manera se evita utilizar un circuito inversor que consume muchos recursos y se usa en vez un circuito exponenciador modular aprovechando el circuito multiplicador montgomery modular.

El circuito exponenciador se puede realizar basándose en el algoritmo cuadrado y multiplicación repetitivas [18].

b) Circuito Sumador-Restador modular

El sumador-restador modular se puede implementar de manera eficiente utilizando el "método de Omura". Con este método, la suma modular se implementa como lo indica la ecuación 3.1 y la resta modular como la 3.2. La ventaja de esta forma es que permite la implementación de ambas operaciones utilizando el mismo circuito si los números son representados en complemento a dos.

$$S = A + B \,(mod\, p) = \begin{cases} A + B - p, & A + B \geq 2^m \\ A + B & otherwise \end{cases} \quad \text{..... (3.1)}$$

$$S = A - B \,(mod\, p) = \begin{cases} A - B + p, & if\ A - B < 0 \\ A - B & otherwise \end{cases} \quad \text{..... (3.2)}$$

c) Circuito multiplicador modular Montgomery

La multiplicación modular es el circuito que define en mayor medida la eficiencia del diseño. Por ello es esencial un circuito optimizado para operaciones con números con grandes tamaños de palabra. Tenca and Koç [19] introducen un algoritmo basado en palabras para la multiplicación montgomery al que llamó Multiple-Word Radix-2 Montgomery Multiplication (MWR2MM) Fig. 3.3. En este algoritmo, Y y M se dividen en partes de w bits de esta manera cada bit de X puede multiplicar por partes a Y.

Input: odd $M, n = \lfloor \log_2 M \rfloor + 1$, word size w, $e = \lceil \frac{n+1}{w} \rceil$, $X = \sum_{i=0}^{n-1} x_i \cdot 2^i$, $Y = \sum_{j=0}^{e-1} Y^{(j)} \cdot 2^{w \cdot j}$, $M = \sum_{j=0}^{e-1} M^{(j)} \cdot 2^{w \cdot j}$, with $0 \leq X, Y < M$
Output: $Z = \sum_{j=0}^{e-1} S^{(j)} \cdot 2^{w \cdot j} = \mathrm{MP}(X, Y, M) \equiv X \cdot Y \cdot 2^{-n} \pmod{M}$, $0 \leq Z < 2M$

```
S = 0
for i = 0 to n − 1 do
    q_i = (x_i · Y_0^(0)) ⊕ S_0^(0)
    (C^(1), S^(0)) = x_i · Y^(0) + q_i · M^(0) + S^(0)
    for j = 1 to e do
        (C^(j+1), S^(j)) = C^(j) + x_i · Y^(j) + q_i · M^(j) + S^(j)
        S^(j−1) = (S_0^(j), S_{w−1...1}^(j−1))
    S^(e) = (0, S_{w−1...1}^(e))
return Z = S
```

Fig. 3.3: Algoritmo MWR2MM

Utilizando este algoritmo se pueden paralelizar operaciones al utilizar una arquitectura de matriz sistólica que permite un número variable de elementos de procesamiento (PE) que se ejecutan en paralelo. Además, en [3], Huang et al. propone una arquitectura optimizada para este algoritmo que permite realizar la multiplicación modular de números de n bits de precisión en aproximadamente n ciclos de reloj.

Recomendaciones y Trabajos futuros

- En un futuro se plantea realizar el diseño e implementación de una arquitectura de un procesador criptográfica de Curvas Elípticas en un FPGA a partir de las conclusiones obtenidas de este trabajo.

- Se recomienda el uso de la multiplicación Montgomery como algoritmo de multiplicación modular, ya que permite operar número de grandes tamaños sin requerir gran cantidad de área. Esto lo vuelve ideal para dispositivos de bajos recursos como WISP.

- En el caso que se llegara a hacer una implementación, se recomienda verificar el funcionamiento del procesador criptográfico en un FPGA junto con la plataforma WISP. Para ello se debería modificar parte del código core de WISP que realiza la comunicación con la antena para integrar la encriptación con el protocolo de comunicación. Además, se recomienda evaluar experimentalmente la distancia máxima que se puede alcanzar al tener el PFGA en la plataforma WISP y cómo influye en su consumo energético.

Referencias

[1] A. Salman *et al.*, “A Scalable ECC Processor Implementation for High-Speed and Lightweight with Side-Channel Countermeasures.”

[2] L. Batina, B. Preneel, J. Vandewalle, and S. B. Ors, “Hardware Implementation of an Elliptic Curve Processor over GF (p),” pp. 1–11, 2003.

[3] M. Huang, K. Gaj, and T. El-ghazawi, “New Hardware Architectures for Montgomery Modular Multiplication Algorithm,” vol. 60, no. 7, pp. 923–936, 2011.

[4] Statista Research Department, “Global Internet of Things market 2014-2020, by industry,” *Global Internet of Things market 2014-2020, by industry*, 2015. https://www.statista.com/statistics/512673/worldwide-internet-of-things-market/ (accessed Jul. 22, 2020).

[5] X. Jia, Q. Feng, T. Fan, and Q. Lei, “RFID technology and its applications in Internet of Things (IoT),” in *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, Apr. 2012, pp. 1282–1285, doi: 10.1109/CECNet.2012.6201508.

[6] S. S. Anjum *et al.*, “Energy Management in RFID-Sensor Networks: Taxonomy and Challenges,” *IEEE Internet Things J.*, vol. 6, no. 1, pp. 250–266, 2019, doi: 10.1109/JIOT.2017.2728000.

[7] D. J. Yeager, “Development and Application of Wirelessly-Powered Sensor Nodes,” 2009.

[8] S. Naderiparizi, A. N. Parks, Z. Kapetanovic, B. Ransford, and J. R. Smith, “WISPCam: A battery-free RFID camera,” *2015 IEEE Int. Conf. RFID, RFID 2015*, pp. 166–173, 2015, doi: 10.1109/RFID.2015.7113088.

[9] D. J. Yeager, J. Holleman, R. Prasad, J. R. Smith, and B. P. Otis, “NeuralWISP: A wirelessly powered neural interface with 1-m range,” 2009, doi: 10.1109/TBCAS.2009.2031628.

[10] C. Pendl, M. Pelnar, and M. Hutter, “Elliptic curve cryptography on the WISP UHF RFID tag,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7055 LNCS, pp. 32–47, 2012, doi: 10.1007/978-3-642-25286-0_3.

[11] A. D. Rayome, “As IoT attacks increase 600% in one year, businesses need to up their security,” *Security*, 2018. .

[12] A. Ibrahim and G. Dalkiliç, “An Advanced Encryption Standard Powered Mutual

Authentication Protocol Based on Elliptic Curve Cryptography for RFID, Proven on WISP," *J. Sensors*, vol. 2017, 2017, doi: 10.1155/2017/2367312.

[13] E. R. Naru, H. Saini, and M. Sharma, "A recent review on lightweight cryptography in IoT," 2017, doi: 10.1109/I-SMAC.2017.8058307.

[14] B. Rashidi, "A Survey on Hardware Implementations of Elliptic Curve Cryptosystems," no. December, pp. 1–61, 2017, [Online]. Available: http://arxiv.org/abs/1710.08336.

[15] D. Yeager, F. Zhang, A. Zarrasvand, N. T. George, T. Daniel, and B. P. Otis, "A 9 μA, addressable Gen2 sensor tag for biosignal acquisition," *IEEE J. Solid-State Circuits*, vol. 45, no. 10, pp. 2198–2209, 2010, doi: 10.1109/JSSC.2010.2063930.

[16] M. Philipose, J. R. Smith, B. Jiang, A. Mamishev, S. Roy, and K. Sundara-Rajan, "Battery-free wireless identification and sensing," *IEEE Pervasive Comput.*, vol. 4, no. 1, pp. 37–45, 2005, doi: 10.1109/MPRV.2005.7.

[17] R. R. Goundar, M. Joye, and A. Miyaji, "Co- Z Addition Formulæ and Binary Ladders on (Extended Abstract)," pp. 65–79.

[18] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. 1996.

[19] A. F. Tenca and Ç. K. Koç, "A scalable architecture for montgomery multiplication," 1999, doi: 10.1007/3-540-48059-5_10.

[20] J. Cavanagh, *Verilog HDL*. 2017.

[21] X. Tan, M. Dong, C. Wu, K. Ota, J. Wang, and D. W. Engels, "An Energy-Efficient ECC Processor of UHF RFID Tag for Banknote Anti-Counterfeiting," *IEEE Access*, vol. 5, no. c, pp. 3044–3054, 2017, doi: 10.1109/ACCESS.2016.2615003.

[22] A. Shantha, J. Renita, and N. Edna Elizabeth, "Analysis and implementation of ECC algorithm in lightweight device," *Proc. 2019 IEEE Int. Conf. Commun. Signal Process. ICCSP 2019*, pp. 305–309, 2019, doi: 10.1109/ICCSP.2019.8697990.

[23] D. Hein and M. Br, "Elliptic Curve Cryptography ASIC for Radio Frequency Authentication Master Thesis," 2008.

[24] D. Ma, N. Saxena, T. Xiang, and Y. Zhu, "Location-aware and safer cards: Enhancing RFID security and privacy via location sensing," *IEEE Trans. Dependable Secur. Comput.*, vol. 10, no. 2, pp. 57–69, 2013, doi: 10.1109/TDSC.2012.89.