Arquitectura Hardware de un Criptosistema de Curva Elíptica con Compresión de Datos

Miguel Morales Sandoval, Claudia Feregrino Uribe
Coordinación de Ciencias Computacionales
mmorales@inaoep.mx, cferegrino@inaoep.mx

RESUMEN

La investigación que se presenta consiste en diseñar e implementar una arquitectura hardware de un sistema de cifrado/descifrado de datos mediante criptografía de curva elíptica que opere en tiempo real. Los datos a cifrarse se comprimen primero de manera transparente mediante un algoritmo de compresión sin perdida también implementado en hardware. Una de las ventajas de este enfoque es el poder mejorar el rendimiento del sistema al reducir la cantidad de información a cifrarse, además de beneficiarse de transmitir más información con el mismo ancho de banda disponible.

I INTRODUCCIÓN

La criptografía convierte información inteligible en algo totalmente ininteligible permitiendo la confidencialidad de los datos. La compresión reduce la cantidad de información y con ello, los costos de espacio de almacenamiento y tiempo de transferencia. La combinación de estas dos tecnologías importantes en las aplicaciones en red no ha sido explotada ampliamente. Compresión y criptografía han sido consideradas como tecnologías opuestas: mientras que los algoritmos de criptografía esconden patrones de información predecible, los algoritmos de compresión buscan esos patrones de información para sustituirlos por tokens de longitud menor [9]. La compresión y la criptografía deben realizarse en el orden correcto, se aplica primero compresión a los datos y después se realiza el proceso de cifrado. Si primero se realizará el cifrado de los datos y a continuación la compresión, no se tendrían buenos resultados debido a que los datos cifrados presentan muy poca redundancia con lo que la compresión sería pobre.

El uso de curvas elípticas en criptografía de llave pública [2] fue propuesto independientemente por Victor Miller y Neal Koblitz en 1985 [1]. Criptografía de Curva Elíptica (ECC) ofrece la principal ventaja sobre el criptosistema ampliamente usado y aceptado RSA [2] de operar con llaves de longitud de hasta 6 veces más cortas, ofreciendo los mismos niveles de seguridad que RSA con longitud de llave de 1024 bits. A pesar de las ventajas que ECC ofrece, entre matemáticos y científicos aún existe escepticismo alrededor del uso de ECC en aplicaciones grandes de criptografía. La seguridad de ECC no ha sido probada, sino que la fortaleza radica en la incapacidad de encontrar ataques. ECC ofrece los más altos niveles de seguridad dentro de los criptosistemas de llave pública ya que el mejor algoritmo que se conoce para resolver el problema matemático, en el que ECC basa su seguridad, es de complejidad exponencial.

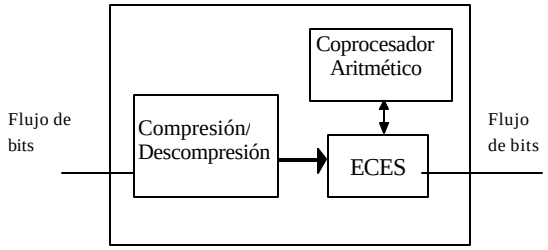

Figura 1 Arquitectura propuesta

Tanto los algoritmos de compresión como los algoritmos criptográficos realizan operaciones computacionales intensivas y sus implementaciones en software no satisfacen los requerimientos para sistemas donde el procesamiento de los datos debe realizarse "al vuelo".

La investigación que aquí se realiza consiste en implementar en hardware de manera eficaz una arquitectura que realice el cifrado/descifrado de datos en tiempo real aplicando criptografía de curva elíptica. Una de las principales razones para realizar una implementación en hardware es la velocidad de operación ya que los algoritmos criptográficos son caros computacionalmente y se ejecutan ineficientemente en procesadores de propósito general. Un diagrama general de la arquitectura propuesta se muestra en la figura 1.

Para mejorar el rendimiento del sistema, la información a cifrarse se comprime primero mediante un algoritmo de compresión sin pérdida. Además de reducirse la información a procesarse, la compresión desaparecerá patrones en los datos de entrada que pudieran ser empleados por criptoanalistas para recuperar la información original. El diseño del sistema se realizará con un lenguaje de descripción de hardware y la implementación de la arquitectura hardware se realizará en un dispositivo lógico programable (FPGA).

II COMPRESIÓN DE DATOS

La compresión de datos reduce la cantidad de información reduciendo los costos de almacenamiento y transmisión. En la literatura se han propuesto varias técnicas para realizar compresión de datos [3], ninguna de estas técnicas es óptima para todos los tipos de datos. Sin embargo, existen técnicas que se han propuesto y probado para diferentes tipos de datos obteniendo una razón de compresión promedio de alrededor del 50%. Los métodos de compresión sin pérdida permiten recuperar íntegramente la información original a partir de la información comprimida. Existen dos enfoques para realizar compresión sin pérdida: mediante métodos estadísticos y mediante métodos basados en diccionarios. Entre los métodos estadísticos destacan Codificación de Huffman, Codificación Aritmética y PPM [3]. Los métodos estadísticos consiguen las mejores razones de compresión asignando códigos de longitud variable a los símbolos de entrada de acuerdo a su frecuencia. Debido a que se requieren las estadísticas de los símbolos antes de realizar la codificación, los requerimientos de espacio y tiempo elevados [10].

Por otra parte, los métodos basados en diccionario realizan la compresión sustituyendo ocurrencias de subcadenas en la entrada por índices de un diccionario donde se almacenan las subcadenas que ya han aparecido previamente. Los métodos basados en diccionarios no logran las razones de compresión de los métodos estadísticos pero sus implementaciones son menos costosas en cuanto tiempo y espacio. Una característica importante de los métodos basados en diccionario es la rapidez para descomprimir los datos. La mayoría de los métodos basados en diccionarios son variantes de los trabajos de Jacob Ziv y Abraham Lempel en 1977 y 1978 [3].

III CRIPTOGRAFÍA DE CURVA ELÍPTICA

Una curva elíptica es un conjunto de pares ordenados que satisfacen una relación. Esta relación se define sobre elementos de un campo finito [1]. En cuestiones de criptografía, los campos finitos sobre los que se definen las curvas elípticas son el campo de los números primos F_p y el campo de las cadenas de m bits F_2^m. Las curvas elípticas definidas sobre los campos F_p o F_2^m, forman un grupo abeliano sobre la operación de suma. Con esto, la suma de cualquiera dos puntos de la curva elíptica resulta ser otro punto de la curva elíptica. Otra característica importante de las curvas elípticas es que el problema del logaritmo discreto definido sobre los puntos de la curva elíptica es intratable, es decir, el mejor algoritmo conocido es de complejidad exponencial. La curva elíptica sobre F_p está dada por la ecuación (1) mientras que aquella para F_2^n está dada por la ecuación (2). Los campos finitos, tanto F_p como F_2^m, tienen reglas bien definidas para realizar las operaciones de suma, multiplicación e inversión entre sus elementos. Cada una de estas reglas es diferente de acuerdo al tipo de representación que se maneje para los elementos del campo finito. La forma de realizar las operaciones de suma entre los puntos de una curva elíptica varía dependiendo del tipo de representación que se utilice, la cual puede ser mediante coordenadas proyectivas, affine o jacobianas [1].

$$y^2 = x^3 + ax + b \qquad (1)$$
$$y^2 + xy = x^3 + ax^2 + b \qquad (2)$$

El algoritmo para cifrado ECES (Elliptic Curve Encryption Standard) se encuentra descrito en el primer borrador del estándar IEEE P1363 (Standard Specifications For Public-Key Cryptography). El algoritmo ECES requiere de una curva elíptica C y un punto particular de esa curva P. Suponga que A desea enviar un mensaje cifrado a B. Ambas entidades cuentan una llave privada (un elemento del campo finito) y una llave pública (un punto de la curva elíptica). Sean K_A y K_B las llaves privadas de A y B respectivamente. Sean P_A y P_B las llaves públicas de A y B respectivamente.

1. Realiza la operación $K_A * P_B = (Xc, Yc)$
2. Genera una cadena M' de bits de longitud L utilizando una Función de Generación de Máscara a partir de Xc.
3. El mensaje cifrado es: M XOR M'.

El receptor B descifra la información aplicando las siguientes acciones:

1. Realiza la operación $K_B * P_A = (Xc, Yc)$

2. Genera M' de longitud L utilizando una Función de Generación de Máscara.
3. El mensaje se descifra realizando M XOR M'.

La operación K*P = P+P+P+…+P, es la operación más demandante computacionalmente del algoritmo ECES e involucra operaciones entre puntos de la curva elíptica y operaciones entre los elementos del campo finito. Esta operación suele implementarse como una adición repetitiva del mismo punto P, K veces. Por ejemplo, si el campo finito sobre el que se está trabajando es F_2^n, la operación P+ P = P', donde P = (x_1,y_1) y P' = (x_2,y_2) se realiza de acuerdo a las ecuaciones (3-5).

$$x_2 = \lambda^2 + \lambda + a \quad (3)$$

$$y_2 = \lambda(x_1 + x_2) + x_2 + y_1 \quad (4)$$

$$\lambda = x_1 + (x_1/y_1) \quad (5)$$

La operación K*P requiere de K-1 sumas entre dos puntos de una curva elíptica, en cada suma, se requiere de una inversión, dos multiplicaciones, una elevación al cuadrado y seis sumas en el campo finito empleado. La operación de inversión es la de mayor costo computacional, en implementaciones software, equivale a la realización de 24 mu ltiplicaciones [4].

IV TRABAJO RELACIONADO

Las arquitecturas propuestas para compresión de datos sin pérdida pueden dividirse en dos enfoques principales: mediante CAM'S (Content Addressable Memory) y mediante arreglos sistólicos. En [5], se implementa el método estadístico de codificación de Huffman dinámico, el cual, realiza la compresión asignando códigos de longitud variable a los símbolos de entrada. Los códigos se asignan recorriendo un árbol binario que se construye conforme los datos se leen. Los símbolos menos frecuentes serán hojas de las ramas más profundas del árbol y tendrán los códigos de longitud más grande. Los símbolos más frecuentes se situarán en las hojas de las ramas menos profundas y tendrán códigos más cortos. La arquitectura propuesta se basa en CAM's, contiene alrededor de 17,700 compuertas, opera a 40 MHz, tiene un rendimiento de 40 Mbps y logra una razón de compresión del 50%. En [6], se implementa el algoritmo LZ77, el cual es un método basado en diccionario. Se tiene un arreglo de elementos de procesamiento que buscan la subcadena de mayor longitud en el diccionario. La arquitectura ocupa alrededor de 11000 transistores con un rendimiento de 90 Mbps.

No existen implementaciones reportadas de algoritmos de cifrado mediante ECC. Solo se han realizado implementaciones, tanto en hardware como en software, de otros protocolos como lo es el intercambio de llaves y la firma digital [7].

En la literatura se han presentado varios trabajos que realizan la operación kP, donde k es un elemento de F_p o F_2^m, y P es un punto de la curva elíptica. En [8] se ha reportado la implementación en FPGA más rápida para realizar la operación kP. La arquitectura propuesta explota las características del cómputo reconfigurable para poder operar con diferentes curvas y ambos campos finitos F_p y F_2^m. La operación kP se realiza en 0.21 ms operando a 75 MHz.

En lo que se refiere a la conjunción de compresión con criptografía, solo se tienen algunos trabajos realizados por empresas como HiFn, PKWare y CISCO. En PKWare, al programa de compresión PKZip v6 se le agregan capacidades de cifrado. El algoritmo utilizado para realizar las operaciones criptográficas es RSA. En HiFn, se aborda el problema que se tiene al aplicar cifrado a la carga útil de los paquetes que se transmiten en red. El algoritmo empleado para compresión es LZS [9], un método basado en diccionario y los algoritmos para el cifrado de la información son el DES y el 3DES, ambos, algoritmos de criptografía de llave privada [2]. En CISCO se incorporan módulos para compresión y cifrado junto con enrutadores para redes de comunicación para mejorar el rendimiento en la transmisión de datos. La carga útil de los paquetes se comprime según el protocolo de compresión IPPCP [11] o mediante el algoritmo LZS. La información comprimida se cifra con el algoritmo AES (llave privada) con longitudes de llave de 128, 192 y 256 bits.

La arquitectura que se propone en esta investigación es inovativa ya que no existe una implementación en hardware reportada de un algoritmo de cifrado mediante ECC. De igual forma, no se han reportado arquitecturas para cifrado que integren compresión en el mismo chip.

V TRABAJO REALIZADO

A la fecha se ha realizado una evaluación de métodos de compresión sin perdida, tanto estadísticos como basados en diccionarios. Las métricas de evaluación son los requerimientos de memoria, el tiempo de ejecución y la razón de compresión obtenida. Esta evaluación ayuda a decidir el tipo de algoritmo que se implementará en la arquitectura propuesta. De los resultados obtenidos, una solución basada en diccionarios parece ser la mejor, ya que obtiene razón de compresión cercana a la obtenida por compresores estadísticos (la diferencia es alrededor de 0.1).

.

Se han revisado y analizado diferentes implementaciones en hardware de métodos de compresión sin pérdida, tanto estadísticos como basados en diccionarios. Se han identificado el área necesaria, el rendimiento obtenido y el diseño de las arquitecturas. Por otra parte, se ha realizado la implementación en software del algoritmo ECES. De esta implementación, se ha medido el tiempo de ejecución de forma modular. .Se han identificado las partes más pesadas computacionalmente. Esta implementación será una base para comenzar a realizar el diseño de la arquitectura hardware que implemente el algoritmo ECES. En la tabla 1 se muestran los tiempos de ejecución para el cifrado de dos archivos con y sin compresión previa.

Tamaño	Tc	Tamaño Comprimido	Tg	Tcf	Tsc
152,089	0.05	72.322	2.22	0.04	0.1
8,673,375	3.18	4.204.995	2.57	2.37	4.65

Tabla 1. Tiempo de ejecución de ECES

En la tabla 1, Tc es el tiempo en realizar la compresión, Tg es el tiempo en generar la curva elíptica y las llaves pública y privada. Tcf es el tiempo en realizar el cifrado de los datos con compresión y Tsc es el tiempo en realizar el cifrado sin compresión. Los tamaños de los archivos están dados en bytes y los tiempos de ejecución están expresados en segundos. Al reducir la cantidad de información, el tiempo para realizar el cifrado se reduce también cerca del 50%. De la tabla 1 se observa que los tiempos para comprimir y cifrar son elevados. La generación de la curva, el cálculo de K*P y la compresión de la información son las partes que más tiempo consumen y donde se hace evidente una optimización mediante soluciones en hardware.

VI CONCLUSIONES

El estudio de la conjunción de compresión con criptografía no ha sido explotado ampliamente. La compresión ofrece dos ventajas a la criptografía al reducir la cantidad de información que el cifrador procesará y al desaparecer patrones predecibles en los datos que pudieran emplearse para descifrar la información. Los avances logrados indican que realizando compresión en tiempo real y realizando las operaciones de ECC en hardware especializado, es posible tener un sistema de cifrado con la más alta seguridad que realice el procesamiento de los datos “al vuelo”.

VII AGRADECIMIENTOS

El autor le agradece al CONACyT el apoyo otorgado a través de la Beca para Estudios de Maestría #171577. Asimismo, agradece al Instituto Nacional de Astrofísica Óptica y Electrónica por los recursos ofrecidos para el desarrollo de esta investigación.

VIII REFERENCIAS

[1] Dahab, R., and López, J., “An Overview of Elliptic Curve Criptography”, Technical Report, IC-00-10, May 2000. Disponible en http://citeseer.nj.nec.com/333066.html

[2] Stallings, William., Cryptography and Network Security, Prentice Hall, NJ, 1999.

[3] Salomon, D., A Guide to Data Compression Methods, Springer-Verlag, New York, 2002.

[4] Lopez, J., and Dahab, R., Improved Algorithms for Elliptic Curve Arithmetic in GF(2^n), SAC’98, LNCS 1556, pp. 201-212, Springer-Verlag, 1998.

[5] Lian-Ying, L et al., CAM-Based VLSI Architectures for Dinamyc Huffman Coding, IEEE Transactions on Consumer Electronics, Vol. 40, No. 3, August 1994.

[6] Jung, B., and Burleson, W., Efficient VLSI for Lempel-Ziv Compression in Wireless Data Communication Networks, IEEE Transactions on Communications, Volume 47, Issue: 9, pp. 1278-1283, September 1999.

[7] Hankerson, D., et al., Software Implementation of elliptic Curve Cryptography Over Binary Fields. Proceedings of CHES2000, pp 1-12.

[8] Orlando, G., and Paar, C., A High-Performance Reconfigurable Elliptic Curve Processor for GF(2^m), CHES 2000, Springer-Verlag, Lecture Notes in Computer Science 1965, 2000

[9] HiFn Inc., The First Book of Compression and Encryption, Hifn Whitepaper, January 2001. Disponible en www.hifn.com/cgi-bin/DocLoc.pl

[10] Witten, I. H., Moffat, A., and Bell, T. C., Managing Gigabytes: Compressing and Indexing Documents and Images, Morgan Kaufmann Publishing, San Francisco, 1999

[11] Shacham, A., et al., RFC 2393-IP Payload Compression Protocol, December 1998. Disponible en www.faqs.org/rfcs/