REVISTA CUBANA
DE TRANSFORMACIÓN DIGITAL

RNPS 2487 • ISSN 2708-3411
Vol. 4 • Nro. 2 • abril-junio 2023 • e198

ARTÍCULO ORIGINAL

Protocolos Asimétricos con Criptografía No Conmutativa y Matrices Elementales

Asymmetric Protocols with Non-Commutative Cryptography and Elementary Matrices

Ernesto Rafael Carbonell Rigores
ernesto@ceis.cujae.edu.cu • https://orcid.org/0000-0001-5722-103X
UNIVERSIDAD TECNOLÓGICA DE LA HABANA "JOSÉ ANTONIO ECHEVERRÍA", CUJAE, CUBA

Ricardo Araujo Rodríguez
ricardo24Dev@gmail.com • https://orcid.org/0000-0001-5743-1045

Recibido: 2023-01-26 • Aceptado: 2023-03-25

RESUMEN

En este trabajo se presenta una implementación de cuatro protocolos criptográficos asimétricos: transporte de claves, intercambio de claves, firma digital y cifrado de mensajes, basada en una propuesta teórica realizada por Juan Pedro Hecht, de la Universidad de Buenos Aires, Argentina. En los cuatro casos se utiliza criptografía no conmutativa (poscuántica), mediante el grupo algebraico de matrices de Hill. Como función de una vía se utiliza el Problema de la Descomposición Simétrica Generalizada (GSDP, por sus siglas en inglés), considerado computacionalmente intratable, lo que obliga a aplicar «fuerza bruta» para vulnerarlo. En la implementación de estos protocolos se necesita solo aritmética de precisión simple, lo que los hace utilizables en dispositivos de bajas prestaciones computacionales. En la propuesta original se requiere generar matrices generales (sin una estructura particular), "aleatorias" no singulares, mediante un método tradicional que implica un elevado costo computacional. En este trabajo se propone, en su lugar, la generación «aleatoria» de matrices elementales, imponiendo ciertas condiciones que garantizan su no singularidad. Para almacenar estas matrices se utilizan solo dos vectores y un escalar, luego, para su generación se realizan menos operaciones. Por otro lado, para el producto de una matriz elemental por una general (o viceversa) y la potenciación de una elemental, se precisa de menos esfuerzo, que para multiplicar dos matrices generales y obtener la potencia de una general, respectivamente. Adi-

cionalmente, para las matrices elementales se transfieren menos datos por la red que para las generales. Estos protocolos son utilizados, de forma práctica, en aplicaciones reales que realizan menos operaciones y almacenan y transfieren por la red menor cantidad de datos.

Palabras clave: criptografía asimétrica, criptografía no conmutativa, matriz elemental, Problema de la Descomposición Simétrica Generalizada, protocolo criptográfico.

ABSTRACT

This paper presents an implementation of four asymmetric cryptographic protocols: key transport, key exchange, digital signature and message cyphering, based on a theoretical proposal made by Juan Pedro Hecht, from the University of Buenos Aires, Argentina. In all four cases, non-commutative (post-quantum) cryptography is used by means of the algebraic group of Hill matrices. As a one-way function, the Generalized Symmetric Decomposition Problem is used, considered as computationally intractable, which requires applying "brute force" to violate it. They require only single-precision arithmetic, which makes them usable on low-performance computational devices. In the original proposal, it is required to generate non-singular "random" general matrices (without a particular structure), by means of a traditional method that implies a high computational cost. In this paper, instead, the "random" generation of elementary matrices is proposed, imposing certain conditions that guarantee their invertibility. To store these matrices, only two vectors and one scalar are used, so fewer operations are performed to generate them. On the other hand, the product of an elementary matrix by a general one (or vice versa) and the power of an elementary one require less effort than to multiply two general matrices and to obtain the power of a general one, respectively. Additionally, for elementary matrices, less data is stored and transferred over the network than for general ones. These protocols are used, in a practical way, in real applications that perform fewer operations and store and transfer less data over the network.

Keywords: *Asymmetric cryptography, cryptographic protocol, elementary matrix, Generalized Symmetric Decomposition Problem, non-commutative cryptography.*

INTRODUCCIÓN

En la actualidad, el diseño de casi todos los sistemas computacionales impone, como requisitos, aspectos de seguridad informática. En ese proceso, los criptógrafos se afanan en preservar la información y los criptoanalistas en vulnerarla, para obtener algún beneficio de ella. Es por eso que los métodos para ocultar y trasmitir información han debido ser cada vez más sofisticados y robustos, en aras de poder hacer frente a los ataques para destruirlos. En tal sentido, las matemáticas y la tecnología se han convertido en pilares fundamentales de la criptografía, para brindar los servicios de seguridad de la información (confidencialidad, integridad, autenticación y no repudio), a través de herramientas (primitivas) criptográficas (Bishop, 2003; Dixon, 1981; Hecht, 2015; Hecht, 2016; Kallenberg, 1975; Mieres, 2009; Pomerance, 1996; Sevillano Castellano, 2018; Sicard, 1999).

Este trabajo tiene como antecedente la propuesta teórica presentada en Hecht (2015), solución novedosa consistente en un conjunto integrado de cinco protocolos criptográficos asimétricos, arbitrados, compactos y seguros, para intercambio y transporte de claves, cifrado de mensajes, firma digital y autenticación, respectivamente. El conjunto resulta integrado, dado que todos comparten un único núcleo algebraico; son protocolos arbitrados, porque incorporan una Tercera Parte de Confianza (TPC) que genera y publica información necesaria para el trabajo de estos; compactos, ya que manipulan matrices de orden ocho, cuyas entradas son congruencias módulo 251; seguros, debido a la aplicación, sobre esas matrices, del Problema de la Descomposición Simétrica Generalizada (GSDP, por sus siglas en inglés), para el que no se han encontrado, hasta el momento, reportes de ataques con algoritmos eficientes en computación clásica ni cuántica. Vale aclarar que, por la elección del módulo 251 (mayor primo representable en 8 bits), las operaciones sobre las entradas matriciales requieren solo aritmética de precisión simple, lo que hace posible implementarlos en aplicaciones para dispositivos de bajas prestaciones computacionales, como ordenadores de 8 o 16 bits, teléfonos móviles y dispositivos de Internet de las Cosas, entre otros. Por otra parte, ninguno trasmite información secreta o sensible por la red, por lo que pueden ser utilizados en redes no seguras (Hecht, 2015).

Hay, sin embargo, algunos aspectos que deben considerarse. En todos ellos se realizan productos, potencias e inversiones matriciales, y cálculos de determinantes, operaciones de elevado esfuerzo computacional; por otro lado, todas las matrices utilizadas son generales (no responden a una estructura o distribución particulares de sus entradas, que pueda aprovecharse para acelerar algunos cálculos) (Golub & Van Loan, 1996; Meyer, 2000).

Para dar solución a estas problemáticas aquí se traza el objetivo de presentar una variante de implementación de los protocolos para intercambio y transporte de claves, firma digital y cifrado de mensajes, así como de aplicaciones reales que ejemplifican el uso de estos en la solución de problemas concretos.

En la implementación que se presenta, se utilizan matrices elementales en sustitución de algunas generales que intervienen en la propuesta original de los protocolos. La representación interna para las matrices elementales consta de dos vectores y un escalar, como se explica más adelante. Luego, para una matriz elemental de orden n se requiere almacenar y transferir por

la red menos valores *(2*n + 1)* que para una general *(n^2)* de su mismo orden. El uso de matrices elementales, además, propicia la reducción de la cantidad de operaciones que realizan los protocolos. Por ejemplo, el producto de una matriz elemental por una general y viceversa tiene orden computacional cuadrático, en contraste con el orden cúbico del producto de dos generales; la inversión de una matriz elemental es de orden lineal, mientras que la de una general es de orden cúbico; la potenciación de una transformación de similitud formada con una matriz diagonal y una elemental tiene orden cuadrático, mientras que la de una formada por una matriz diagonal y una general es de orden cúbico. En esta propuesta, las matrices elementales se introducen para algunos productos e inversiones matriciales y en todas las potenciaciones de transformaciones de similitud. La potencia de dichas transformaciones es la que permite el uso de GSDP.

Luego, el contraste entre el uso de matrices elementales contra el de generales, resulta favorable al primero, porque logra reducir la cantidad de operaciones, y datos para almacenar y transferir por la red en todos los protocolos (Meyer, 2000; Sun, 1996).

METODOLOGÍA

En esta sección se presentan aspectos metodológicos utilizados para abordar la investigación. Se muestran algunos hitos importantes en la historia de la criptografía; además, se describen aspectos relevantes de la propuesta original y los considerados para lograr una implementación eficiente.

Algunos hitos importantes en la historia reciente de la criptografía

En la evolución de los sistemas informáticos y las redes de computadoras se ha hecho cada vez más necesario preservar y trasmitir, de forma segura, la información. Los sistemas han de garantizar que se dé cumplimiento a tres principios fundamentales para la seguridad: la confidencialidad, la integridad y la disponibilidad. En ese sentido, la criptografía juega un rol preponderante (Delgado, De Abiego, Gallegos, & Cabarcas, 2019; Menezes, Van Oorschot, Vanstone, & Rosen, 1996).

La publicación de Diffie & Hellman (1976) se considera uno de los resultados más sorprendentes en la historia de la criptografía, ya que introdujo el concepto de criptografía de clave pública (asimétrica o de dos claves) y ofreció un nuevo método para el intercambio de claves. Este método basa su seguridad en la intratabilidad computacional del Problema del Logaritmo Discreto (DLP, por sus siglas en inglés) (Menezes *et al.,* 1996).

En 1978 se descubre el primer esquema práctico de clave pública de cifrado y firma, conocido como RSA (por las iniciales de sus autores). La seguridad de este esquema se sustenta en el Problema de la Factorización de Enteros (IFP, por sus siglas en inglés), considerado computacionalmente difícil (Menezes *et al.,* 1996; Rivest, Shamir, & Adleman, 1978). En 1985, se presenta otro esquema de clave pública, basado en DLP (ElGamal, 1985; Menezes *et al.,* 1996). A partir de estos trabajos y de forma tradicional, los protocolos criptográficos asimétri-

cos han basado su seguridad en los problemas DLP e IFP aplicados sobre campos numéricos de naturaleza conmutativa. Sin embargo, se han reportado ataques subexponenciales para muchos de ellos y se presume la existencia de ataques en tiempo polinomial contra algunos (Bishop, 2003; Dixon, 1981; Hecht, 2015; Hecht,2016; Kallenberg, 1975; Mieres, 2009; Pomerance, 1996; Sevillano Castellano, 2018; Sicard, 1999).

En Shor (1994) se despliegan algoritmos cuánticos para resolver DLP e IFP, lo que condujo a la exploración de nuevas tendencias que conforman la hoy denominada criptografía poscuántica (Delgado *et al.,* 2019; Gil, 2020; Hecht, 2015; Hecht, 2016; Hernández, 2022). En ese contexto se han desarrollado soluciones como las multicuadráticas, las basadas en *hash* y las criptografías algebraicas no conmutativa y no asociativa (Hecht, 2016). Desde 2016, el Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés), del Departamento de Comercio de Estados Unidos, lleva a cabo un proceso de normalización poscuántica, con el afán de elaborar normas y directrices en el desarrollo de primitivas seguras, eficientes y resistentes a ataques cuánticos (Delgado *et al.,* 2019; Gil, 2020; Salgado, 2021; Lemus, 2021). Todo ello denota la importancia que merece la criptografía, para la seguridad informática.

Propuesta original de Hecht

En la vertiente no conmutativa de la criptografía poscuántica se enmarca la propuesta presentada en Hecht (2015), antecedente de este trabajo. A los efectos de describir el funcionamiento de esos protocolos, su autor expone las bases estructurales que sustentan la propuesta, en virtud de las cuales se elige el grupo general lineal $GL(8,Z_{251})$, como espacio no conmutativo común para todos ellos, al que denota por M_8. Al subgrupo conmutativo de M_8 lo denota por P_8. A $GL(n,Z_p)$ se le conoce en criptografía como grupo de las matrices de Hill de orden n y módulo p. En este trabajo, a las entradas de esas matrices se les denomina valores modulares. La seguridad de todos los protocolos se garantiza por aplicar GSDP sobre $GL(8,Z_{251})$, lo que los convierte en protocolos poscuánticos. El mencionado problema se enuncia como: dados un grupo no conmutativo G, un subgrupo abeliano $S \subset G$, el par $(x, y) \in G \times G$ y los enteros m y n, encontrar $z \in S$, tal que $y = z^m \cdot x \cdot z^n$".

En todos los protocolos intervienen dos entidades que deben generar sus llaves pública y privada. Solo en el caso del protocolo de firma digital, la entidad que verifica la firma no genera llave alguna, sino que usa la pública de la otra en la verificación. Una vez creadas dichas llaves, se desencadena la ejecución correspondiente, en la que se obtienen productos, potencias e inversas matriciales. Para estas últimas, el autor propone un método de cálculo del determinante módulo p. Todas estas operaciones requieren de un elevado esfuerzo computacional (Golub & Van Loan, 1996; Meyer, 2000).

En las mencionadas bases estructurales se establece que ciertas matrices de M_8 sean generadas al azar, a la vez que sean no singulares. Para cada una de ellas, el autor propone se generen al azar sus 64 entradas (en Z_{251}) y el cálculo de su determinante. La generación de todas las entradas debe repetirse mientras el determinante calculado resulte nulo (Hecht, 2015), lo cual garantiza su no singularidad pero no garantiza la eficiencia en el proceso (Golub & Van

Loan, 1996; Meyer, 2000). Para dar solución a esta problemática, en este trabajo se propone el uso de matrices elementales invertibles que, por su estructura, ofrecen beneficios relacionados con la eficiencia.

Matrices elementales

Las matrices (o transformaciones) elementales juegan un rol clave en la teoría y los cálculos matriciales. Se utilizan, tanto en el desarrollo de algoritmos para problemas computacionales, como para ofrecer pruebas constructivas de muchos resultados analíticos, como la eliminación gaussiana (Meyer, 2000; Sun, 1996).

Una matriz elemental A de orden n responde a la forma:

$$A = I - uv^T, \qquad (1)$$

donde u y v son vectores columna de orden n, de un cierto espacio F^n, e I es la matriz identidad de orden n. Lu inversa A^{-1} se obtiene mediante la expresión:

$$A^{-1} = (I - uv^T)^{-1} = I - \frac{uv^T}{v^T u - 1}, \qquad (2)$$

siempre que el escalar $v^T u \neq 1$. De lo contrario, la matriz no es invertible (Meyer, 2000; Sun, 1996). Nótese que el escalar:

$$k = \frac{1}{v^T u - 1}, \qquad (3)$$

denominado en este trabajo factor de inversión, multiplica a uv^T en (2) y requiere $2*n + 2$ operaciones: n productos y n sumas para el producto interior $v^T u$, una sustracción y una división, lo que resulta en un orden lineal de esfuerzo computacional. Luego, para la inversa $A^{-1} = (I - u'v'^T)$ de una matriz A de la forma (1) solo se necesita obtener los vectores $u' = ku$ y $v' = v$, con $2*n$ operaciones para hallar k, n para obtener u' y otras n para obtener v'. El orden de la inversión es lineal, en contraste con el cúbico de invertir una matriz general y no requiere del determinante.

Estos elementos determinaron la propuesta de representación interna expuesta en la sección introductoria: se almacenan solo los vectores u y v y el factor de inversión ($2*n + 1$ valores), espacio significativamente menor que el requerido para los n^2 valores de la respectiva general. Se propone, además, obtener el factor de inversión $k' = 1/(v'^T u' - 1)$, al hallar la inversa A^{-1} de A. Por su parte, para los productos matriz-matriz y matriz-vector, así como para la potenciación, se realizan menos operaciones al involucrar y aprovechar la estructura de matrices elementales (Golub & Van Loan, 1996; Meyer, 2000), en aras de simplificar las ecuaciones matriciales.

Para generar «al azar» una matriz elemental no singular A se propone seguir los siguientes pasos:

1. Generar las respectivas $n - 1$ primeras entradas de u y v
2. A la par, hallar la suma parcial $s = \sum v^T_i * u_i \ (1 \leq i \leq n - 1)$

3. Generar la *n*-ésima entrada de *u*
4. Generar la *n*-ésima entrada de *v*
5. Calcular $h = v^T_n * u_n$
6. Si $h + s = 1$, regresar al paso 4
7. Hallar $k = 1/(h + s - 1)$.

Adicionalmente, se imponen las restricciones de que $u_i \neq 0$ y $v_i \neq 0$ ($1 \leq i \leq n$), pues se anularían filas o columnas de la matriz que se genera, que $u_i \neq u_j$ y $v_i \neq v_j$ ($i \neq j$), para evitar que hayan varias filas o columnas iguales, que $u_i * v_i \neq 1$ ($1 \leq i \leq n$), para que no se anulen entradas diagonales, dado que, según (1), $u_i * v_i$ se sustrae de $I_{i,i,}$ y que $u_i \neq v_i$ ($1 \leq i \leq n$) para aumentar la "aleatoriedad" de la matriz. Con estas restricciones, la matriz generada no es realmente aleatoria, razón por la cual se han colocado comillas alrededor de los términos aleatoria y al azar.

En el caso de que la matriz elemental "aleatoria" de M_8 no deba ser invertible, se generan las *n* entradas u_i y v_i, con $u_i \neq 0$ y $v_i \neq 0$, $u_i \neq v_i$ ($1 \leq i \leq n$) y $u_i \neq u_j$ y $v_i \neq v_j$ ($i \neq j$). Por último, a la par, se obtiene la suma $s = \sum v^T_i * u_i$ ($1 \leq i \leq n$) y se halla $k = 1/(s-1)$, o $k = 0$, si $s = 1$.

Implementación de los protocolos

En esta sección se describe la variante de implementación adoptada para los protocolos de transporte e intercambio de claves, firma digital y cifrado de mensajes, propuestos teóricamente en (Hecht, 2015). De forma consciente y sistemática, se aprovechó la estructura de las matrices elementales para realizar la menor cantidad posible de operaciones durante la ejecución de los protocolos. Las aplicaciones se implementaron en el Ambiente de Desarrollo Integrado QtCreator, en su versión 5.7.0, en el lenguaje C++ y se hizo uso del protocolo de comunicaciones TCP/IP.

Como se expresó antes, en todos los protocolos se utiliza el grupo M_8, así como del subgrupo conmutativo $P_8 \subset M_8$ generado mediante transformaciones de similitud de la forma PDP^{-1}, donde *P* es una matriz invertible de vectores propios y *D* es la matriz diagonal de los valores propios correspondientes. Resulta simple demostrar que la potencia *m* de esa transformación se obtiene por $(PDP^{-1})^m = PD^mP^{-1}$. Excepto que se especifique cómo se obtienen, las matrices y exponentes son generados «al azar».

En este trabajo, se propone que *P* sea elemental, generada mediante el proceder descrito en la sección anterior. En cambio, la matriz *D* debe satisfacer $D_{i,i} \neq 0$ ($1 \leq i \leq n$) y $D_{i,i} \neq D_{j,j}$ ($i \neq j$). Cada entidad genera su propia matriz *D* y debiera obtener su clave privada $A_{Pr} = PDP^{-1}$. Dado que ambas entidades usan la misma matriz *P*, el producto de las llaves privadas sería conmutativo, lo que sirve para demostrar la validez del funcionamiento de los protocolos. Se usa, además, una matriz auxiliar *G*, que no necesita ser invertible y, también, se propone sea elemental, $G = (I - rs^T)$, y generada mediante el proceder descrito en la sección anterior para matrices no necesariamente invertibles. Con ella, cada entidad obtiene su llave pública $A_{Pu} = A_{Pr}{}^m G A_{Pr}{}^n$, con $m \neq n$ y $m,n \in [2,1024]$, denominado intervalo de confianza. Sin embargo, en lugar de obtener la clave pública $A_{Pu} = A_{Pr}{}^m G A_{Pr}{}^n = PD^mP^{-1}GPD^nP^{-1} = (I-uv^T)D^m(I-kuv^T)$

$(I\text{-}rs^T)(I\text{-}uv^T)D^n(I\text{-}kuv^T)$, se tomó en cuenta la estructura matricial de las elementales *P* y *G*, para aplicar propiedades y reglas algebraicas que permitieron simplificar tal expresión. Se obtuvo, entonces, un algoritmo que aquí se denota por la función $f(P,D,G,k,m,n)$, donde *k* es el factor de inversión de *P*. Con este algoritmo no es necesario obtener ni almacenar la llave privada; de algún modo, la matriz *D* funge como tal. El análisis consciente, en aras de reducir la cantidad de operaciones de los protocolos y, por consiguiente, sus tiempos de ejecución, arrojó que algunas matrices se obtienen a través de expresiones similares que involucran a las llaves privadas pero que, en lugar de A^mGA^n, tienen la forma A^mHA^n, donde *H* es una matriz general o un producto de tales matrices. Para esos casos, se siguió un razonamiento análogo y se obtuvo otro algoritmo, denotado aquí por la función $g(P,D,H,k,m,n)$, siendo *k* el factor de inversión de *P*. Se obtuvo, también, un algoritmo para hallar una transformación de similitud de la forma $A = PDP^{-1}$, que aquí se denota por $h(P,D,k)$, donde *k* es el factor de inversión de *P*. En este trabajo, al análisis para la simplificación de todas esas expresiones se le denomina enfoque sistemático.

Todos los protocolos se ejecutan en varias fases. En la primera, de preparación, se genera y publica la matriz de vectores propios *P* y, según cuál sea el protocolo, se generan y publican otros datos. Las entidades participantes son nombradas Alice y Bob, y generan sus respectivas llaves privada y pública y otros datos, según cuál sea el protocolo. Por otro lado, en ningún caso se exponen los datos secretos de las entidades.

Para cada protocolo se implementaron dos aplicaciones (una para cada entidad) que utilizan matrices elementales con el enfoque sistemático, y otras dos (una para cada entidad) que hacen uso de matrices generales con operaciones matriciales tradicionales.

A continuación, se presenta la variante de implementación, con matrices elementales, adoptada para cada protocolo. Para ello, en sustitución de la TPC, se designa a una de las entidades para generar y enviar (publicar) a la otra, ciertas matrices o datos.

Protocolo de transporte de claves

Este protocolo implementa una versión generalizada del de Baumslag (Gerritzen, Goldfeld, Kreuzer, Rosenberger, & Shpilrain, 2006): Alice desea trasmitir una clave secreta K a Bob, sin transferir secreto alguno por la red (Hecht, 2015).

En la fase de preparación, Bob genera y envía a Alice la matriz elemental no singular *P* y genera la matriz diagonal secreta D_B. Alice genera la clave secreta *K* (matriz elemental), así como la matriz diagonal secreta D_A. En la fase 2, Alice elige dos enteros *k1* y *k2* ($k1 \neq k2$, $k1,k2 \in [2,1024]$) y obtiene y envía a Bob la matriz $T_A = A^{k1}KA^{k2}$ (mediante $T = f(P,D_A,K,k,k1,k2)$). En la fase 3, Bob elige dos enteros *r1* y *r2* ($r1 \neq r2$, $r1,r2 \in [2,1024]$) y obtiene y envía a Alice la matriz $T_B = B^{r1}T_AB^{r2}$ ($T_B = g(P,D_B,T_A,k,r1,r2)$). En la fase 4, Alice obtiene y envía a Bob la matriz $S = A^{-k1}T_BA^{-k2}$ ($S = g(P,D_A,T_B,k,-k1,-k2)$). En la quinta y última fase, Bob calcula la clave secreta $K = B^{-r1}SB^{-r2}$ ($K = g(P,D_B,S,k,-r1,-r2)$).

En Hecht (2015) se demuestra que las matrices *K* obtenidas por Alice y por Bob son iguales.

Protocolo de intercambio de claves

En este caso se implementa una versión generalizada del protocolo de Diffie-Hellman (Menezes *et al.,* 1996): Alice y Bob generan, sin compartir secreto alguno, una misma clave de sesión aleatoria (Hecht, 2015).

En la fase de preparación, Alice genera y envía a Bob la matriz elemental invertible P y genera su matriz diagonal secreta D_A. Por su parte, Bob genera y envía a Alice la matriz elemental G y genera la matriz diagonal secreta D_B. En la segunda fase, Alice elige dos enteros $k1$ y $k2$ $(k1 \neq k2, k1,k2 \in [2,1024])$ y obtiene y envía a Bob el testigo $T_A = A^{k1}GA^{k2}$ $(T_A = f(P,D_A,G,k,k1,k2))$. En la fase 3, Bob elige dos enteros $r1$ y $r2$ $(r1 \neq r2, r1,r2 \in [2,1024])$ y obtiene y envía a Alice el testigo $T_B = B^{r1}GB^{r2}$ $(T_B = f(P,D_B,G,k,r1,r2))$. En la fase 4, Alice obtiene y envía a Bob la matriz $K_A = A^{k1}T_BA^{k2}$ $(K_A = g(P,D_A,T_B,k,k1,k2))$. En la quinta y última fase, Bob obtiene la matriz $K_B = B^{r1}K_AB^{r2}$ $(K_B = g(P,D_B,K_A,k,r1,r2))$.

En Hecht (2015) se demuestra que $K_A = K_B$.

Protocolo de cifrado de mensajes

Este protocolo implementa una versión generalizada del protocolo de ElGamal (ElGamal, 1985): Alice cifra un mensaje M destinado a Bob (Hecht, 2015).

En la fase de preparación, Bob genera y envía a Alice la matriz elemental invertible P, así como los enteros m y n $(m \neq n, m,n \in [2,1024])$ y genera la matriz diagonal secreta D_B. Alice genera y envía a Bob la matriz elemental G. Bob obtiene y envía a Alice su clave pública $B' = B^mGB^n$ $(B' = f(P,D_B,G,k,m,n))$. Alice genera la matriz diagonal secreta D_A, obtiene su clave pública $A' = A^mGA^n$ $(A' = f(P,D_A,G,k,m,n))$, así como la matriz elemental secreta M que cifra el mensaje original, genera otra matriz diagonal secreta D_K, de valores propios asociados a las columnas (vectores propios) de P, para generar la matriz secreta de sesión $K = PD_KP^{-1}$ que, como las llaves privadas, no es necesario obtener. En la segunda fase (de cifrado) Alice obtiene y envía a Bob las matrices $Y_1 = K^mGK^n$ $(Y_1 = f(P,D_K,G,k,m,n))$ y $Y_2 = M(K^mB'K^n)$ (mediante $Y_2 = Mg(P,D_K,B',k,m,n)$). En la tercera y última fase (de descifrado) Bob obtiene la matriz que cifra el mensaje original $M = Y_2(B^mY_1B^n)^{-1}$, para lo cual se aplica la ley del orden reverso para la inversión de productos matriciales, en virtud de la cual $M = Y_2(B^{-n}Y_1^{-1}B^{-m})$, luego, $M = Y_2g(P,-D_B,Y_1^{-1},k,-n,-m))$.

En Hecht (2015) se prueba la igualdad de la matriz M generada por Alice y la obtenida por Bob.

Protocolo de firma digital

En este protocolo se acuerda una función pública y criptográficamente segura (*hashing),* que transforme irreversiblemente una cadena binaria finita $msg \in \{0,1\}^n$ en una matriz $H(msg)$ del subgrupo conmutativo generado por la matriz invertible de vectores propios P. Se requiere de una función que genera matrices diagonales de M_8, a partir de una cadena binaria. Para generar la firma digital, se necesitaría una matriz secreta L que debe ser de uso único (Hecht, 2015). En este protocolo, Alice genera la firma digital y cualquier entidad que posea los elementos públicos (en este caso Bob) puede verificarla.

En la fase de preparación, Alice genera la matriz elemental invertible *P,* elige los enteros *m* y *n* (*m* ≠ *n, m,n* ∈ *[2,1024]),* genera la matriz diagonal secreta D_A, genera una matriz diagonal secreta D_L de valores propios asociados a los vectores propios (columnas) de *P,* debiera generar la matriz secreta auxiliar $L = PDLP^{-1}$ pero esta resulta innecesaria, dado que Alice genera y envía a Bob la matriz $A' = A^mLA^n$, para la que resulta relativamente simple probar que $A' = h(P,D_A^mD_LD_A^n,k)$. Por último, convierte el mensaje original a la cadena binaria finita *msg* y, a partir de ella, genera y envía a Bob la matriz *H.* En la fase 2, Alice genera y envía a Bob la matriz correspondiente a la firma $F = A^{-n}L^{-1}HA^{-m}$ (mediante $F = h(P,D_A^{-n}D_L^{-1},k)Hh(P,D_A^{-m},k)$), así como el mensaje original *msg.* En la tercera y última fase, Bob obtiene la matriz *H´,* a partir de aplicar la función acordada sobre el mensaje *msg,* verifica la firma, mediante la comparación de *H'* con *FA'.* Si son iguales, el mensaje es auténtico.

RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados obtenidos en la implementación de los protocolos. Para la validación de estos, se contrastaron las mediciones de los tiempos de ejecución, la cantidad de datos a almacenar y a transferir por la red, para la variante que utiliza matrices elementales y el enfoque sistemático, con respecto a la que hace uso de matrices generales y operaciones matriciales tradicionales. Como unidad experimental se utilizó un ordenador portátil Lenovo, con procesador Intel® Core® i3-6006U, 8.00GB de RAM y 3.00MB de caché. Los tiempos tomados para el análisis son la media de las mediciones realizadas en 45 ejecuciones consecutivas de cada variante. Como las potencias matriciales deben ser diferentes, se midieron los tiempos para los órdenes matriciales 4, 8, 16, 32, 64 y 128, y las combinaciones de potencias *m = 2* y *n = 3* (rotuladas «Potencias mínimas» en las tablas 1 a 4) y *m* = 1 023 y *n* = 1 024 (rotuladas «Potencias máximas» en las propias tablas).

Protocolo de transporte de claves

En la tabla 1 se muestran los resultados de las mediciones realizadas para las dos variantes del protocolo de transporte de claves. Se analizan estos resultados y se contrasta el espacio requerido por el protocolo para almacenar los datos y la cantidad de datos a transferir por la red en cada variante.

Tabla 1. Mediciones de tiempo (en milisegundos) para el protocolo de transporte de claves, con diferentes órdenes y potencias matriciales

Orden matricial	Tiempos de ejecución (msec)			
	Potencias mínimas		Potencias máximas	
	Matriz general	Matriz elemental	Matriz general	Matriz elemental
4	<1	<1	18,3	15,3
8	<1	<1	94,5	62,3
16	16,4	<1	625,2	484,0
32	78,0	16,3	4 563,0	3 812,8
64	703,2	110,5	40 981,8	31 220,4
128	8 110,4	844,6	382 862,4	241 808,9

Nótese que para los órdenes 4 y 8 y las potencias mínimas, los tiempos no alcanzan el milisegundo en ninguna variante, por lo que no es posible determinar cuál es más rápida. En el resto de los casos, la variante con matrices elementales resulta más veloz. En cuanto al espacio, la variante con matrices elementales almacena 34 valores modulares: 17 de *P* y 17 de *K* y, de ellos, transfiere solo los 17 de *P.* La variante con matrices generales almacena 128: 64 de *P* y 64 de *K,* de los que transfiere los 64 de *P.* Para las restantes matrices, ambas variantes utilizan el mismo espacio. Por la red se transfieren las matrices *P,* T_A, T_B y *S,* ninguna de las cuales es secreta.

Protocolo de intercambio de claves

La tabla 2 contiene los resultados de las mediciones realizadas para el protocolo de intercambio de claves. Se presenta el análisis de los resultados, relativos a los tiempos de ejecución, a la cantidad de datos a almacenar y a transferir por la red, por ambas variantes de implementación de este protocolo.

Tabla 2. Mediciones de tiempo (en milisegundos) para el protocolo de intercambio de claves, con diferentes órdenes y potencias matriciales

Orden matricial	Tiempos de ejecución (msec)			
	Potencias mínimas		Potencias máximas	
	Matriz general	Matriz elemental	Matriz general	Matriz elemental
4	<1	<1	31,3	25,5
8	<1	<1	125,6	110,3
16	<1	<1	875,2	391,3
32	63,4	3,1	4 734,5	516,2
64	500,5	9,0	47 455,8	641,4
128	8 406,4	16,3	436 411,2	781,8

Se observa que para los órdenes 4, 8 y 16 y potencias mínimas, los tiempos de ambas variantes no alcanzan el milisegundo y no se puede determinar qué variante es más rápida. Para el resto de los órdenes y potencias, la variante con matrices elementales resulta más veloz. La variante con matrices elementales almacena y transfiere por la red 34 valores modulare: 17 de *P* y 17 de *G,* cifra menor que los 128 (64 para cada una) que almacenarían y transferirían, si fuesen generales. Para el resto de las matrices se almacena y transfiere la misma cantidad de datos en ambas. Además de *P* y *G,* el protocolo transfiere las matrices T_A, T_B y K_A y ninguna de ellas constituye información secreta para ninguna entidad.

Protocolo de cifrado de mensajes

La tabla 3 muestra las mediciones de tiempo realizadas al protocolo de cifrado. Se presenta el análisis realizado, acerca de los tiempos de ejecución, al espacio requerido y a la cantidad de datos a transferir, en las dos variantes de este protocolo.

Para el orden 4 y potencias mínimas, el tiempo de ejecución no alcanzó el milisegundo en ninguna de las variantes, por lo que no se puede determinar cuál es más rápida. En el resto de los casos, la variante con matrices elementales resulta más veloz. Para las matrices P, G y M, esa propia variante almacena y transfiere un total de 51 valores modulares (17 de cada una),

Tabla 3. Mediciones de tiempo (en milisegundos) para el protocolo de cifrado de mensajes, con diferentes órdenes y potencias matriciales

Orden matricial	Tiempos de ejecución (msec)			
	Potencias mínimas		Potencias máximas	
	Matriz general	Matriz elemental	Matriz general	Matriz elemental
4	<1	<1	47,3	16,7
8	16,6	<1	218,8	63,6
16	24,6	9,4	1 594,3	406,3
32	94,8	49,7	7 767,8	3 094,4
64	734,2	362,3	45 190,8	24 863,3
128	5 329,6	965,6	464 421,4	242 637,6

mientras que la otra almacena y transfiere 192 (64 de cada matriz). Ambas variantes requieren del mismo espacio y transfieren la misma cantidad de datos para las restantes matrices. El protocolo transfiere los enteros m y n, así como las matrices *P, G, B´, Y_1* y *Y_2*, que no constituyen información secreta alguna.

Protocolo de firma digital

La tabla 4 muestra las mediciones de tiempo realizadas a este protocolo.

Tabla 4. Mediciones de tiempo (en milisegundos) para el protocolo de firma digital, con diferentes órdenes y potencias matriciales

Orden matricial	Tiempos de ejecución (msec)			
	Potencias mínimas		Potencias máximas	
	Matriz general	Matriz elemental	Matriz general	Matriz elemental
4	<1	<1	16,0	13,3
8	1,2	<1	156,7	63,2
16	3,3	<1	1 063,0	422,2
32	79,8	32,5	8 094,2	3 313,1
64	594,5	220,3	64 300,4	48 815,4
128	6 813,8	4 626,3	444 099,6	364 329,1

Para el orden 4, los tiempos en ambas variantes no alcanzan el milisegundo, por lo que no se puede saber cuál es más veloz. En el resto de los casos, la variante con matrices elementales se ejecuta en menos tiempo. Con la variante con matrices elementales se utiliza menos espacio y se transfieren menos datos por almacenar la matriz *P* como elemental: 17 valores modulares, en contraste con los 64 valores que la formaran, si esta fuese general. Por la red se transfieren las matrices *A´, F* y *H,* así como el mensaje original *msg,* elementos todos que no constituyen información secreta de Alice.

CONCLUSIONES

Se logró una implementación práctica de los protocolos para transporte e intercambio de claves, firma digital y cifrado de mensajes, mediante el uso de matrices elementales, con las que se logra mayor eficiencia en cuanto al espacio para almacenar las matrices, al tiempo de ejecución y a la cantidad de datos a transferir, que la obtenida con matrices generales y operaciones algebraicas tradicionales. Con estos protocolos no se comprometen las llaves privadas ni datos secretos, por lo que pueden ser utilizados en redes no seguras.

AGRADECIMIENTOS

Los autores desean agradecer al Dr. Roberto Sepúlveda Lima, por haber propiciado su incursión en el uso del álgebra computacional en el contexto de la criptografía.

REFERENCIAS

Bishop, M. (2003). What is computer security? *IEEE Security & Privacy, 1*(1): 67-69.

Delgado Vargas, K. A., de Abiego L´Eglisse, A. F., Gallegos-García, G., & Cabarcas, D. (2019). Un acercamiento a la línea del tiempo de los algoritmos criptográficos. *Revista Digital Universitaria, Universidad Nacional Autónoma de México, 20*(5): 10. doi: http://doi.org/10.22201/codeic.16076079e.2019.v20n5.a7.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE Transaction on Information Theory, 22*(6): 644-654.

Dixon, J. D. (1981). Asymptotically fast factorization of integers. *Mathematics of computation, 36*(153): 255-260.

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory, 31*(4): 469-472.

Gerritzen, L., Goldfeld, D., Kreuzer, M., Rosenberger, G., & Shpilrain, V. (2006). Algebraic Methods in Cryptography: AMS/DMV Joint International Meeting, June 16-19, 2005, Mainz, Germany: International Workshop on Algebraic Methods in Cryptography, November 17-18, 2005, Bochum, Germany (Vol. 10): American Mathematical Soc.

Gil Rosell, B. (2020). CSIDH: criptografía postcuántica basada en isogenias de curvas Elípticas. Tesis de Grado en Matemáticas, Universitat de Barcelona, España.

Golub, G. H., & Van Loan, C. F. (1996a). Matrix Computations: Baltimore, The Johns Hopkins University Press.

Hecht, J. P. (2015). A post-quantum set of compact asymmetric protocols using a general linear group. Actas del VIII Congreso Iberoamericano de Seguridad Informática CIBSI, 15, 96-101.

Hecht, J. P. (2016). Post-Quantum Cryptography: generalized ElGamal cipher over GF(2518). *Theoretical and Applied Informatics, 28*(4): 14. doi: 10.20904/284001

Hernández Basco, B. E. (2022). Test de primalidad y algoritmos de factorización en criptografía: aspectos matemáticos y computacionales. Tesis de Grado en Ingeniería en Computación y en Licenciatura en Matemáticas, Universidad de La República, Uruguay.

Kallenberg, O. (1975) Random measures. Berlin and London: Akademie-Verlag.

Menezes, A. J., van Oorschot, P. C., Vanstone, S. A., & Rosen, K. (1996). Handbook of Applied Cryptography: CRC Press.

Meyer, C. D. (2000b). Matrix Analysis and Applied Linear Algebra. Society for Industrial and Applied Mathematics.

Mieres, J. (2009). Ataques informáticos. Debilidades de seguridad comúnmente explotadas. Recuperado de: http://proton.ucting.udg.mx/tutorial/hackers/hacking.pdf.

Miguel Salgado, A. (2021). Criptografía Postcuántica. Tesis de Grado en Matemáticas, Universidad del País Vasco, España.

Pomerance, C. (1996). A tale of two sieves. Paper presented at the Notices Amer. Math. Soc.

Rambaut Lemus, D. F. (2021). Introducción a la Criptografía post-cuántica basada en teoría de códigos. Tesis de Grado en Matemáticas Aplicadas y Ciencias de la Computación, Universidad del Rosario, Colombia.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM, 21*(2): 120-126.

Sevillano Castellano, E. (2018). Leyes de reciprocidad (cuatro demostraciones de la Ley de Reciprocidad Cuadrática).

Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring Proceedings 35th annual symposium on foundations of computer science (pp. 124-134): Ieee.

Sicard, A. (1999). Algunos elementos introductorios acerca de la computación cuántica. Memorias VII Encuentro ERM, Universidad de Antioquia, Medellín, 23.

Sun, X. (1996). Aggregations of Elementary Transformations.