Improving a Compact Cipher Based on Non Commutative Rings of Quaternions

Jorge Alejandro Kamlofsky

[1] CAETI - Universidad Abierta Interamericana
Av. Montes de Oca 725 – Buenos Aires – Argentina
Jorge.Kamlofsky@uai.edu.ar

Abstract. Asymmetric cryptography is required to start encrypted communications. Most protocols are based on modular operations over integer's rings. Many are vulnerable to sub-exponential attacks or by using a quantum computer. Cryptography based on non-commutative algebra is a growing trend arising as a solid choice that strengthens these protocols. In particular, Hecht (2009) has presented a key exchange model based on the Diffie-Hellman protocol using matrices of order four with elements in Z_{256}, that provides 128-bits keys also to devices with low computing power. Quaternions are four-component's vectors. These also form non-commutative rings structures, with compact notation and lower run-times in many comparable operations. Kamlofsky et al (2015) presented a model using quaternions with elements in Z_{256}. To provide a 128-bit key is required 4 rounds of 32-bits. However, a gain of 42% was obtained. This paper presents an improvement of this cipher that reduces even more the run-times.

Keywords: Asymmetric cryptography, quaternion's cipher, non-commutative cryptography, post-quantum cryptography.

1 Introducción

1.1 Trabajos Relacionados

La Criptografía es una rama de la Matemática. Trata el problema de enviar información confidencial por un medio inseguro. Para ello, se cifra la información de manera que, aun cuando se encuentre disponible para cualquiera, no pueda ser utilizada, a menos que alguien autorizado la descifre. En una comunicación cifrada, entonces, pueden presentarse dos instancias diferentes: el intercambio seguro de claves y luego, con ello, el cifrado y descifrado del mensaje [1]. La Criptografía se

divide en dos grandes ramas: de clave privada o simétrica, que cifra y descifra los mensajes y de clave pública o asimétrica, que logra el intercambio seguro de claves.

Diffie y Hellman fueron los pioneros de la criptografía asimétrica: en 1976, en [2] presentaron el revolucionario concepto de criptografía de clave pública cuya seguridad radica en el problema de la intratabilidad del logaritmo discreto [3] (*DLP: Discrete Logarithm Problem*). Sin embargo, por facilidad de implementación práctica, el esquema criptográfico de clave pública hoy más usado es RSA [4]: su seguridad radica en el problema de la intratabilidad de la factorización de grandes números enteros *(IFP: Integer Factorization Problem)*.

En 1993 Peter Shor presentó en [5] un algoritmo que reduce la complejidad computacional del problema IFP mediante una computadora cuántica. A pesar que este dispositivo aún no se había inventado, solo la existencia del algoritmo, debilitó a esta rama de la criptografía. Hoy su existencia es un hecho: la empresa D-Wave Systems ya vendió computadoras cuánticas a Lockheed Martin, al laboratorio Los Alamos, a Google y a la NASA, entre otros [6]. Además, IBM por su lado, ofrece servicios en la nube con su computadora cuántica [7].

Desde inicios de este siglo ha crecido el interés por el desarrollo de cripto-sistemas asimétricos alternativos que sean resistentes a ataques de complejidad sub-exponencial y ataques vía computadora cuántica [8 – 9]. A la mayoría de estos esquemas se los denomina colectivamente como criptografía post-cuántica [10] o bien, por su naturaleza algebraica, se los denomina criptografía no conmutativa [11]. Sobre esta línea, no se conocen ataques que hayan logrado resultados concretos. Dentro de esta línea, en [12] se presentó un esquema de distribución de claves Diffie-Hellman basado en un anillo de polinomios matriciales. Al sistema se lo denominó compacto debido a que no se requieren librerías de precisión extendida, lo que hace posible su uso en procesadores de menor porte. En [13] se implementó dicho esquema en un anillo de polinomios de cuaterniones, lo cual permitió la obtención de claves de la misma longitud (128 bits) con una mejora de 42,59% en los tiempos de ejecución.

En este trabajo, se presentan resultados que muestran mejoras aún mucho mayores en los tiempos de ejecución usando cuaterniones con elementos de otros conjuntos numéricos, obteniendo claves de la misma longitud que las presentadas en [12, 13].

1.2 Motivación y Alcance

En [13] se logra el intercambio de claves de 128 bits a partir de cuaterniones con elementos de 8 bits en 42,59% menos tiempo de ejecución. Para ello, en cada instancia se obtiene un cuaternión que trae consigo 4 x 8 bits = 32 bits. El algoritmo se repite 4 veces para obtener los 128 bits. Sin embargo, en otro trabajo [14], donde se compara el desempeño de aplicaciones que utilizan matrices cuadradas y cuaterniones, se observa que la ganancia en tiempos de ejecución obtenida con

cuaterniones es notoriamente superior, lo cual permitió intuir la existencia de un amplio margen para mejorar aún más los tiempos de ejecución de este cifrador.

El destino del protocolo presentado en [12] es para procesadores de pequeño porte: 16 bits. Sin embargo, se trabaja con elementos de 8 bits. De aquí se puede obtener parte de ese margen disponible solo trabajando con elementos de 16 bits.

Como los primeros ensayos obtenidos fueron favorables, se permite pensar en ampliar la idea a elementos de 32 bits, considerando que hoy también pueden llamarse a los procesadores de 32 bits como de menor porte.

1.3 Objetivo del Trabajo

La finalidad de este trabajo es mostrar que mediante el uso de cuaterniones en el conjunto numérico adecuado pueden obtenerse importantes mejoras en los tiempos de ejecución en el esquema de intercambio de claves Diffie Hellman Compacto.

1.4 Relevancia del tema

La existencia de la computadora cuántica es un hecho. Con ello, la criptografía asimétrica clásica, se encuentra muy debilitada, ya que RSA, el algoritmo más usado ha sido quebrado. Los nuevos desarrollos en criptografía post-cuántica permiten mitigar esta debilidad. Y esquemas más veloces permiten que en la práctica, éstos puedan ser implementados de manera extendida.

1.5 Estructura del Trabajo

En la Sección 2 se presenta el marco teórico. En la Sección 3 se presenta la solución propuesta con datos experimentales. En la Sección 4 se presentan las conclusiones.

2 Marco Teórico

2.1 La Importancia de la Criptografía en la Seguridad de las Comunicaciones

Nociones Básicas de Criptografía Simétrica. La Criptografía se ocupa de asegurar la integridad y confidencialidad en las comunicaciones a través de un canal inseguro. Para ello, el mensaje se transforma en el punto de emisión mediante operaciones matemáticas de manera que sea imposible de interpretar mientras viaja en el canal

inseguro, o bien su costo en tiempo y/o recursos sean tan altos que su descubrimiento carezca de sentido.

Se usan algoritmos criptográficos altamente robustos que permiten además que la información se encripte bit a bit (cifradores de flujo) o en grupos de n-bits (cifradores de bloque) permitiendo que puedan cifrarse comunicaciones en tiempo real [3]. Estos cifradores usan la misma clave para el cifrado y descifrado del mensaje. A estos cifradores se los clasifica como Criptografía Simétrica. Muchos cripto-sistemas simétricos seguros (AES, DES, Trivium) pueden iniciarse con claves de 128 bits.

Nociones Básicas de Criptografía Asimétrica. La criptografía asimétrica o de clave pública, usa elementos públicos que se comparten, y elementos privados que se mantienen en secreto. Generalmente usan propiedades y operaciones de aritmética modular en estructuras algebraicas de anillos de números enteros.

Ésta brindó soluciones al problema de presentar en forma segura claves para su uso en cifradores simétricos: mientras RSA [4] permite que se pueda enviar la clave simétrica cifrada a otro usuario usando su clave pública, con Diffie-Hellman [2] y Elgamal [15] ambas partes pueden generar la misma clave intercambiando elementos.

Amenaza a la Criptografía: El Algoritmo de Shor y la Computación Cuántica. En 1995 Peter Shor presentó un algoritmo para computación cuántica basado en la transformada rápida de Fourier (FFT) que logra resolver en tiempo polinómico el problema IFP [5]. Es decir, permite reducir drásticamente la complejidad del problema (considerado de clase NP) a niveles atacables [16].

Una computadora cuántica usa qubits en lugar de bit. Un qubit posee los estados 0, 1 y la superposición de ambos: 0 y 1 a la vez. Por ello, se puede realizar una cantidad exponencial de operaciones en paralelo en relación con la cantidad de qubits del computador cuántico.

En 2001 se implementó el algoritmo de Shor en la primer computadora cuántica. La computación cuántica prácticamente arrasa con todo lo conocido en la criptología actual: con ello desaparecen de escena todos los criptosistemas de clave pública: RSA y todas las variantes de ElGamal y Diffie-Hellman [16].

Criptografía Post-Cuántica Basada en Anillos no Conmutativos. Se utilizan estructuras de anillos de matrices cuadradas o de cuaterniones, entre otros, con elementos finitos, por lo tanto, su seguridad radica en la complejidad del tratamiento del problema DLP. Algunos esquemas como el presentado en [17] se basan en la dificultad de resolver el problema SDP (*Simple Descomposition Problem*) en un anillo no conmutativo de polinomios matriciales.

Desde el punto de vista criptográfico, solo se necesita estar seguro que no existe fórmula que permita reducir la complejidad del problema DLP (incluso con

computadora cuántica). Y esto está garantizado ya que en los anillos no conmutativos no existe forma de relacionar el determinante de una matriz o bien sus eigenvalores con la potencia de la matriz [18], parte de la clave privada, independientemente de la cantidad de qubits que pudiera tener una computadora cuántica que ejecute el ataque.

En [12] se muestran más consideraciones de la seguridad de estos esquemas.

2.2 El Anillo no Conmutativo de Cuaterniones

Anillos no Conmutativos. Un anillo (A ; + ; .) es una estructura algebraica (un conjunto A con las operaciones suma y producto) donde (A ; +) forman estructura de grupo, y (A ; .) de semigrupo. Un anillo será no conmutativo si no se verifica la propiedad conmutativa entre todos los elementos de A para la operación producto.

El primer anillo de división (cuyos elementos no nulos son inversibles) no conmutativo fue el anillo de los cuaterniones. Otro ejemplo es el conjunto de matrices cuadradas de orden n con coeficientes en A (simbolizado por $M_n(A)$), es un anillo con respecto a las operaciones usuales de suma y producto de matrices. Si $n > 1$, entonces $M_n(A)$ no es conmutativo. Otros ejemplos de anillos de división no conmutativos son: los Octoniones, Sedeniones, Tessarines, cocuaterniones o bicuaterniones.

Definición: Cuaternión. Sea (A ; + ; .) un anillo conmutativo con unidad. Un cuaternión con coeficientes en A es un número hiper-complejo q de la forma: $q = a + b.i + c.j + d.k$ donde $a, b, c, d \in A$; i, j, k son unidades imaginarias que verifican que: $i^2 = j^2 = k^2 = - i . j . k = - 1$ y además: $i . j = -j . i = k$; $j . k = -k . j = i$; $i . k = -k . i = j$.

Fueron creados en 1843 por William Hamilton [19]. Forman una estructura de Anillo de división no conmutativo. Tienen una notación compacta y resultan muy sencillos para trabajar. Son muy eficientes: requieren menor cantidad de operaciones básicas y menor espacio de almacenaje en comparación con la operación de matrices.

Operaciones Básicas con Cuaterniones. Sean los cuaterniones $q = (w , x , y , z)$, $q_1 = (w_1 , x_1 , y_1 , z_1)$ y $q_2 = (w_2 , x_2 , y_2 , z_2)$

Norma del Cuaternión q: $|q| = \sqrt{q . \bar{q}} = \sqrt{\bar{q} . q} = \sqrt{w^2 + x^2 + y^2 + z^2}$

Suma, Resta y Producto de un Escalar por un Cuaternión: Se realiza de la misma forma que con cualquier vector de 4 dimensiones.

Producto: $q_1.q_2 = (w_1.w_2 - x_1.x_2 - y_1.y_2 - z_1.z_2, w_1.x_2 + x_1.w_2 + y_1.z_2 - z_1.y_2, w_1.y_2 - x_1.z_2 + y_1.w_2 + z_1.x_2, w_1.z_2 + x_1.y_2 - y_1.x_2 + z_1.w_2)$. Notar que el producto entre cuaterniones no es conmutativo.

Cociente: $\frac{q_1}{q_2} = q_1 . (q_2)^{-1} = q_1 . \left(\frac{q_2}{|q|^2} \right)$ con $q_2 \neq (0,0,0,0)$.

Potencia: $q_1^n = |q_1|^n . (\cos(n\frac{\alpha}{2}) + \breve{v} . sen(n\frac{\alpha}{2}))$.

2.3 Sistemas Compactos de Intercambio de Claves Diffie-Hellman Basados en Álgebra No Conmutativa

Esquema Diffie - Hellman Compacto con Matrices (DHCM). En [12] se presentó un sistema de intercambio de claves Diffie Hellman [2] sobre anillos de matrices de enteros con elementos en Z_{256}, que utiliza como clave privada un polinomio con coeficientes y exponentes en Z_{16}. Un par de ventajas surgen de ello. Primero: no se requiere el uso de librerías de precisión extendida, por lo tanto, puede ser usado en procesadores de pequeño porte. De allí la calificación de compacto. La otra ventaja se relaciona con el hecho que las matrices conforman estructura de anillo no conmutativo. Gracias a ello, el esquema es inmune a ataques cuánticos y de complejidad sub-exponencial. La clave resultante es una matriz de orden 4 con elementos en Z_{256} conformando así una clave de 128 bits, adecuada para su uso en varios cifradores simétricos seguros.

Resumen del Protocolo. Alice envía a Bob (a través de un canal público e inseguro) dos números enteros aleatorios *m* y *n* en Z_{16}, y dos elementos aleatorios A y B, matrices de orden 4 con elementos en Z_{256}. Elige como clave privada un polinomio entero *f(x)* con coeficientes y exponentes en Z_{16} tal que $f(A) \neq 0$. Bob elige como su clave privada un polinomio entero *h(x)* con coeficientes y exponentes en Z_{16} tal que $h(A) \neq 0$. Luego Alice y Bob calculan sus tokens: $r_A = f(A)^m . B . f(A)^n$ (Alice) y $r_B = h(A)^m . B . h(A)^n$ (Bob); y se los intercambian para el cálculo de las claves: $K_A = f(A)^m . r_B . f(A)^n$ (Alice), $K_B = h(A)^m . r_A . h(A)^n$ (Bob) con $K_A = K_B$.

Esquema Diffie-Hellman Compacto con Cuaterniones (DHCQ8). Diversas aplicaciones pueden implementarse con cuaterniones en lugar de matrices cuadradas, en menores tiempos de ejecución [14]. Ello inspiró el desarrollo de este esquema [13], que resultó en un ahorro de tiempo cercano al 50% bajo condiciones similares.

Consideraciones del Protocolo. Este protocolo es muy similar al de matrices [12]: En su lugar, los elementos A y B son cuaterniones en forma cartesiana con elementos en Z_{256}. Cada cuaternión, entonces, tiene 32 bits. Se incorporan dos instancias de normalización del cuaternión, y una modularización en Z_{256}. Gracias a esto se puede

aprovechar la notable simpleza de potenciar cuaterniones normalizados, en comparación con la complejidad de potenciar matrices.

3 La Mejora Propuesta

3.1 Resumen de la Mejora Propuesta

El protocolo presentado en [13] trabaja con elementos de Z_{256}. En esta propuesta se trabaja con elementos en $Z_{2^{k*16}}$ para $k = 1$ (DHCQ16) y $k = 2$ (DHCQ32).

3.2 El Protocolo Propuesto

Alice elige dos números enteros aleatorios m y n en Z_{16}, y dos cuaterniones aleatorios A y B, con elementos de $Z_{2^{k*16}}$ (con $k = 1$ o bien $k = 2$) y calcula sus normalizaciones: q_A y q_B. Luego elige como clave privada un polinomio entero $f(x)$ con coeficientes y exponentes en Z_{16} tal que $f(q_A) \neq 0$ y envía a Bob por el canal inseguro los elementos m, n, q_A y q_B. Bob elige como clave privada un polinomio entero $h(x)$ con coeficientes y exponentes en Z_{16} tal que $h(q_A) \neq 0$. Alice y Bob realizan las normalizaciones de $f(q_A)$ y $h(q_A)$: $f'(q_A)$ y $h'(q_A)$. Alice calcula su token: $r_A = f'(q_A)^m \,.\, B \,.\, f'(q_A)^n$. Bob calcula el suyo: $r_B = h'(q_A)^m \,.\, B \,.\, h'(q_A)^n$; y se los intercambian para el cálculo de las claves: $k_A = f'(q_A)^m \,.\, r_B \,.\, f'(q_A)^n$ (Alice), $k_B = h'(q_A)^m \,.\, r_A \,.\, h'(q_A)^n$ (Bob), las cuales se modularizan: $K_A = k_A \,.\, 2^{k*16} \pmod{2^{k*16}}$, $K_B = k_B \,.\, 2^{k*16} \pmod{2^{k*16}}$ con $K_A = K_B$.

La clave obtenida para $k = 1$ posee 4 x 16 bits = 64 bits. Para lograr una clave de 128 bits, el proceso debe repetirse. Para $k = 2$, la clave posee 4 x 32 bits = 128 bits.

3.3 Un Ejemplo Numérico

El ejemplo numérico se realiza para $k = 1$ (DHCQ16), y a los fines de esta presentación, la cantidad de decimales se limita a 2.

Alice elige dos números enteros aleatorios $m=7$ y $n=10$, y dos cuaterniones aleatorios $A = (45606, 11140, 21549, 43028)$ y $B = (42679, 56493, 45062, 43484)$, con elementos de Z_{65536} y calcula sus normalizaciones: $q_A = (0.68, 0.17, 0.32, 0.64)$ y $q_B = (0.45, 0.60, 0.48, 0.46)$. Luego elige como clave privada un polinomio entero

$f(x)= 14x^{15} + 12x^{14} + 13x^{13} + 8x^{12} + 5x^{11} + 4x^{10} + 9x^9 + 13x^8 + 7x^7 + + 11x^6 + 4x^5 + 9x^4 + 14x^3 + 7x^2 + 3x + 4$ y envía a Bob por el canal inseguro los elementos m, n, q_A y q_B. Bob elige como su clave privada un polinomio entero $h(x)= 5x^{15} + 6x^{14} + 3x^{13} + 11x^{12} + 14x^{11} + 3x^{10} + 12x^8 + 2x^7 + 3x^6 + x^5 + 14x^4 + 12x^2 + + 9x$. Luego Alice y Bob realizan las normalizaciones de $f(q_A)$ y $h(q_A)$: $f'(q_A) = (0.78, -0.14, -0.27, -0.54)$ y $h'(q_A) = (-0.88, 0.11, 0.21, 0.41)$. Alice calcula su token: $r_A = (-0.37, -0.12, 0.36, 0.26)$, Bob calcula el suyo: $r_B = (-0.17, 0.26, 0.00, 0.39)$; y se los intercambian: $k_A = (-0.25, 0.06, 0.09, -0.11)$ (Alice), $k_B = (-0.25, 0.06, 0.09, -0.11)$ (Bob), las cuales se modularizan: $K_A = k_A \,.\, 65536 \,(mod\ 65536) = (49061, 3662, 5778, 58342)$, $K_B = k_B \,.\, 65536 \,(mod\ 65536) = (49061, 3662, 5778, 58342)$ con $K_A = K_B = K$: la clave.

3.4 Equipamiento Usado

El computador usado contiene un procesador AMD A10-5745M × 4 núcleos de 64 bits y 12Gb de memoria RAM. Se instaló una distribución Ubuntu 15.10, el cual tiene un núcleo Linux Debian. Los algoritmos fueron programados en Python 2.7.10.

3.5 Resultados Experimentales

Se presenta una comparación de los tiempos de ejecución del algoritmo DHCM [12], DHCQ8 [13] y la solución aquí propuesta: para 16 bits y 32 bits ($k = 1$ y $K = 2$) para la obtención de 10.000 claves de 128 bits con polinomios aleatorios con coeficientes y exponentes en Z_{16}. La tabla 1 muestra los resultados experimentales.

Tabla 1. Tiempos de ejecución para la obtención de 10.000 claves de 128 bits mediante las diferentes opciones analizadas.

	CPU Time (s)					CPU Time (s)			
Nº Test	DHCM 8-bits	DHCQ8 8-bits	DHCQ16 16-bits	DHCQ32 32-bits	Nº Test	DHCM 8-bits	DHCQ8 8-bits	DHCQ16 16-bits	DHCQ32 32-bits
1	37,8177	20,0350	11,3851	5,5569	14	37,8728	19,9016	11,6061	5,7219
2	36,4591	19,5770	11,1946	5,7522	15	36,7675	19,3920	11,1752	5,6757
3	37,6444	19,0540	11,3141	5,6158	16	37,7545	19,1195	11,6449	3,9712
4	36,6029	19,4511	11,3369	5,7398	17	37,0721	19,1790	11,3357	5,5914
5	37,3720	20,0530	11,2472	5,6139	18	36,4384	19,6326	11,4020	5,5797
6	37,2393	19,8001	11,0701	5,5895	19	37,0333	19,6599	11,4161	5,5830
7	37,0366	18,9289	11,2918	5,6067	20	37,5093	19,1302	11,1700	5,5174
8	37,3745	19,7008	11,1029	5,7065	21	36,5904	19,6875	11,3885	5,6230
9	37,3369	19,2519	11,2191	5,6430	22	36,8067	19,8909	11,6555	5,7127
10	36,4455	19,7068	11,6477	5,5709	23	37,9800	19,2684	11,4140	5,5620
11	37,6164	19,3665	11,4154	5,5396	24	37,2734	19,4936	11,1356	5,6469
12	37,6196	20,0801	11,1886	5,6568	25	36,9313	19,3517	11,2654	5,5139
13	37,3872	19,7883	11,4547	5,8716					

El tiempo promedio (en segundos) para la obtención de 10000 claves con DHCM fue 37,2, con DHCQ8 fue 19,54, con DHCQ16 fue 11,34 y con DHCQ32 fue 5,57. Como los coeficientes de variación (CV) son CV < 0,1 se acepta al promedio como indicador de tendencia central adecuado. Este experimento generó 1 millón de claves sin error.

3.6 Ventajas de la Solución

Velocidad del Cifrador. Menores tiempos de ejecución evidencian mayor velocidad del esquema de cifrado propuesto en comparación con los esquemas presentados en [12] y [13]: El esquema DHCQ8 logra la generación de 128 bits de clave en el 52% del tiempo en comparación con DHCM lo que implica 1,9 veces su velocidad; DHCQ16 lo hace en el 30% implicando más tres veces su velocidad, y DHCQ32 lo hace en el 15% lo que significa más de 6 veces su velocidad. Si la comparación se hace entre los esquemas propuestos y DHCQ8: DHCQ16 lo hace en el 58% del tiempo haciéndolo 1,72 veces más rápido. Sin embargo, DHCQ32 lo hace en el 29% del tiempo, es decir con 3,5 veces más rapidez.

Solución Apta para Procesadores de Pequeño Porte. La implementación de la solución aquí propuesta puede realizarse sin necesidad de uso de librerías de precisión extendida, lo cual lo hace apto para procesadores de pequeño porte.

Inmunidad Frente a Ataques de Complejidad Sub-Exponencial o de Computadora Cuántica. Los anillos de cuaterniones conforman estructuras algebraicas no conmutativas, y sobre este tipo de estructuras no se conocen aún ataques de estos tipos que hayan sido efectivos y que debiliten su seguridad.

4 Conclusiones

Ya se ha visto que usando cuaterniones se puede obtener aplicaciones más rápidas que con matrices. Y esto también es válido en criptografía. Puede aprovecharse la simpleza de la potencia cuaterniones normalizados para lograr mayor velocidad. Trabajar con los conjuntos numéricos adecuados permite hacer uso más eficiente de esta ventaja, logrando implementaciones criptográficas más veloces.

Al usarse estructuras de anillos no conmutativos, hace a este esquema inmune a ataques de complejidad sub-exponencial o de computadora cuántica.

El esquema propuesto es útil para procesadores de menor porte.

Referencias

1. Marrero Travieso, Yran: La Criptografía como elemento de la seguridad informática. ACIMED 11.6 (2003).
2. Diffie W., Hellman M.E: New directions in cryptography. IEEE Transactions on information theory, 22, 644-654, (1976).
3. Menezes, Alfred J., Paul C. Van Oorschot, and Scott A. Vanstone: Handbook of applied cryptography. CRC press (1996).
4. Rivest, Ronald L., Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21.2, 120-126. (1978)
5. Shor P.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput., 5, 1484-1509 (1997)
6. D-Wave-Systems Press Releases [en línea], (2016). Disponible en: <http://www.dwavesys.com/news/press-releases>. Fecha de consulta: 05/06/2016.
7. IBM: IBM Makes Quantum Computing Available on IBM Cloud to Acceletare Innovation [En Linea], (2016). Disponible en: <https://www-03.ibm.com/ press/ us/ en/ pressrelease/ 49661.wss>. Fecha de consulta: 05/06/2016.
8. Magliveras S.S., Stinson D.R., van Trung T.: New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups, Technical Report CORR, 2000-2049 (2000)
9. Shpilrain V., Zapata G.: Combinatorial group theory and public-key cryptography, Preprint arXiv/math.gr, 0410068 (2004)
10. Barreto,P. et al: Introdução à criptografia pós-quântica, Minicursos do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, SBSeg, (2013).
11. Gerritzen L. et al (Editors): Algebraic Methods in Cryptography, Contemporary Mathematics, AMS, Vol. 418, (2006)
12. Hecht J.: Un modelo compacto de criptografía asimétrica empleando anillos no conmutativos. V Congreso Iberoamericano de Seguridad Informática CIBSI, Montevideo (2009).
13. Kamlofsky J.A., Hecht J.P, Abdel Masih S., and Hidalgo Izzi, O.: A Diffie-Hellman Compact Model Over Non-Commutative Rings Using Quaternions. VIII Congreso Iberoamericano de Seguridad Informática CIBSI, Quito (2015).
14. Kamlofsky J., Bergamini L.: Cuaterniones en Visión Robótica. V Congreso de Matemática Aplicada, Computacional e Industrial MACI, Tandil (2015).
15. Elgamal, Taher. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. En Advances in cryptology. Springer Berlin Heidelberg, pp. 10–18 (1984).
16. Hecht, JP.: Fundamentos de Computación Cuántica. Editorial Académica Española. ISBN 978-3-8484-7529-2 (2005).
17. Cao Z., Xiaolei D., Wang L.: New public-key cryptosystems using polynomials over non-commutative rings, Preprint arXiv/cr, eprint.iacr.org/2007/009.pdf (2007).
18. Eftekhari, M.: A Diffie–Hellman key exchange protocol using matrices over noncommutative rings. Groups-Complexity-Cryptology, 4(1), pp. 167–176 (2012).
19. Hamilton, W. R.: Lectures on Quaternions: Containing a Systematic Statement of a New Mathematical Method, Hodges and Smith, (1853)