Informe eSAMCid sobre criptografía post-cuántica

Diciembre de 2017

Resumen

En este documento se hace una presentación del estado actual de la llamada *criptografía post-cuántica*, área que engloba los diseños criptográficos susceptibles de mantener su seguridad en presencia de adversarios cuánticos. El objetivo de nuestra exposición es proporcionar una aproximación informal a este área, que permita a los no-expertos entender qué tipo de herramientas se están desarrollando dentro de este ámbito y cuáles son los procesos de criba que, eventualmente, permitirán estandarizar herramientas post-cuánticas para su uso masivo.

Comenzaremos por explicar qué es la criptografía post-cuántica y por qué surge con tanta fuerza en los últimos años, como contrapunto a la apuesta por la criptografía cuántica. Detallaremos los paradigmas matemáticos alrededor de los que surgen construcciones post-cuánticas, comentando qué tipo de aplicaciones se están desarrollando dentro de cada paradigma y cual es el éxito esperado de estas propuestas. En concreto, hablaremos de construcciones basadas en hashes, retículos, teoría de grupos, álgebra multivariable y teoría de códigos, por ser éstas las áreas señaladas por la comunidad científica como más prometedoras en este ámbito.

Subvencionado por el Ministerio Español de Economía y Universidades mediante el projecto MTM2013-41426-R.

Índice

1. Introducción

La criptología es la ciencia que se ocupa de la gestión, trasmisión, compartición y almacenamiento de información en entornos hostiles. Tiene una vertiente constructiva, llamada criptografía, y otra destructiva, el criptanálisis. La criptografía se encarga del diseño eficiente de esquemas y protocolos seguros, mientras que el criptanálisis es el análisis crítico de los mismos.

El desarrollo de herramientas cuánticas de computación ha supuesto una revolución dentro del escenario criptográfico. Gracias a ellas aparecen nuevas técnicas criptanalíticas que son capaces de vulnerar la seguridad desde distintos niveles. Más concretamente:

- Permiten la resolución en tiempo *razonable* de problemas matemáticos que sólo son resolubles en tiempo inabordable con computación clásica.
- Aceleran las búsquedas de elementos en conjuntos no estructurados, lo cual facilita mucho el cribado de claves posibles dentro de un conjunto grande (clásicamente inabordable).

El ejemplo paradigmático del primer aspecto es el algoritmo cuántico propuesto por Peter Shor [25], mientras que el algoritmo de Grover [15] es el diseño esencial para aumentar la velocidad en búsquedas sobre conjuntos sin estructura.

La *criptografía post-cuántica* es el tipo de criptografía que se desarrolla bajo la hipótesis de que el adversario es capaz de implementar algoritmos cuánticos. Esta posibilidad puede materializarse bien a través de un *ordenador cuántico*, es decir, una máquina de cómputo *universal* cuyas unidades de trabajo sean bits cuánticos o *qbits* o, en una concepción más realista, a través de hardware específico para implementar un algoritmo concreto bajo ciertos parámetros bien definidos.

Los diseños criptográficos post-cuánticos se fundamentan en problemas matemáticos para los que no se conoce ningún algoritmo cuántico eficiente; son diseños por tanto derivados del análisis de los algoritmos cuánticos conocidos más eficientes. Estudiando qué tipo de problemas son fáciles de resolver a través de estos algoritmos, la comunidad científica ha sido capaz de identificar el tipo de construcciones susceptibles de mantener su robustez ante adversarios cuánticos.

2. El proceso de estandarización del NIST

En la actualidad, el National Institute for Standards and Technology (NIST) está desarrollando un proceso para solicitar, evaluar y estandarizar herramientas post-cuánticas de clave pública. En concreto, se persigue la estandarización de esquemas de cifrado, firma digital e intercambio de clave. La planificación temporal del proceso puede verse en el Cuadro 1.

Temporalización	Hito
Diciembre 2016	Llamada a la participación
30 Noviembre 2017	Deadline propuestas
Abril 2018	Primer Workshop
3-5 años	Fase de análisis (1-2 Workshops)
2025	publicación primeros borradores de los estándares

Cuadro 1: Proceso de estandarización - NIST

En su informe para criptografía post-cuántica [9], se resume el impacto previsto para las herramientas más extendidas, citando las predicciones que estiman que un ordenador cuántico será una realidad en aproximadamente 15 años, eso sí, con un coste estimado de un billón de dólares y un consumo energético similar a la producción de una central nuclear. Independientemente de lo aventurado de dichas estimaciones, lo cierto es que se supone un impacto elevado en las herramientas actuales.

En concreto, en el caso de clave pública, se considerarán totalmente inseguras todas las herramientas basadas en teoría de números y problemas asociados (factorización, logaritmo discreto en grupos cíclicos asociados a cuerpos finitos o curvas elípticas). Herramientas como los esquemas RSA (para firma, cifrado o intercambio de clave), los basados en curvas elípticas para firma e intercambio de claves (ECDSA, ECDH) y los esquemas basados en el logaritmo discreto (como DSA, DH) se consideran, por tanto, obsoletos.

De manera similar, y como consecuencia directa de la reducción de complejidad que se deriva del algoritmo de Grover (ver [15]), la recomendación dada es doblar el tamaño de clave en los cifrados simétricos (como el AES) y el rango de las funciones hash utilizadas (recomendándose SHA-3). Es imprescindible seguir de cerca el proceso de estandarización del NIST y estar al tanto de la evolución de las propuestas que se anuncien en los próximos meses, de cara a tomar las mejores decisiones en cuanto a la migración a herramientas post-cuánticas. Toda la información está disponible

en la página web del proceso de estandarización (disponible en NIST PQ-standarization).

3. Planteamientos

Existen distintas áreas desde las que se enfocan las principales construcciones post-cuánticas, algunas procedentes de escenarios matemáticos clásicos que no han sido, hasta ahora, explotados en aplicaciones criptográficas. Muchas de las áreas que mencionaremos sólo sirven para plantear herramientas de un cierto tipo (por ejemplo, cifrado o firma digital), y algunas ya se consideran descartadas a la luz de los resultados que arroja la primera ronda de evaluación procedente de la competición del NIST.

3.1. Criptografía basada en Grupos

En los últimos quince años se han propuesto de manera continua herramientas criptográficas que tomaban como base problemas matemáticos descritos en grupos no abelianos. Muchas veces, el argumento de su resistencia a algoritmos cuánticos se ha esgrimido como un acicate a este área. Sin embargo, existen serias dudas de la seguridad de muchas de estas construcciones (incluso sin necesidad de recurrir a criptanálisis cuánticos).

Muchas de estas propuestas se basan en la idea de *sustituir* construcciones basadas en teoría de números, como el esquema de intercambio de claves de Diffie-Hellman, por una especie de diseño análogo no-abeliano, donde las exponenciaciones modulares se sustituyen por conjugaciones en el grupo utilizado. Tales diseños se han propuesto sobre grupos de trenzas o grupos de matrices, siempre con escaso éxito (como señalamos en el trabajo [14]). El monográfico [13] recoge las principales propuestas en ese ámbito, y señala los problemas de seguridad detectados en las mismas.

Por último, creemos oportuno mencionar que existe una vía abierta de trabajo, las construcciones simétricas basadas en el *Hidden Shift Problem* [1], mucho más prometedora que las propuestas que hemos mencionado.

3.2. Criptografía basada en Códigos

En esta sección hablaremos de criptografía basada en códigos, centrando nuestro análisis en el esquema de cifrado de clave pública McEliece [20] y

sus variantes. A grandes rasgos, el problema subyacente a este tipo de construcciones es el de decodificar una palabra codificada a través de un código lineal desconocido. Dicho código se describe a través de 3 parámetros: n y k, que hacen referencia a la longitud y *dimensión* del código, y t, que es el número de errores que es posible corregir en una codificación errónea. Así, una matriz G de dimensión $n \times k$ sirve para generar palabras codificadas: las palabras (vectores binarios de longitud k) se codifican como vectores de longitud n al multiplicarse por G. De la misma forma, si existe un error en la transmisión traducible en un vector e binario con peso de Hamming[1] acotado por t, existe un algoritmo de decodificación asociado a G que permite recuperar la palabra original.

Para construir un cifrado de clave pública a partir de esta idea, se siguen los siguientes pasos:

- Alice selecciona una matriz G asociada a un código de parámetro (n, k, t) y la *ofusca* transformándola en una matriz $\hat{G}$. La clave secreta servirá para revertir este proceso, recuperando G a partir de $\hat{G}$. La clave pública es el par $(\hat{G}, t)$.
- Bob cifra un mensaje $m \in \{0, 1\}^k$ como $c = m\hat{G}+e$, siendo e un vector aleatorio que contiene exactamente t unos.
- Alice recupera m usando el algoritmo de corrección-decodificación asociado a G.

3.2.1. Cifrado McEliece

Es el ejemplo más destacado de cifrado basado en códigos. Fue propuesto en 1998 y se fundamenta en la idea anterior, siendo G la matriz asociada a ciertos códigos lineales llamados *códigos de Goppa*. Los parámetros sugeridos originalmente eran $n = 1024, k = 524$ y $t = 50$, si bien las propuestas actuales para seguridad de 80 bits son distintas, resultando en claves públicas de tamaños desorbitados (en torno a 500.000 bits). Para los parámetros sugeridos con el fin de conseguir seguridad post-cuántica, $n = 6960$, $k = 5413$ y $t = 119$, el tamaño de las claves usando códigos de Goppa está por encima de los ocho millones de bits, siendo éste el principal inconveniente de este tipo de cifrado. Por otro lado, las operaciones de cifrado y descifrado son relativamente eficientes, y los resultados de seguridad resultan esperanzadores y permiten establecer pautas claras para la generación de claves ajustada al nivel de seguridad perseguido.

[1] Número de entradas no nulas.

3.2.2. Criptanálisis

Aunque el principal obstáculo a la extensión del esquema de MacEliece tiene que ver con la eficiencia, en su larga historia han aparecido multitud de ataques (casi siempre a implementaciones concretas) y contramedidas asociadas, que pueden resultar de utilidad a la hora de tomar decisiones hacia un nuevo desarrollo. En concreto, destacamos las siguientes lineas de ataque/investigación:

- **Algoritmos para resolver el problema general de decodificación de códigos binarios.** El ataque de Stern y la mejora posterior de Canteaut-Chabaud (ver [5]) utilizando esta técnica obligó a cambiar los parámetros originalmente propuestos para McEliece y a ampliar los tamaños de clave de 88 a 130 KBytes. Desde 1995 apenas ha habido avances en estas lineas, aunque líneas de trabajo como la llamada "decodificación iterativa" [12] o "decodificación estadística" [19] son susceptibles de dar resultados notables.

- **Algoritmos para decodificar otros códigos.** Si en un intento de conseguir claves más cortas se reemplazan los códigos Goppa utilizados habitualmente en las implementaciones por otros, los algoritmos de decodificación pueden resultar más efectivos, y por tanto los esquemas resultan más débiles. Así se ha demostrado, por ejemplo, para esquemas propuestos con códigos GRS y códigos de Reed-Muller (ver [21]).

- **Ataques por canales colaterales (side-channel attacks).** Fijada una implementación concreta, en ocasiones este tipo de ataques son muy efectivos, extrayendo valiosa información a través de patrones dependientes de la implementación (o incluso del dispositivo concreto que la ejecuta). Los más exitosos en cuanto a MacEliece son los ataques de tipo DPA (que miden diferencias en cuanto al consumo eléctrico) y otros llamados verticales/horizontales asociados a ciertas implementaciones (ver [6, 8]). Existen técnicas de *enmascaramiento* para corregir las implementaciones vulnerables a estos ataques (ver por ejemplo [7]), si bien toda implementación desarrollada debería (como indican las buenas prácticas criptográficas) revisarse en este sentido evitando la filtración de patrones.

3.3. Criptografía Multivariable

La criptografía multivariante se articula alrededor de problemas asociados a la resolución de sistemas de ecuaciones no lineales en varias variables sobre cuerpos finitos. Típicamente, se publican m polinomios $p_1, \ldots, p_m$ de n variables y grado bajo d sobre un cuerpo finito F (caso más habitual: $d = 2$). Para descifrar, autenticarse o firmar digitalmente un usuario se enfrenta al reto de, dado $z = (z_1, \ldots, z_m) \in F^m$ encontrar una solución $w = (w_1, \ldots, w_n)$ para el sistema asociado:

$$\begin{cases} p_1(w_1, \ldots, w_n) = z_1 \\ p_2(w_1, \ldots, w_n) = z_2 \\ \quad \vdots \\ p_m(w_1, \ldots, w_n) = z_m \end{cases}$$

Algunos de los esquemas más destacados en esta línea son:

- La firma QUARTZ [22], que destaca por producir firmas muy cortas (100 bits).
- El cifrado ZHFE [23], para el que aún no se dispone de una demostración de seguridad.

Pese a las ventajas que estos esquemas pueden presentar (como su flexibilidad para ser implementados en distintas plataformas), existen serias dudas sobre la seguridad que alcanzan. Otro inconveniente (menor) que presentan es que el tamaño de sus claves es bastante grande. Para más información, ver [11].

3.4. Criptografía basada en Retículos

Desde 1998, se ha evaluado la utilidad de problemas difíciles sobre retículos a la hora de construir herramientas criptográficas. Las técnicas computacionales para retículos de enteros fueron fundamentales para el criptanálisis de los primeros esquemas combinatorios de cifrado, así como para evaluar la robustez de las claves utilizadas por el esquema RSA o las funciones hard-core asociadas a funciones unidireccionales. En los últimos años, sin embargo, su papel ha sido distinto al proporcionar construcciones para cifrado homomórfico y potencialmente resistente a criptanálisis cuántico.

3.4.1. Fundamentos matemáticos

La mayoría de los esquemas de cifrado de clave pública basados en retículos se basan en el problema designado con las siglas LWE (*learning with errors*).

El problema LWE**.** Sea $n \in \mathbb{N}$ y consideremos q un entero positivo (cuyo tamaño en bits es similar a n). Consideremos n vectores $\vec{b}_1, \ldots, \vec{b}_n$ cuyas coordenadas están en $\mathbb{Z}_q$. El retículo Λ generado por la base de vectores $B = \{\vec{b}_1, \ldots, \vec{b}_n\}$ queda definido por:

$$\Lambda = \{\Sigma_{i=1}^{n} z_i \cdot \vec{b}_i \mid z_i \in \mathbb{Z}\}.$$

Con frecuencia, para hacer explícita la base, Λ se denota $\mathcal{L}(B)$.

Una *instance* del problema LWE se plantea de la siguiente manera: consideremos fijado un vector secreto $\vec{s}$ con n coordenadas en $\mathbb{Z}_q$.[2]

1. Elegimos un vector $\vec{a} \in \mathbb{Z}_q^n$.
2. Elegimos un error e al azar según una distribución T, donde T es una cierta distribución de probabilidad.[3]
3. Calculamos el producto escalar de $\vec{a}$ y $\vec{s}$, que denotaremos $\langle \vec{a}, \vec{s} \rangle$.
4. Definimos $t = \langle \vec{a}, \vec{s} \rangle + e \pmod{q}$.
5. Damos como salida el par $(\vec{a}, t) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Se plantea entonces el problema de como recuperar $\vec{s}$ dada una colección de pares $\{(\vec{a}_i, t_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q\}_{i=1,\ldots m}$ construidos mediante el proceso anterior. La dificultad de este problema depende de cómo elijamos la distribución T y los parámetros n y q. Con frecuencia, la distribución T es una distribución llamada *distribución Gaussiana discreta* que depende de un parámetro real positivo s. Así, lo habitual es definir cada instancia concreta de LWE a partir de los tres parámetros (n, q, s).[4] Hay una versión *decisional* del problema anterior, que plantea distinguir los valores $t_1, \ldots, t_m$ de valores seleccionados al azar en $\mathbb{Z}_q$.

[2] Matemáticamente el conjunto que contiene esos vectores se denota $\mathbb{Z}_q^n$, usaremos en lo sucesivo dicha notación.

[3] La definición de T es relativamente compleja y deberá explicitarse en cada implementación concreta.

[4] Típicamente, tanto q como s son de la forma n^α para distintas constantes α, es decir: estos tres parámetros no son independientes.

Desafortunadamente, no resulta sencillo dar una evaluación de seguridad estricta que permita conocer cómo influyen los parámetros (n, q, s) en la dificultad del problema anterior. Aunque esta cuestión es objeto de numerosos trabajos actuales (ver, por ejemplo [16]), los expertos no han conseguido explicitar qué impacto tiene tomar, por ejemplo, dimensiones n mayores o menores en la seguridad de los esquemas criptográficos asociados. Existen recomendaciones basadas esencialmente en ataques heurísticos (ver por ejemplo [26]).

3.4.2. Propuestas

Existen distintas propuestas para construir cifrado de clave pública a partir de este problema y de problemas relacionados, como el llamado *problema del vector más próximo* o *closest vector problem*. Mencionamos los más destacados:

- Esquema de Regev [24]: primera propuesta basada en LWE. Esencialmente académica.
- NTRU [17]: propuesto en 1998, no es completamente homomórfico pero mantiene bien la estructura para un número prefijado de cálculos con dos operaciones no demasiado grande (según implementaciones).
- BGV [4]: completamente homomórfico, implementado en la librería HELib. Es una variante del aclamado esquema de Gentry que usa el problema LWE sobre anillos (llamado también RLWE).

Hacemos mención especial al trabajo reciente [2], que puede servir como puente entre nuestros recientes resultados para la construcción de claves en grupos utilizando hash-proof systems y el escenario post-cuántico. Más concretamente, muchas construcciones criptográficas basadas en hash-proof systems pasarían a ser post-cuánticas gracias estos resultados (por ejemplo, nuestro esquema para la intersección privada propuesto en [10]).

3.5. Criptografía basada en Hashes

Una función hash o función resumen, es simplemente una aplicación que transforma cadenas de bits de longitud arbitraria en cadenas de longitud prefijada. Estas funciones se utilizan ampliamente en criptografía, esencialmente para construir pruebas de integridad o acelerar la comparación de

valores. En el primer caso, un valor $H(m)$, trasmitido junto a un mensaje, proporciona una etiqueta para verificar si m se ha modificado en el proceso de trasmisión. En el segundo, por ejemplo, es frecuente almacenar hashes de contraseñas (en lugar de contraseñas de usuarios) a la hora de establecer mecanismos de control de acceso. Un requerimiento imprescindible para la mayoría de los usos de funciones hash, es que sea difícil encontrar *colisiones*, es decir, dos valores distintos cuyos *resumenes* (imagenes por la función hash H) coincidan.

Muchas de las funciones hash utilizadas en la actualidad serían vulnerables a ataques cuánticos que utilizasen el algoritmo de Grover. La manera trivial de evitar dichos ataques, neutralizando la ventaja cuadrática en búsquedas no estructuradas que da el algoritmo de Grover, es doblar el rango de los hashes para cualquiera de sus usos. La criptografía basada en funciones hash tiene especial interés en el escenario post-cuántico dentro del escenario de la firma digital. En principio, las firmas construidas con hashes usando los llamados árboles de Merkle constituyen los candidatos más sólidos para firmas post-cuánticas, destacando los esquemas XMSS y SPHINCs (ver [18, 3])— siempre con claves de 256 bits.

Informalmente un árbol de Merkle se describe como una estructura de datos en árbol, binario o no, de modo que cada nodo que no es una hoja está etiquetado con el hash de la concatenación de las etiquetas o valores de sus hijos. De este modo, se posibilita que un gran número de datos separados puedan ser ligados a un único valor de hash, el hash del nodo raíz del árbol. A través de esta estructura puede definirse por tanto un método de verificación segura y eficiente de los contenidos de grandes estructuras de datos.

Referencias

[1] G. Alagic, A. Russell. "Quantum-secure symmetric-key cryptography based on hidden shifts". Cryptology ePrint Archive: Report 2016/960. 2016. Online: `http://eprint.iacr.org/2016/960`. 3

[2] F. Benhamouda, O. Blazy, L. Ducas, W. Quach. "Hash proof systems over lattices revisited". En Proc. PKC '18, Lecture Notes in Computer Science **10770**, 644–674. Springer, 2018. 8

[3] D.J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, Z. Wilcox-O'Hearn. "SPHINCS: Practical stateless hash-based signatures". En Proc. EUROCRYPT '15, Lecture Notes in Computer Science **9056**, 368–397. Springer, 2015. 9

[4] Z. Brakerski, C. Gentry, V. Vaikuntanathan. "(Leveled) Fully homomorphic encryption without bootstrapping". ACM Trans. Computation Theory **6(3)**, 13:1–13:36, 2014. 8

[5] A. Canteaut, F. Chabaud. "A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense BCH codes of length 511". IEEE Trans. Information Theory **44(1)**, 367–378, 1998. 5

[6] C. Chen, T. Eisenbarth, I. von Maurich, R. Steinwandt. "Differential power analysis of a mceliece cryptosystem". En Proc. ACNS '15, Lecture Notes in Computer Science **9092**, 538–556. Springer, 2015. 5

[7] C. Chen, T. Eisenbarth, I. von Maurich, R. Steinwandt. "Masking large keys in hardware: A masked implementation of McEliece". En Proc. SAC '15, Lecture Notes in Computer Science **9566**, 293–309. Springer, 2015. 5

[8] C. Chen, T. Eisenbarth, I. von Maurich, R. Steinwandt. "Horizontal and vertical side channel analysis of a McEliece cryptosystem". IEEE Trans. Information Forensics and Security **11(6)**, 1093–1105, 2016. 5

[9] L. Chen, S. Jordan, Y-K. Liu, D. Moody, R. Peralta, R. Pelner, D. Smith-Tone. "Report on post-quantum cryptography". 2016. Online: `https://csrc.nist.gov/publications/detail/nistir/8105/final`. 2

[10] P. D'Arco, M.I. Gonzalez Vasco, A.L. Pérez del Pozo, C. Soriente, R. Steinwandt. "Private set intersection: New generic constructions and feasibility results". Adv. in Math. of Comm. **11(3)**, 481–502, 2017. 8

[11] J. Ding, B-Y. Yang. "Multivariate Public Key Cryptography". En Post-Quantum Cryptography, 193–241, Springer, Berlin, Heidelberg, 2009. 6

[12] M. P. C. Fossorier, K. Kobara, H. Imai. "Modeling bit flipping decoding based on nonorthogonal check sums with application to iterative decoding attack of McEliece cryptosystem". IEEE Trans. Information Theory **53(1)**, 402–411, 2007. 5

[13] M.I. González Vasco, R. Steinwandt. "Group Theoretic Cryptography". Chapman & Hall/CRC, 1st edition. 2015. 3

[14] M.I. González Vasco, A.L. Pérez del Pozo, P. Taborda Duarte, J.L. Villar. "Cryptanalysis of a key exchange scheme based on block matrices". Information Sciences **276**, 319–331, 2014. 3

[15] L.K. Grover. "A fast quantum mechanical algorithm for database search". En Proc. the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, 212—219, ACM, New York, NY, USA, 1996. 1, 2

[16] G. Herold, E. Kirshanova, A. May. "On the asymptotic complexity of solving LWE". Designs, Codes and Cryptography **86**, 55–83, 2018. 8

[17] J. Hoffstein, J. Pipher, J.H. Silverman. "NTRU: A ring-based public key cryptosystem". En Proc. ANTS '98, Lecture Notes in Computer Science **1423**, 267–288. Springer, 1998. 8

[18] A. Huelsing, D. Butin, S-L. Gazdag, J. Rijneveld, A. Mohaisen. "XMSS: Extended Hash-Based Signatures". Internet-Draft draft-irtf-cfrg-xmss-hash-based-signatures-10, Internet Engineering Task Force, July 2017 (Work in Progress). Online: `https://tools.ietf.org/html/draft-irtf-cfrg-xmss-hash-based-signatures-10`. 9

[19] A.Kh. Al Jabri. "A statistical decoding algorithm for general linear block codes". En Proc. IMA '01, Lecture Notes in Computer Science **2260**, 1–8. Springer, 2001. 5

[20] R.J. McEliece. “A Public-Key Cryptosystem Based On Algebraic Coding Theory”. Deep Space Network Progress Report **44**, 114–116, 1978. 3

[21] L. Minder, A. Shokrollahi. “Cryptanalysis of the Sidelnikov cryptosystem”. En Proc. EUROCRYPT ’07, Lecture Notes in Computer Science **4515**, 347–360. Springer, 2007. 5

[22] J. Patarin, N. Courtois, L. Goubin. “Quartz, 128-bit long digital signatures”. En Proc. CT-RSA ’01, Lecture Notes in Computer Science **2020**, 282–297. Springer, 2001. 6

[23] J. Porras, J. Baena, J. Ding. “ZHFE, a new multivariate public key encryption scheme”. En Proc. PQCrypto ’14, Lecture Notes in Computer Science **8772**, 229–245. Springer, 2014. 6

[24] O. Regev. “On lattices, learning with errors, random linear codes, and cryptography”. En Proc. 37th Annual ACM Symposium on Theory of Computing, 84–93, ACM, New York, NY, USA, 2005. 8

[25] P.W. Shor. “Polynominal time algorithms for discrete logarithms and factoring on a quantum computer”. En Proc. ANTS ’94, Lecture Notes in Computer Science **877**, 289–289. Springer, 1994. 1

[26] D. Stehlé, R. Steinfeld. “Making NTRU as secure as worst-case problems over ideal lattices”. En Proc. EUROCRYPT ’11, Lecture Notes in Computer Science **6632**, 27–47. Springer, 2011. 8