UN MODELO COMPACTO DE CRIPTOGRAFÍA ASIMÉTRICA EMPLEANDO ANILLOS NO CONMUTATIVOS

Juan Pedro Hecht

Carrera de Especialización en Criptografía y Seguridad Teleinformática – EST (IESE) *
Carrera de Especialización y Maestría en Seguridad Informática – UBA **

* Escuela Superior Técnica – Instituto de Enseñanza Superior del Ejército (IESE)
Av. Cabildo 15 - (C1426AAA) - Buenos Aires – Argentina
** Facultad de Cs. Exactas y Naturales/Facultad de Ingeniería/Facultad de Cs. Económicas
Universidad de Buenos Aires – Argentina
phecht@dc.uba.ar
http://cripto.iese.edu.ar/

Resumen. La criptografía asimétrica (AC) ha estado presente desde los trabajos pioneros de Diffie, Pohlig, Merkle y Hellman. Prácticamente todos los protocolos criptográficos AC están vinculados a funciones trampa de una vía. Hoy día las primitivas de mayor difusión están vinculadas al problema de la factorización de enteros o al problema del logaritmo discreto. Ambos están íntimamente vinculados y pertenecen en general a operaciones aritméticas en estructuras algebraicas conmutativas (grupos, anillos y campos). Una desventaja reside en que se conocen ataques de complejidad subexponencial (y tal vez existan métodos de resolución en tiempo polinómico) y ataques eficientes a través de algoritmos cuánticos, de modo que hay serios motivos para desconfiar de los mismos. Para reemplazar de raíz a la AC convencional (y su aplicación en criptografía de clave pública), ha surgido en años recientes un interés creciente en la aplicación de estructuras algebraicas no conmutativas. Con este objetivo en mente, se desarrollaron sistemas AC usando grupos de distinta naturaleza (trenzas, Thompson, nilpotentes, simétricos, policíclicos) y recientemente anillos de polinomios matriciales. Hay que enfatizar que todos usan, de una forma u otra, bibliotecas de precisión múltiple de manera que se complica su aplicación en plataformas de baja capacidad computacional. En este trabajo presentamos una solución (la definimos como *modelo compacto*) y que consiste en la aplicación de polinomios matriciales en un formato particular. Estas matrices de orden cuatro poseen elementos de 8 bits y todas las operaciones se llevan a cabo en aritmética Z_{256} de manera que pueden ser manejadas por procesadores de 16 bits (tarjetas inteligentes, teléfonos celulares, llaves criptográficas USB, etc.). Tal como se ilustra en este trabajo, su simplicidad no afecta a la seguridad de empleo, siempre que se tomen los recaudos del caso. Este enfoque puede servir para avanzar en el desarrollo de nuevas soluciones *compactas* de AC.

Palabras clave: criptografía asimétrica, álgebra no conmutativa, anillos.

1 ANTECEDENTES

En 1976, Diffie y Hellman concibieron su histórico algoritmo de intercambio de claves y Pohlig-Hellman un algoritmo asimétrico de encripción [1]. Esa brillante idea fue absorbida y perfeccionada por Rabin, Shamir y Adleman con el desarrollo de la criptografía de clave pública (PKC) [2], la que maduró luego en la plataforma tecnológica PKI (*public-key infrastructure*). El fundamento de la criptografía asimétrica (AC) consiste en el empleo de funciones trampa de una vía [1]. Hoy día, las primitivas más exitosas de la AC están basadas en problemas computacionalmente difíciles (de complejidad NP o al menos cuando no se conoce algoritmos de ataque de complejidad P) que con una pieza de información secreta se transforman en resolubles (complejidad P) [3]. En particular el énfasis ha sido puesto en el problema de la factorización de enteros positivos (IFP) y el problema del logaritmo discreto (DLP) [1] en *grupos conmutativos* de dimensiones suficientemente elevadas acorde con la tecnología de ataque estado-del-arte.

Resumiendo, los anillos más empleados son $\mathbb{Z}_n$ y su extensión $\mathbb{Z}_n[x]$. Así fueron desarrollados protocolos de amplia difusión como RSA [2], Rabin-Williams [4], el sistema LUC [5] o versiones RSA basadas en curvas elípticas KMOV [6]. Otro ejemplo adecuado es la familia ElGamal incluyendo su versión básica [7], criptosistemas de curvas elípticas (ECC) e hiperelípticas (HECC) [3], el intercambio de claves Diffie-Hellman [8], el estándar de firma digital DSS que emplea ElGamal [1] o el esquema McCurley [9], todos basados en la dificultad de resolver logaritmos discretos en un campo finito generado por un primo de tamaño suficiente y por supuesto incluidos en un *grupo conmutativo*. En particular para ECC, no existen siquiera algoritmos de tiempo subexponencial para poder atacarlo.

Tradicionalmente, los fundamentos de la AC y la PKC se hallaban mas en el ámbito de la teoría de números que en el espacio del álgebra abstracta [3], [10]. En computación cuántica [11] tanto IFP y DLP, como DLP sobre curvas elípticas (ECDLP) pueden ser eficazmente atacados gracias a los algoritmos de Shor [12], Kitaev [13] y Proos-Zalka [14]. A pesar que una computadora cuántica realmente eficaz a estos fines esté tal vez unas décadas más allá del horizonte actual, la mera noción de esa debilidad crea desconfianza en los protocolos vigentes [15]. Es por este motivo que en los últimos años se está enfatizando la búsqueda de nuevos algoritmos de naturaleza más algebraica que numérica [15]. Resumamos las principales líneas exploradas:

- En 1985, Wagner [16] propuso un protocolo PKC basado en el problema indecidible de las palabras en *grupos* y *semigrupos*. Esta idea fue perfeccionada por otros investigadores [17].
- En 1999, Anshel [18] propuso un protocolo PKC basado en la dificultad de resolver ecuaciones en *grupos no conmutativos*. En su trabajo propusieron incluso el uso de la teoría de trenzas como alternativa viable.
- En 2000, Ko [19] presenta por primera vez un protocolo PKC fundamentado en el empleo de la teoría de trenzas. Este tema alcanzó una popularidad inmediata [18], [20]. Sin embargo, desde 2001 a 2003, el criptoanálisis exitoso frenó ese en-

tusiasmo [21]. Incluso algunos autores anunciaron la muerte prematura de la PKC basada en *grupos* de trenzas [21]. Seguramente no está dicha la última palabra sobre este tema [22].

- En 2001, Paeng [23] publica un esquema PKC basado en *grupos finitos no conmutativos*. El método emplea el problema DLP en el *automorfismo* definido por la operación de *conjugación*. Específicamente se trata de la dificultad de resolver la búsqueda del *elemento conjugador* (CSP). Este sistema ha sido ulteriormente perfeccionado como protocolo MOR [24].
- Simultáneamente Magliveras [10] presenta nuevos esquemas usando funciones trampa de una vía sobre *grupos finitos*. Así presenta dos sistemas PKC basados en la dificultad de factorizar productos en *grupos finitos*, denominados MST1 y MST2. En 2002, González Vasco [25] demuestra que generalizando lo necesario, los conceptos de factorización empleados, se obtiene una plataforma general que unifica criptosistemas disímiles como MST1, MST2, MOR, ElGamal y teoría de trenzas.
- En 2002, Grigoriev y Ponomarenko [26] desarrollaron criptosistemas *homomorficos* en *anillos conmutativos* de *matrices enteras* y los extendieron [27] a *semigrupos finitos* basándose en la dificultad del problema de la *pertenencia* en *grupos* de *matrices* Z^*_n.
- En 2004, Eick y Kahrobaei [28] proponen generar sistemas PKC usando *grupos policíclicos*, una generalización natural de los grupos cíclicos pero de estructura más compleja. Esta línea puede llegar a ser muy promisoria en el futuro.
- En 2004 y 2005, Shpilrain y Ushakov [29], [30] sugieren el uso del *grupo de Thompson* para desarrollos PKC. Se trata de un *grupo no-abeliano infinito* con representación finita. Allí se plantea por primera vez la dificultad del problema de la *descomposición simétrica* (SDP) del que hablaremos más adelante en este trabajo y que es una generalización del problema CSP.
- En 2007, Cao, Dong y Wang [31] presentan un interesante esquema general de PKC basándose en la dificultad de resolver SDP en un *anillo no conmutativo* de *polinomios matriciales* (PSDP). Esto conduce directamente a protocolos de intercambio de claves Diffie-Hellman generalizados (GDHP) y a criptosistemas ElGamal generalizados.
- En 2008, Mahalanobis [32] propone con igual finalidad el empleo de presentaciones finitas de *grupos nilpotentes de clase 2*.
- En 2008, Thomas y Lal [33] presentaron por primera vez un criptosistema genérico de de firma digital asegurada (UDS o *undeniable digital signature*) embebido en un protocolo de prueba de conocimiento cero (ZKP). Se basaron en la dificultad de los problemas CSP y SDP. En su desarrollo proponen el uso de *álgebra no conmutativa* pero no definen ninguna estructura en particular aunque sugieren como alternativas el empleo de *grupos de trenzas*, *grupos policíclicos* o el *grupo de Thompson*.
- En 2008, Anjaneyulu, Reddy y Reddy [34] aplicaron la teoría desarrollada por Cao [31] generando un protocolo UDS basándose en el problema DSDP.

Sintetizando, existe una fuerte tendencia al reemplazo de AC y PKC basado en *sistemas algebraicos numéricos y conmutativos* por otros de naturaleza *no conmutativa*. Podemos resumir que en cada caso se buscan funciones trampa de una vía que condu-

cen a generalizaciones del problema DLP. Este problema de *logaritmo discreto generalizado* (GDLP) en el marco de una *estructura algebraica no conmutativa* (NCAS) es especialmente atractivo por dos motivos: a) no existen algoritmos de tiempo subexponencial para atacarlos, léase Index Calculus [1] y b) aún (hoy día) son inmunes al ataque con computadoras cuánticas [11]. Sin embargo hay que poner mucha atención en la clase de *estructura* y *operación* que se elige, sirva como ejemplo la fugaz moda de la llamada "criptografía de *trenzas*" [35].

2 OBJETIVO DEL PRESENTE TRABAJO

Hay ciertos inconvenientes que subsisten en la criptografía NCAS, uno de los cuales es la necesidad de empleo de bibliotecas de precisión extendida y el uso de *hardware* que permita usarlas con fluidez [1]. Un desafío abierto es hallar un modelo NCAS que sea criptográficamente seguro pero capaz de emplear recursos computacionales escasos, lo que permitirá una eficaz operación en plataformas de baja potencia. Estos modelos que definiremos como ***compactos***, emplearán procesadores de 16 bits, en contenedores inteligentes como *smartcards*, teléfonos celulares o llaves inteligentes USB.

Este trabajo, inspirado en el desarrollo de Cao [31], explora como alternativa el uso de un modelo compacto de NCAS empleando el *anillo no conmutativo* de *matrices* $M_4[\mathbb{Z}_{256}]$ que no requieren el empleo de bibliotecas numéricas de precisión extendida, independientemente de su extensión espacial (memoria ocupada = 16 bytes por matriz) dado que todas las operaciones numéricas se efectuarán dentro del anillo numérico $\mathbb{Z}_{256}$. Tal como lo propuso Cao en su trabajo, la dificultad de ataque al protocolo requiere operar con un módulo RSA ($n=p.q$, primos fuertes) y por ende no sólo requiere bibliotecas de precisión extendida sino que además queda expuesto a los ataques del problema IFP. Esto se corrige con el modelo que aquí se presenta. Como ejemplo, se aplicará la estructura no conmutativa en un protocolo Diffie-Hellman generalizado (GDHP), aunque queda claro por lo ya descripto en la bibliografía que no existe inconveniente en trasladarlo a un criptosistema ElGamal, a un esquema de firma digital [31] o tal vez a un protocolo ZKP [33]. El cardinal del espacio de matrices propuesto es $|M_4[\mathbb{Z}_{256}]| = 2^{128}$ pero como se verá más adelante su dimensión no incide en la seguridad del protocolo.

3 FUNDAMENTOS ALGEBRAICOS Y COMPUTACIONALES

3.1. Problemas complejos en grupos (y semigrupos) no conmutativos

En un *grupo no conmutativo* G, dos elementos (x, y) son *conjugados*, representado como $x \sim y$, si $y = z^{-1}x\,z$ para algún $z \in G$. Aquí a z o a z^{-1} se lo denomina

el elemento *conjugador*. Así se definen problemas que a su manera, representan instancias algebraicas del problema generalizado del *logaritmo discreto* (GDLP) .

(CSP) Problema de la búsqueda del conjugador: dados $(x, y) \in G \times G$, hallar $z \in G$ tal que $y = z^{-1} x\ z$.

(DP) Problema de la descomposición: dados $(x, y) \in G \times G\ y\ S \subseteq G$, hallar $(z_1, z_2) \in S$ tal que $y = z_1 x z_2$.

(SDP) Problema de la descomposición simétrica: dados $(x, y) \in G \times G$, y $(m, n) \in \mathbb{Z}$, hallar $z \in G$ tal que $y = z^m x\ z^n$.

(GSDP) Problema de la descomposición simétrica generalizada: dados $(x, y) \in G \times G\ y\ S \subseteq G$ y $(m, n) \in \mathbb{Z}$, hallar $z \in S$ tal que $y = z^m x\ z^n$.

Acorde a los conocimientos presentes, todos los problemas aquí aplicados sobre *grupos no conmutativos* generales son lo suficientemente difíciles como para ser empleados con seguridad criptográfica [31]. Están ordenados por grado de complejidad creciente. En particular GSDP es al menos tan complejo como SDP. Concluimos que no se conoce (y casi seguramente no exista) algoritmo probabilístico de tiempo P capaz de resolver GSDP si la escala dimensional es la adecuada. Dado que GSDP puede ser considerado una instancia del GDLP, podemos agregar:

(CDH) Problema computacional Diffie-Hellman en el grupo no conmutativo G (con respecto a algún subgrupo abeliano S): Computar $x^{z_1 z_2}$ (= $x^{z_2 z_1}$) para un dado $(x,\ x^{z_1}, x^{z_2})$ tal que $x \in G$, $(z_1, z_2) \in S$ y $S \subseteq G$.

Hoy día no hay ninguna idea acerca de cómo extraer z_1 (o z_2) de los datos provistos [34] y eso es alentador si se lo compara con el empleo tradicional de *grupos numéricos conmutativos*. Las mismas consideraciones son válidas si G fuese un *semigrupo no conmutativo*, basta con restringir las condiciones de los exponentes a valores enteros positivos $(m, n) \in \mathbb{Z}_{>0}$ o $(z_1, z_2) \in \mathbb{Z}_{>0}$.

3.2. Problemas complejos en anillos no conmutativos

Para ejemplificar los siguientes problemas computacionalmente complejos trabajaremos sobre *anillos no conmutativos de polinomios con coeficientes enteros positivos*. Sea R un anillo $(R, +, 0)$ y $(R, \cdot, 1)$, donde la suma forma un *grupo abeliano* y el producto forma un *semigrupo no abeliano*.

Definición 1. El producto escalar se define para $k \in \mathbb{Z}_{>0}$ y $r \in R$ y vale

$$(k)r \triangleq \langle r + \cdots + r \rangle_{k\ veces} \tag{1}$$

Cuando $k \in \mathbb{Z}_{<0}$ vale

$$(k)r \triangleq (-k)(-r) = \langle (-r) + \cdots + (-r) \rangle_{k\ veces} \tag{2}$$

Para $k = 0$ es natural que $(k)r = 0$. Similarmente, vale $r^k = \langle r \cdot ... \cdot r \rangle_{k\ veces}$.

Teorema 1. $(a)r^m.(b)r^n = (ab)r^{m+n} = (b)r^n.(a)r^m, \quad \forall a,b,m,n \in \mathbb{Z} \ \ y \ \forall r \in R$, lo que se demuestra elementalmente aplicando la *propiedad asociativa del anillo* y la *propiedad conmutativa* de la *suma*.

Lema 1. Nótese que en general $(a)r.(b)s \neq (b)s.(a)r \quad si \quad r \neq s$ dada la *no conmutatividad* del *semigrupo producto*. Esta distinción es crucial para el modelo desarrollado.

Definición 2. Se define el anillo de polinomios de coeficientes enteros positivos para todo elemento $r \in R$

$$f(r) = \sum_{i=0}^{n} (a_i)r^i = (a_0).1 + (a_1)r + \cdots + (a_n)r^n \quad donde \ \ \forall i \mid a_i \in \mathbb{Z}_{>0} \tag{3}$$

Aquí *r* puede considerarse como una simple variable (*x*) y el conjunto de los polinomios *f(r)* puede considerarse como la extensión polinómica del anillo $\mathbb{Z}_{>0}[r]$ para todo polinomio con coeficientes enteros positivos *f(r)* y todo $r \in R$.

Teorema 2. Dados dos polinomios cualesquiera de la extensión del anillo $f(r),\ h(r) \in \mathbb{Z}_{>0}[r]$, se demuestra

$$f(r).h(r) = h(r).f(r) \tag{4}$$

Es elemental la demostración usando el Teorema 1. y la *propiedad distributiva* del *producto* respecto de la *suma*.

Lema 2. Nótese que en general $f(r).h(s) \neq h(s).f(r) \quad si \quad r \neq s$ dada la *no conmutatividad* del *semigrupo producto*. También esta distinción resulta importante para fundamentar al modelo desarrollado.

Definición 3. Dado un *anillo no conmutativo* $(R, +, \cdot)$ y cualquier elemento $a \in R$ elegido al azar, definimos un *subanillo* $P_a \subseteq R$ por

$$P_a \triangleq \{ f(a) : f(x) \in \mathbb{Z}_{>0}[x] \} \tag{5}$$

Es decir P_a abarca al conjunto de todos los polinomios de la extensión aplicados sobre la misma variable *a*.

Ahora podemos definir nuevas versiones de los problemas SDP y CDH definidos en el apartado 3.1.

(PSD) Problema de la descomposición simétrica sobre el anillo no conmutativo de polinomios con coeficientes enteros positivos *R*: dados $(a, x, y) \in R^3$, y $(m, n) \in \mathbb{Z}$, hallar $z \in P_a$ tal que $y = z^m x\ z^n$.

(PDH) Problema computacional Diffie-Hellman en el anillo no conmutativo de polinomios con coeficientes enteros positivos *R*: Computar $x^{z_1 z_2}$ (= $x^{z_2 z_1}$) para un dado (a, x, x^{z_1}, x^{z_2}) tal que $a \neq x$, $(a, x) \in R^2$ y $(z_1, z_2) \in P_a$.

Nótese que P_a es precisamente el *subanillo conmutativo* necesario para poder implementar el protocolo de intercambio de claves Diffie-Hellman que se presenta

en el siguiente punto. No importa que instancias polinómicas *f(a), h(a)* se elijan, su producto será independiente del orden. Pero también es cierto que esto deja de ser válido si el producto de esos mismos polinomios se aplica a distintos elementos *f(a), h(b)*, ver el Teorema 2. y su Lema. Resulta evidente que si valen todos los antecedentes previamente expuestos, ambos problemas (PSD y PDH) son computacionalmente intratables, o sea no existe (o al menos no se conoce) un algoritmo probabilístico de tiempo P que pueda resolverlos con precisión no despreciable con respecto a la escala dimensional del problema [3] (lo que se consigue con cardinales $|R|$ y $|P_a|$ suficientemente grandes como para no ser resuelto con los recursos computacionales vigentes).

4 PROTOCOLO DE INTERCAMBIO DE CLAVES DIFFIE-HELLMAN USANDO ANILLOS NO CONMUTATIVOS

Se han expuesto los fundamentos necesarios y ahora se está en condiciones de presentar un protocolo (*ejemplo*) de intercambio de claves Diffie-Hellman empleando *anillos no conmutativos de polinomios matriciales con coeficientes enteros positivos*.

(a) ALICE envía dos números enteros positivos aleatorios $(m,n) \in \mathbb{Z}_{16} > 0$ y dos elementos aleatorios $(a,b) \in R$ a BOB a través de un canal público e inseguro. Cada elemento aleatorio es una matriz $M_4[\mathbb{Z}_{256}]$ de orden 4 y valores en $\mathbb{Z}_{256}$.

(b) ALICE elige un polinomio entero al azar no nulo $f(x) \in \mathbb{Z}_{16}[x]$ tal que $f(a) \neq 0$ y sus coeficientes y exponentes sean enteros positivos ($\in \mathbb{Z}_{16} > 0$) y lo define como su clave privada.

(c) BOB elige un polinomio entero al azar no nulo $h(x) \in \mathbb{Z}_{16}[x]$ tal que $h(a) \neq 0$ y sus coeficientes y exponentes sean enteros positivos ($\in \mathbb{Z}_{16} > 0$) y lo define como su clave privada.

(d) ALICE computa $r_A = f(a)^m . b . f(a)^n$ y se lo envía por igual canal a BOB.

(e) BOB computa $r_B = h(a)^m . b . h(a)^n$ y se lo envía por igual canal a ALICE.

(f) ALICE computa la clave de sesión compartida $K_A = f(a)^m . r_B . f(a)^n$

(g) BOB computa la clave de sesión compartida $K_B = h(a)^m . r_A . h(a)^n$

Observar que $K_A = K_B$ dado que se verifica

$$K_A = f(a)^m . h(a)^m . b . h(a)^n . f(a)^n = h(a)^m . f(a)^m . b . f(a)^n . h(a)^n = K_B \quad (6)$$

En la práctica los pasos (a) (b) (d) pueden ser simultáneos y requieren un solo mensaje por parte de ALICE. Luego los pasos (c)(e) son simultáneos y BOB envía un único mensaje. Finalmente (f)(g) son autónomos. Hay que advertir dos detalles acerca del muestreo y enmascaramiento de los datos transmitidos:

(1) La selección de los elementos *(a,b)* debe ser aleatoria e uniforme dentro de su espacio. Dado que se trata de matrices numéricas con 16 valores del intervalo [0 - 255] no resultará difícil llevar adelante la selección. La misma condición debe regir para el resto de los parámetros del sistema. Algunas de estas matrices modulares resultan ser *nilpotentes*, por lo cual deben ser descartadas por ALICE porque generan claves de sesión débiles.

(2) Una medida excepcional a considerar consiste en el enmascaramiento de los datos en tránsito. Si se pretendiese armar un (micro) entorno de espacio reducido con el objeto de incrementar la velocidad en una plataforma de recursos escasos (lo cual es en principio desaconsejable) será conveniente enmascarar los datos en tránsito para dificultar el ataque [21], [36].

5 CONSIDERACIONES ACERCA DE LA SEGURIDAD COMPUTACIONAL DEL PROTOCOLO PROPUESTO

Queda claro que apoyándose en la dificultad computacional de los problemas PSD y PDH y las claves privadas de tamaño *adecuado*, el protocolo será potencialmente inmune a los ataques.

Se debe resaltar que incrementar el orden de los parámetros (m,n) o de las matrices (a,b) no aporta seguridad al protocolo dado que son elementos públicos. La seguridad reside exclusivamente en el cardinal del espacio de las claves privadas, es decir en el grado y los coeficientes de los polinomios que elijan las partes. Si se considera que un *ataque de fuerza bruta* debe trabajar simultáneamente con ambas claves privadas, la complejidad de dicho ataque será proporcional al cardinal de pares de polinomios privados.

Grado de los polinomios y sus coeficientes. Considerando que hay $O(m^2)$ polinomios enteros $\mathbb{Z}_m[x]$ de grado $\leq m$ y coeficientes en $\mathbb{Z}_m$, al elegir $\mathbb{Z}_{16}[x]$ en el protocolo ejemplo existen $O(2^{16})$ pares de polinomios. Obviamente no se puede usar este ejemplo en aplicaciones prácticas, pero incrementando el espacio $\mathbb{Z}_m[x]$ a valores *razonables* para transacciones de vida útil *breve*, por ejemplo con polinomios $\mathbb{Z}_{256}[x]$, el número de pares de claves privadas se incrementa a $O(2^{32})$. En aplicaciones corrientes no sería aventurado pensar en operar con claves privadas del orden $\mathbb{Z}_{4096}[x]$ en cuyo caso la complejidad del ataque será $O(2^{48})$ para lograr un nivel de seguridad computacional aceptable.

A pesar de ello se prefirió presentar al protocolo con claves privadas $\mathbb{Z}_{16}[x]$ con el solo objeto de mostrarlo aquí como ejemplo.

6 UN EJEMPLO DEL PROTOCOLO

(a) Todas las operaciones numéricas se efectúan en $\mathbb{Z}_{256}$. ALICE envía a BOB (m=12, n=7) y los elementos

$$a=\begin{pmatrix}17&3&128&45\\255&12&1&65\\136&90&28&13\\201&166&8&28\end{pmatrix};\ b=\begin{pmatrix}244&25&67&117\\9&25&0&59\\11&2&195&219\\104&38&121&44\end{pmatrix}$$

(b) ALICE elige su clave privada $f(x)=11x^{14}+6x^{9}+3x^{7}+13$

(c) BOB elige su clave privada $h(x)=x^{15}+15x^{12}+3x^{8}+11x^{5}+2x+7$

(d) ALICE computa su *token* y se lo envía a BOB (I es la matriz identidad orden 4)

$$f(a)=11\begin{pmatrix}17&3&128&45\\255&12&1&65\\136&90&28&13\\201&166&8&28\end{pmatrix}^{14}+6\begin{pmatrix}17&3&128&45\\255&12&1&65\\136&90&28&13\\201&166&8&28\end{pmatrix}^{9}+3\begin{pmatrix}17&3&128&45\\255&12&1&65\\136&90&28&13\\201&166&8&28\end{pmatrix}^{7}+13.I=\begin{pmatrix}101&70&91&29\\163&208&97&193\\111&46&127&167\\230&1&68&106\end{pmatrix}$$

$$r_A=\begin{pmatrix}101&70&91&29\\163&208&97&193\\111&46&127&167\\230&1&68&106\end{pmatrix}^{12}\begin{pmatrix}244&25&67&117\\9&25&0&59\\11&2&195&219\\104&38&121&44\end{pmatrix}\begin{pmatrix}101&70&91&29\\163&208&97&193\\111&46&127&167\\230&1&68&106\end{pmatrix}^{7}=\begin{pmatrix}214&107&138&118\\4&243&148&80\\12&233&60&176\\73&202&107&91\end{pmatrix}$$

(e) BOB computa su *token* y se lo envía a ALICE

$$h(a)=\begin{pmatrix}17&3&128&45\\255&12&1&65\\136&90&28&13\\201&166&8&28\end{pmatrix}^{15}+15\begin{pmatrix}17&3&128&45\\255&12&1&65\\136&90&28&13\\201&166&8&28\end{pmatrix}^{12}+3\begin{pmatrix}17&3&128&45\\255&12&1&65\\136&90&28&13\\201&166&8&28\end{pmatrix}^{8}+$$

$$+11\begin{pmatrix}17&3&128&45\\255&12&1&65\\136&90&28&13\\201&166&8&28\end{pmatrix}^{5}+2\begin{pmatrix}17&3&128&45\\255&12&1&65\\136&90&28&13\\201&166&8&28\end{pmatrix}+7.I=\begin{pmatrix}197&216&148&122\\234&241&102&222\\64&94&251&214\\164&8&90&197\end{pmatrix}$$

$$r_B=\begin{pmatrix}197&216&148&122\\234&241&102&222\\64&94&251&214\\164&8&90&197\end{pmatrix}^{12}\begin{pmatrix}244&25&67&117\\9&25&0&59\\11&2&195&219\\104&38&121&44\end{pmatrix}\begin{pmatrix}197&216&148&122\\234&241&102&222\\64&94&251&214\\164&8&90&197\end{pmatrix}^{7}=\begin{pmatrix}242&43&53&189\\199&73&72&195\\51&96&51&87\\80&140&255&10\end{pmatrix}$$

(f) ALICE computa la clave de sesión compartida

$$K_A = \begin{pmatrix} 101 & 70 & 91 & 29 \\ 163 & 208 & 97 & 193 \\ 111 & 46 & 127 & 167 \\ 230 & 1 & 68 & 106 \end{pmatrix}^{12} \begin{pmatrix} 242 & 43 & 53 & 189 \\ 199 & 73 & 72 & 195 \\ 51 & 96 & 51 & 87 \\ 80 & 140 & 255 & 10 \end{pmatrix} \begin{pmatrix} 101 & 70 & 91 & 29 \\ 163 & 208 & 97 & 193 \\ 111 & 46 & 127 & 167 \\ 230 & 1 & 68 & 106 \end{pmatrix}^{7} = \begin{pmatrix} 192 & 211 & 72 & 228 \\ 166 & 175 & 90 & 150 \\ 82 & 221 & 174 & 162 \\ 65 & 136 & 163 & 107 \end{pmatrix}$$

(g) BOB computa la clave de sesión compartida

$$K_B = \begin{pmatrix} 197 & 216 & 148 & 122 \\ 234 & 241 & 102 & 222 \\ 64 & 94 & 251 & 214 \\ 164 & 8 & 90 & 197 \end{pmatrix}^{12} \begin{pmatrix} 214 & 107 & 138 & 118 \\ 4 & 243 & 148 & 80 \\ 12 & 233 & 60 & 176 \\ 73 & 202 & 107 & 91 \end{pmatrix} \begin{pmatrix} 197 & 216 & 148 & 122 \\ 234 & 241 & 102 & 222 \\ 64 & 94 & 251 & 214 \\ 164 & 8 & 90 & 197 \end{pmatrix}^{7} = \begin{pmatrix} 192 & 211 & 72 & 228 \\ 166 & 175 & 90 & 150 \\ 82 & 221 & 174 & 162 \\ 65 & 136 & 163 & 107 \end{pmatrix}$$

Resulta evidente la obtención de una clave de sesión común, sin emplear aritmética de precisión extendida. Falta definir cómo se traslada la matriz resultante a la clave numérica, aunque es trivial trasladar la matriz a un vector de 16 bytes (128 bits). Hay *16!* formas de concatenar los bytes de la clave común, sin contar que la misma puede finalmente someterse a un *hashing* estándar. Estos detalles también pueden ser protocolizados en las aplicaciones concretas.

7 APLICACIÓN PRÁCTICA: SEGURIDAD Y TIEMPOS

Habiendo definido de la complejidad computacional del ataque de fuerza bruta, se puede asumir que existe una relación entre los cardinales de los espacios de búsqueda de claves privadas (los polinomios), los tiempos de cómputo y los potenciales rangos de aplicación del protocolo aquí presentado.

Resulta de sumo interés obtener estimaciones del tiempo que lleva generar las claves compartidas, lo que puede llegar a ser limitante para las aplicaciones prácticas. En la Fig. 1. se presenta un gráfico que ilustra la relación entre la dimensión de las claves privadas y una estimación pesimista del tiempo de procesamiento. En este caso se usó un código fuente no compilado y no optimizado en tiempos en un ambiente prototipo, sugiriendo que bajo condiciones adecuadas estos tiempos podrán ser significativamente reducidos.

En la Fig. 2. se presenta otro gráfico que ilustra la relación que existe entre el grado del polinomio usado y el cardinal del espacio de claves. Como es trivial de observar, el cardinal del espacio crece exponencialmente. Considerando los costos computacionales crecientes, se presentan posibles bandas de seguridad de compromiso tiempo-seguridad con las potenciales aplicaciones de uso. La comunicación celular encriptada se refiere a tráfico no almacenado.

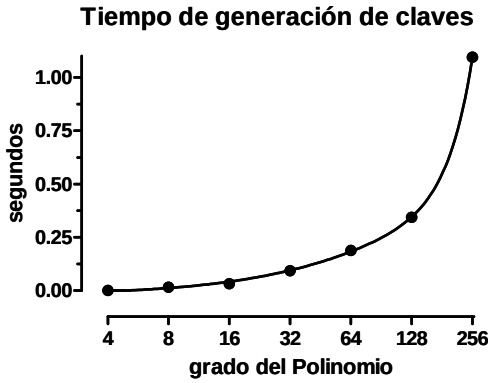

Fig. 1. Tiempos pesimistas de la generación de claves. En este gráfico se observa la curva de crecimiento temporal de la generación de claves según el protocolo presentado. Los valores corresponden a código no compilado y no optimizado en *Mathematica 7.0* corriendo en un equipo WinXP CPU Intel CoreDuo 1.6 GHz y 2 Gb RAM.

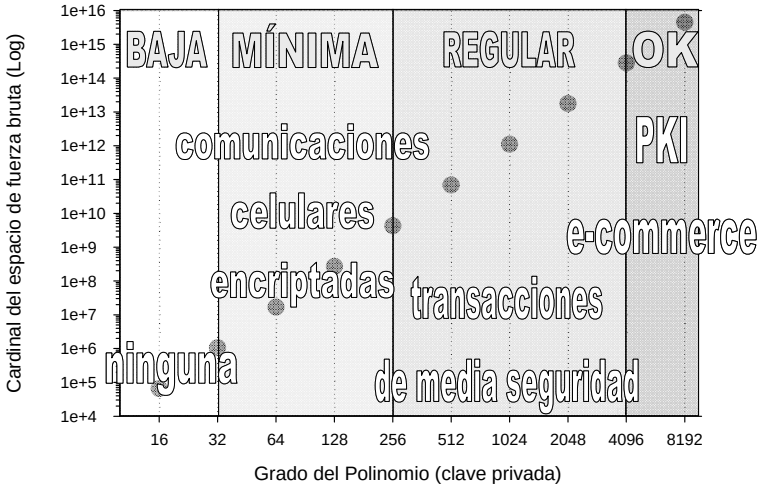

Fig. 2. Bandas potenciales de seguridad y aplicabilidad práctica. Aquí se observan hipotéticas bandas de seguridad y aplicación práctica, definidas por la relación entre el grado del polinomio usado como clave privada y el cardinal de la exploración sistemática. Si los cómputos del ataque procediesen a razón de 10^6 tanteos por segundo, la banda baja se explora en menos de 1 segundo, la banda mínima en menos de 3 horas, la banda regular en menos de tres años y la banda segura en el orden de 3 siglos.

8 CONCLUSIONES

En la última década se han desarrollado algoritmos y protocolos criptográficos basados en *estructuras algebraicas no conmutativas* (NCAS). Lo atractivo de estas soluciones es que no están basadas en la dificultad de la resolución de ciertos problemas que operan en *estructuras conmutativas* como la factorización de enteros (IFT) o el logaritmo discreto (DL) y para los cuales ya existen ataques de complejidad subexponencial en el tiempo. Por otra parte, la eventual aparición de computadoras cuánticas amenaza con destruir los criptosistemas tradicionales, lo que en estos sistemas NCAS no parece ser viable. Por ese motivo se han apelado a las más diversas estructuras como grupos de trenzas, grupos de Thompson, grupos policíclicos, grupos nilpotentes, grupos simétricos, anillos matriciales, etc. La lista seguramente no está agotada y se verán aparecer nuevos y seguramente promisorios enfoques. Lo interesante del caso es que una vez descubierto algún protocolo de criptografía asimétrica (AC) o de clave pública (PKC) casi automáticamente se resuelven temas vinculados a la confidencialidad, autenticación, pruebas de conocimiento cero e intercambio de claves.

Hasta ahora no hay antecedentes bibliográficos acerca de sistemas NCAS compactos. En este trabajo se presenta por primera vez un protocolo de esa clase diseñado específicamente para ser operado en plataformas computacionalmente pobres, es decir que no requiera bibliotecas de precisión extendida y que puedan resolverse cómodamente usando aritmética entera de 16 bits, todo ello sin resignar seguridad criptográfica. Obviamente en ambientes de mayor envergadura su operación puede llegar a competir favorablemente con otros protocolos de seguridad equivalente.

Sintetizando, entendemos que el modelo compacto aquí presentado puede brindar una alternativa útil tanto por su seguridad criptográfica como por su bajo costo computacional que lo hace apto para implementaciones muy reducidas.

8 REFERENCIAS

1. Menezes A.J., van Oorschot P.C., Vanstone S.A.: Handbook of Applied Cryptography. CRC Press (1997)
2. Rivest R.L. Shamir A., Adleman L.: A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM., 21:1, 120-126 (1978)
3. Koblitz N.: Algebraic aspects of cryptography, Springer (1998)
4. Williams H.C.: A modification of the RSA public key encryption procedure, IEEE Transactions on Information Theory, 26:6, 726-729 (1980)
5. Smith P. Lennon M.: LUC - a new public key system, Proceedings of the IFIP- IX International conference on information security, North-Holland, 93, 103-117 (1993)
6. Koyama K., Maurer U.,Okamoto T.,Vanston S.: New public key schemes based on elliptic curves over the ring Zn, Crypto'91, (ed) J. Feigenbaum, Springer (1992)
7. ElGamal T.: A public key cryptosystem and a signature scheme based on discrete logarithms , IEEE Transactions on information theory, 31, 469-472 (1985)
8. Diffie W. Hellman M.E.: New directions in cryptography, IEEE Transactions on information theory, 22, 644-654 (1976)

9. McCurley K.: A key distribution system equivalent to factoring, Journal of Cryptology, 1, 95-100 (1988)
10. 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)
11. Nielsen M.A., Chuang I.L.: Quantum computation and quantum information. Cambridge University Press (2000)
12. Shor P.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput., 5, 1484-1509 (1997)
13. Kitaev A.: Quantum measurements and the abelian stabilizer problem, Preprint arXiv/quant-ph., 9511026 (1995)
14. Proos J., Zalka C.: Shor's discrete logarithm quantum algorithm for elliptic curves, Quantum information and computation, 3, 317-344 (2003)
15. Lee E.: Braid groups in cryptography, IEICE Trans. Fundamentals, E87-A:5, 986-992 (2004)
16. Wagner N.R., Magyarik M.R.: A public-key cryptosystem based on the word problem, Crypto'84 (ed) Blakley G.R. Chaum D. Springer Verlag (1985)
17. Birget J., Magliveras S.S., Sramka M.: On public-key cryptosystems based on combinatorial group theory, Cryptology ePrint archive report 2005/070 (2005)
18. Anshel I., Anshel M., Goldfeld D.: An algebraic method for public-key cryptography, Math. Research Letters, 6, 287-291 (1999)
19. Ko K.H., Lee S.J., Cheon J.H., Han J.W.: New public-key cryptosystem using braid groups, Crypto 2000. 166-183, Springer Verlag (2000)
20. Thomas T., Lal A.K.: Group signatures schemes using braid groups, Preprint arXiv/cs., 0602063 (2006)
21. Dehornoy P.: Braid based cryptography, Contemporary mathematics, 360, 5-33 (2004)
22. Bohli J.M., Glas B., Steinwandt R.: Towards provable secure group key agreement building on group theory, Cryptology ePrint archive, report 2006/79 (2006)
23. Paeng S.H., Ha K.C., Kim J.H., Chee S., Park C.: New public-key cryptosystem using finite non-abelian groups, Crypto 2001, (ed). J. Kilian, Springer Verlag, 470-485 (2001)
24. Paeng S.H., Kwon D., Ha K. C., Kim J.H.: Improved public-key cryptosystem using finite non-abelian groups, Cryptology ePrint archive, Report 2001/066 (2001)
25. González Vasco M. I., Martinez C., Steinwandt R.: Towards a uniform description of several group based cryptographic primitives, Designs, Codes and Cryptography, 33, 215-226 (2004)
26. Grigoriev D., Ponomarenko I.: On non-abelian homomorphic public-key cryptosystems, Preprint arXiv/cs, 0207079 (2002)
27. Grigoriev D., Ponomarenko I.: Homomorphic public-key cruptosystems over groups and rings, Preprint arXiv/cs, 0309010 (2003)
28. Eick B., Kahrobaei D.: Polycyclic groups: a new platform for cryptography, Preprint arXiv/math.gr, 0411077 (2004)
29. Shpilrain V., Ushakov A.: A new key exchange protocol based on the decomposition problem , Preprint arXiv/math.gr,Vol. 0512140 (2005)
30. Shpilrain V., Ushakov A.: Thompson's group and public-key cryptography, Preprint arXiv/math.gr, 0505487 (2005)
31. Cao Z., Xiaolei D., Wang L.: New public-key cryptosystems using polynomials over noncommutative rings, Preprint arXiv/cr, eprint.iacr.org/2007/009.pdf (2007)
32. Mahalanobis A.: The Diffie-Hellman key exchange protocol and non-abelian groups, Preprint arXiv/math.gr, 0602282v3 (2007)
33. Thomas T., Lal A.K.: A Zero-Knowledge undeniable signature scheme in non-abelian group setting, International Journal of Network Security, 6:3, 265-269 (2008)

34. Anjaneyulu G.S., Reddy P.V., Reddy U.M.: Secured digital signature scheme using polynomials over non-conmutative division semirings, IJCSNS International Journal of computer science and network security, 8:8, 278-284 (2008)
35. Mahlburg K.: An overview of braid group cryptography, www.math.wisc.edu/~boston/mahlburg.pdf (2004)
36. Shpilrain V., Zapata G.: Combinatorial group theory and public-key cryptography, Preprint arXiv/math.gr, 0410068 (2004)