Algoritmos de Reducción de Base en Teoría de Números y Criptoanálisis

Ismael Jiménez Calvo
Instituto de Ciencias de la Construcción Eduardo Torroja, CSIC, c/ Serrano Galvache s/n, 28033-Madrid.

***Resumen*—Los algoritmos para la reducción de la base de un retículo se han convertido en una herramienta imprescindible en el criptoanálisis de muchos sistemas de cifrado y en problemas de computación en Teoría de Números. En este trabajo se describe sucintamente el problema de la reducción de base y los problemas relacionados como la búsqueda del vector más corto de un retículo (SVP, *Shortest Vector Problem*) o la del vector del retículo más cercano a uno dado (CVP, *Closest Vector Problem*). Se describen los algoritmos de reducción más eficaces conocidos, se esbozan algunas aplicaciones prácticas de los algoritmos y se muestran algunos resultados obtenidos sobre la conjetura ABC y sobre la reducción de bases de retículos de dificultad probada.**

I. Introducción

La Geometría de los Números es un campo de las matemáticas introducido por Minkowski en la segunda mitad del s. XIX que auna aspectos del Álgebra y de la Teoría de Números, en especial, de la Aproximación Diofántica. El objeto de estudio de esta disciplina son las propiedades de los retículos. Una base de *n* vectores linealmente independientes define un retículo de *n* dimensiones como el conjunto de todas las combinaciones lineales enteras de los vectores de la base. Como quiera que un mismo retículo puede ser generado por un número infinito de bases, un problema fundamental es la de obtener bases con vectores cortos, esto es, el problema de la *Reducción de Base*. Este problema fue, en un principio, planteado como solución para la reducción de formas cuadráticas e identificado como "difícil" ya que requería la enumeración de las combinaciones lineales de los vectores de la base para identificar aquellos de más baja norma. De hecho Emde Boas [2] demostró en 1981 que el problema de encontrar el vector más corto (SVP, *Shortest Vector Problem*), es NP-duro y por tanto no se espera que pueda existir un algoritmo que lo resuelva en tiempo polinómico. Por esta misma causa, se considera que no sería atacable por un ordenador cuántico. Dos textos clásicos que se pueden consultar sobre esta disciplina son [4], [9]. Una exposición del problema de la reducción de base, de muy buen nivel y orientada a los intereses de los que trata este artículo puede encontrarse en la introducción de la tesis de N. Gama [7].

Es famosa la demostración de Minkowski de su teorema sobre el volumen mínimo del cuerpo convexo que encierra al menos un vector no nulo del retículo $\mathcal{L}$ generado por la base $B = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n\}$. Como resultado del teorema se tiene que la norma euclídea del vector más corto es $\lambda(\mathcal{L}) \leq \sqrt{n}\ det(B)^{1/n}$. Si bien el problema de encontrar el vector más corto de un retículo es NP-duro, el celebrado algoritmo LLL [13] de A. Lenstra, H. Lenstra y L. Lovasz, publicado en 1982, es capaz de encontrar en tiempo polinómico un vector del retículo con norma euclídea menor que $2^{(n-1)/4}det(B)^{1/n}$, donde n es la dimensión del retículo. Sobre la base del algoritmo LLL se han desarrollado otros, siendo los más conocidos los ideados por C. P. Schnorr [16]. Estos se describen en la Sección siguiente mostrándose algunas de sus aplicaciones en Teoría de Números y criptoanálisis. Igualmente se muestran algunos resultados computacionales [17] obtenidos sobre la conjetura ABC [14] y algunas modificaciones del algoritmo de C. P. Schnorr y M. Euchner [16] en la que se emplean códigos reflejados en la fase de enumeración de combinaciones lineales del bloque de Korkine-Zolotarev.

La eficacia de los algoritmos de reducción, que en la practica va más allá de lo que teóricamente se puede esperar, puede resolver el SVP en poco tiempo en casos específicos poniendo en duda la eficacia de utilizar este problema en el diseño de algoritmos de cifrado. El problema se puede expresar de la siguiente manera. Encontrar un vector corto en un retículo y demostrar que no existe otro más corto dentro de un determinado rango es un problema difícil computacionalmente pese a lo cual un algoritmo de reducción lo puede encontrar si el tipo del retículo es aleatorio. Sin embargo, A. Ajtai [1] mostró un método para generar bases difíciles de reducir, lo que se ha dado en llamar *"hard instances of lattice problems"*. Esta dificultad se asienta en la prueba de que un algoritmo que en tiempo polinómico encuentre el vector más corto de este tipo difícil de retículos, lo encontrará en cualquier otro retículo de dimensión ligeramente menor.

O. Goldreich, S. Goldwasser y S. Halevi [8] propusieron un sistema de cifrado de clave pública basado en la dificultad de encontrar el vector del retículo más cercano a uno dado, (CVP, *Closest Vector Problem*), que incluye un esquema de firma digital. Sin embargo, Phong Nguyen logró romper un caso de cifrado presentado por los creadores del algoritmo como reto público para poner a prueba su fortaleza [15]. Por otra parte, se ha propuesto un sistema de cifrado asimétrico, el NTRU [10], cuya fortaleza descansa en la dificultad de la reducción de una base del tipo que se asemeja a las propuestas por A. Ajtai [1]. Ejemplos concretos de bases de este tipo de dimensión superior a 500 se han sometido como reto público para su reducción [18] sin que, ni siquiera para las dimensiones más bajas se haya encontrado ningún vector con norma inferior a la cota por debajo de la cual se ha demostrado que existen tales vectores. Estos trabajos computacionales pueden asegurar la fortaleza de los sistemas de cifrado basados en estos problemas

así como los parámetros de las bases que se pueden considerar seguras frente a ataques criptoanalíticos.

Los algoritmos de reducción de base han encontrado muchas aplicaciones como la decodificación de señales MIMO en sistemas de transmisión con múltiples antenas en las etapas de emisión y recepción, pero en este trabajo solo se exponen algunas aplicaciones dentro de la Teoría de Números y el criptoanálisis.

II. Retículos y algoritmos de reducción de base

Sea una base formada por n vectores linealmente independientes del espacio $\mathbb{R}^m$, $B = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n\}$. Se define el retículo generado por la base B como $\mathcal{L}(B) = \{v = a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \cdots, a_n\mathbf{b}_n;\ a_1, a_2, \cdots, a_n \in \mathbb{Z}\}$. En concreto, el retículo canónico sería $\mathbb{Z}^n$. Si la base B es de rango completo, esto es, si $n = m$, tenemos que el hipervolumen que abarca la celdilla unidad del retículo es $det(B)$. En caso contrario, este volumen vendría dado por $\sqrt{det(B \cdot B^T)}$. Consideramos una matriz unimodular entera U, esto es, una matriz en la que todos los items son enteros y en la que $|det(U)| = 1$. Podemos observar que otra base $B' = U \cdot B$ genera el mismo retículo porque $B = U^{-1} \cdot B'$, donde la matriz inversa de U también sería entera y unimodular. Por otra parte, al ser U unimodular, el determinante de cualquier base del retículo no varía, esto es, es un invariante del retículo que representamos por $|\mathcal{L}|$. En las aplicaciones que aquí se tratan, las bases de los retículos está formada por vectores en $\mathbb{Z}^n$.

Se plantea el problema de encontrar la base que esté formada por los *mínimos sucesivos del retículo* cuyas normas son $\lambda_1, \lambda_2, \cdots, \lambda_n$, donde λ_1 es la norma de un vector del retículo tal que no existe otro con norma (generalmente se toma la norma euclídea) menor que él. Los siguientes mínimos sucesivos serán linealmente independientes con los anteriores y de norma lo más baja posible. Encontrar esta base supone conocer la norma de todos los vectores del retículo, problema que tiene complejidad exponencial en n, la dimensión del retículo. Sólo para $n = 2$ se conoce un algoritmo ejecutable en tiempo polinómico, el algoritmo de Gauss. Sea una base $B = \{\mathbf{b}_1, \mathbf{b}_2\}$ en la que $|\mathbf{b}_1| \leq |\mathbf{b}_2|$. Podemos tomar una nueva base $\{\mathbf{b}_1, \mathbf{b}_2 - \lfloor\mu\rceil\, \mathbf{b}_1$ donde $\mu = < \mathbf{b}_2, \mathbf{b}_1 > / < \mathbf{b}_1, \mathbf{b}_1 >$, donde $< \cdot, \cdot >$ denota el producto escalar de dos vectores. Podemos ver que la matriz correspondiente a esta transformación es unimodular, por lo que la nueva base sigue siendo la base del retículo. De esta manera reducimos al máximo la proyección de $\mathbf{b}'_2 = \mathbf{b}_2 - \lfloor\mu\rceil\, \mathbf{b}_1$ sobre $\mathbf{b}_1$ y obtenemos una nueva base más reducida y más cercana a la ortogonalidad, conceptos que resultan ser equivalentes en este caso. Si $|\mathbf{b}'_2| > (1 + \epsilon)|\mathbf{b}_1|$ se termina el algoritmo o se intercambian ambos vectores y se repite el paso anterior. La constante ϵ es necesaria para evitar que el algoritmo caiga en un ciclo infinito. También se puede apreciar la similitud de este algoritmo con el de Euclides para el cálculo de *Máximo Común Divisor* cambiando enteros por vectores.

II-A. El algoritmo LLL

Sobre la base, patente en el algoritmo de reducción de Gauss, de la equivalencia entre reducción de base y *ortogonalización*, A. K. Lenstra, H. W. Lenstra, and L. Lovász [13] publicaron un algoritmo de reducción que en tiempo polinómico obtiene una base reducida con las siguientes características:

$$|\mathbf{b}_1| \leq 2^{(n-1)/4}|\mathcal{L}|^{1/n}$$

$$|\mathbf{b}_1| \leq 2^{(n-1)/2}\lambda_1(\mathcal{L})$$

$$|\mathbf{b}_1| \cdot |\mathbf{b}_2| \cdots |\mathbf{b}_n| \leq 2^{\binom{n}{2}/2}|\mathcal{L}|.$$

El eje del algoritmo gira sobre el *proceso de ortogonalización de Gram-Schmidt* que hace de guía para la reducción y que se describe como sigue. Dada una base $B = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n\}$ se obtiene otra base ortogonal $\{\mathbf{b}_1{}^*, \mathbf{b}_2{}^*, \cdots, \mathbf{b}_n{}^*\}$ con el algoritmo recursivo siguiente.

$$\mathbf{b}_1{}^* = \mathbf{b}_1,$$

$$\mathbf{b}_i{}^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j{}^*, \quad \mu_{i,j} = \frac{< \mathbf{b}_i, \mathbf{b}_j{}^* >}{< \mathbf{b}_j{}^*, \mathbf{b}_j{}^* >}, \quad i = 2, \cdots, n.$$

Si formamos una matriz con los coeficientes $\mu_{i,j}$ y llamamos B^* a la base de Gram-Schmidt, vemos que

$$B = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \mu_{2,1} & 1 & 0 & \cdots & \cdots & 0 \\ \mu_{3,1} & \mu_{3,2} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \ddots & \ddots & 0 \\ \vdots & \vdots & \cdots & \ddots & \ddots & 0 \\ \mu_{n,1} & \mu_{n,2} & \cdots & & \mu_{n,n-1} & 1 \end{pmatrix} B^*$$

Hay que hacer notar que la matriz B^* depende del orden en que se tomen los vectores. Si obviamos cuestiones de cálculo y análisis de complejidad, el esquema del algoritmo LLL resulta sencillo. Se parte de la matriz ortogonal B^* y la matriz de coeficientes $\{\mu_{i,j}\}$. Empezamos reduciendo el vector $\mathbf{b}_2$ de forma que pasa a ser $\mathbf{b}_2 \leftarrow \mathbf{b}_2 - \lfloor\mu_{2,1}\rceil\, \mathbf{b}_1$. En la etapa i, hacemos $\mathbf{b}_i \leftarrow \mathbf{b}_i - \sum_{j=1}^{i-1} \lfloor\mu_{ij}\rceil\, \mathbf{b}_j$. Esta etapa, denominada *reducción en tamaño*, si bien no encuentra el vector cuya proyección sobre $\mathbf{b}_i{}^*$ sea la mínima posible, al menos reduce los coeficiente $\mu_{i,j}$ de esa fila a un valor menor o igual a $1/2$ una vez recalculados. Posteriormente se comparan las proyecciones de $\mathbf{b}_i$ y $\mathbf{b}_{i-1}$ sobre el espacio ortogonal al generado por los vectores $(\mathbf{b}_1, \cdots, \mathbf{b}_{i-2})$. Si esta proyección de $\mathbf{b}_i$ es menor dentro de un factor ε necesario para evitar inestabilidades como en el caso de la Reducción de Gauss, se intercambian ambos vectores y se continúa el algoritmo en la fila anterior, pasando a la posterior en caso contrario. Vemos que este algoritmo debido a Lovász obtiene una base reducida en tiempo polinómico con las características anteriormente reseñadas.

Si los vectores de la base a reducir tiene sus coeficientes enteros, tanto la base de Gram-Schmidt como los coeficientes $\mu_{i,j}$ resultan ser racionales y es posible ejecutar el algoritmo en aritmética exacta. Si se utiliza representaciones de tipo *float*

se acelera notablemente la ejecución del algoritmo LLL y de las variantes que se describen más adelante. Sin embargo, es a costa de que aparezcan problemas de "estabilidad" en su ejecución. En primer lugar, en el cálculo de la matriz de Gram-Schmidt, al ser recursiva, se acarrean errores que conducen a una merma en la calidad de la reducción y a que el algoritmo pueda caer en ciclos. Para evitarlo se puede emplear un coeficiente ε más bajo en las comparaciones de los tamaños de los vectores. También se ha propuesto emplear las rotaciones de Givens o la reflexiones de Houeseholder en el cálculo de la matriz ortogonal porque no tienen los problemas de propagación de errores que tiene el método de Gram-Schmidt o también ligeras modificaciones del método de Gram-Schmidt que le dan más estabilidad. Si los algoritmos se programan utilizando bibliotecas multiprecisión, se puede aumentar la precisión cuando los problemas de estabilidad aparecen, tal como propuso Schnorr. El autor de este trabajo ha comprobado que resulta eficaz programar métodos para salir de los ciclos más sencillos una vez que se detecta que el programa ha caído en ellos.

La eficacia del algoritmo LLL reside en las etapas de intercambio de los vectores puesto que la etapa de *reducción en tamaño* es insuficiente. C.-P. Schnorr and M. Euchner propusieron una modificación del algoritmo llamada de *inserciones profundas (deep insertions)* que, a de costa emplear un tiempo exponencial en su ejecución, consigue una mayor calidad en la reducción. El algoritmo LLL solo considera un intercambio de filas contiguas. Sin embargo, parece natural poder permitir un intercambio con una fila más profunda, digamos de los vectores $\mathbf{b}_i$ y $\mathbf{b}_j$, $i > j$, comparando sus proyecciones sobre el espacio ortogonal al generado por los vectores $(\mathbf{b}_1, \cdots, \mathbf{b}_{j-1})$. Para limitar el tiempo de ejecución, el algoritmo de *inserciones profundas* limita el intercambio de vectores a índices en los rangos $0 \cdots \delta$ y $i - \delta \cdots i$. En la práctica, es suficiente ajustar el coeficiente de comparación ε a un valor suficientemente bajo para que no se dispare el tiempo de ejecución del algoritmo, consiguiéndose resultados similares.

III. La reducción de Korkine-Zolotarev y el algoritmo de enumeración de Kannan

La base formada por los *mínimos sucesivos* $\boldsymbol{\lambda}_i$ definidos por Minkowski es la más reducida. Encontrarla exige la enumeración de los vectores del retículo y sobre este concepto no se puede construir un algoritmo de reducción eficaz. Hermite, Korkine y Zolotarev definieron otro tipo de base reducida más fácil de trasladar a la concepción de algoritmos. El primer mínimo coincide con el de Minkowski. Para el vector mínimo de índice i se toman los vectores de la base linealmente independientes con los vectores $\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_{i-1}$ y se busca aquella combinación lineal tal que su proyección sobre el espacio ortogonal al expandido por los vectores $\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_{i-1}$ sea mínimo. Como la enumeración de todas las combinaciones lineales posibles es impracticable, C.-P. Schnorr and M. Euchner [16] limitaron las combinaciones a un bloque de vectores. Este algoritmo se considera actualmente como el más eficaz. Se ejecuta sobre la base del algoritmo LLL. Hay que observar que el algoritmo LLL aplicado sobre un conjunto de vectores en el que no todos son linealmente independientes, encontrará los vectores nulos que, eliminados, darán lugar a una base de vectores linealmente independiente. Suponemos que estamos en la etapa i del algoritmo. En ese momento enumeramos todas las combinaciones lineales del bloque de vectores $\mathbf{b}_i, \cdots, \mathbf{b}_{i+\delta}$ y se agrega aquella combinación de vectores que tenga una proyección sobre el espacio ortogonal al espacio expandido por los vectores $(\mathbf{b}_1, \cdots, \mathbf{b}_{i-1})$ más corta. La subsiguiente etapa de reducción LLL eliminará aquel vector que sea linealmente dependiente de los demás.

El problema de encontrar los límites en la enumeración de las combinaciones lineales dentro del bloque lo resolvió R. Kannan [12] de la siguiente manera. Suponemos que tenemos un bloque de vectores $(\mathbf{a}_1, \cdots, \mathbf{a}_n)$. Calculamos la matriz de Gram-Schmidt del bloque $(\mathbf{a}_1^* = \mathbf{a}_1, \cdots, \mathbf{a}_n^*)$. El vector $\mathbf{a}_n^*$ es ortogonal a $\mathbf{a}_1$ y a todos los demás excepto $\mathbf{a}_n$. Si tomamos k veces el vector $\mathbf{a}_n$, se debe cumplir que $|k\mathbf{a}_n^*| \leq |\mathbf{a}_1|$ pues, de otra forma el vector resultante de la combinación lineal sería más largo que $\mathbf{a}_1$. Así,

$$k = \left\lfloor \frac{|\mathbf{a}_1|}{|\mathbf{a}_n^*|} \right\rceil .$$

De esta manera, en la enumeración de las combinaciones de vectores, los coeficientes del vector $\mathbf{a}_n$ solo deben variar entre $-k$ y $+k$. Se puede observar que si el bloque es de sólo dos vectores la reducción de Korkine-Zolotarev en bloque es equivalente a la reducción LLL.

El autor de este trabajo propone la idea, aún no experimentada, de acelerar la enumeración de los vectores empleando una secuencia de ellos dada por un *código reflejado*. Al igual que en los códigos de Gray sólo cambia un dígito binario al pasar de un término a otro consecutivo, se pueden construir códigos reflejados no binarios en los que cada paso de la enumeración de vectores solo implica una adición o sustracción de un vector con la consiguiente aceleración del algoritmo. En este aspecto, se está trabajando en un algoritmo de reducción y se está aplicando sobre bases de retículos "difíciles", inspiradas en el trabajo de M. Ajtai [1] que ya se mencionó en la introducción. Por ahora, y en el momento de redactar este trabajo, el autor ocupa el nº 13 en este reto público [18] ya que Phong Nguyen y Yuanmi Chen copan los doce primeros puestos.

IV. Aplicaciones de los Algoritmos de Reducción de Base

IV-A. Aproximación diofántica, relaciones enteras y problemas de la "mochila"

Dirichlet planteó el problema de la aproximación diofántica simultanea. Dado un conjunto de constantes reales, $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ queremos encontrar sus mejores aproximaciones racionales con un mismo denominador $\{p_1/q, p_2/q, \cdots, p_n/q\}$. Para ello utilizamos el retículo

definido por los vectores que forman las filas de la matriz

$$\begin{array}{c} p_1 \\ p_2 \\ \vdots \\ \vdots \\ p_n \\ q \end{array} \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ -\alpha_1 & -\alpha_2 & -\cdots & -\alpha_n & \epsilon/Q \end{pmatrix}, \quad \epsilon > 0.$$

Si $(p_1, p_2, \cdots, p_n, q)$ son los coeficientes de la combinación lineal entera correspondiente a un vector corto del retículo respecto a la norma L_∞, tenemos que $|p_i - q\alpha_i|, \quad i \in \{1, \cdots, n\}$ es pequeño. La calidad de la aproximación simultánea óptima viene dada por la cota de Minkowski para el vector más corto, pero una buena aproximación a ella se puede obtener en tiempo polinómico aplicando el algorimo LLL.

Queremos ahora encontrar una *relación entera*, es decir, una combinación lineal entera de varias constantes reales $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$, tal que su suma sea pequeña. Podemos reducir el retículo definido por los vectores que forman las filas de la matriz

$$\begin{array}{c} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{array} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & N\alpha_1 \\ 0 & 1 & 0 & \cdots & 0 & N\alpha_2 \\ 0 & 0 & 1 & \cdots & 0 & N\alpha_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & N\alpha_n \end{pmatrix}, \qquad (1)$$

donde N es una constante elevada, y buscar un vector corto. Sean los coeficientes de esta relación $\{c_1, c_2, c_3, \cdots, c_n\}$, así que tal vector tiene la forma $(c_1, c_2, c_3, \cdots, c_n, N\sum_{i=1}^{n} c_i\alpha_i)$, que nos da la relación buscada. Supongamos que queremos conocer si una constante α es algebraica. Para ello basta encontrar una relación entera de las potencias de la constante $1, \alpha, \alpha^2, \cdots, \alpha^n$ que se anule. Si para cierto n esta relación entera nula no se encuentra, podemos afirmar que no es raíz de un polinomio de grado igual o menor que n.

No es de extrañar que las primeras víctimas criptoanalíticas de los algoritmos de reducción de base fuesen los sistema de cifrado de clave pública basado en el problema de la "mochila" (*knapsacks*), como el de Merkle y Hellman. En estos sistemas de encriptado, la palabra cifrada C es una relación entera (binaria) de los enteros que forman parte de la clave pública $\{w_1, w_2, \cdots, w_n\}$. Sustituyendo la última columna en la matriz (1), por $(w_1, w_2, \cdots, w_n, C)$ obtendremos la secuencia de bits que forman el mensaje original.

IV-B. Conjetura ABC

La conjetura ABC de Oesterlé-Masser [14] establece a grandes rasgos que, en una relación de enteros coprimos $A + B = C$, el producto de los primos distintos que dividen ABC, que llamamos radical R, no puede ser mucho menor que el producto ABC. Si definimos la "calidad" de una terna A, B, C como $q = log(C)/log(R)$, podemos expresar exactamente la conjetura de la forma siguiente: *Para cualquier $\varepsilon > 0$, sólo existe un número finito de ternas en las que $q > 1 + \varepsilon$*. Esta conjetura, muy relacionada con el *Último Teorema de Fermat*, se considera muy difícil de demostrar y necesitada de evidencias computacionales para las cuales se han puesto en marcha campañas de computación cooperativa y retos publicados en la red. Una forma de computar ejemplos ayudándose de los algoritmos de reducción de base es la búsqueda de dos números a y b coprimos, muy próximos entre sí, que factoricen en potencias altas de primos. Esto es, si $a/b \approx 1$, $a - b = e$, siendo e un entero relativamente pequeño. Estos números *a, b y e* constituyen una terna A, B, C que puede cumplir la condición $q > 1$. Consideramos una base formada por los n primeros primos $\mathcal{P} = \{2, 3, 5, \cdots, p_n\}$ y sean los números

$$a = \prod_{p_i \in \mathcal{P}} p_i^{c_i}, \quad b = \prod_{p_j \in \mathcal{P}} p_j^{c_j}, \quad p_i \neq p_j, \quad c_i, c_j \in \mathbb{Z}^+ + \{0\}.$$

Si $a/b = 1 + \epsilon$, tomando logaritmos tenemos que

$$\sum c_i log(p_i) - \sum c_j log(p_j) = e,$$

e pequeño en valor absoluto. Para encontrar candidatos apropiados para la conjetura ABC se pueden buscar relaciones enteras de los números reales $\alpha_i = log(p_i)$ mediante la reducción de bases del tipo (1) con distintas constantes N. De esta forma, el autor posee el record con la siguiente terna ABC con $q > 1$ y de mayor tamaño y "mérito", hasta el momento de escribir este trabajo [17].

$$A = 3^3 \cdot 31^3 \cdot c;$$

$$B = 5^{362} \cdot 7^{109} \cdot 11^7 \cdot 17^{326} \cdot 37^{11} \cdot 53^{33} \cdot 59^{179} \cdot 67^{137} \cdot 79^{76} \cdot 103^{348} \cdot 109^{12} \cdot 113^{103} \cdot 131^{42} \cdot 151^{12} \cdot 163^{166};$$

$$C = 2^{465} \cdot 13^{76} \cdot 19^{57} \cdot 23^{611} \cdot 29^{19} \cdot 41^{11} \cdot 43^{98} \cdot 61^{84} \cdot 71^{13} \cdot 73^{250} \cdot 83^{30} \cdot 89^{10} \cdot 97^{80} \cdot 101^{45} \cdot 127^7 \cdot 137^8 \cdot 139^3 \cdot 167^{253} \cdot 173^{25};$$

donde c es un entero sin factores cuadráticos.

IV-C. Raices de polinomios módulo un entero de factorización desconocida

En el campo de los polinomios, la publicación del algoritmo LLL se presentó como herramienta para componer el primer algoritmo que en tiempo polinómico era capaz de factorizar polinomios con coeficientes racionales. Técnicas parecidas, que se explican a continuación, permiten encontrar las raíces de un polinomio modular cuando no es conocida la factorización del módulo. Estas técnicas fueron empleadas por Coppersmith [5] para un ataque al RSA en casos en que los bits altos de la palabra cifrada son conocidos. Este es el caso en que la palabra a cifrar es $M + x$ donde M es conocido y x pequeño y desconocido. Si e es el exponente de cifrado y N el módulo de factorización desconocida que forman la parte pública de la clave RSA, la recuperación de x equivale a encontrar una raiz pequeña del polinomio $(M + x)^e - C \equiv 0 \pmod{N}$, siendo C la palabra cifrada. Veremos que los algoritmos de reducción de base pueden abordar estos problemas de criptoanálisis y otros parecidos.

Queremos calcular una raíz pequeña x_0 de $f(x) \pmod N$ cuando la factorización de N es desconocida. Observamos que x_0 también es raíz de los polinomios

$$f_{i,j}(x) = N^{d-j} x^i f(x)^j \pmod{N^d}, \quad j = 1 \cdots d.$$

y cualquier combinación lineal de ellos, $F(x)$, tendrá como raíz $x_0 \pmod{N^d}$. Se consideran los vectores cuyas coordenadas son los monomios del polinomio dotados de peso X^i, donde X es una constante elevada.

Una vez hecha la reducción LLL obtenemos un vector de baja norma. Dividiendo cada término por X^i recuperamos los coeficientes del polinomio $F(x)$, también de baja norma, que tiene como raíz x_0. Si la suma de los valores absolutos de los monomios de $F(X)$ es menor que N^d, esto nos garantiza que, resuelto el polinomio sobre los enteros, obtenemos la raíz buscada x_0.

Un ejemplo simplificado es el siguiente. Buscamos una raíz $x_0 < X$ del polinomio $f(x) = x^3 + ax^2 + bx + c \pmod N$. Se consideran los polinomios como vectores, cuyas entradas son los monomios dotados de peso X^i. Tomamos los polinomios

$$f_i(x) = x^i f(x) \pmod N, \quad i = 0, 1, 2, 3.$$

Aplicamos el algoritmo LLL sobre la base formada por las filas de la siguiente matriz en la que cada columna se ha multiplicado por X^i de tal forma que cada término representa el valor de cada monomio en cada uno de los polinomios, tomando por valor de la variable al entero X.

$$\begin{array}{c} X^6\ X^5\ X^4\ X^3\ X^2\ X\ 1 \\ \left(\begin{array}{ccccccc} 1 & a & b & c & 0 & 0 & 0 \\ 0 & 1 & a & b & c & 0 & 0 \\ 0 & 0 & 1 & a & b & c & 0 \\ 0 & 0 & 0 & 1 & a & b & c \\ 0 & 0 & 0 & 0 & N & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & N & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & N \end{array} \right), \end{array}$$

Después de obtener una combinación lineal de baja norma, se divide cada término por el valor correspondiente de X^i, recuperándose los coeficientes del polinomio. Teniendo en cuenta que $|\mathcal{L}| = N^3 X^{21}$, aplicando la cota para el vector más corto encontrado por el algoritmo LLL en la Sección *II- A*, vemos que éste debe ser de norma inferior a $kN^{3/7}X^3$. El valor del polinomio para la raíz buscada no debe sobrepasar N para que pueda ser resuelto en los enteros. En consecuencia, una raíz $x_0 < X$ de $O(N^{4/21})$, podrá ser calculada sin conocer la factorización de N.

Así, el ataque ideado por Coppersmith es capaz de recuperar en tiempo polinómico los $1/e$ bits de menor peso del mensaje, conociendo el resto de los bits. Otro ataque de este tipo que, no utiliza algoritmos de reducción de base, se puede encontrar en [11]. Otro ataque, a casos específicos del RSA, que utiliza estas técnicas se puede encontrar en [3].

Igualmente, los algoritmos de reducción de base se han aplicado a la reconstrucción de secuencias pseudoaleatorias obtenidas por generadores lineales modulares del tipo $x_{i+1} \leftarrow ax_i + c \pmod N$ en la que solo se revelan los bits mas altos de la secuencia. A. M. Frieze, J. Hastad, R. Kannan, J. C. Lagarias y A. Shamir [6] demostraron que en tiempo polinómico se podía reconstruir la secuencia, demostrando así que no eran seguras para aplicaciones criptográficas.

REFERENCIAS

[1] M. Ajtai, "Generating hard instances of lattice problems", En Proc. of 28th STOC, pp. 99–108. ACM, 1996.

[2] P. van Emde Boas, "Another NP-complete partition problem and the complexity of computing short vectors in lattices". Report 81-04, Mathematische Institut, University of Amsterdam. 1981.

[3] D. Boneh and G. Durfee. "Cryptanalysis of RSA with private key d less than $N^{0,292}$", Proc. of Eurocrypt '99, volume 1592 of Lecture Notes in Computer Science, pp. 1–11. IACR, Springer, 1999.

[4] J.W.S. Cassels, "An Introduction to the Geometry of Numbers", Spinger-Verlag, Berlin/Heidelberg, 1971.

[5] D. Coppersmith, "Small solutions to polynomial equations, and low exponent RSA vulnerabilities", J. of Cryptology, vol. 10, pp. 233–260, 1997.

[6] A. M. Frieze, J. Hastad, R. Kannan, J. C. Lagarias y A. Shamir, "Reconstructing Truncated Integer Variables Satisfying Linear Congruences", SIAM J. Comput. 17(2), pp. 262-280, 1988

[7] N. Gama, "Geometrie des nombres et cryptanalyse de NTRU", Tesis Doctoral, 2008.

[8] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-key cryptosystems from lattice reduction problems", Proc. of Crypto '97, volume 1294 of LNCS, pages 112–131. IACR, Springer- Verlag, 1997.

[9] P. M. Gruber and C. G. Lekkerkerker, "Geometry of Numbers", Elsevier, 1987.

[10] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, "NTRU: A Ring Based Public Key Cryptosystem", Algorithmic Number Theory (ANTS III), J.P. Buhler (ed.), Lecture Notes in Computer Science 1423, Springer-Verlag, Berlin, 1998, 267-288.

[11] I. Jiménez Calvo y G. Sáez, "Approximate Power roots in Z_m", 4^{th} International Conference ISC'2001 (Information Security), 1-3 Octubre de 2001, Málaga (Spain), LNCS 2200, pp. 310-323.

[12] R. Kannan, "Minkowski's convex body theorem and integer programming", Mathematics of Operations Research", vol. 12, pp. 415–440, 1987.

[13] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients", Math. Ann, vol. 261, pp. 515–534, 1982.

[14] J. Oesterlé, "Nouvelles approaches du 'théorème de Fermat'.", Sém. Bourbaki 2/1988, exposé #695.

[15] Phong Q. Nguyen, "Cryptanalysis of the Goldreich–Goldwasser–Halevi cryptosystem from crypto'97". Proc. of Crypto'99: LNCS vol. 1666, pages 288–304, London, UK, 1999. Springer-Verlag.

[16] C.-P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems", Math. Programming, vol. 66, pp. 181–199, 1994.

[17] ABC triples: `http://www.math.leidenuniv.nl/~desmit/abc/index.php?set=1`

[18] TU Darmstadt Lattice challenge: `http://www.latticechallenge.org/halloffame.php`