15

UN SISTEMA DE CIFRADO SIMÉTRICO Y ALGUNAS CONSIDERACIONES SOBRE LA SEGURIDAD COMPUTACIONAL[1]

José M. Sempere
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Email:jsempere@dsic.upv.es

Resumen: En el presente capítulo proponemos un sistema de cifrado simétrico basado en la Teoría de Lenguajes Formales. El sistema se basa en algunos problemas de equivalencia y ambigüedad sobre gramáticas incontextuales que se han demostrado como indecidibles. En consecuencia, la seguridad del sistema propuesto se fundamenta no en la Teoría de la Complejidad, y más en concreto de la NP-completitud, sino en una rama de la Teoría de la Computabilidad como es la Teoría de la Indecidibilidad. A partir del criptoanálisis del sistema proponemos algunas conclusiones de validez general para otros sistemas de cifrado simétrico y asimétrico.

1. INTRODUCCIÓN

La Teoría de Lenguajes Formales ha realizado distintas aportaciones a la Criptografía en su sentido más amplio. Entre otras aportaciones podemos nombrar desde sistemas de cifrado simétricos y de clave pública hasta sistemas de identificación y autenticación. Ejemplos de estas aportaciones se pueden encontrar en trabajos tales como [2, 10, 11 y 12] y, por citar algunos de ellos. Es bien sabido que la teoría de la complejidad, como una especialización de la teoría de lenguajes formales, y más en concreto la teoría de la NP-completitud es uno de los pilares básicos de la criptografía de clave pública así como de la formulación de protocolos criptográficos como los que se muestran en [5] y en [3].

En el presente trabajo proponemos un sistema de cifrado simétrico basado en algunos aspectos relacionados con la teoría de lenguajes formales y, más en concreto, con algunos problemas formulados sobre gramáticas incontextuales. El análisis de seguridad del sistema propuesto se basa no en la teoría de la complejidad sino en la teoría de la indecidibilidad. Por lo tanto, planteamos un nuevo escenario para el análisis de la seguridad de sistemas de cifrado que, en buena medida, supera al de la seguridad computacional planteada tradicionalmente. La estructura del presente trabajo se fundamenta en una primera sección donde se proporcionan las definiciones básicas procedentes de la teoría de lenguajes que posteriormente se utilizarán. A continuación se plantea un sistema de cifrado simétrico y se realiza su análisis de seguridad frente a diversos ataques con protocolos conocidos. Por último, planteamos una serie de conclusiones acerca de la seguridad computacional que servirán de referente para futuros trabajos.

[1] Trabajo parcialmente financiado por el Ministerio de Ciencia y Tecnología bajo el proyecto TIC2003-09319-C03-02.

2. CONCEPTOS BÁSICOS

Los conceptos básicos sobre la teoría de lenguajes formales que vamos a proporcionar pueden consultarse en [6] y en [9]. En primer lugar, Σ denotará un alfabeto, es decir un conjunto finito y no vacío de elementos denominados *símbolos*. Una cadena x definida sobre un alfabeto Σ es una secuencia finita y ordenada de símbolos con o sin repetición. Σ^* denota el conjunto infinito de todas las posibles cadenas definidas sobre Σ. De entre todas las cadenas, la *cadena vacía* se define como la que no tiene símbolos y la denotaremos como λ. El producto o concatenación de dos cadenas x e y, denotado como xy, se define como la cadena formada por la secuencia de x seguida de la secuencia de y. Un lenguaje L, definido sobre el alfabeto Σ, es un conjunto (finito o infinito) de cadenas de Σ^* y por lo tanto se cumple que $L \subseteq \Sigma^*$.

Una gramática es una tupla $G = (N,\Sigma,P,S)$ donde, N y Σ son dos alfabetos disjuntos de símbolos auxiliares y terminales respectivamente, P es un conjunto finito de pares de la forma $\alpha\rightarrow\beta$ donde $\alpha\in(\Sigma\cup N)^*N(\Sigma\cup N)^*$ y $\beta\in(\Sigma\cup N)^*$, que denominaremos *producciones*, y $S \in N$ es el axioma de la gramática. Habitualmente, si tenemos el conjunto de producciones $\alpha\rightarrow\beta_1$, $\alpha\rightarrow\beta_2$, ..., $\alpha\rightarrow\beta_n$ escribiremos $\alpha\rightarrow\beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$ con el ánimo de economizar notación. Podemos establecer la *relación de derivación directa* entre cadenas formadas por símbolos auxiliares y terminales, que denotaremos por $\underset{G}{\Rightarrow}$, de la siguiente forma: $x \underset{G}{\Rightarrow} y$ sii $x = x_1\alpha x_2$, $y = x_1\beta x_2$ y $\alpha\rightarrow\beta \in P$. Denominaremos *derivación* a la secuencia $\alpha_1 \underset{G}{\Rightarrow} \alpha_2 \underset{G}{\Rightarrow} \ldots \underset{G}{\Rightarrow} \alpha_n$. La clausura reflexiva y transitiva de la relación de derivación directa, que denominaremos *relación de derivación*, la denotaremos por $\overset{*}{\underset{G}{\Rightarrow}}$ o por $\overset{*}{\Rightarrow}$ si la gramática G está definida previamente. El lenguaje de una gramática G lo definiremos como el conjunto $L(G) = \{ x \in \Sigma^* : S \overset{*}{\underset{G}{\Rightarrow}} x \}$.

Una gramática $G = (N, \Sigma, P, S)$ diremos que es *incontextual* si las producciones de P toman la forma $A\rightarrow\alpha$ con $A \in N$ y $\alpha\in (N \cup\Sigma)^*$. Además, diremos que está en *forma normal de Chomsky* si todas y cada una de sus producciones son de la forma $A\rightarrow BC$ o $A\rightarrow a$ con $A,B,C \in N$ y $a \in\Sigma$. De igual forma, diremos que un lenguaje es incontextual si existe una gramática incontextual que lo pueda generar. Dada una gramática incontextual, diremos que la secuencia $\alpha_1 \underset{G}{\Rightarrow} \alpha_2 \underset{G}{\Rightarrow} \ldots \underset{G}{\Rightarrow} \alpha_n$ es una *derivación por la izquierda* si en cada paso de la derivación la producción se aplica sobre el símbolo auxiliar situado más a la izquierda de cada α_i. Podemos asignar a cada producción de la gramática una etiqueta de la forma $\sigma{:}P\rightarrow Lab$. Así, la secuencia $\alpha_1 \underset{G}{\Rightarrow}_{\sigma_1} \alpha_2 \underset{G}{\Rightarrow}_{\sigma_2} \ldots \underset{G}{\Rightarrow}_{\sigma_n} \alpha_n$ especificará las producciones aplicadas. La anterior secuencia de derivación la podemos denotar como $\alpha_1 \overset{\Pi}{\underset{G}{\Rightarrow}} \alpha_n$ siendo $\Pi = \sigma_1\,\sigma_2 \ldots \sigma_n$. Además, denotaremos por $\alpha_1 \overset{\Pi}{\underset{G}{\Rightarrow}}_{l} \alpha_n$ la derivación por la izquierda obtenida aplicando las reglas de Π. Las anteriores definiciones originan el *lenguaje de Szilard* asociado a cualquier gramática incontextual y que se define como el

conjunto $Sz(G) = \{ \Pi \in Lab^* : S \overset{\Pi}{\underset{G}{\Rightarrow}} x, \ x \in \Sigma^* \}$. De igual forma, $Szl(G) = \{ \Pi \in Lab^* : S \overset{\Pi}{\underset{G}{\Rightarrow}}_l x, \ x \in \Sigma^* \}$. Un resultado conocido acerca de los lenguajes de Szilard es que tomando cualquier gramática incontextual G se cumple que $Szl(G)$ es, a su vez, incontextual.

Una caracterización alternativa de los lenguajes de Szilard definidos para derivaciones por la izquierda se la debemos a Mäkinen [8]. En su trabajo, Mäkinen define la caracterización de los lenguajes $Szl(G)$ en términos de gramáticas puras. Una gramática incontextual pura se define por la tupla (Σ, P, S) donde la única salvedad con respecto a las producciones de una gramática incontextual es que el alfabeto de definición es único. Se ha demostrado que si G cumple ciertas restricciones de recursividad entonces $Szl(G)$ es un lenguaje incontextual puro [8]. Podemos establecer que para cualquier alfabeto Σ se cumple que el lenguaje Σ^* puede ser generado por una gramática incontextual G cuyo lenguaje de Szilard $Szl(G)$ es incontextual puro.

Para mayor conocimiento sobre las gramáticas incontextuales, sus derivaciones y los lenguajes de Szilard recomendamos el trabajo realizado por Mäkinen [8]

La identificación de gramáticas es un problema que fundamenta el ámbito de investigación de la inferencia gramatical como paradigma del aprendizaje automático. Han sido numerosos los resultados proporcionados en este área. Diremos que un algoritmo A identifica una gramática G si dada una secuencia de datos de entrada, el algoritmo produce como salida la gramática. Se han estudiado diversos protocolos de información de entrada tales como secuencias de palabras del lenguaje generado por la gramática, secuencias de palabras del alfabeto de definición, estructuras gramaticales en forma de árboles, etc. Posteriormente, veremos la importancia de estos problemas.

Un problema diremos que es *indecidible* si no existe un algoritmo para su resolución. Existen algunos problemas característicos sobre las gramáticas incontextuales que se han estudiado a lo largo del tiempo. Citaremos algunos de ellos.

1. El problema de la pertenencia
   Dada una gramática incontextual G y una cadena w consiste en establecer si $w \in L(G)$. Este problema es decidible y tiene una resolución polinómica con la longitud de la cadena cuando la gramática se encuentra en forma normal de Chomsky

2. El problema de la equivalencia
   Dadas dos gramáticas incontextuales G_1 y G_2 establecer si $L(G_1) = L(G_2)$. Este problema fue demostrado en su momento como un problema indecidible.

3. El problema de la ambigüedad
   Dada una gramática incontextual G establecer si es ambigua, es decir si existen al menos dos derivaciones por la izquierda distintas que produzcan la misma cadena final. Este problema también fue demostrado en su momento como indecidible.

4. El problema de la identificación a partir de datos positivos
   Dada una gramática incontextual pura G y un conjunto (finito o infinito) S^+ de cadenas pertenecientes a $L(G)$ no existe ningún algoritmo que identifique G a partir de S^+. El caso concreto de lenguajes incontextuales puros fue estudiado por Koshiba, Mäkinen y Takada [7].

3. UN SISTEMA DE CIFRADO SIMÉTRICO

A continuación vamos a proponer un sistema de cifrado simétrico o de clave secreta que se basa en la aplicación de algunos conceptos expuestos en la sección anterior.

En primer lugar tomemos una gramática incontextual en Forma Normal de Chomsky $G_1 = (N_1, \Sigma_1, P_1, S_1)$ cumpliendo las restricciones de recursividad expuestas en la sección anterior y un etiquetado de sus producciones $\sigma_1: P_1 \rightarrow Lab_1$. Se cumple que $L(G_1) = \Sigma_1^*$. A partir de la anterior gramática definimos una gramática $G_2 = (N_2, \Sigma_2, P_2, S_2)$ de forma que $Szl(G_1) = L(G_2)$. Obsérvese que la construcción de G_2 es posible dado que $Szl(G_1)$ es incontextual puro tal y como se ha expuesto en la sección anterior. Además, se aplica también un etiquetado sobre las producciones de G_2 definido como $\sigma_2: P_2 \rightarrow Lab_2$.

El esquema de cifrado y descifrado se detalla a continuación.

Supongamos que el emisor A y el receptor B comparten como clave secreta el anterior par de gramáticas $<G_1,G_2>$. En este caso, dado un mensaje x suponemos que $x \in \Sigma_1^*$. A partir de aquí se toman las siguientes acciones

1. A somete el mensaje x a un análisis sobre la gramática G_1 recuperando la secuencia de producciones Π^1_x que cumple que $S_1 \underset{G_1}{\overset{\Pi^1_x}{\Rightarrow_l}} x$

2. A somete la secuencia Π^1_x al análisis sobre la gramática G_2 recuperando una secuencia de producciones Π^2_x que cumple que $S_2 \underset{G_2}{\overset{\Pi^2_x}{\Rightarrow}} \Pi^1_x$

3. A transmite $y = \Pi^2_x$ a B

Las acciones que toma B sobre el mensaje cifrado y son las siguientes

1. B aplica la secuencia de producciones de G_2 definida por y, es decir por Π^2_x, y obtiene la cadena Π^1_x

2. B aplica la secuencia de producciones de G_1 definida por Π^1_x y obtiene x

3.1. Criptoanálisis del sistema

Plantearemos en primer lugar ataques del tipo *sólo texto cifrado*, es decir, un criptoanalista C dispone de los textos cifrados que le envía A a B o viceversa. En primer lugar, C no puede aplicar un análisis de tipo frecuencial para asignar a los símbolos de y una correlación con los símbolos de x. Obsérvese, por ejemplo, el siguiente trío de derivaciones asociado a los mensajes *aa*, *ab* y *bb*

$$S \underset{G_1}{\Rightarrow_1} A_1 A_1 \underset{G_1}{\Rightarrow_2} a A_1 \underset{G_1}{\Rightarrow_2} aa$$

$$S \underset{G_1}{\Rightarrow}_4 A_2 A_2 \underset{G_1}{\Rightarrow}_5 a A_2 \underset{G_1}{\Rightarrow}_6 ab$$

$$S \underset{G_1}{\Rightarrow}_7 A_3 A_3 \underset{G_1}{\Rightarrow}_8 b A_3 \underset{G_1}{\Rightarrow}_8 bb$$

los mensajes obtenidos por la gramática G_1 serían 122, 456 y 788. Posteriormente, los mensajes anteriores se asocian con las siguientes derivaciones en G_2

$$S \underset{G_2}{\Rightarrow}_1 A_1 A_1 \underset{G_2}{\Rightarrow}_2 A_2 A_3 A_1 \underset{G_2}{\Rightarrow}_3 1 A_3 A_1 \underset{G_2}{\Rightarrow}_4 12 A_1 \underset{G_2}{\Rightarrow}_5 122$$

$$S \underset{G_2}{\Rightarrow}_1 A_1 A_1 \underset{G_2}{\Rightarrow}_2 A_2 A_3 A_1 \underset{G_2}{\Rightarrow}_7 4 A_3 A_1 \underset{G_2}{\Rightarrow}_8 45 A_1 \underset{G_2}{\Rightarrow}_9 456$$

$$S \underset{G_2}{\Rightarrow}_1 A_1 A_1 \underset{G_2}{\Rightarrow}_2 A_2 A_3 A_1 \underset{G_2}{\Rightarrow}_a 7 A_3 A_1 \underset{G_2}{\Rightarrow}_b 78 A_1 \underset{G_2}{\Rightarrow}_c 788$$

Los mensajes obtenidos a partir de la gramática G_2 serían respectivamente 12789, 12346 y 12abc. Obsérvese la imposibilidad de intentar establecer partes del mensaje cifrado junto con el mensaje sin cifrar.

Para el resto de ataques, recurriremos a la inferencia gramatical como herramienta de análisis tal y como se explica a continuación.

3.2. El criptoanálisis como un problema de inferencia gramatical

El problema al que se enfrenta el criptoanalista C es intentar inferir las gramáticas G_1 y G_2 a partir de distintas fuentes de información. No olvidemos que $L(G_2) = Szl(G_1)$ es incontextual pura y que los textos cifrados que intercepta C son cadenas generadas por $Sz(G_2)$. La gramática G_2 pudiera tener asociados lenguajes de Szilard no incontextuales, lenguajes de Szilard incontextuales puros o sencillamente lenguajes de Szilard incontextuales. La decisión de cómo se construye G_2 depende fundamentalmente de la elección de G_1 (que pudiera tener incluso ambigüedad) y del tipo de lenguaje de Szilard que queramos asociar a G_2 (por la izquierda $Szl(G_2)$ o no restringido $Sz(G_2)$).

Debemos hacer constar que, en cualquiera de los casos, C no podrá nunca inferir la gramática G_2 a partir de los datos cifrados ya que este problema es una instancia del problema de identificación de gramáticas demostrado como indecidible y expresado en la sección anterior. Por otra parte, C se enfrentaría a un segundo problema que sería de nuevo obtener la gramática G_1 a partir de los datos proporcionados por la gramática G_2. Uno podría argumentar que ante este problema, C contaría no sólo con datos positivos sino también con datos negativos (es decir, cadenas que G_2 no pudiera generar). Ante este escenario, podemos plantear los siguientes argumentos:

1. La identificación de gramáticas incontextuales a partir de datos positivos y negativos se ha demostrado como un problema *NP*-completo [4].

2. El establecimiento de si G_1 es o no ambigua es indecidible tal y como se ha expresado en la sección anterior. Por lo tanto, C debería establecer si G_1 es o no ambigua dado que un mismo mensaje podría tener asociadas varias derivaciones por la izquierda

Las anteriores consideraciones, impiden el ataque por parte de C con un protocolo de sólo texto cifrado. La siguiente cuestión sería si un cambio de protocolo facilitaría la labor de C. Por ejemplo, supongamos que C dispone de los pares $<x,y>$ de texto cifrado y sin cifrar. De nuevo nos encontraríamos ante un escenario similar al anterior, es decir, el conocimiento acerca de x no facilita la labor de C para la identificación de G_1 debido a dos factores:

1. G_1 genera todos los posibles mensajes sobre el alfabeto de definición, por lo que x no resulta un rasgo distintivo del lenguaje $L(G_1)$.
2. G_1 pudiera plantear ambigüedad y, además, la identificación de G_1 no se podría establecer a partir de datos positivos.

Por las anteriores consideraciones, no se podría establecer un ataque por parte de C que pudiera resultar factible.

4. ALGUNAS CONSIDERACIONES SOBRE LA SEGURIDAD COMPUTACIONAL

Tradicionalmente, se ha considerado que un sistema de cifrado era computacionalmente seguro si la cantidad de recursos computacionales para realizar un ataque con éxito era prohibitivo. Aquí el concepto prohibitivo se entendía en términos de crecimiento exponencial y, de forma más pragmática, en términos de resolución de un problema *NP*-completo.

El anterior sistema de cifrado que hemos planteado soporta un tipo de seguridad computacional que, en buena medida, supera al concepto utilizado anteriormente ya que aquí el ataque no puede fructificar ni siquiera con un número de recursos computacionales ilimitado. De esta forma, podemos definir que un sistema presenta *seguridad computacional completa* si la ruptura del sistema (esto es el establecimiento de sus claves a partir de texto cifrado/sin cifrar) implica la resolución de un problema indecidible.

Queda por establecer, lo cual dejaremos para trabajos futuros, si la seguridad computacional completa implica seguridad incondicional (en términos de teoría de la información) y viceversa.

Agradecimientos

El autor agradece la información facilitada por Erkki Mäkinen acerca de las ideas expuestas sobre la teoría de lenguajes

Referencias

[1] V. Amar and G. Putzolu. On a Family of Linear Grammars. *Information and Control* 7, pp 283-291. 1964.
[2] M. Andrasiu, G. P□un, J. Dassow, A. Salomaa. Language-theoretic problems arising from Richelieu cryptosistems. *Theoretical Computer Science* 116, pp 339-357. 1993.

[3] P. Caballero, C. Hernández. *La Teoría de Grafos como Alternativa para el Diseño de Criptoprotocolos*. Actas de la VII Reunión Española sobre Criptología y Seguridad de la Información (Tomo I) pp 109-121. Servicio de Publicaciones de la Universidad de Oviedo 2002

[4] C. de la Higuera. Characteristic Sets for Polynomial Grammatical Inference. *Machine Learning* 27, pp 125-138. 1997.

[5] C. Hernández, P. Caballero. *Conocimiento nulo sobre problemas NP-completos*. Actas de la VI Reunión Española sobre Criptología y Seguridad de la Información pp 65-72. Ed Ra-Ma 2000.

[6] J Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation* Addison-Wesley Publishing Co. 1979.

[7] I Koshiba, E. Mäkinen, Y. Takada. Inferring pure context-free languages from positive data. *Acta Cybernetica* 14 pp 469-477. 2000

[8] E. Mäkinen. *On context-free derivations*. Acta Universitatis Tamperensis ser A vol. 198 University of Tampere. 1985.

[9] A. Salomaa. *Formal Languages*. Academic Press. 1973.

[10] A. Salomaa, S Yu On a public-key cryptosystem based on iterated morphisms and substitutions. *Theoretical Computer Science* 48, pp 283-296. 1986.

[11] R. Siromoney, L. Mathew. A Public Key Cryptosystem based on Lyndon Words. *Information Processing Letters* 35, pp 33-36. 1990.

[12] R. Tao, S Chen On finite automaton public-key cryptosystem *Theoretical Computer Science* 226, pp 143-172. 1999