Estructuras algebraicas aplicables en criptografía

Ignacio Gallego Sagastume

Facultad de Informática, Universidad Nacional de La Plata
ignaciogallego@gmail.com

Resumen

Existen diferentes estructuras algebraicas que son muy útiles en el ámbito de la seguridad informática y en particular en aplicaciones criptográficas. Algunas de estas estructuras son los cuadrados Latinos (CLs o LSs) y los quasi-grupos (QGs). Ellos son convenientes para la implementación de algoritmos de cifrado, de autenticación y protocolos de comunicación seguros.

Más específicamente, los CLs aleatorios sirven como claves de algoritmos de cifrado simétrico; además, son difíciles de generar y son inviables los ataques por fuerza bruta para adivinar un CL, dado el gigantesco espacio de claves que conforman (ver [MW05]).

El primer problema que se presenta es el de generar CLs aleatorios y uniformemente distribuidos, mediante algoritmos eficientes. Este es un problema parcialmente resuelto por [JM96] en el año 1996, pero existen más alternativas posibles según el grado de aleatoriedad deseado y la eficiencia computacional requerida en la generación. Se han implementado métodos propios con éxito en el proyecto [Gal15c]. La línea de investigación debe continuar con la exploración e implementación de diversos métodos ya conocidos para generación de estas estructuras, como pueden ser [Kos02], [MW91] y [SVT14]. También se prevé trabajar sobre otras aplicaciones prácticas de estas estructuras en criptografía.

Palabras clave: generación, cuadrados Latinos, aleatorios, quasigrupos, QGs, LSs, CLs, distribución uniforme

Contexto

El proyecto en el que se enmarca esta investigación, comenzó a principios de 2012 con el trabajo de tesis de Magister [Gal15d], presentado en agosto de 2015 en la Facultad de Informática de la UNLP. El trabajo de tesis fue desarrollado bajo la dirección de la Dra. Claudia Pons (LIFIA, UNLP).

Desde aquél momento y hasta la actualidad, éste es un trabajo de investigación independiente y unipersonal, que carece de fuentes de financiamiento.

Introducción

Un cuadrado Latino (CL) de orden n es una matriz de dimensiones $n \times n$, que se completa utilizando n símbolos (por ejemplo los números de 0 a $n-1$), y en donde cada símbolo aparece exactamente una vez en cada fila y una vez en cada columna. Por ejemplo:

1 3 2
2 1 3
3 2 1

es un CL de orden 3. De un CL puede decirse si es aleatorio o no. Por ejemplo,

1 2 3 4
2 3 4 1
3 4 1 2
4 1 2 3

no es aleatorio, pues sus símbolos aparecen siempre en el mismo orden (informalmente no está "mezclado").

Un algoritmo genera CLs aleatorios y uniformemente distribuidos si cualquier CL posible tiene la misma probabilidad de obtenerse como resultado (no hay tendencias en la generación). Esta propiedad la tiene el algoritmo de Jacobson & Matthews [JM96] (abreviado J&M), donde se demuestra que los resultados son *aproximadamente* uniformes.

La cantidad de todos los LSs posibles de orden *n*, llamada *L(n)*, es tan grande que en la actualidad solo se conoce hasta *L(11)*. A medida que *n* crece, *L(n)* crece exponencialmente. Para órdenes mayores a 11 sólo se conocen cotas superiores e inferiores de *L(n)* (ver [MW05]). Calcular este valor para cualquier *n* se trata de un problema de combinatoria sumamente complejo que en la actualidad permanece sin resolver. Esto implica que, para órdenes grandes (digamos, mayores a 16), un ataque por fuerza bruta probando todos los CLs resulta imposible (con el poder de cómputo disponible actualmente).

Los CLs pueden utilizarse como claves de algoritmos simétricos para cifrar y descifrar información en un protocolo criptográfico de comunicación. Los CLs aleatorios de orden 256 son de particular importancia, porque permiten cifrar y descifrar cualquier carácter de la tabla de códigos ASCII, utilizando un recorrido secuencial a la manera de Gibson [Gib12].

Estas claves deben ser generadas con frecuencia para evitar posibles ataques. El primer problema que se presenta entonces es el de obtener un algoritmo que genere CLs aleatorios y aproximadamente uniformemente distribuidos de orden 256 y en una forma lo más eficiente posible.

Este problema fue parcialmente resuelto por J&M, pero, al implementar su algoritmo para órdenes tan grandes, el mismo resulta bastante ineficiente (se desempeña en $O(n^3)$).

Se han propuesto varias alternativas propias a este método en el trabajo de tesis [Gal15d], las cuales se describen en la sección de resultados. Algunas de éstas son más eficientes aunque menos uniformes. En la siguiente sección se detallan las líneas abiertas de investigación y objetivos a corto plazo, mientras que en la sección de resultados y objetivos se dan los objetivos a largo plazo.

Líneas de Investigación, Desarrollo e Innovación

Actualmente, se está trabajando en los algoritmos propuestos por McKay y Wormald [MW91] (pioneros en el área), que datan de antes del método propuesto por J&M y dan una implementación de tiempo de $O(nk^3)$, dado $k=o(n^{1/3})$, para generaciones de rectángulos Latinos de *k* filas por *n* columnas. Los resultados de este método son uniformemente distribuidos.

También se está trabajando en un método reciente propuesto por Selvi (et. al.) en [SVT14], que corrige un método

de O'Carroll de 1963 ([Car63]) que fallaba para algunos casos.

Otro enfoque al problema de generación es el de Fontana ([Fon13]), que extrae un CL uniformemente de todos los CLs posibles mediante un método todavía ineficiente en la práctica (sólo se ha llegado a generar CLs de orden 7 con este método). Pero aunque no sea práctico, los elementos expuestos en este trabajo resultan enriquecedores en el área y aportan elementos de análisis al problema de la generación.

La comprensión de estos trabajos en profundidad es esencial para su implementación en un lenguaje de programación, y para su posterior optimización y puesta en producción.

Resultados y Objetivos

Se han implementado diversos métodos (conocidos y también propios) de generación de CLs en el proyecto Java en GitHub.com [Gal15c]; algunos mejoran la eficiencia con respecto al método de J&M, aunque no obtienen CLs tan uniformemente distribuidos. Estas implementaciones son útiles cuando la uniformidad no es una condición que se requiera estrictamente. Podrían además utilizarse los distintos métodos alternativamente, como por ejemplo J&M para el inicio de la comunicación y el método del grafo de reemplazos en generaciones posteriores. De esta manera, se lograría una negociación entre tiempo y uniformidad de los resultados, aprovechando las ventajas de cada algoritmo.

Utilizando un concepto de producto de CLs de Koscielny, en [Gal13] se dio una implementación de un algoritmo muy eficiente ($O(n^2)$) para generar CLs de cualquier orden, aunque los resultados no son uniformes.

En el trabajo [Gal14] se da una implementación optimizada de J&M, que puede ejecutarse paso a paso para ver los resultados intermedios y finales mediante gráficos en 3D (utilizando la librería "OpenGL"). Esta propuesta se desarrolló con fines didácticos y educativos.

El método propuesto en [Gal15a] (no publicado), también detallado en el capítulo 5 de la tesis [Gal15d], converge en forma muy eficiente ($O(n^3)$ pero menor al algoritmo de J&M) a CLs prácticamente uniformes.

Por otra parte, el método de grafo de reemplazos publicado en [Gal15b], supera ampliamente en eficiencia (en un factor de 5 a 1) a la implementación dada del método de J&M, y la uniformidad de los CLs alcanzada es aceptable, lo cual lo hace útil en la práctica.

Los objetivos a largo plazo son investigar otras estructuras algebraicas como las propuestas por Koscielny (en [Kos95], [Kos02] y [Kos04]). Por ejemplo, los anillos, grupos, campos de Galois "adulterados" y otras estructuras son útiles para implementar soluciones eficientes en lenguajes de programación modernos como Java, C++ o Python. Esta línea se propone porque Koscielny da implementaciones en Maple 7, que es un lenguaje matemático muy específico, no disponible para sistemas que usen lenguajes de propósito general como los antes mencionados.

Además de generar CLs, se pretende estudiar la utilización de estas estructuras en aplicaciones prácticas (criptográficas o no), como por ejemplo en mecanismos de autenticación, algoritmos de cifrado/descifrado (alternativos al citado [Gib12]), protocolos de conocimiento cero, generadores de números pseudo-aleatorios (pseudo-random number generators o PRNGs), utilizar CLs como key-stream generators en stream ciphers (es decir, como generadores de claves), y

estudiar posibles ataques cuando se usan CLs en criptografía.

Formación de Recursos Humanos

Se prevé seguir trabajando sobre las líneas de investigación inconclusas y en nuevas líneas no consideradas hasta la actualidad. Se pretende publicar los resultados en congresos de seguridad informática o eventos especializados en criptografía a nivel nacional (como se hizo desde el inicio del proyecto). Si es posible, se desea también enviar trabajos a congresos o revistas especializadas a nivel internacional; el objetivo de esto último es interactuar con especialistas en criptografía para enriquecer el estudio. Si se consiguen resultados a nivel internacional, se considerará incluirlos en una posible tesis de doctorado en Ciencias de la Computación.

También sería muy conveniente conformar un equipo de trabajo que contribuya en las actividades de I+D+I en el tema propuesto. Se desea también buscar posibles formas de financiamiento del proyecto, lo cual permitiría brindar al equipo un feedback positivo en temas relacionados con la seguridad informática, y especialmente en criptografía.

Referencias

[Gib12] Steve Gibson, *Off the grid* (Gibson Research online), https://www.grc.com/offthegrid.htm (2012).

[JM96] Mark T. Jacobson and Peter Matthews, *Generating uniformly distributed random Latin squares*, J. Combin. Des. 4 (1996), no. 6, 405-437. MR MR1410617 (98b:05021)

[MW91] Brendan D. McKay and Nicholas C. Wormald, *Uniform generation of random Latin rectangles*, J. Combin. Math. Combin. Comput. 9 (1991), 179-186. MR 1111853 (92b:05013).

[MW05] Brendan D. Mckay and Ian M. Wanless, *On the number of latin squares*, Ann. Combin. 9 (2005), 335-344.

[SVT14] D. Selvi, G. Velammal, and Thevasahayam Arockiadoss, *Modified method of generating randomized Latin squares*, Journal of Computer Engineering (IOSR-JCE) 16 (2014), 76-80.

[Kos95] Czeslaw Koscielny, *Spurious Galois fields*, Int. J. Appl. Math. Comput. Sci. 5 (1995), no. 1, 169-188.

[Kos02] Czeslaw Koscielny, *Generating quasigroups for cryptographic applications*, Int. J. Appl. Math. Comput. Sci. 12 (2002), no. 4, 559-569.

[Kos04] Czeslaw Koscielny, *Spurious multiplicative group of GF(pm): a new tool for cryptography*, no. 12, 61-73.

[Fon13] Fontana R.: *Random Latin squares and Sudoku designs generation*. ArXiv e-prints, 2013.

[Car63] O'Carroll, F, *A Method of Generating Randomized Latin Squares*, Biometrics. December 1963, 652-653.

[Gal13] Ignacio Gallego Sagastume, *Un método para la generación de cuadrados latinos de orden 256*, Congreso Nacional de Ingeniería Informática y Sistemas de Información (CoNaIISI). UTN Regional Córdoba, Argentina. (2013).

[Gal14] Ignacio Gallego Sagastume, *Generation of Latin squares step by step and graphically*, Congreso Nacional de Ciencias de la Computación (CACIC) 2014. Universidad de La Matanza, Buenos Aires, Argentina. (2014).

[Gal15a] Ignacio Gallego Sagastume, *Generation of Random Latin Squares Using a Random-Swapping Technique* (no publicado), 2015.

[Gal15b] Ignacio Gallego Sagastume, *Generación de cuadrados latinos de orden 256 utilizando un grafo de reemplazos*, Congreso Nacional de Ingeniería Informática y Sistemas de Información (CoNaIISI). UTN Regional Buenos Aires, Argentina (2015).

[Gal15c] Ignacio Gallego Sagastume, *Proyecto "igs-lsgp" (latin square generation package) en github*, https://github.com/bluemontag/igs-lsgp/wiki, 2015.

[Gal15d] Ignacio Gallego Sagastume, tesis de Magister en Ing. De Software "*Análisis de algoritmos para generación de cuadrados Latinos aleatorios para criptografía*", agosto de 2015. http://sedici.unlp.edu.ar/handle/10915/47577