Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Departamento de Ingeniería Eléctrica

Sección de Computación

Métodos Computacionales para Esquemas de Compartición de Secretos Ideales

Tesis que presenta

Leonor Vázquez González

Supervisada por: Guillermo Morales Luna

Septiembre 2004

Resumen

Los Esquemas de Compartición de Secretos (ECS) controlan el acceso a sistemas de información, distribuyendo la responsabilidad entre varios usuarios. De esta manera, una transacción se lleva a cabo si y sólo si conjuntos particulares de usuarios acceden al sistema y autorizan esa transacción. La colección de esos conjuntos particulares es la estructura de acceso del ECS. Estos se basan en modelos matemáticos, algunos en matroides representables. Ejemplos típicos de matroides los constituyen las colecciones de conjuntos linealmente independientes en espacios vectoriales. Un matroide es representable si es isomorfo a uno de esa forma. Las estructuras de acceso y los matroides se corresponden en cuanto a que los conjuntos complementarios de una estructura de acceso conforman un matroide. Con matroides representables, un conjunto de usuarios queda partido como una unión de clanes disjuntos, y a los elementos de cada clan se les asigna llaves privadas de acceso, o fragmentos del secreto. Con esto, una transacción se realiza si y sólo si al menos se reúnen dos participantes en clanes distintos para acceder al sistema. Esta condición de "umbral 2" no es relevante, pues de manera similar puede construirse ECS de cualquier umbral. Esta tesis presenta un método para obtener, de manera aleatoria, matroides representables (aquí la aleatoriedad de la selección de matroides representables, coincide con la selección aleatoria de subespacios lineales de un espacio vectorial: cada tal subespacio da lugar a un matroide de vectores l.i.), los procedimientos de distribución y recuperación de fragmentos en un ECS ideal, así como las relaciones algorítmicas entre las nociones de matroides, estructuras de acceso y ECS. Realizamos algunos procedimientos de conteo en estructuras relevantes, tales como espacios vectoriales sobre campos finitos, bases y matroides. Hacemos también consideraciones en cuanto a la seguridad lograda y las complejidades de las construcciones realizadas. El presente trabajo desarrolla un prototipo computacional para realizar, en la práctica, experimentos de resultados bien conocidos. El énfasis de nuestro trabajo está en el análisis de los métodos. Por un lado, al generar un matroide sobre un conjunto de participantes, ha de darse como la lista de todas sus bases. En la práctica, este recuento es innecesario y aquí es de interés puramente analítico. Por otro lado, nuestra implementación de los ECS, además de efectiva, es, en efecto, incorporable a un sistema de explotación práctica.

Abstract

Secret Sharing Schemes (SSS) provide access control to information systems, by distributing the responsibility among several users. Transactions are carried out if and only if users at special sets provide their keys (which are just pieces of knowledge) in order to gain admittance to the system. The collection of those special sets is the *SSS access structure* indeed. The SSS are based on combinatorial notions, such as representable matroids. The prototypical matroids are families of linearly independent sets in vector spaces, and their isomorphic images are called *representable*. It is rather common that the access structure of a SSS is such that the collection of complements of its access sets forms a matroid. If a representable matroid is built over a set of participants, then this set can be partitioned as a disjoint union of cliques. Participants at each clique receive a share, and the secret can be recovered whenever two participants at different cliques provide their shares. This scheme, called of threshold 2 can be generalized to schemes with higher threshold. Here we produce a method to randomly generate representable matroids (the randomness of the selection consists basically on the random selection of linear subspaces in finite dimensional vector spaces over finite fields), and procedures to distribute shares and to recover secrets in order to get ideal SSS. We analyze from the computing point of view the relations between access structures, representable matroids and ideal SSS. We develop and implement some counting procedures in vector spaces over finite fields, basis and matroids. We estimate the time complexities of the involved algorithms and their connection with security. The developed computational prototype is aimed to provide an experimental platform for theoretical results. Hence the emphasis in our thesis is in the analytical side. In practice, the exhaustive listing of all basis in a matroid is useless, however the implementation of this part allows us to effectively test our counting procedures. The second part of our implementation consists in sharing secrets and distributing shares among participants. This implementation is affective and efficient and able to be embedded in an automatic SSS release to the general public.

Agradecimientos

En primer lugar agradezco a CONACYT por el apoyo económico brindado durante el programa de maestría.

Agradezco a mis padres Abel Vázquez y Leonor González por estar siempre a mi lado apoyándome en todo.

Agradezco a mis hermanos (Juan y Verónica) por su apoyo durante estos años de estudio.

A Darwin por el apoyo que me ha brindado en todo momento.

Agradezco a mi asesor por el apoyo brindado en el desarrollo de mi tesis, así como por poner a mi disposición el equipo necesario para el desarrollo de esta investigación.

A los profesores Francisco José Rodríguez Henríquez por su apoyo en mis estudios.

A Sofía Reza Cruz por su ayuda incondicional.

A Graciela Meza C., Raúl Montaño M. y David Torres F. por sus atenciones brindadas en todas las ocasiones que acudí a la Biblioteca de Ingeniería Eléctrica.

A todos los profesores y compañeros de la Sección de Computación por su amistad durante estos dos años.

Índice General

Agradecimientos **i**

Índice **iv**

1 Introducción **1**

Introducción **1**

2 Antecedentes **7**
2.1 Teoría de Campos . . . 7
2.1.1 El Campo Finito Primo . . . 9
2.1.2 Cómputo en Campos de Galois . . . 11
2.2 Teoría de Matroides . . . 11
2.2.1 Definiciones y Resultados Básicos . . . 11
2.2.2 Matroides Representables . . . 13
2.2.3 Circuitos de un Matroide . . . 14
2.2.4 Teoría de Gráficas . . . 15
2.2.5 Descomposición en Sumas . . . 16
2.3 Esquemas de Compartición de Secretos . . . 16
2.3.1 Esquemas Ideales . . . 20
2.3.2 Esquema de Shamir . . . 20
2.3.3 Esquema de Brickell-Davenport . . . 25

3 Selección Aleatoria de Matroides Representables **29**
3.1 El Problema de la Selección Aleatoria . . . 29
3.2 Algoritmos para la Obtención de un Matroide Representable . . . 31
3.3 Complejidad de los Algoritmos . . . 36
3.4 Métodos Alternativos para la Selección de Matroides . . . 37

4 El Sistema Propuesto **41**
4.1 ECS Ideal mediante Gráficas y Matroides . . . 42
4.1.1 Estructura de Acceso . . . 43
4.2 Construcción del ECS . . . 46
4.2.1 La Partición del Secreto . . . 47
4.2.2 La Recuperación del Secreto . . . 48
4.2.3 Idealidad y Perfección . . . 49

4.2.4 Ejemplo del Sistema 50

5 Resultados y Discusión **53**

6 Conclusiones y Trabajo a Futuro **61**

Apéndice 1. Algoritmos **65**

Apéndice 2. Composición del CD **71**

Bibliografía **74**

Capítulo 1

Introducción

No sólo es la curiosidad que conduce el deseo de introducir a los secretos. En una guerra, la táctica (secreta) de la inteligencia es vital para alcanzar la victoria. En el mercado financiero, el conocimiento de información oculta significa ganancias. De hecho, está a los intereses de muchos individuos el poder obtener una cierta información confidencial, o el *secreto*. Por tanto, debemos poder proteger los secretos contra estos intrusos. Existen muchas maneras de proteger un secreto.

La motivación para compartir secretos fue (y sigue siendo) el manejo seguro de llaves. En algunas situaciones, hay generalmente una llave secreta que proporciona el acceso a muchos archivos importantes. Si se pierde tal llave (se le olvida a la persona que la sabe, o se destruye la computadora que la almacena), todos los archivos importantes llegan a ser inaccesibles. Esta es la idea básica que motivó la creación de los Esquemas de Compartición de Secretos (ECS) inventados independientemente por Blakley [4] y Shamir [20]. Tales esquemas dividen la llave secreta en pedazos, mismos que distribuyen a ciertas personas, de modo que ciertos subconjuntos de las mismas pueden reunirse para recuperar la llave. Esto fuerza a posibles adversarios a atacar múltiples "partes", para conocer el secreto o bien, para destruirlo.

Formalmente, los ECS son protocolos criptográficos que protegen la privacidad y la integridad de la información (el secreto).

Los ECS en general, están basados en diversos modelos matemáticos. El modelo creado inicialmente para los ECS se llama *esquema* (m, n)*-umbral* (o *esquema* m*-de-entre-*n), donde n es el número de personas en el esquema y m es el mínimo número de ellas que puede recuperar el secreto. Las diversas elecciones para los valores de n y m reflejan la compensación entre seguridad y confiabilidad.

Blakley en su esquema utilizó geometría proyectiva, mientras que Shamir basó su modelo en la interpolación de polinomios.

Hasta la fecha se han inventado muchos ECS, los cuales se dedican principalmente a

la caracterización de la colección de conjuntos que pueden recuperar el secreto, es decir, de la propia estructura de acceso. Dada tal colección, la mayoría de los procedimientos para fragmentar se basan en el esquema de Shamir, véase [24], [16].

Por otra parte Brickell y Davenport [7], caracterizaron a los ECS ideales (la noción de ideal se refiere a que el tamaño en bits de los fragmentos es el mismo que el del secreto). Tal construcción está basada en una estructura bien estudiada en Combinatoria conocida como *matroide representable*.

Los matroides nacen de las nociones de "matrices" e "independencia lineal", conceptos bien estudiados en Álgebra Lineal. Así un *matroide* es una colección de objetos llamados *independientes* que satisfacen las tres reglas siguientes:

- El vacío es independiente.
- Cualquier subconjunto de un conjunto independiente es independiente.
- Dados dos conjuntos independientes donde uno posee un elemento más que el otro, entonces es posible encontrar un elemento del conjunto mayor, que cuando se le añade al menor, resulta un conjunto independiente.

A los conjuntos que no son independientes, los conocemos como conjuntos *dependientes* (en analogía con el Álgebra Lineal).

El campo de estudio de los matroides es extenso, comprende nociones de rango, dependencia, dualidad, esquemas voraces, Teoría de Códigos y Teoría de Gráficas entre otras.

Tenemos interés en matroides que son representables sobre campos finitos. Estos últimos siempre son de la forma: $\mathbb{F}_q$, donde $q = p^k$ es una potencia de un número primo p. Los espacios vectoriales sobre ese campo son de la forma $\mathbb{F}_q^n$ para un entero $n \geq 1$ llamado *dimensión* del espacio. La colección de conjuntos

$$\mathcal{C}_{m,n,q} = \{L \mid L \text{ es l. i. en } \mathbb{F}_q^n \text{ y } \mid L \mid = m\}$$

donde por $\mid L \mid$ denotamos al número de elementos del conjunto L, forma un matroide llamado de *conjuntos linealmente independientes*. De particular interés, son los matroides de la forma $\mathcal{B}_{n,n,q} = \mathcal{C}_{n,n,q}$ (aquí $m = n$) pues éstos tienen como bases precisamente a las bases de $\mathbb{F}_q^n$ en el sentido de álgebra lineal (en el ejemplo 2.2.10, presentamos a unos matroides denotados mediante $\mathcal{B}_{m,n,q}$, con $m \leq n$, y son exactamente de la forma $\mathcal{C}_{m,m,q}$ relativos a subespacios $V < \mathbb{F}_q^n$ de dimensión m).

Un matroide isomorfo a uno de la forma $\mathcal{B}_{m,n,q}$ es representable. En tales matroides, la colección de conjuntos dependientes proporciona (mediante el teorema de Brickell-Davenport [7]) esquemas ideales.

El resultado de Brickell-Davenport afirma que dado el matroide representable $\mathcal{B}_{m,m,q}$ el conjunto de participantes del ECS ideal coincide con $\mathbb{F}_q^n - \{0\}$ (el $\{0\}$ hará las veces de "encargado" de distribuir los secretos) y la estructura de acceso para el ECS es precisamente:

$$\mathcal{A} = \{D \mid D \text{ es dependiente en el matroide representable}\}.$$

Aunque en su presentación original, este teorema no era constructivo (no se decía cómo construir el ECS correspondiente) es posible ciertamente componer diversos procedimientos parciales para construir el ECS ideal.

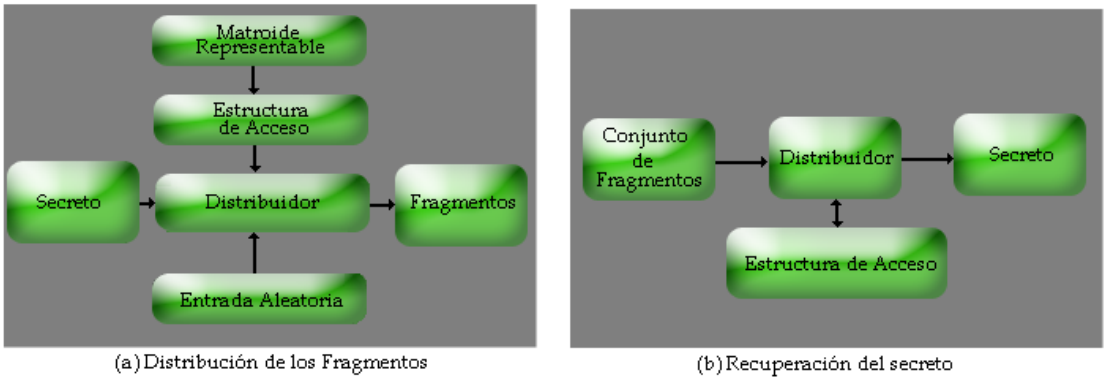
Por otro lado, si $m < n$ un espacio vectorial de dimensión m, sobre un campo fijo, puede identificarse de muchas maneras como un subespacio del espacio vectorial de dimensión n sobre el mismo campo. Mediante una tal identificación la construcción que se haga para $\mathcal{B}_{m,m,q}$ se traduce directamente en una construcción en $\mathcal{B}_{m,n,q}$. Así es posible construir un ECS ideal cuya estructura de acceso es la colección de conjuntos dependientes respecto a $\mathcal{B}_{m,n,q}$, sobre el conjunto de participantes $\mathbb{F}_q^n - \{0\}$.

Naturalmente si P es un conjunto con no más de q^n elementos, entonces P se pone en correspondencia con un subconjunto de $\mathbb{F}_q^n - \{0\}$ y $\mathcal{B}_{m,n,q}$ con un subconjunto de "bases" B de un matroide definido sobre P.

Este trabajo está dividido en dos partes fundamentales: primero proporcionamos una manera de seleccionar aleatoriamente matroides representables de la forma $\mathcal{B}_{m,n,q}$ (esto se restringe a elegir aleatoriamente un subespacio $V < \mathbb{F}_q^n$, de dimensión m, y construir sobre él el correspondiente $\mathcal{B}_{m,n,q}$), y por otro lado proponemos un protocolo de ECS ideal, el cual requiere como parámetros el secreto s, el número de participantes N (el cual determinará la característica p del campo finito, a la potencia respectiva $q = p^k$ y a la dimensión n) así como a la colección de conjuntos dependientes respecto a un matroide $\mathcal{B}_{m,n,q}$ para realizar a esa colección como estructura de acceso. Mediante la identificación descrita al final del párrafo anterior, a la cual nos referiremos como la "inyección" de P en $\mathbb{F}_q^n$, esta construcción es transportable a un conjunto P de participantes. Esto liga a ambas partes de la tesis: se podrá generar un matroide sobre P que sea representable, y luego considerando los conjuntos dependientes, un ECS correspondiente sobre P.

La estructura de acceso creada a través de matroides de un ECS dificulta los ataques de seguridad, pues no es fácil determinar el matroide representable que ha sido utilizado para el ECS, pues éste, según vimos antes, depende de la inyección elegida de P en $\mathbb{F}_q^n$.

Ilustramos la estructura del sistema elaborado en la figura 1.1: Así como al hablar de conjuntos independientes en matroides hemos distinguido a los maximales y los hemos llamado bases, es importante distinguir a los minimales dependientes, los cuales reciben



(a) Distribución de los Fragmentos (b) Recuperación del secreto

Figura 1.1: El Sistema ECS Propuesto

el nombre de *circuitos*. En el teorema de Brickell-Davenport, los circuitos son sumamente importantes y permiten partir a un conjunto de participantes como una unión disjunta de subconjuntos no vacíos. Cualesquiera dos participantes en un tal subconjunto están relacionados entre sí, por lo que esos subconjuntos adquieren estructuras de gráficas completas. Llamemos pues *clan* a cada uno de ellos.

Para efectuar la fragmentación del secreto s, primero proporcionamos la partición del conjunto de participantes, como una unión de clanes disjuntos, donde a cada elemento de un clan le asignamos un fragmento del secreto mediante el esquema de Shamir.

En la implementación del sistema desarrollado nos hemos restringido a considerar sólo campos finitos primos $\mathbb{F}_p$ donde p es primo, ya que en ellos la aritmética coincide con la de los enteros módulo p. Sin embargo, los procedimientos pueden transportarse directamente a campos compuestos $\mathbb{F}_q$, con $q = p^k$ una potencia de un primo, realizando la aritmética de polinomios módulo un polinomio irreducible de grado k sobre $\mathbb{F}_p$. En nuestra presentación usaremos el símbolo p para denotar a un primo y el símbolo q para denotar a una potencia de un primo. Así los resultados de la implementación actual hacen recomendable que el número de participantes sea de la forma $p^n - 1$, donde p es un número primo y n relativamente pequeño (del orden de decenas). Esto ofrece, sin embargo, un esquema de seguridad con alto grado de inviolabilidad. También hemos comprobado que la construcción es eficiente.

Nuestro trabajo tiene como propósito realizar un análisis de métodos de conteo de diversos componentes de matroides representables y de la efectividad de construcción de ECS basados en el Teorema de Brickell-Davenport, así como la realización de un prototipo computacional que nos permita comprobar la corrección de los análisis desarrollados. El énfasis de este trabajo está entonces en la parte analítica. La generación aleatoria de matroides tiene un interés puramente formal: en la práctica resulta ocioso hacer una enumeración exhaustiva de todos los conjuntos de acceso en un ECS. Las enumeraciones

que puede hacer nuestra implementación nos permiten verificar los métodos de conteo desarrollados. Sin embargo, como dijimos antes, la construcción de los ECS que realizamos es eficiente, y esta parte bien puede incorporarse a un ulterior desarrollo con fines de explotación.

Esta tesis está organizada de la siguiente manera: en el capítulo 2 proporcionamos un breve repaso a los fundamentos teóricos necesarios para la comprensión del texto: Teoría de Campos, Teoría de Matroides y Esquema de Compartición de Secretos, en el capítulo 3 describimos el proceso de la selección aleatoria de matroides representables y describimos la elaboración de los algoritmos involucrados; en el capítulo 4 describimos el sistema que proponemos para la creación de un ECS ideal con el uso del matroide representable $\mathcal{B}_{n,n,q}$ y detallamos los algoritmos involucrados en la construcción del ECS ideal, en el capítulo 5 planteamos la discusión de los resultados obtenidos, en el capítulo 6 proporcionamos las conclusiones y sugerimos un posible trabajo futuro, finalmente, colocamos la bibliografía consultada.

Capítulo 2

Antecedentes

2.1 Teoría de Campos

En este capítulo proporcionamos las nociones básicas de la Teoría de Campos.

Definición 2.1.1 *Un* grupo *es una estructura algebraica* $(G, \circ)$*, donde* G *es un conjunto no vacío dotado con una operación binaria,* $\circ : G \times G \to G$*, tal que se verifican las siguientes propiedades:*

1. *Para cualesquiera* $a, b, c \in G$, $\circ$ *es asociativa, es decir:* $a \circ (b \circ c) = (a \circ b) \circ c$.
2. *En* G *existe un elemento* $e \in G$ *(llamado el* neutro*) que cumple* $e \circ a = a \circ e = a$*, para todo* $a \in G$;
3. *Para todo* $a \in G$ *existe* $a^{-1} \in G$ *(llamado el* inverso *de* a*) tal que* $a \circ a^{-1} = a^{-1} \circ a = e$.

Definición 2.1.2 *Un grupo* $(G, \circ)$ *donde verificamos* $a \circ b = b \circ a$ *para todo* $a, b \in G$ *se dice ser* abeliano *o* conmutativo.

Definición 2.1.3 *A un grupo* $(G, \circ)$ *lo llamamos* finito *si el conjunto* G *es finito, de otra forma, lo llamamos grupo* infinito.

Denotaremos a un grupo $(G, \circ)$ simplemente por G, mientras esto no nos cause confusión.

Ejemplo 2.1.4 *Las siguientes estructuras son grupos:*

- $(\mathbb{R}, +)$ *es un grupo abeliano.* $\mathbb{R}$ *es el conjunto de los números reales y* $+$ *es la suma usual.*
- $(\mathbb{R} - \{0\}, \cdot)$ *es un grupo abeliano.* $\mathbb{R}$ *es el conjunto de los números reales y* $\cdot$ *es el producto usual. El cero no tiene inverso multiplicativo, por eso lo excluimos. Por otro lado,* $\mathbb{R} - \{0\}$ *denota la diferencia de los conjuntos* $\mathbb{R}$ *menos* $\{0\}$.
- $(\mathbb{Z}/n\mathbb{Z}, +)$ *es un grupo, donde* $\mathbb{Z}/n\mathbb{Z}$ *es el conjunto de los enteros módulo* n*,* $n \in \mathbb{Z}$ *y* $n > 0$*. Usualmente escribimos* $\mathbb{Z}/n\mathbb{Z} = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}$.

Los grupos $(\mathbb{R}, +)$ y $(\mathbb{R}, \cdot)$ son infinitos. El grupo $\mathbb{Z}/n\mathbb{Z}$ es finito (posee n elementos).

Definición 2.1.5 *Un* anillo *es una estructura $(R, +, \cdot)$, donde $+$ y $\cdot$ son dos operaciones binarias tales que:*

1. *$(R, +)$ es un grupo conmutativo, cuyo neutro (llamado* neutro aditivo*) denotamos por $0_R \in R$.*
2. *$\cdot$ es asociativa y posee* unidad*, es decir, existe $1_R \in R$ tal que: $1_R \cdot a = a = 1_R \cdot a$ para todo $a \in R$.*
3. *Vale la* ley distributiva *del producto sobre la suma:*

$$(a + b) \cdot c = a \cdot c + b \cdot c,$$

$$a \cdot (b + c) = a \cdot b + a \cdot c.$$

Usualmente omitimos el símbolo "$\cdot$" cuando esto no causa confusión. Decimos que un anillo conmutativo*, si su multiplicación es conmutativa.*

Definición 2.1.6 *Un* campo *es una estructura $(K, +, \cdot)$, donde $+$ y $\cdot$ son dos operaciones binarias tales que:*

1. *$(K, +)$ es un grupo conmutativo.*
2. *$(K^*, \cdot)$, es un grupo conmutativo y $K^* = K - \{0\}$.*
3. *Vale la ley distributiva del producto sobre la suma.*

Ejemplo 2.1.7 *Los siguientes conjuntos son campos:*

- *$(\mathbb{Q}, +, \cdot)$ es el campo de los números racionales.*
- *$(\mathbb{R}, +, \cdot)$ es el campo de los números reales.*
- *$(\mathbb{C}, +, \cdot)$ es el campo de los números complejos.*
- *Si p es un primo, $\mathbb{Z}/p\mathbb{Z}$ es un campo. A este campo lo solemos denotar por $\mathbb{F}_p$ y es claramente finito.*
- *En particular, el campo más pequeño, y uno de los más importantes, es cuando $p = 2 : \mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z} = \{0, 1\}$.*

Definición 2.1.8 *Decimos que dos campos $(K_1, +_1, \cdot_1), (K_2, +_2, \cdot_2)$ son* isomorfos *si existe una función:*

$$f : K_1 \to K_2,$$

tal que f es un homomorfismo respecto a la suma, es decir $\forall a, b \in K_1, f(a +_1 b) = f(a) +_2 f(b)$, también f es un homomorfismo respecto al producto: $\forall a, b \in K_1, f(a \cdot_1 b) = f(a) \cdot_2 f(b)$, y además tal función es una función biyectiva.

La Teoría de Galois Clásica (que explicamos en la siguiente sección) es parte de la herramienta algebraica que aplicamos a nuestro trabajo. Su estudio no es una abstracción innecesaria, ya que los campos finitos aparecen de manera natural en áreas como la Geometría Algebraica, Teoría de Códigos y Teoría de Números.

2.1.1 El Campo Finito Primo

En este apartado precisaremos el concepto de característica de un campo y el de subcampo primo. Es importante hacer la observación de que sólo los campos finitos tienen característica no nula.

Definición 2.1.9 *Sea K un campo. Llamamos* característica *de un elemento $x \in K - \{0\}$, al mínimo entero positivo p tal que $p \cdot x = 0$. Un elemento no nulo del campo, digamos x, es de* característica nula *si $\forall p \in K,\ p \cdot x = 0$ implica $p = 0$.*

Proposición 2.1.10 (Véase [10]) *Sea K un campo.*

1. *Todos los elementos de K tienen la misma característica, que representa un invariante del campo. A este número lo llamamos la* característica del campo *y lo denotamos por: car K.*
2. *La característica de un campo es cero, o bien, es un número primo.*
3. *Todo $x \in K - \{0\}$ cumple:*
   - *Si car $K = 0$, entonces el subgrupo aditivo generado por x es infinito.*
   - *Si car $K = p$, entonces el subgrupo aditivo generado por x es finito de orden p.*

Definición 2.1.11 *El subcampo del campo K, generado por la unidad $1_K \in K$, recibe el nombre de* subcampo primo *de K. Tal subcampo lo representaremos en adelante por la expresión $\Pi_K = \{0, 1_K, 2 \cdot 1_K, \ldots, (p-1) \cdot 1_K\}$.*

El subcampo primo es, por construcción, el mínimo subcampo de K, contenido en cualquier otro subcampo de K.

Observación 2.1.12 *Sea K un campo de característica p y sea Π_K el subcampo primo de K. Entonces Π_K es isomorfo al conjunto $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$, el cual es también un campo. Así, tenemos que:*

$$\Pi_K \cong \mathbb{F}_p, \text{ si } \Pi_K \text{ es finito.} \tag{2.1}$$

Si K es un campo finito, entonces su característica es positiva (porque $(K, +)$ es un grupo finito), y como mencionamos anteriormente, la característica (si es no nula) es siempre un número primo. El siguiente resultado limita el orden que puede tener un campo finito:

Proposición 2.1.13 (Véase [10]) *Sea K un campo finito de característica p. Entonces $|K| = p^n$ para algún entero $n > 0$.*

Observación 2.1.14 *En el Teorema del Binomio:*

$$(a+b)^n = \binom{n}{0}a^n + \binom{n}{1}a^{n-1}b + \ldots + \binom{n}{n}b^n$$

vale en cualquier anillo conmutativo. Si p es un número primo, entonces p divide a $\binom{n}{k}$, $1 \leq k \leq p-1$. En consecuencia, cuando un campo K tiene característica p, tenemos que:

$$(a+b)^p = a^p + b^p.$$

El siguiente resultado da también una limitación sobre la estructura de un campo finito:

Proposición 2.1.15 *Si K es un campo finito con $|K| = p$ (p un número primo), entonces el grupo multiplicativo, $K^* = K - \{0\}$ es cíclico (i.e. esta generado por un sólo elemento $1_K \in K$) y es isomorfo a $\mathbb{F}_{p-1}$.*

A continuación estudiamos someramente a los campos de Galois hasta lograr establecer que todo campo de Galois, K, es el campo de descomposición del polinomio $p(x) = x^q - x$.

En primer lugar recordemos algunos tópicos de la resolución de ecuaciones algebraicas.

Definición 2.1.16 *Un* polinomio *en la indeterminada x y coeficientes en un campo K es de la forma:*

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0.$$

Denotaremos por $K[x]$ al conjunto de todos estos polinomios en la indeterminada x y coeficientes en el campo K.

Definición 2.1.17 *Si $p(x)$ es un polinomio no constante, entonces decimos que $p(x)$ es un polinomio* irreducible *o* primo *en $K[x]$, si siempre que $p(x) = q(x)r(x)$ con $q(x), r(x) \in K[x]$, entonces $q(x)$ es constante ó $r(x)$ lo es.*

Todo polinomio de primer grado es irreducible. Por ejemplo, $p(x) = x^2 - 2 \in \mathbb{Q}[x]$ es irreducible en $\mathbb{Q}[x]$.

Definición 2.1.18 *A un campo con un número finito de elementos lo llamamos* campo finito*, o bien,* campo de Galois*. El* orden *de un campo de Galois, $O(K)$, es la cardinalidad del campo.*

Podemos observar que la característica de un campo de Galois es un número primo, por otra parte, el orden de un campo de Galois es una potencia de su característica, es decir: si K es un campo de Galois entonces existe $n \in \mathbb{N}$ tal que $O(K) = (carK)^n$.

Proposición 2.1.19 *Sea K un campo de Galois de $q = p^n$ elementos (p primo). Entonces, K es, salvo isomorfismo, el campo de descomposición del polinomio $p[x] = x^q - x$. Además, K es isomorfo a todo campo de Galois con el mismo número de elementos.*

Proposición 2.1.20 *Dado p un número primo y n un entero positivo, existe un campo de Galois K, tal que $O(K) = p^n$.*

Las demostraciones de estas proposiciones se encuentran en [10].

2.1.2 Cómputo en Campos de Galois

En un campo de Galois, la operaciones suma, resta, multiplicación y división por elementos no cero están bien definidas. Existe el neutro aditivo, el 0, y la unidad, el 1. Cada número no cero posee un inverso el cual es único en el campo. Valen las leyes de asociatividad, conmutatividad y distributividad del producto respecto a la suma.

La aritmética en un campo de Galois es ampliamente usada en Criptografía. Las operaciones de la Teoría de Números son magníficas: poseemos números de tamaño finito, y la división no tiene ningún error de redondeo. Muchos criptosistemas están basados en $\mathbb{F}_p$, donde p es un número primo muy grande.

Para sistemas más complicados, los criptógrafos hacen uso de la aritmética módulo polinomios *irreducibles*, cuyos coeficientes son enteros módulo p, donde p es un primo. Estos campos (como vimos anteriormente) los denotamos por: $\mathbb{F}_q$, donde $q = p^n$, para algún entero $n \in \mathbb{N}$. Hacemos toda la aritmética (de polinomios, suma, resta, multiplicación y división) módulo $p(x)$, donde $p(x)$ es un polinomio irreducible de grado n.

Por ejemplo, el campo de Galois $\mathbb{F}_{2^3}$ que posee a los elementos $0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1$. Las operaciones en tal campo debemos efectuarlas con la aritmética de polinomios, módulo un polinomio irreducible de grado 3 y coeficientes en el campo $\mathbb{F}_2 = \{0, 1\}$. Proponemos al polinomio irreducible como: $p[x] = x^3 + x + 1$ (este polinomio no tiene raíces en el campo $\mathbb{F}_2$, pues $p[0] \neq 0 \neq p[1]$).

Entonces, hablando en términos de polinomios, reemplazamos el término "primo" por "irreducible".

En particular, los cálculos en el campo $\mathbb{F}_{2^n}$ son efectuados rápidamente haciendo la implementación en hardware y eligiendo polinomios irreducibles adecuados. Para campos de este tipo en criptografía se han utilizado polinomios de la forma $p(x) = x^n + x + 1$ como módulo, cuando éste es irreducible en $\mathbb{F}_2$, ya que la cadena larga de ceros entre su coeficiente principal y el del término lineal hace fácil la implementación de la multiplicación rápida.

2.2 Teoría de Matroides

En esta sección nos encargamos de mostrar las definiciones elementales sobre la Teoría de Matroides (para más detalle véase en [17] y [23]).

2.2.1 Definiciones y Resultados Básicos

El nombre de matroide sugiere una estructura relativa a las matrices. Los matroides fueron introducidos por Witney en 1935 y proveen de un tratamiento abstracto de la Independencia Lineal de Álgebra y Teoría de Gráficas.

Definición 2.2.1 *Un* matroide $\mathcal{M}$ *es una estructura combinatoria* $(E, \mathcal{I})$*, donde* E *es un conjunto de* n *elementos,* $\mathcal{I} \subset \mathcal{P}(E)$ *(el conjunto potencia de* E*, o las partes de* E*), a cuyos elementos los conocemos como* conjuntos independientes*, mismos que satisfacen:*

1. $\emptyset \in \mathcal{I}$.
2. *Si* $X \in \mathcal{I}$ *y* $Y \subset X$ *entonces* $Y \in \mathcal{I}$.

3. Si $X, Y \in \mathcal{I}$ tales que $|X| = |Y|+1$, entonces existe $x \in X - Y$ tal que: $Y \cup \{x\} \in \mathcal{I}$.

Un *conjunto dependiente* en un matroide es cualquier subconjunto de E que no es independiente (que no pertenece a $\mathcal{I}$).

Ejemplo 2.2.2 *Sea E un conjunto que contiene n elementos e $\mathcal{I}$ la colección que consiste de todos los subconjuntos de E, que poseen a lo más k elementos para algún $k \leq n$. Claramente $\mathcal{I}$ satisface los 3 axiomas de la definición 2.2.1. Por lo tanto $(E, \mathcal{I})$ es un matroide conocido como el* matroide uniforme *$U_{k,n}$.*

Ejemplo 2.2.3 *Consideremos a la matriz:*

$$A = \begin{pmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & 1 \end{pmatrix},$$

definida sobre el campo de los números reales $\mathbb{R}$. Sea $E = \{1, 2, 3, 4\}$ el conjunto de índices que utilizamos para enumerar a las columnas de A. Entonces $\mathcal{I}$ es la colección de subconjuntos $X \subset E$, para los cuales el conjunto de vectores columna de A (con índices en E) es un conjunto linealmente independiente (l. i.) en $\mathbb{R}^2$. Por lo tanto:

$$\mathcal{I} = \{\emptyset, \{1\}, \{3\}, \{4\}, \{1, 3\}, \{1, 4\}, \{3, 4\}\}.$$

Entonces la pareja $(E, \mathcal{I})$ es un matroide.

Definición 2.2.4 *Dos matroides $\mathcal{M}_1 = (E_1, \mathcal{I}_1)$ y $\mathcal{M}_2 = (E_2, \mathcal{I}_2)$ se dicen ser* isomorfos*, y se escribe $\mathcal{M}_1 \simeq \mathcal{M}_2$, si existe una biyección $\psi : E_1 \to E_2$ tal que, para todo $X \subset E_1$, $\psi(X)$ es un conjunto independiente en E_2 si y sólo si X es un conjunto independiente en E_1.*

En espacios vectoriales tenemos la noción de "subespacio vectorial". En los matroides definimos el término *submatroide*, que consiste de la pareja $(E, \mathcal{I}')$, donde $\mathcal{I}'$ es una subcolección de $\mathcal{I}$, que a su vez es un matroide.

El siguiente resultado es una generalización de la definición de matroide y se usa extensivamente en la literatura.

Teorema 2.2.5 (El teorema de aumento) *Sea $(\mathcal{M}, \mathcal{I})$ un matroide, y $X, Y \in \mathcal{I}$ tales que $|X| < |Y|$ (aquí, $|X|$ denota la* cardinalidad *de X). Entonces existe $Z \subset Y - X$ tal que $|X \cup Z| = |Y|$ y $X \cup Z \in \mathcal{I}$.*

Una consecuencia inmediata de este teorema es el siguiente resultado, que es análogo a la propiedad de bases de un espacio vectorial, pues los matroides son obtenidos de manera única mediante "conjuntos básicos" como mostramos enseguida.

Sea $\mathcal{B}$ la colección que coincide con los conjuntos maximales $B \in \mathcal{I}$. Los elementos de $\mathcal{I}$ que no están en $\mathcal{B}$, son precisamente los conjuntos independientes $A \subset B\, (A \neq B)$, tales que $B \in \mathcal{B}$.

Definición 2.2.6 *A los elementos $B \in \mathcal{B}$ se les llamamos las* bases *del matroide $\mathcal{M}$.*

A partir de aquí, cuando hagamos referencia a un matroide nos referiremos indistintamente a la colección $\mathcal{I}$ ó $\mathcal{I}'$, en caso de ser necesario, haremos la aclaración correspondiente.

Teorema 2.2.7 (Véase [17]) *La colección $\mathcal{B}$ de un matroide $\mathcal{M}$ satisface las siguientes condiciones:*

1. *$\mathcal{B} \neq \emptyset$.*
2. *Si $B_1, B_2 \in \mathcal{B}$ entonces $\mid B_1 \mid = \mid B_2 \mid$, es decir, las bases de un matroide tienen todas una misma cardinalidad, digamos r. El número r se llama el* rango *del matroide $\mathcal{M}$.*

Por otra parte, dado un matroide $\mathcal{M} = (E, \mathcal{I})$ definimos también el *rango* del matroide $\mathcal{M}$ como una función $r : \mathcal{P}(E) \to \mathbb{Z} \cup \{0\}$ (donde $\mathcal{P}(E)$ es el conjunto potencia de E) que satisface:

1. $r(\emptyset) = 0$.
2. Si $X \subset E$ y $x \in E$, entonces $r(X) \leq r(X \cup x) \leq r(X) + 1$.
3. Si $X \subset E$ y $x, y \in E$ son tales que si $r(X \cup x) = r(X \cup y) = r(X)$, entonces $r(X \cup \{x, y\}) = r(X)$.

El siguiente teorema proporciona una caracterización de las bases de los matroides:

Teorema 2.2.8 (Véase [17]) (Axiomas de Bases) *Una colección no vacía de $\mathcal{B}$ de subconjuntos de E es el conjunto de bases de un matroide sobre E si y sólo si se satisface la siguiente condición:*

- *Para cada $B_1, B_2 \in \mathcal{B}$ existe un elemento $x \in (B_1 - B_2)$ y un elemento $y \in (B_2 - B_1)$, tales que $[(B_1 \cup \{y\}) - \{x\}] \in \mathcal{B}$.*

Con este teorema, se puede obtener un nuevo axioma de la definición de matroide:

Teorema 2.2.9 *Una colección $\mathcal{I}$ de subconjuntos de E es la colección de conjuntos independientes de un matroide sobre E, si y sólo si $\mathcal{I}$ satisface las condiciones (I1) y (I2) de la definición 2.2.1 e (I3) se reemplaza por la condición:*
(I3′) Para cada $A \subset E$, todos los subconjuntos maximales $Y \subset A$ (con $Y \in \mathcal{I}$) tienen la misma cardinalidad.

2.2.2 Matroides Representables

Observemos el siguiente ejemplo:

Ejemplo 2.2.10 *Sean $n, r \in \mathbb{N}$ y $q = p^r$ (p un número primo). $\mathbb{F}_q^n$ es el espacio vectorial de dimensión n, sobre el campo de Galois $\mathbb{F}_q$. Sea $P \subset \mathbb{F}_q^n$ y sea $V < \mathbb{F}_q^n$ el subespacio vectorial generado por los vectores en P. Sea $m \leq n$ la dimensión de V. Consideremos la colección:*

$$\mathcal{L}_V = \{L \subset V \mid L \text{ es l. i. en } V\}, \tag{2.2}$$

entonces la pareja $(\mathbb{F}_q^n, \mathcal{L}_V)$ forma un matroide que se denota por $\mathcal{B}_{m,n,q}$.

Para los fines de este trabajo, necesitamos una definición muy importante sobre la estructura de matroide.

Definición 2.2.11 *Un matroide $\mathcal{M} = (E, \mathcal{I})$ se dice ser* representable *sobre un campo finito $\mathbb{F}_q$, si existen números $q = p^r$ (p primo), $n \in \mathbb{N}$, y $m \leq n$, tales que $\mathcal{M}$ es isomorfo a un matroide de la forma $\mathcal{B}_{m,n,q}$ (definido en el ejemplo 2.2.10).*

El rango de un matroide representable $\mathcal{B}_{m,n,q}$ es m, pues las bases de este matroide poseen m elementos.

Por otra parte cabe mencionar que los matroides representables también son utilizados en la elaboración de códigos lineales [17]; dado un código lineal, es posible obtener un matroide representable; y de un matroide representable es posible obtener una familia de códigos.

2.2.3 Circuitos de un Matroide

Definición 2.2.12 *Sea $\mathcal{M} = (E, \mathcal{I})$ un matroide. Un* circuito *es un subconjunto de E, cuyos subconjuntos propios son subconjuntos independientes.*

Una definición equivalente para los circuitos es la siguiente:

Definición 2.2.13 *Sea $\mathcal{M} = (E, \mathcal{I})$ un matroide. Un conjunto minimal dependiente en $\mathcal{M}$ se conoce como un* circuito *de $\mathcal{M}$.*

Con la siguiente proposición proporcionamos una caracterización para la colección de los circuitos de un matroide $\mathcal{M}$.

Proposición 2.2.14 (Véase [17, 23]) *Una colección finita $\mathcal{C}$ de subconjuntos de un conjunto E, es la colección de circuitos de un matroide $\mathcal{M} = (E, \mathcal{I})$ si y sólo si se satisfacen las siguientes condiciones:*

1. *$\emptyset \notin \mathcal{C}$.*
2. *Ningún elemento C de $\mathcal{C}$ está contenido en otro C' de $\mathcal{C}$.*
3. *Si C y C' son elementos distintos de $\mathcal{C}$ que contienen a un elemento en común e, entonces algún subconjunto de $(C \cup C') - \{e\}$, es un elemento de $\mathcal{C}$.*

La condición (3) se llama la *condición de eliminación de circuitos.* La siguiente proposición describe una condición más "fuerte" que la número (3) anterior pero que resultan ser equivalentes.

Proposición 2.2.15 (Véase [17, 23]) *Una colección finita $\mathcal{C}$ de subconjuntos de un conjunto finito E es la colección de circuitos de un matroide $\mathcal{M} = (E, \mathcal{I})$ si y sólo si se satisfacen las siguientes condiciones:*

1. *$\emptyset \notin \mathcal{C}$.*
2. *Ningún elemento C de $\mathcal{C}$ está contenido en otro C' de $\mathcal{C}$.*
3. *Si C y C' son elementos distintos de $\mathcal{C}$ que contienen a un elemento en común e, y f es un elemento de C que no está en C', entonces algún subconjunto de $(C \cup C') - \{e\}$, es un elemento de $\mathcal{C}$ que contiene a f.*

Si queremos obtener a los elementos independientes de $\mathcal{I}$ de un matroide $\mathcal{M}$, a partir de los circuitos $\mathcal{C}$ del matroide, observamos que $A \subset E$ es independiente si y sólo si no contiene a ningún circuito.

También observamos que: $\mathcal{C}(M) = \emptyset$ si y sólo si $M \cong U_{n,n}$.

2.2.4 Teoría de Gráficas

En esta sección proporcionamos las definiciones y resultados básicos de la Teoría de Gráficas y su relación con la Teoría de Matroides.

Definición 2.2.16 *Una* gráfica simple G *es una pareja* (V, A) *donde* V *es un conjunto no vacío a cuyos elementos llamamos* vértices *y* A *es una colección de subconjuntos de dos elementos distintos de* V*, a los que llamamos* aristas *o* lados. *El* orden *de* G *es el número de vértices y lo denotamos por* $v(G)$. *El* tamaño *de* G *es el número de aristas y denotamos por* $e(G)$.

Denotaremos a una arista $\{x, y\} \in A$ por $xy \in A$. El tamaño de una gráfica de orden n tiene la propiedad $0 \leq e(G) \leq \binom{n}{2}$.

Definición 2.2.17 *Una gráfica de orden* n *y tamaño* $\binom{n}{2}$ *se llama* n-gráfica completa *y se denota por* K_n.

Intuitivamente una gráfica completa K_n es la gráfica en la que cualesquiera dos vértices están unidos por una arista.

Por ejemplo, K_0 es la gráfica de orden $v(G) = 1$ y tamaño $e(G) = 0$. Es decir, es la gráfica de un solo vértice. K_2 es la gráfica de orden $v(G) = 2$ y tamaño $e(G) = 1$, la única arista une a los dos vértices. K_3 es la gráfica de orden $v(G) = 3$ y tamaño $e(G) = 3$. Gráficamente K_3 corresponde a un triángulo.

Definición 2.2.18 *Sea* $G = (V, A)$ *una gráfica. La* gráfica complementaria *de* G *es* $G^c = (V, V^2 - A)$, *donde* $V^2 = \{U \subset V : |U| = 2\}$. *Observamos también que el número de aristas de* G^c *es* $\binom{n}{2} - e(G)$.

Definición 2.2.19 *Un* n*-clan es una gráfica completa con* n *vértices.*

Definición 2.2.20 *Dos vértices* $x, y \in V$ *se dicen ser* adyacentes*, si* xy *es una arista de* G*, es decir si* $\{x, y\} \in A$. *Dos aristas se dicen ser* adyacentes*, si tienen un vértice en común. Parejas de aristas o vértices no-adyacentes se dicen ser* independientes.

Definición 2.2.21 *Sean* $G = (V, A)$ *y* $G' = (V', A')$ *dos gráficas. Decimos que* G *y* G' *son* gráficas isomorfas *si existe una función biyectiva* $\phi : G \to G'$ *con* $xy \in A \Leftrightarrow \phi(x)\phi(y) \in A'$ *y escribimos* $G \cong G'$. ϕ *es llamado un* isomorfismo *entre* G *y* G'.

Definición 2.2.22 *Una gráfica acíclica (es decir que no contiene ciclos), se conoce como* bosque. *Un bosque conexo se conoce como un* árbol *(además, los vértices en los cuales incide una sola arista, se les conoce como* hojas*).*

Entonces es fácil ver que un bosque es una gráfica cuya componentes son árboles.

Ejemplo 2.2.23 *Sea G una gráfica con conjunto de vértices $V \neq \emptyset$. Si definimos*

$$\mathcal{I} = \{I \subset V : I \text{ induce una gráfica acíclica en } G\}$$

entonces $(V, \mathcal{I})$ es un matroide que se denota por $\mathcal{M}(G)$, y se llama el matroide cíclico *de G. Cualquier matroide isomorfo a un matroide gráfico $\mathcal{M}(G)$ se conoce como* matroide gráfico.

En Combinatoria se demuestra que si $\mathcal{M}$ es un matroide gráfico, podemos encontrar una gráfica conexa G tal que $\mathcal{M} \simeq \mathcal{M}(G)$ [17, 8].

Otro resultado interesante es que si dado un matroide gráfico $\mathcal{M}(G)$ entonces un subconjunto $X \subset E$ es una base si y sólo si el bosque generado por X es maximal y G es un conjunto conexo.[17, 8].

2.2.5 Descomposición en Sumas

En esta sección mostramos un resultado que la Teoría de Matroides que se conoce como la *descomposición de matroides*. Dado un matroide $\mathcal{M} = (E, \mathcal{I})$, podemos obtener una partición del conjunto $E = \cup_i E_i$ (con $E_i \neq \emptyset$), la restricción $\mathcal{M}$ cada uno de los elementos de la partición resulta ser un matroides que es uniforme. Tal partición nos ayuda a definir la EA del sistema que proponemos.

En primer lugar definimos a la relación $\sim$ sobre el conjunto E, como sigue: $\forall e, e' \in E$,

$$e \sim e' \Leftrightarrow \exists\, C \in \mathcal{C} : e, e' \in C. \tag{2.3}$$

Esto quiere decir que dos elementos de E están relacionados si y sólo si, pertenecen a un mismo circuito.

Proposición 2.2.24 *La relación $\sim$ definida anteriormente es una relación de equivalencia.*

La relación $\sim$ induce una partición del conjunto E en clases de equivalencia, digamos $E_1, \ldots, E_N$. El matroide $\mathcal{M}$ restringido a cada clase de equivalencia, forma un matroide que es uniforme. A tales matroides las conocemos como las *componentes* del matroide $\mathcal{M}$. El matroide $\mathcal{M}$ se dice ser *conexo* si sólo tiene una componente, y se llama *disconexo* si $\mathcal{M}$ posee más de una componente.

Teorema 2.2.25 (Véase [17]) *Sea $\mathcal{M} = (E, \mathcal{I})$ un matroide. Si las clases de equivalencias son E_1, E_2, $\ldots$, E_N, entonces:*

$$\mathcal{M} = (\mathcal{M} \mid_{E_1}) \oplus (\mathcal{M} \mid_{E_2}) \oplus \ldots \oplus (\mathcal{M} \mid_{E_N}).$$

2.3 Esquemas de Compartición de Secretos

En esta sección proporcionamos un estudio sobre los Esquemas de Compartición de Secretos (que denotamos de aquí en adelante por *ECS*). Los protocolos de compartición de secretos fueron creados de manera independiente por Blakley [4] y Shamir [20].

La motivación principal de la compartición de secretos ha sido que en algunas situaciones existe una llave secreta que provee acceso a muchos archivos de actividades muy importantes.

Los ECS surgen de la necesidad del manejo de llaves robusto. Dada una llave criptográfica queremos generalizar la noción de control de acceso fragmentado. Podemos usar el término ECS pues esto enfatiza la propiedad de fragmentación, a diferencia de los esquemas de umbral creados inicialmente y que son más restrictivos, pues requieren la concurrencia de un número fijo de participantes.

La construcción de los esquemas de Blakley y Shamir realizan esquemas de umbrales k-de-entre-n, más aún podemos mencionar que sus construcciones son fundamentalmente distintas.

Si la persona que conoce la llave la pierde, o bien se borra o destruye la memoria de la computadora que almacena la llave, entonces los archivos importantes se vuelven inaccesibles. Por esto, se requiere una forma de evitar esta situación. Un método consiste de tener muchas copias de la llave y distribuirlas a mucha gente, o almacenarlas en muchos lugares. Esto no es deseable porque cuanto mayor es el número de copias, mayor es el riesgo que implica en la seguridad.

Los ECS tratan este problema y nos permiten tener confiabilidad sin riesgo creciente, como veremos más adelante.

Definición 2.3.1 *Sea P un conjunto finito de n* participantes*. Un elemento adicional, que denotamos por $\mathfrak{p}_0 \notin P$ es el* distribuidor*, es quien se encarga de fragmentar y distribuir un secreto.*

Dentro de un ECS deseamos determinar a la colección de subconjuntos de P que pueden calcular el secreto. Esta colección debe satisfacer algunas propiedades básicas, como la que enseguida definimos.

Definición 2.3.2 *Una colección $\mathcal{A} \subset \mathcal{P}(P)$ es* monótona *si para todo $B \in \mathcal{A}$, siempre que $B \subset C$, se tiene que $C \in \mathcal{A}$.*

Definición 2.3.3 *Una* estructura de acceso (EA) *es una colección monótona de subconjuntos no vacíos de P a la que denotamos por Γ. Cada elemento de $\mathcal{A}$ es llamado* conjunto de acceso *(o conjunto* reconstructible*).*

Consideremos entonces un conjunto de participantes P, un distribuidor $\mathfrak{p}_0$ y un conjunto finito de secretos $S = \{0, 1, \ldots, m-1\}$ (también conocido como *dominio de secretos*). Sea $\Gamma \subset \mathcal{P}(P)$ una EA y sea R un conjunto finito de entradas aleatorias de tal forma que $\{\mu_s : R \to [0,1]\}_{s \in S}$ es un conjunto de distribuciones de probabilidad. Entonces:

Definición 2.3.4 *Un* esquema de compartición de secretos (ECS) Π*, con dominio de secretos S es una función $\Pi : (S \times R) \to (S_1 \times S_2 \times \ldots \times S_n)$, del producto de los secretos y las entradas aleatorias, a una n-ada (donde cada entrada es un* fragmento *o* parte*, que toma valor en un dominio de fragmentos). Denotamos al fragmento del i-ésimo participante por $\Pi_i(s, r)$ (la proyección en la i-ésima coordenada), además el ECS Π y la EA Γ satisfacen:*

Podemos reconstruir el secreto s con cualquier conjunto en Γ. Esto es, para cualquier subconjunto $B \in \Gamma$ ($B = \{i_1, \ldots, i_{|B|}\}$), existe una función $h_B : S_{i_1} \times \cdots \times S_{i_{|B|}} \to S$, de modo que para cualquier entrada aleatoria r, si $\Pi(s, r) = (s_1, s_2, \ldots, s_n)$, entonces $h_B(s_{i_1}, \ldots, s_{i_{|B|}}) = s$.

Un ejemplo de un ECS es el siguiente:

Ejemplo 2.3.5 *Consideremos como secreto al número $s \in \mathbb{Z}_p$. Entonces veremos a s como un secreto que deseamos compartir entre k participantes. El proceso de fragmentación por parte del distribuidor es el siguiente:*

- $\mathfrak{p}_0$ *selecciona aleatoriamente $k-1$ números $a_1,\ a_2,\ \ldots, a_{k-1} \in \mathbb{Z}_p$.*
- $\mathfrak{p}_0$ *asigna a cada participante $p_i,\ 1 \leq i \leq k$, el número $s_k = s - \sum_{i=1}^{k-1} a_i \ mod\ p$.*

Para poder recobrar el secreto s, es necesario que todos los participantes se reúnan y coloquen sus fragmentos ($s = \sum_{i=1}^{k} s_i \ mod\ p$).

Definición 2.3.6 *Un ECS se dice ser* perfecto *(con respecto a la EA Γ) si se cumplen las siguientes dos condiciones:*

1. *Si los participantes de un subconjunto $B \in \Gamma$ reúnen sus fragmentos, entonces podemos determinar el secreto.*
2. *Si $B \notin \Gamma$, entonces los participantes en B no pueden determinar ninguna información sobre el secreto. De manera formal:*

$$\forall b \in B,\ \ p(s|b) = p(s),$$

esto dice que s y B son estadísticamente independientes, es decir, la probabilidad de "adivinar" el secreto s, dado que se conoce el fragmento b, es simplemente $p(s)$ (la probabilidad de adivinar el secreto).

Ejemplo 2.3.7 *Un esquema de (n, k)-umbral tiene como estructura de acceso al conjunto:*

$$\Gamma = \{B \subset P : |B| \geq k\}.$$

Definición 2.3.8 *$B \in \Gamma$ es un conjunto de acceso* minimal *si para todo $A \notin \Gamma$ cuando $A \subset B$ entonces $A \neq B$. Denotamos por Γ_0 a la colección de conjuntos de acceso minimales de Γ y a sus elementos los llamamos* bases de Γ.

También definimos la cerradura *de una colección de conjuntos de P, digamos Λ, como:*

$$cl(\Lambda) = \{C \subset P | B \subset C,\ B \in \Lambda\}.$$

Si Γ es monótona, la cerradura de Γ_0 resulta ser Γ, pues:

$$\Gamma = cl(\Gamma_0) = \{C \subset P : B \subset C,\ B \in \Gamma_0\}.$$

Como los ECS juegan un rol muy importante en el campo de distribución de llaves y computación multiparte, existe un número muy grande de esquemas introducidos en las ultimas dos décadas. Listamos algunos de estos esquemas a continuación:

1. Esquemas de Umbral, véase [21, 14].
2. Esquemas de Estructura de Acceso General, véase [14].
3. Compartición de Secretos Verificable, véase [9, 14].
4. ECS Perfectos, véase [1, 14].
5. Compartición de Secretos Proactivos, véase [14].

Los esquemas de umbral son básicamente los introducidos por Shamir (mediante Interpolación de Polinomios) y Blakley (el cual se basa en la Geometría Proyectiva).

En los ECS de estructura de acceso general, sólo los miembros de la EA pueden reconstruir el secreto. Este esquema es más general que cualquier otro. En particular, podemos representar a los esquemas de umbral como una EA general.

Los ECS verificables son aquellos en los cuales se prueba la validez del fragmento.

Los ECS proactivos son esquemas en los cuales los fragmentos son actualizados por los participantes para evitar ataques móviles. La motivación principal se debe a que en la compartición de secretos normal, sólo distribuimos una vez los fragmentos del secreto y permanecen fijos en adelante. Los ECS proactivos también se caracterizan por el hecho de que la EA permanece igual antes de y después de la actualización.

Podemos ver la relación entre los diferentes esquemas la mostramos en la figura 2.1. La línea punteada indica esquemas que son derivados de los ECS más generales.

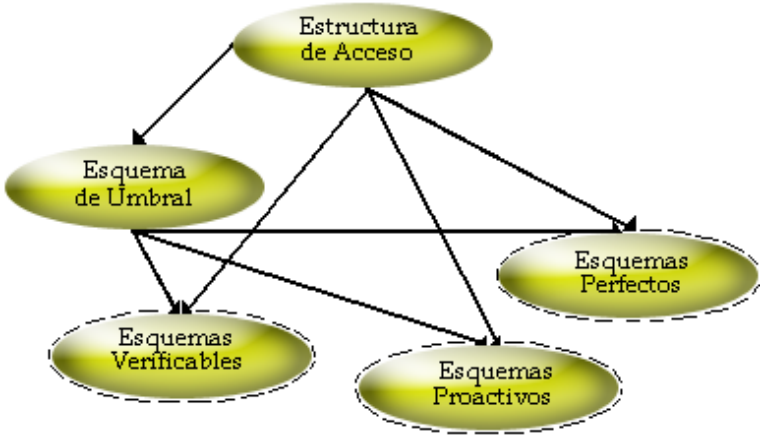


Figura 2.1: Relación entre diversos ECS

Existen diversas caracterizaciones de cada esquema de acuerdo a sus propiedades. Entonces a cada una de ellas, las podemos clasificar bajo una o más de las siguientes categorías con respecto a sus características (véanse [14, 21, 16]).

1. Los ECS basados en las técnicas usadas, podemos clasificarlos en esquemas Geométricos. Por ejemplo, el Esquema de Blakley.
2. Los ECS basados en la complejidad computacional, existen dos categorías: Teoría de la Información y Teoría de Números. Por ejemplo, el esquema (t, n)-umbral de Shamir (Véase [22]).

3. Los ECS basados en la perfección: los ECS perfectos y los no perfectos. Por ejemplo los esquemas umbral de Shamir o Blakley son esquemas perfectos.

4. Los ECS verificables y los no verificables, que están basados en el aspecto de la verificación de la validez de los fragmentos.

2.3.1 Esquemas Ideales

En esta sección introduciremos la noción de ECS *ideales*.

Definición 2.3.9 *Para un ECS, la* tasa de información ρ_{p_i} *para un participante* $p_i \in P$ *es el cociente:*

$$\rho_{p_i} = max\left\{\frac{\log_2 |s|}{\log_2 |s_{p_i}|} : s \in S\right\}, \tag{2.4}$$

donde $|s|$ *es el tamaño del secreto compartido,* S *es el dominio de secretos y* $|s_{p_i}|$ *es el tamaño del fragmento del participante* p_i*. Definimos a la* tasa de información del esquema *como*

$$\rho = min\{\rho_{p_i} : 1 \leq p_i \leq \mid P \mid\}.$$

Definición 2.3.10 *Los ECS se dicen* ideales *si:* $\rho = 1$.

Ejemplo 2.3.11 *El esquema de Umbral de Shamir (que mostraremos en detalle en la siguiente sección), es un esquema ideal.*

Teorema 2.3.12 (Véase [16]) *En cualquier ECS perfecto, todos los participantes tienen un fragmento más grande que el secreto mismo, es decir,* $\rho_{p_i} \leq 1$, $p_i \in P$.

Definición 2.3.13 *Sea* $\mathcal{M} = (E, \mathcal{I})$ *un matroide y sea* Γ *una estructura de acceso sobre el conjunto* E*. Decimos que el matroide* $\mathcal{M}$ *es* apropiado *para la estructura* Γ *si:*

$$\Gamma = \{A \subset E : A \notin \mathcal{I}\}.$$

2.3.2 Esquema de Shamir

El ECS de Shamir es un tipo de esquema de umbral, el cual está basado en la interpolación de polinomios.

Consideramos a un polinomio de grado $t-1$ sobre el campo finito K :

$$p[x] = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}, \tag{2.5}$$

este polinomio lo construimos de tal manera que el coeficiente a_0 es el secreto y los demás coeficientes son elementos aleatorios en el campo K.

Cada uno de los n fragmentos serán los puntos $(x_i, p[x_i])_{1 \leq i \leq n}$, donde:

$$x_i \neq x_j \text{ para todo } i \neq j \text{ y } p[0] = a_0 \ . \tag{2.6}$$

Tales puntos son elementos de la curva definida por el polinomio anterior sobre el plano $\mathbb{R}^2$ (figura 2.2).

Dados t fragmentos, podemos determinar de manera única al polinomio, y en consecuencia, podemos calcular a a_0 (el secreto) mediante la sustitución $x = 0$ en la función polinomial.

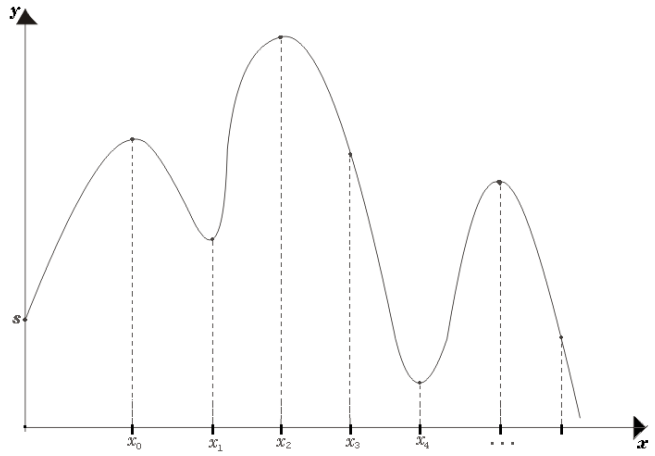


Figura 2.2: Los fragmentos del esquema de Shamir en el plano $\mathbb{R}^2$

Ejemplo 2.3.14 *El caso especial en que $t = 2$, sólo requerimos dos fragmentos para recobrar el secreto. En consecuencia, la ecuación del polinomio describe una línea recta. El secreto es el punto en el que la línea intersecta al eje y. Ilustramos esto en la figura 2.3.*

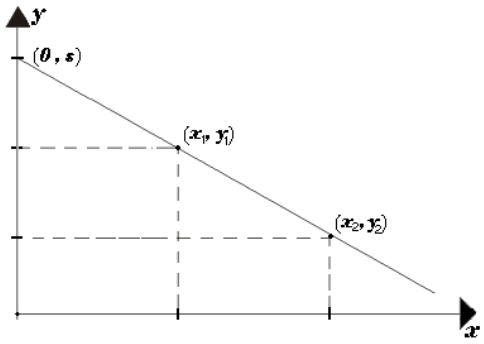


Figura 2.3: El esquema de Shamir con umbral $t = 2$

El esquema de Shamir está basado en la interpolación de polinomios por el hecho de que el polinomio invariante $p[x]$ de grado $t-1$ está determinado de manera única por t puntos (x_i, s_i) con distintos x_i pues estos puntos definen t ecuaciones independientes con t incógnitas en los coeficientes a_i.

En los algoritmos 1, 2 y 3 describimos las fases del mecanismo de construcción del esquema umbral de Shamir.

Algoritmo 1 Fase Inicial del distribuidor $\mathfrak{p}_0$

Require: $s \geq 0$, el secreto y t es el umbral.
Ensure: a_i los coeficientes del polinomio.

1: $\mathfrak{p}_0$ elige un primo $p > max(s, n)$ y define $a_0 = s$;
2: $\mathfrak{p}_0$ selecciona aleatoriamente $(t-1)$ coeficientes $a_1, \ldots, a_{t-1}$, con $0 \leq a_i \leq p-1$, que define al polinomio aleatoriamente sobre $\mathbb{F}_p$.

Algoritmo 2 Distribución de los fragmentos

Require: El polinomio $p[x]$ e índices x_i para los participantes.
Ensure: $(x_i, s_i = p[x_i])$ los fragmentos para cada participante.
1: $\mathfrak{p}_0$ calcula $s_i = P[x_i] \bmod(\mathbb{F}_p)$ para cada participante x_i, y hace la transferencia segura del fragmento con índice público x_i.

Algoritmo 3 Reunión de los fragmentos

Require: Un grupo de t o más fragmentos $(x_i, s_i)_{1\leq t}$.
Ensure: El secreto s.
1: Calculamos los coeficientes a_j, $1 \leq j \leq t-1$ del polinomio $p[x]$ mediante interpolación de Lagrange. Recuperamos el secreto sustituyendo $x = 0$ en $p[x]$, es decir $p[0] = a_0 = s$.

A continuación mostramos un ejemplo de este esquema:

Ejemplo 2.3.15 *Sean $p = 23$, $t = 3$, $n = 6$ y las coordenadas públicas $x_i = i$ para cada participante p_i, para $i = 1, \ldots, 6$.*

Supongamos que los participantes de $B = \{p_1, p_3, p_5\}$ reúnen sus fragmentos, los cuales son respectivamente $6, 16$ y 11. Escribimos al polinomio como:

$$p[x] = a_0 + a_1 x + a_2 x^2, \tag{2.7}$$

calculando $p[1], p[3]$ y $p[5]$, obtenemos la siguientes tres ecuaciones lineales en $\mathbb{F}_{23}$

$$a_0 + a_1 + a_2 = 06$$

$$a_0 + 3a_1 + 9a_2 = 16$$

$$a_0 + 5a_1 + 25a_2 = 11.$$

Este sistema posee solución única $a_0 = 4, a_1 = 1$ y $a_2 = 1$ (el secreto compartido es $a_0 = 4$).

Para demostrar que el esquema de Shamir es perfecto, necesitamos que el sistema de t ecuaciones linealmente independientes siempre tenga una única solución. Efectivamente esto es cierto.

En general tenemos dados t fragmentos:

$$s_{i_k} = p[x_{i_k}],\ 1 \leq k \leq t,$$

donde:

$$p[x] = a_0 + a_1 x + \ldots + a_{t-1} x^{t-1},\ \text{con } a_0 = s,$$

el sistema de ecuaciones lineales en $\mathbb{F}_p$ es el siguiente:

$$\begin{array}{rcl} a_0 + a_1 x_{i_1} + a_2 x_{i_1}^2 + & \ldots + a_{t-1} x_{i_1}^{t-1} & = s_{i_1} \\ a_0 + a_1 x_{i_2} + a_2 x_{i_2}^2 + & \ldots + a_{t-1} x_{i_2}^{t-1} & = s_{i_2} \\ \vdots & \vdots & \\ a_0 + a_1 x_{i_t} + a_2 x_{i_t}^2 + & \ldots + a_{t-1} x_{i_t}^{t-1} & = s_{i_t} \end{array} \tag{2.8}$$

Este sistema visto en un sistema matricial es:

$$\begin{pmatrix} 1 & x_{i_1} & x_{i_1}^2 & \dots & x_{i_1}^{t-1} \\ 1 & x_{i_2} & x_{i_2}^2 & \dots & x_{i_2}^{t-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{i_t} & x_{i_t}^2 & \dots & x_{i_t}^{t-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} s_{i_1} \\ s_{i_2} \\ \vdots \\ s_{i_t} \end{pmatrix} \tag{2.9}$$

La matriz de la izquierda (que denotamos por A) se conoce como la *matriz de Vandermonde.* Para tal matriz existe una fórmula conocida para su determinante:

$$det(A) = \Pi_{1 \leq j < k \leq t}(x_{i_k} - x_{i_j}) mod\ p.$$

Como los x_{i_k} son distintos entre sí, los términos $(x_{i_k} - x_{i_j})$ son distintos de cero. Por lo tanto $det(A) \neq 0$. Esto indica que el sistema 2.8 posee solución única en el campo $\mathbb{F}_p$.

Esto demuestra que cualquier grupo de t participantes puede recuperar el secreto. Por otro lado, cuando $t-1$ participantes reúnen sus fragmentos, obtienen un sistema de $t-1$ ecuaciones. Es claro que existe una solución no única en el campo $\mathbb{F}_p$. Por lo tanto el esquema de Shamir es perfecto.

Hasta ahora hemos analizado el esquema de Shamir desde el punto de vista de resolución de un sistema de ecuaciones lineales sobre el campo $\mathbb{F}_p$. Existe un método para la resolución de este sistema basado en los polinomios de Lagrange. El método de interpolación de Lagrange da una fórmula explícita para el polinomio de grado a lo más $t-1$ como sigue: los coeficientes de tal polinomio $p[x]$ de grado menor que t, definido por los puntos (x_i, s_i), $1 \leq i \leq t$ será:

$$p[x] = \sum_{i=1}^{t} s_i \prod_{1 \leq j \leq t, j \neq i} \frac{x - x_j}{x_i - x_j}. \tag{2.10}$$

Dado que el secreto es $p[0]$, si evaluamos $x = 0$ en la ecuación 2.10 obtenemos:

$$p[0] = s = \sum_{i=1}^{t} c_i s_i, \tag{2.11}$$

donde:

$$c_i = \prod_{1 \leq j \leq t, j \neq i} \frac{x_j}{x_j - x_i}. \tag{2.12}$$

Observemos que los c_i son constantes para un grupo fijo de t participantes. Ilustremos esto con un ejemplo:

Ejemplo 2.3.16 *Consideremos el ejemplo 2.3.15. Para los participantes* $\{p_1, p_3, p_5\}$*, tenemos los fragmentos* $6, 16$ *y* 11 *respectivamente. Entonces los índices asociados a estos participantes son,* $x_1 = 1$ *(para el participante* p_1*),* $x_2 = 3$ *(para el participante* p_3*),* $x_3 = 5$

(para el participante p_5). De acuerdo a la fórmula 2.12, obtenemos:

$$\begin{aligned} c_1 &= \prod_{1\leq j\leq 3, j\neq 1} \frac{x_j}{x_j - x_1} \; mod\; 23 \\ &= \frac{x_2}{x_2 - x_1}\frac{x_3}{x_3 - x_1} \; mod\; 23 \\ &= \frac{3\cdot 5}{(2)(4)} \; mod\; 23 \\ &= \frac{15}{8} \; mod\; 23 \\ &= (15)(3) \; mod\; 23 \\ &= 45 \; mod\; 23 \\ &= 22, \end{aligned}$$

$$\begin{aligned} c_2 &= \frac{x_1 \cdot x_3}{(x_1 - x_2)(x_3 - x_2)} \; mod\; 23 \\ &= \frac{5}{-4} \; mod\; 23 \\ &= (5)(17) \; mod\; 23 \\ &= 16, \end{aligned}$$

$$\begin{aligned} c_3 &= \frac{x_1 \cdot x_2}{(x_1 - x_3)(x_2 - x_3)} \; mod\; 23 \\ &= \frac{3}{8} \; mod\; 23 \\ &= (3)(3) \; mod\; 23 \\ &= 9. \end{aligned}$$

Con tales coeficientes, y de acuerdo a la ecuación 2.11, el secreto es:

$$\begin{aligned} s = p[0] &= \sum_{i=1}^{3} c_i \; s_i \; mod\; 23 \\ &= c_1 \cdot s_1 + c_2 \cdot s_2 + c_3 \cdot s_3 \; mod\; 23 \\ &= 22 \cdot 6 + 16 \cdot 16 + 9 \cdot 11 \; mod\; 23 \\ &= 487 \; mod\; 23 \\ &= 4. \end{aligned}$$

Propiedades de Esquema de Shamir

Como el esquema umbral de Shamir es mundialmente reconocido posee aplicaciones muy importantes. De hecho, el esquema de Shamir es la base de casi todos los esquemas umbral existentes. En seguida listamos las propiedades más importantes que posee el esquema de Shamir (y en general, que cualquier esquema umbral posee).

1. *Es perfecto:* si conocemos $t-1$ o menos fragmentos, la probabilidad de encontrar el valor para el secreto compartido en el campo $\mathbb{F}_p$, es siempre la misma (ver definición 2.3.6).

2. *Es ideal:* Como los fragmentos y el secreto pertenecen al campo $\mathbb{F}_p$, entonces poseen la misma longitud en bits.

3. *Es escalable:* Es posible expandir para nuevos usuarios, esto significa que podemos calcular y distribuir nuevos fragmentos. Por otro lado podemos observar que si un usuario borra su fragmento, el sistema sigue funcionando normalmente, pero hace inaccesible el acceso para este participante.

4. *Es controlable por niveles:* Si proveemos a un usuario con múltiples fragmentos, le proporcionamos más control para la recuperación del secreto, pues requiere un número menor (que el umbral k) de participantes para su reconstrucción. Esto implica una estructura de acceso distinta.

5. *Es seguro:* La seguridad del esquema de Shamir se basa principalmente en la elección del umbral.

2.3.3 Esquema de Brickell-Davenport

Como hemos mencionado anteriormente, para construir un ECS los elementos básicos son el secreto y la estructura de acceso. Brickell y Davenport [7] desarrollaron un modelo más general que caracteriza a los ECS ideales, sin embargo en su artículo original no proporcionaron construcción alguna de tales esquemas.

El teorema es el siguiente.

Teorema 2.3.17 *Si $\mathcal{M} = (E, \mathcal{I})$ es un matroide que es representable sobre un campo finito K y $v_0 \in E$ es un punto, entonces existe un ECS ideal Π de tal forma que el dominio de secretos es K, el distribuidor $\mathfrak{p}_0$ coincide con v_0 y el conjunto de participantes P coincide con E, y además los conjuntos dependientes del matroide forman la estructura de acceso de Π.*

En esta sección analizamos algunos ECS que posteriormente nos servirán de apoyo para proponer un esquema ideal con las condiciones del teorema 2.3.17 (véase el capítulo 4). También recordamos la construcción del correspondiente ECS presentada por Brickell mismo en [6].

Construcción de Brickell Basada en un Espacio Vectorial

La construcción de Brickell basada en un espacio vectorial generaliza el esquema de Shamir. Supongamos que Γ es una estructura de acceso, $P = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_m\}$ el conjunto de m participantes y $\mathfrak{p}_0$ el distribuidor. Sea $\mathbb{F}_q$ un campo finito, digamos de característica p, donde p es un número primo y, en consecuencia, $q = p^k$ para algún entero $k \geq 1$. Sea $n \geq 2$ un número entero, y sea $\mathbb{F}_q^n$ el espacio vectorial de dimensión n sobre el campo $\mathbb{F}_q$, consistente precisamente de su n-ésima potencia cartesiana. Para cada participante $\mathfrak{p}_i \in P$, supongamos elegido un vector $x_{\mathfrak{p}_i} \in \mathbb{F}_q^n$. Para cada subconjunto de participantes $B \subset P$,

sea $L(B) = \langle x_{\mathfrak{p}} | \mathfrak{p}_i \in B \rangle$ el subespacio vectorial generado por los vectores correspondientes a los participantes en B. Supongamos, además, que para cualquier subconjunto $B \subset P$ se cumple la relación

$$e_1 \in L(B) \Leftrightarrow B \in \Gamma \tag{2.13}$$

donde $e_1 = (1, 0, \ldots, 0)$ es el primer vector de la base canónica de $\mathbb{F}_q^n$. La relación 2.13 es un predicado que nos permite decidir cuando un subconjunto B es de acceso.

Supongamos también que tanto el dominio de secretos como el dominio de fragmentos coinciden con $\mathbb{F}_q$. Fijo un vector $a \in \mathbb{F}_q^n$, sea $f_a : \mathfrak{p} \mapsto (a \cdot x_{\mathfrak{p}})$ la función de "distribución" $P \to \mathbb{F}_q$ que a cada participante le asocia como fragmento el producto interno de a con el vector asociado a ese participante.

En los algoritmos 4, 5 y 6 (éste último se presenta al final de esta sección) describimos las fases del mecanismo de construcción del esquema de espacio vectorial de Brickell.

Algoritmo 4 Fase Inicial del distribuidor $\mathfrak{p}_0$

Require: Los parámetros n y q del espacio vectorial $\mathbb{F}_q^n$.

Ensure: Un vector $x_{\mathfrak{p}} \in \mathbb{F}_q^n$, para cada participante $\mathfrak{p} \in P$. La colección $\{x_{\mathfrak{p}} | \mathfrak{p} \in P\}$ se hace pública.

Algoritmo 5 Distribución de los fragmentos

Require: Un secreto $s \in \mathbb{F}_q$.

Ensure: El secreto $f_a(\mathfrak{p})$ es entregado al participante $\mathfrak{p}$.

1: El distribuidor $\mathfrak{p}_0$ elige, en privado, $n-1$ elementos $a_2, \ldots, a_n \in \mathbb{F}_q$, y forma el vector $a = (s, a_2, \ldots, a_n) \in \mathbb{F}_q^n$.

2: A cada participante $\mathfrak{p} \in P$, el distribuidor $\mathfrak{p}_0$ le entrega como fragmento el valor $f_a(\mathfrak{p}) = (a \cdot x_{\mathfrak{p}})$ (denotamos por $s_{\mathfrak{p}}$ a $f_a(\mathfrak{p})$).

Vemos entonces que en el procedimiento anterior, ya que el secreto s está fijo, hay q^{n-1} posibilidades de elegir el resto del vector a en el punto 2.(a) del algoritmo anterior. Sea $\mathcal{F}_s = \{f_a | a$ se construye como en 2.$(a)\}$ la familia de funciones de distribución de ese algoritmo.

Supongamos ahora que la probabilidad de elegir cualquier elemento en $\mathcal{F}_s$ en el punto 2.(a) es uniforme, es decir, que, para cualquier $a \in \mathbb{F}_q^n$ cuya primera coordenada coincida con s:

$$\text{prob}_{\mathcal{F}_a}(f = f_a) = \frac{1}{p^{n-1}}.$$

Proposición 2.3.18 *Si la familia de vectores $\{x_{\mathfrak{p}} | \mathfrak{p} \in P\}$ satisface la relación (2.13) entonces la familia $\mathcal{F}_{\mathbb{F}_q} = \bigcup\{\mathcal{F}_s | s \in \mathbb{F}_q\}$ comprende un esquema ideal que realiza a la estructura de acceso Γ.*

Demostración. Veamos que el algoritmo anterior efectivamente permite recuperar el secreto cuando los participantes de un conjunto de acceso $B \in \Gamma$ proporcionan todos sus fragmentos. En efecto, por la relación (2.13), si $B \in \Gamma$ entonces existen coeficientes

$(c_{\mathfrak{p}})_{\mathfrak{p}\in B} \subset \mathbb{F}_q$, tales que $e_1 = \sum_{\mathfrak{p}\in B} c_{\mathfrak{p}} x_{\mathfrak{p}}$. En consecuencia,

$$s = (e_1 \cdot a) = \left(\sum_{\mathfrak{p}\in B} c_{\mathfrak{p}} x_{\mathfrak{p}} \cdot a \right) = \sum_{\mathfrak{p}\in B} c_{\mathfrak{p}} \left(x_{\mathfrak{p}} \cdot a \right) = \sum_{\mathfrak{p}\in B} c_{\mathfrak{p}} s_{\mathfrak{p}}, \tag{2.14}$$

así pues, al conocer los fragmentos de los participantes en B se recupera S, tomando en cuenta que se puede calcular los coeficientes $(c_{\mathfrak{p}})_{\mathfrak{p}\in B}$.

Ahora bien, si $B \notin \Gamma$, entonces los coeficientes anteriores $(c_{\mathfrak{p}})_{\mathfrak{p}\in B} \subset \mathbb{F}_q$ no existen. Pero aún más, sea $n_B = \dim L(B)$ la dimensión del espacio generado por los vectores asociados a los participantes en B. Claramente, $n_B \leq |B|$. Para cada $s \in \mathbb{F}_q$, al plantear el sistema de ecuaciones lineales $(e_1 \cdot a) = k$, $(x_{\mathfrak{p}} \cdot a) = s_{\mathfrak{p}}$, $\mathfrak{p} \in B$, se tiene que éste posee una matriz de coeficientes de rango $n_B + 1$ (pues $e_1 \notin L(B)$). Si acaso el sistema de ecuaciones tuviese solución $a \in \mathbb{F}_q^n$, el espacio de soluciones sería de dimensión $n - n_B - 1$. Así pues, para cada tal $s \in \mathbb{F}_q$ hay q^{n-n_B-1} funciones de distribución de la forma f_a que darían los mismos fragmentos. Por tanto la probabilidad de que s sea el secreto elegido coincide con la probabilidad condicional de que lo sea, dado que los fragmentos de participantes en B sean $(s_{\mathfrak{p}})_{\mathfrak{p}\in B}$. El conocimiento entonces de los fragmentos no incrementa el conocimiento del secreto.

En este segundo caso, el sistema planteado posee una solución que es precisamente el vector a elegido por el distribuidor $\mathfrak{p}_0$. □

La manera de recuperar un secreto en el esquema de espacio vectorial de Brickell se encuentra en el algoritmo 6.

Algoritmo 6 Reunión de los fragmentos

Require: Un subconjunto B de participantes que satisfacen la ecuación 2.13.
Ensure: El secreto s.
1: Se calculan los coeficientes $c_{\mathfrak{p}}$ de $e_1 = \sum_{\mathfrak{p}\in B} c_{\mathfrak{p}} x_{\mathfrak{p}}$.
2: El secreto esta dado por la expresión de ecuación 2.14.

Proposición 2.3.19 *Si $G = (V, A)$ es una gráfica complementaria a una unión de clanes entonces hay un esquema ideal que realiza a la estructura de acceso $cl(A)$ sobre el conjunto de participantes V.*

Demostración. Si $V_1, \ldots, V_N$ es la colección de clanes, sea $q > card\, V$ una potencia de un número primo y sean $y_1, \ldots, y_N \in \mathbb{F}_q$ elementos distintos a pares. A cada participante $\mathfrak{p}_{i,j} \in V$, que es elemento del clan V_i se le asocia el vector $x_{i,j} = (y_i, 1) \in \mathbb{F}_q^2$.

Se tiene que el primer vector de la base canónica de $\mathbb{F}_q^2$ estará en el espacio generado por los vectores asociados a dos participantes, en símbolos $(1, 0) \in \langle x_{i_1,j_1}, x_{i_2,j_2} \rangle$, si y sólo si $i_1 \neq i_2$ y en tal caso

$$(1, 0) = (y_{i_1} - y_{i_2})^{-1} \left(x_{i_1,j_1} - x_{i_2,j_2} \right).$$

Así pues la colección A (que es la colección de aristas cuyos vértices se encuentran en clanes distintos) satisface la relación (2.13) y, por lo anterior, su cerradura se realiza de acuerdo con la construcción en la proposición previa. □

Ilustremos, mediante un pequeño ejemplo, la construcción mostrada. Supongamos que C_1 es una arista, C_2 es un triángulo y C_3 es un cuadrado (con sus dos diagonales principales incluidas). Entonces el número de participantes es $2+3+4=9$: $P=(\mathfrak{p}_j)_{j=1}^{9}$, los dos primeros forman C_1, los siguientes tres C_2 y los últimos cuatro C_3. Consideremos $q=4=2^2$. Los cuatro elementos de $\mathbb{F}_2^2$ son $0, 1, X, X+1$ y las operaciones son polinomiales reducidas módulo el polinomio $p(X)=X^2+X+1$. Sean respectivamente $y_1=1$, $y_2=X$ e $y_3=X+1$. A los dos primeros participantes se les asocia el vector $x_1=(1,1)$, a los siguientes tres $x_2=(X,1)$, y a los últimos cuatro $x_3=(X+1,1)$. Dos participantes de un mismo clan sólo aportan un único vector y $(1,0)$ no es ningún múltiplo de él. En cambio, cuando los participantes son de clanes distintos, entonces $(1,0)$ sí se realiza como una combinación lineal de los vectores correspondientes:

$$\begin{aligned}(1,0) &= X(1,1)+X(X,1)=Xx_1+Xx_2\\ &= (X,1)+(X+1,1)=x_1+x_3\\ &= (X+1)(1,1)+(X+1)(X+1,1)=(X+1)x_2+(X+1)x_3\end{aligned}$$

La familia Γ_0 constará entonces de las parejas de participantes $(\mathfrak{p}_j,\mathfrak{p}_k)$ tales que el primero está en un clan y el segundo en otro. Explícitamente, las $6+8+12=26$ parejas son las siguientes:

Arista-triángulo. $(\mathfrak{p}_1,\mathfrak{p}_3)$, $(\mathfrak{p}_1,\mathfrak{p}_4)$, $(\mathfrak{p}_1,\mathfrak{p}_5)$, $(\mathfrak{p}_2,\mathfrak{p}_3)$, $(\mathfrak{p}_2,\mathfrak{p}_4)$, $(\mathfrak{p}_2,\mathfrak{p}_5)$,

Arista-cuadrado. $(\mathfrak{p}_1,\mathfrak{p}_6)$, $(\mathfrak{p}_1,\mathfrak{p}_7)$, $(\mathfrak{p}_1,\mathfrak{p}_8)$, $(\mathfrak{p}_1,\mathfrak{p}_9)$, $(\mathfrak{p}_2,\mathfrak{p}_6)$, $(\mathfrak{p}_2,\mathfrak{p}_7)$, $(\mathfrak{p}_2,\mathfrak{p}_8)$, $(\mathfrak{p}_2,\mathfrak{p}_9)$,

Triángulo-cuadrado. $(\mathfrak{p}_3,\mathfrak{p}_6)$, $(\mathfrak{p}_3,\mathfrak{p}_7)$, $(\mathfrak{p}_3,\mathfrak{p}_8)$, $(\mathfrak{p}_3,\mathfrak{p}_9)$, $(\mathfrak{p}_4,\mathfrak{p}_6)$, $(\mathfrak{p}_4,\mathfrak{p}_7)$, $(\mathfrak{p}_4,\mathfrak{p}_8)$, $(\mathfrak{p}_4,\mathfrak{p}_9)$, $(\mathfrak{p}_5,\mathfrak{p}_6)$, $(\mathfrak{p}_5,\mathfrak{p}_7)$, $(\mathfrak{p}_5,\mathfrak{p}_8)$, $(\mathfrak{p}_5,\mathfrak{p}_9)$.

Es claro que éstas son las únicas parejas que cumplen con la relación (2.13). La unión de todas estas parejas forma sobre P una gráfica tripartita que es precisamente la complementaria a la unión de la arista, el triángulo y el cuadrado. □

El sistema construido es de umbral con 2 participantes como mínimo para reconstruir el secreto. Pueden construirse sistemas de ECS con umbrales mayores pasando de la noción de gráficas a la de hipergráficas. Si se quiere tener un umbral k se ha de considerar hiperaristas de grado k, las que son conjuntos de k elementos. Sobre esto volveremos en la subsección 4.1.1 más adelante.

De manera similar que la proposición anterior, existe un ECS ideal para una estructura de acceso que es igual a la clausura de los clanes de una gráfica.

Proposición 2.3.20 *Si $G=(V,A)$ es una gráfica que es una unión de clanes entonces hay un esquema ideal que realiza a la estructura de acceso $cl(A)$ sobre el conjunto de participantes V.*

Demostración. Si $V_1,\ldots,V_N$ es la colección de clanes, sea $q>card\ V$ una potencia de un número primo y sean $y_1,\ldots,y_N\in\mathbb{F}_q$ elementos distintos a pares que serán asignados a cada clan V_i.

Observemos que los elementos de un clan V_i describen puntos sobre una línea recta en $\mathbb{R}^2$. Entonces, con al menos dos puntos en un mismo clan V_i, podemos recuperar el secreto s.

Así pues la estructura de acceso es la cerradura de la colección A (quien es la colección de aristas cuyos vértices se encuentran en un mismo clan). □

Capítulo 3

Selección Aleatoria de Matroides Representables

El esquema general que proponemos en esta tesis está constituido de dos fases. La primera consiste de inicializar los parámetros para la distribución de los fragmentos (figura 3.1).

Figura 3.1: Fase Inicial

En esta fase necesitamos proporcionar al distribuidor los parámetros necesarios para obtener un ECS ideal. Para poder definir a la estructura de acceso de este esquema primero obtenemos un matroide representable.

3.1 El Problema de la Selección Aleatoria

Comenzamos con el problema de seleccionar un matroide representable de tal forma que tenga una distribución uniforme. Esto equivale a que dado un campo finito $\mathbb{F}_q,\ q = p^k,\ p$ primo, deseamos determinar un submatroide que sea representable.

Recordemos que un espacio vectorial $\mathbb{F}_q^n$, sobre un campo finito $\mathbb{F}_q$, junto con la colección de conjuntos independientes, es por definición un matroide representable.

Supongamos que $m \leq n$, como antes. Es necesario considerar los siguientes hechos:

1. El campo $\mathbb{F}_q$ posee q elementos entre los cuales podemos efectuar operaciones aritméticas. Si $k = 1$, es decir, $q = p$ es un primo, todas las operaciones de $\mathbb{F}_q$ se aplican módulo q, y también existen algoritmos efectivos para calcular inversos en el campo, a saber, el algoritmo de Euclides para calcular máximos comunes divisores. Si $q = p^k$, con $k > 1$, entonces el trabajo por hacer es implementar la aritmética de polinomios (sumas, multiplicaciones y divisiones),pues las operaciones son módulo un polinomio irreducible en $\mathbb{F}_q[X]$.

2. El campo $\mathbb{F}_q$ posee exactamente q elementos, por lo que por medio de una enumeración convencional se identifica con el conjunto de índices $\{0, 1, 2, \ldots, q-1\}$. Escribamos $\mathbb{F}_q = (z_j)_{j=0}^{q-1}$.

3. El espacio vectorial $\mathbb{F}_q^n$ posee q^n elementos:

$$\begin{aligned}
(z_0, z_0, \ldots, z_0) &\sim 0, \\
(z_0, z_0, \ldots, z_1) &\sim 1, \\
(z_0, z_0, \ldots, z_2) &\sim 2, \\
&\vdots \\
(z_0, z_0, \ldots, z_{q-1}) &\sim q-1, \\
(z_0, z_0, \ldots, z_1, z_0) &\sim q, \\
(z_0, z_0, \ldots, z_1, z_1) &\sim q+1, \\
&\vdots \\
(z_{q-1}, z_{q-1}, \ldots, z_{q-1}) &\sim q^n - 1.
\end{aligned}$$

   De donde es claro que podemos poner en correspondencia a $\mathbb{F}_q^n$ con el conjunto: $[\![0, q^n - 1]\!] := \{0, 1, 2, \ldots, q^n - 1\}$. Dado un vector $(a_{n-1}, \ldots, a_1, a_0) \in \mathbb{F}_q^n$, el correspondiente número en $[\![0, q^n - 1]\!]$ es $\sum_{i=0}^{n-1} a_i \, q^i$. Recíprocamente, dado un número en $[\![0, q^n - 1]\!]$, obtenemos al vector asociado colocando los dígitos de la representación del número en base q, con los más significativos hacia la izquierda.

4. Un vector $v \in \mathbb{F}_q^n$ posee $q - 1$ múltiplos: $z_j v$, $j = 0, \ldots, q-1$.

5. Por conveniencia es mejor observar a los vectores $v \in \mathbb{F}_q^n$ como vectores columna, $v = [v_1 \ \cdots \ v_n]^T$.

6. Una colección $\{v_1, \ldots, v_k\}$ de vectores en $\mathbb{F}_q^n$ es *linealmente independiente* (l. i.) si:

$$a_1 v_1 + \ldots a_k v_k = 0 \Rightarrow a_i = 0, \ i = 1, \ldots, k.$$

   Un conjunto que no es linealmente independiente naturalmente ha de ser *linealmente dependiente* (l. d.).

7. Una base del espacio $\mathbb{F}_q^n$ es un conjunto de vectores linealmente independientes que generan al espacio, es decir, que expresan a cualquier vector del espacio como una combinación lineal de ellos con coeficientes en el campo.

8. Una base en el espacio $\mathbb{F}_q^n$ posee n vectores.

9. Si dada una base $\{b_1, b_2, \ldots, b_n\}$ de $\mathbb{F}_q^n$, colocamos los vectores en ella como columnas de una matriz, $M = [b_1\, b_2\, \ldots, b_n^T]$, resultará que M es no-singular (su determinante es distinto de cero), lo cual es un resultado elemental de Álgebra Lineal, ver, por ejemplo [11].

10. En consecuencia, es posible contar a las matrices no-singulares en $\mathbb{F}_q^{n\times n}$ de la siguiente manera: para seleccionar una típica matriz no-singular M, tenemos $(q^n - 1)$ posibilidades para la primera columna (que es cualquier vector no-nulo); luego, para cada $j \leq n$, en la j-ésima columna podemos colocar cualquier vector que no sea una combinación lineal de las anteriores $j-1$ columnas (y hay q^{j-1} tales combinaciones), por lo que hay exactamente $(q^n - q^{j-1})$ posibilidades para la j-ésima columna. Así pues el número total de matrices no-singulares es

$$s_{q,n} = \prod_{j=1}^{n}(q^n - q^{j-1}) = \prod_{j=0}^{n-1}(q^n - q^j) \tag{3.1}$$

y en consecuencia el número de bases es:

$$c(q,n) = \frac{s_{q,n}}{n!} = \prod_{j=0}^{n-1}\left(\frac{q^n - q^j}{j+1}\right). \tag{3.2}$$

Con estas observaciones, construiremos el matroide representable considerando la siguiente organización: Dados m, n, q,

- consideramos el espacio F_q^n;
- consideramos a un conjunto de m vectores linealmente independientes en $\mathbb{F}_q^n$ (seleccionado aleatoriamente);
- consideramos el subespacio vectorial V generado por estos vectores; y
- devolvemos como matroide representable a la colección de conjuntos linealmente independientes de V.

En la siguiente sección describimos los algoritmos que hacen posible la selección de matroides aleatorios dadas las indicaciones anteriores.

3.2 Algoritmos para la Obtención de un Matroide Representable

En esta sección describimos los procedimientos necesarios para implementar la selección aleatoria. La descripción precisa de los algoritmos aquí propuestos están en el apéndice 1.

Para esta parte del sistema, los siguientes parámetros son necesarios:

- n, m números enteros tales que $m \leq n$.
- q una potencia de un primo.

Algoritmo 8: ComoNumero$[x, n, q]$. Dado un vector $\mathbf{x} \in \mathbb{F}_q^n$, podemos convertirlo a un número entero, como indicamos en el hecho 3 de la sección anterior:

$$n = \sum_{i=0}^{n-1} x_i \, q^i.$$

Algoritmo 9: ComoVector$[a]$. De esta forma, si tomamos un entero $n \in [\![0, q^n - 1]\!]$, construimos el vector correspondiente de la siguiente manera: representamos el número n en base q y colocamos a los dígitos obtenidos como entradas del vector a devolver.

Algoritmo 10: VectorSiguiente$[x]$. Recordando que es posible ordenar al conjunto $\mathbb{F}_q^n$, entonces, dado un vector, es posible dar el vector siguiente de acuerdo con este orden. Si x es el último vector $(q-1, q-1, \ldots, q-1)$, devolvemos $\emptyset$.

Algoritmo 11: InsercionDeColumna$[M, x]$. Por conveniencia, veremos un conjunto de vectores como un arreglo matricial, pues efectuamos diversas operaciones con tales matrices. En particular, insertamos un vector x a una matriz M de k vectores, como una columna de tal matriz:

$$M \cup x = \begin{pmatrix} M_{1,1} & M_{1,2} & \ldots & M_{1,k} & x_1 \\ M_{2,1} & M_{2,2} & \ldots & M_{2,k} & x_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ M_{n,1} & M_{n,2} & \ldots & M_{n,k} & x_n \end{pmatrix}$$

Algoritmo 12: TriangulacionGaussiana$[M]$. Uno de los resultados del álgebra lineal, es el método de Gauss el cual consiste en transformar a una matriz de coeficientes en el campo $\mathbb{F}_q$, en una matriz triangular superior mediante operaciones elementales. Evidentemente, las operaciones son realizadas en el campo $\mathbb{F}_q$.

- *Operaciones elementales.* Son las operaciones que podemos aplicar a una matriz sin que su rango cambie.

  Las operaciones que no cambian el rango son: permutar 2 filas ó 2 columnas, multiplicar o dividir una fila por un número no cero, sumar o restar a una fila, otra fila multiplicada por un número en el campo, suprimir las filas o columnas nulas y suprimir a las filas o columnas que sean múltiplos de alguna otra fila o columna.

- *Método de Gauss.* El método de Gauss consiste en aplicar operaciones elementales a una matriz para hacer cero a los elementos que están por debajo de la diagonal principal ($a_{ij} = 0, i > j$). La triangulación Gaussiana se logra dejando en la diagonal principal de la matriz, elementos no cero, salvo que la fila contenga sólo ceros.

  El rango de una matriz triangulada es el número de filas no nulas de la matriz obtenida.

Algoritmo 13: RevisionLI$[M, x]$. Dado un conjunto de vectores l.i., en una matriz M, y un vector x, es necesario determinar si tal vector esta en el espacio generado por vectores dados en M. Para realizar esto podemos proceder de la siguiente manera: unimos x a M, y aplicamos el método de Gauss a la matriz resultante, si las entradas en la diagonal de esta matriz son distintas de cero, entonces, los vectores son l. i., con lo cual, x no esta en el espacio generado por los vectores en M.

Algoritmo 14: ListaDeLIs$[M]$. Dada una matriz M, cuyos vectores forman un conjunto que es l. i., se puede determinar a la colección (digamos una lista LI) de vectores que satisfacen: $\forall x \in LI$, los vectores de la matriz M y x forman un conjunto l.i..

Primero tenemos la lista vacía, $LI = \emptyset$, y luego, $\forall x \in F_q^n$, verificamos si $M \cup x$ es un conjunto l.i., de ser así, entonces añadimos el vector x a la lista LI.

Todos estos vectores nos ayudarán a formar todos los conjuntos l.i. que son distintos entre sí.

Algoritmo 15: ListadoMNS$[q, n]$. Utilizando el procedimiento anterior, mediante *backtracking* podemos listar a todas las matrices no-singulares (recordamos que tales matrices siempre son cuadradas: $n \times n$).

El algoritmo es recursivo y requiere de n pilas que contendrán a los vectores como ilustramos en la figura 3.2.

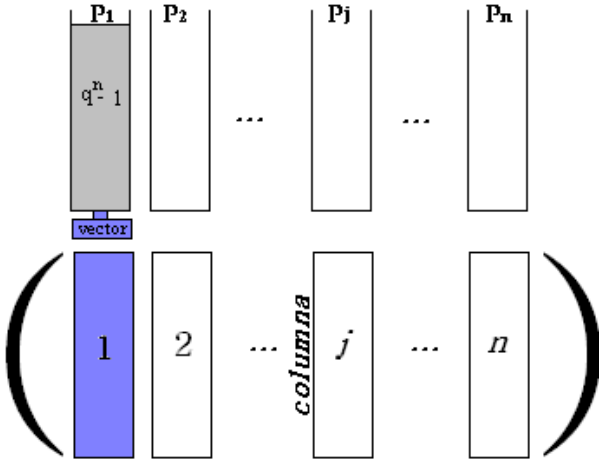


Figura 3.2: El primer vector

Para seleccionar al primero de los n vectores de cada matriz, tenemos $q^n - 1$ posibles vectores (éste es cualquier vector no nulo) que colocamos en la pila P_1, el primer vector, es entonces, el elemento en el tope de P_1 (desempilamos), como observamos en la figura 3.2. Otra observación importante es que el proceso que proponemos enseguida, debe efectuarse para cada vector en la pila P_1.

Para el segundo vector, debemos seleccionar un vector, de tal forma que sean l.i. con el primer vector, de esta forma, podemos elegir tal vector de $q^n - q$ posibles vectores (quitando los q múltiplos del primer vector). Colocamos a tales vectores en la pila P_2. Entonces, tomamos el segundo vector del tope de la pila P_2 (desempilamos), como podemos observar en la figura 3.3. De esta forma, los vectores en la matriz serán l.i.

Luego, para cada columna $j < n$, hemos de elegir (como indica el hecho 10 mencionado

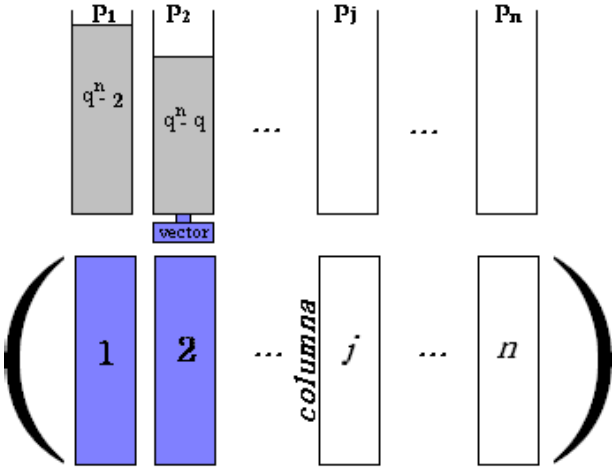


Figura 3.3: El segundo vector

anteriormente) un vector que no esté en el espacio generado por los $j-1$ vectores ya seleccionados, son el fin de que los vectores sean l.i. En total, el número de posibles vectores para la j-ésima columna es $q^n - q^j$, mismos que colocamos en la pila P_j. Tomamos el j-ésimo vector del tope de la pila P_j.

Finalmente, cuando llegamos a la elección de la columna n, (véase la figura 3.4). Podemos elegir a este vector de $q^n - q^{n-1}$ posibles vectores que forman un conjunto l.i. Colocamos a tales vectores en la pila P_n.

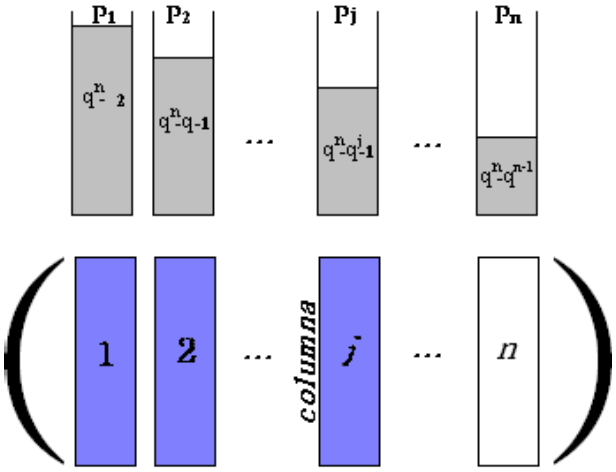


Figura 3.4: La pila P_n

En este punto listamos $q^n - q^{n-1}$ matrices no singulares: la matriz base posee $n-1$ columnas; por lo que formamos las matrices colocando en la posición de la última columna, cada vector de la pila P_n. Finalmente la pila P_n queda vacía. En general, cuando tenemos vacía la pila P_i, procedemos a tomar un elemento de la pila P_{i-1}. El proceso finaliza cuando la pila P_1 queda vacía.

Este procedimiento genera una lista de $s_{q,n}$ matrices no-singulares (véase la ec. 3.1), entre la cuales, existen matrices con conjunto de columnas que son iguales, (esto pues, la permutaciones de una colección de n columnas produce $n!$ matrices distintas). Entonces

es necesario quitar tales repeticiones, ya que sólo tenemos interés en las bases producidas por tales matrices. Por lo tanto, el número de matrices requeridas se reduce a $c(q,n)$ (véase la ec. 3.2).

Para quitar las repeticiones, primero representamos a las columnas de cada matriz como un conjunto de números (en $\mathbb{F}_q^n$), para después, proceder a eliminar a los conjuntos (ordenados) que estén repetidos.

Algoritmo 16: MNSAleatoria$[q,n]$. Con la idea del procedimiento anterior, ahora seleccionamos aleatoriamente una matriz no-singular.

La manera de proceder es la siguiente: la primer columna de la matriz M, seleccionamos (aleatoriamente) tomando un vector, de los q^n-1 vectores posibles de la pila P_1 (como mostramos en la figura 3.2); y para cada $j=2,\ldots,n$, elegimos el j-ésimo vector (aleatoriamente) de una pila P_j, que consiste de los q^n-q^j vectores que obtenidos del **Algoritmo 14**, aplicado a la matriz M actual.

Algoritmo 17: BaseComoNumeros$[M]$. Dada una matriz $M\in\mathbb{F}_q^{n\times m}$, donde sus columnas forman un conjunto l. i. de $\mathbb{F}_q^n$, podemos representar a estos vectores como un conjunto de m números enteros de $[\![0,q^n-1]\!]$. Si M es no-singular, este algoritmo transforma a M en un conjunto de n números que representan a una base en $\mathbb{F}_q^n$. La salida es un arreglo de números ordenados.

Algoritmo 18: ListaBasesComoNumeros$[q,n]$. Dada la lista de matrices no-singulares (obtenida por ListadoMNS$[q,n]$), a cada matriz en ella le aplicamos el procedimiento anterior. Ordenamos la lista obtenida lexicográficamente. Toda base de $\mathbb{F}_q^n$ es vista como un conjunto, por lo que es irrelevante el orden en que se escribe a sus elementos. En cambio, en las matrices, el orden en el que aparecen las columnas es sumamente relevante. Así, dos matrices que difieran por una mera permutación de sus columnas representarán una misma base. Obviamente, cada base dará origen entonces a $n!$ matrices. Por tanto, al suprimir repeticiones (módulo permutaciones) de los elementos de la lista anterior, obtenemos la lista de bases de $\mathbb{F}_q^n$.

Algoritmo 7: MatroideRepAleatorio$[q,m,n]$. Con los algoritmos previos, podemos seleccionar aleatoriamente un matroide representable, basta con elegir aleatoriamente una matriz no-singular $M^{n\times n}(\mathbb{F}_q)$ y de entre sus columnas elegimos un subconjunto de m vectores. Todas las bases de este subespacio formarán el conjunto de bases de un matroide representable en $\mathbb{F}_q^n$. El algoritmo para tal selección es el siguiente:

Algoritmo 7 MatroideRepAleatorio$[q,m,n]$

Require: $m,n\in\mathbb{N},\ m\leq n,\ q=p^k,\ p$ primo.
Ensure: Un matroide representable cuyas bases constan de m elementos cada una.
1: $M\leftarrow$ MNSAleatorio$[q,n]$;
2: $J\subset[\![0,q^n-1]\!]^{(m)}$ (conjunto de m índices);
3: $M_J\in\mathbb{F}_q^{n\times m}$ (la matriz obtenida de M al extraer columnas con índices en J);
4: $L_m\leftarrow$ ListadoMNS$[q,m]$;
5: $L_n\leftarrow\{M_JM|M\in L_m\}$ (una lista);
6: Regresar L_n.

3.3 Complejidad de los Algoritmos

En esta sección proporcionamos un análisis de complejidad de los algoritmos propuestos. Esta complejidad la calculamos a partir de los peores casos. Los tiempos de ejecución considerados para las operaciones en el campo $\mathbb{F}_q$ (suma, producto, inversión, etc.) los estamos considerando, acaso arbitrariamente, como de tiempo unitario, es decir, de tiempo constante. Como ya lo dijimos anteriormente, si $k = 1$ y q es un primo, que de hecho es el caso correspondiente a nuestra implementación, las operaciones son módulo q, en tanto que para $k > 1$ las operaciones son polinomiales y naturalmente ejecutables en tiempo proporcional a k. En el análisis que aquí presentamos las operaciones de $\mathbb{F}_q$ se suponen como primitivas, por lo que las complejidades que mencionaremos han de entenderse "relativizadas" a las operaciones del campo.

Algoritmo 8: ComoNumero$[x, n, q]$. Realizamos n sumas y n productos, pues sólo se está haciendo una evaluación de un polinomio de grado n, mediante el esquema de Horner, en $\mathbb{F}_q$, por lo tanto la complejidad es: $O(n)$.

Algoritmo 9: ComoVector$[a]$. Aquí debemos efectuar a operaciones en $\mathbb{F}_q^n$, por lo tanto la complejidad es: $O(n)$.

Algoritmo 10: VectorSiguiente$[x]$. Aquí efectuamos dos operaciones: **ComoNumero**$[x, n, q]$ y **ComoVector**$[a]$ por lo tanto la complejidad es también: $O(n)$.

Algoritmo 11: InsercionDeColumna$[M, x]$. La complejidad es: $O(n)$.

Algoritmo 12: TriangulacionGaussiana$[M]$. Aquí, supongamos que la matriz es de tamaño $n \times n$, entonces, La complejidad de este algoritmo es: $O(n^3)$.

Algoritmo 13: RevisionLI$[M, x]$. El peor caso sucede cuando M posee $n - 1$ columnas. La complejidad de este algoritmo es la misma del anterior, pues, debemos diagonalizar la matriz $M \cup x$, y después hacemos la revisión de los elementos no cero en la diagonal. Por lo tanto: $O(n^3)$.

Algoritmo 14: ListaDeLIs$[M]$. El peor caso sucede cuando M posee $n - 1$ columnas, aquí efectuamos una triangulación gaussiana por cada elemento en la última pila (que posee $q^n - q^{n-1}$), por lo tanto: $O(q^n n^3)$. Vale la pena mencionar aquí que, de acuerdo con la relación (3.1), la longitud de esta lista es del orden $O(q^n)$, de ahí que la enumeración exhaustiva de esos conjuntos que realiza este procedimiento conlleva en su complejidad el crecimiento de la longitud de la lista.

Algoritmo 15: ListadoMNS$[q, n]$. Por cada una de las $s_{q,n}$ (ec. 3.1) matrices, efectuamos una **ListaDeLIs**$[M]$. Por tanto en el peor caso tenemos: $O((q^n)^n \cdot n^3 n) = O(q^{n^2} n^4)$.

La complejidad mencionada es inmensa. Por un lado, conlleva la misma complejidad del procedimiento **ListaDeLIs**$[M]$, y por otro lado el procedimiento de *backtracking* utilizado para obtener, como condiciones de paro, listas de n vectores linealmente inde-

pendientes.

Algoritmo 16: MNSAleatoria$[q, n]$. Para encontrar una matriz no singular, hay que utilizar n veces a **ListaDeLIs**$[M]$, por tanto: $O(q^n n^4)$.

Algoritmo 17: BaseComoNumeros$[M]$. Son básicamente operaciones en $\mathbb{F}_q$. Hay que transformar a cada columna de una matriz $n \times n$, a su representación en $[\![0, q^n - 1]\!]$. $O(n)$.

Algoritmo 18: ListaBasesComoNumeros$[q, n]$. Transformamos cada una de las $c(q, n)$ (ec. 3.2) matrices en una lista de enteros en $[\![0, q^n - 1]\!]$. Por tanto: $O(q^{n^2} n)$.

Algoritmo 7: MatroideRepAleatorio$[q, m, n]$. Aquí, finalmente, seleccionamos una matriz aleatoria $n \times n$, de la cual tomamos m columnas. Posteriormente, el matroide consiste de la matrices obtenidas de multiplicar a la matriz formada por las m columnas seleccionadas, por cada una de las matrices obtenidas del algoritmo **ListadoMNS**$[q, m]$. Por lo tanto: $O(q^{n^2} n^4)$.

Como mencionamos en la presentación de algunas complejidades arriba, las de los algoritmos (14)-(7) tienen crecimientos vertiginosos, y dependen esencialmente de la misma estructura combinatoria de los problemas que resuelven. Esos procedimientos los hemos incluido aquí debido a que aparecen como meros corolarios de los métodos de conteo de la sección 3.1. En la práctica, tales recuentos exhaustivos no son de interés, y , en el sistema que hemos desarrollado, en su estado actual, sólo se utilizan el (15) y el (7) para la generación de un matroide aleatorio. Sin embargo, tal generación puede omitir realizar la cuenta exhaustiva, pero la modificación necesaria de la implementación ya sale del alcance de esta tesis.

3.4 Métodos Alternativos para la Selección de Matroides

Como mencionamos anteriormente, en la generación aleatoria de matroides deseamos omitir realizar la cuenta exhaustiva pues provoca una complejidad alta. En esta sección proponemos una alternativa para la selección aleatoria de matroides y mostramos algunas de sus ventajas y desventajas.

Sean n y q los parámetros dados. Para generar un matroide de rango n sobre el campo $\mathbb{F}_q$, debemos tener las $s_{q,n} = \frac{1}{n!} \prod_{j=0}^{n-1} (q^n - q^i)$ bases que lo constituyen (pues la colección de conjuntos independientes del matroide son caracterizados por sus bases). A cada base que posee n vectores, podemos identificarla con una matriz de orden $n \times n$, colocando como columnas a los vectores de la base. Entonces una matriz es base si y sólo si, la matriz asociada es no singular.

La alternativa propuesta es obtener aleatoriamente a la colección de los $s_{q,n}$ conjuntos (con n vectores en $\mathbb{F}_q^n - \{0\}$ cada uno), de tal forma que las matrices generadas, sean esencialmente distintas, es decir, que (salvo orden) el conjunto de columnas de tales matrices son siempre distintas entre sí.

Entonces para generar al matroide siguiendo esta idea, debemos considerar lo siguiente:

1. El número total de matrices $n \times n$ que se pueden generar con columnas no nulas es:

$$nm = (q^n - 1)^n,$$

pues, la primer columna la podemos escoger como cualquier vector no nulo, por lo que el número de posibilidades es $q^n - 1$, la segunda columna la podemos escoger como cualquier vector no cero, por lo que el número de posibilidades es $q^n - 1$, etc. Denotemos por $\mathcal{M}_n(\mathbb{F}_q)$ a la colección de tales matrices (con columnas no cero).

2. La probabilidad de que una matriz $M \in \mathcal{M}_n(\mathbb{F}_q)$ sea una matriz no singular (MNS) es:

$$p(M \text{ es MNS}) = \frac{\text{casos posibles en } \mathcal{M}_n(\mathbb{F}_q)}{\text{total de casos en } \mathcal{M}_n(\mathbb{F}_q)} = \frac{\prod_{j=0}^{n-1}(q^n - q^j)}{(q^n - 1)^n},$$

donde el número de casos posibles, ya lo habíamos calculado con anterioridad: la primer columna de una tal matriz debe ser no cero (por lo que hay $(q^n - 1)$ vectores posibles para la primer columna), la segunda columna, junto con la primera deben formar un conjunto l.i. (por lo que hay $(q^n - q)$ vectores posibles para la segunda columna), etc.

Como esta es una probabilidad uniforme, entonces el valor esperado (de la selección aleatoria de matrices de esta forma) hasta encontrar una matriz no singular es: $\epsilon =$(el número total de posibilidades)$\times$(la probabilidad de que sea no singular), es decir:

$$\epsilon = ((q^n - 1)^n) \left(\frac{\prod_{j=0}^{n-1}(q^n - q^j)}{(q^n - 1)^n} \right) = \prod_{j=0}^{n-1}(q^n - q^j).$$

3. En el inciso previo calculamos la probabilidad de que una sola matriz sea no singular, tomando aleatoriamente a sus columnas del conjunto $\mathbb{F}_q^n - \{0\}$. Entonces, dada una colección de $s_{q,n}$ matrices (seleccionadas aleatoriamente), digamos $MS = \{M_0, M_1, \ldots, M_{s_{q,n}}\}$, deseamos saber si:

   - tales matrices son no singulares, y
   - si tal colección posee matrices cuyos conjuntos de vectores asociados son distintos entre sí.

   Para el primero punto, la probabilidad de que *todas* las matrices sean no singulares es:

$$p(MS) = p((M_0 \text{ es MNS}) \wedge (M_1 \text{ es MNS}) \wedge \cdots \wedge (M_{s_{q,n}-1} \text{ es MNS}))$$

$$\begin{aligned} p(MS) &= p(M_0 \text{ es MNS}) p(M_1 \text{ es MNS}) \cdots p(M_{s_{q,n}-1} \text{ es MNS}) \\ &= \left(\frac{\prod_{j=0}^{n-1}(q^n - q^j)}{(q^n - 1)^n} \right)^{s_{q,n}}, \end{aligned}$$

el cual, podemos observar, es también un número muy pequeño.

Más aún, la probabilidad de que todas las matrices sean *no singulares* y que además sean *distintas* (MD) es:

$$\begin{aligned} p(MD \wedge MS) &= \frac{\text{No. de posibilidades de ser distintas y no singulares}}{\text{Total de colecciones de } s_{q,n} \text{ matrices}} \\ &= \frac{s_{q,n} \times (s_{q,n} - n!) \times \cdots (s_{q,n} - (s_{q,n} - 1)n!)}{\binom{(q^n-1)^n}{s_{q,n}}}. \end{aligned}$$

(Este producto también produce probabilidades considerablemente bajas.) Es decir si una colección de matrices no singulares (seleccionada aleatoriamente como hemos explicado) satisface el hecho de que todas sus matrices son distintas (salvo orden de sus columnas), entonces hemos encontramos el matroide de conjuntos l. i. sobre $\mathbb{F}_q$.

Entonces, en primer lugar para poder encontrar *una* matriz no singular, podríamos esperamos aproximadamente ϵ matrices seleccionadas aleatoriamente, donde observamos que ϵ es un número muy grande (del orden de q^n).

Por otra parte, también observamos que la probabilidad de encontrar las $s_{q,n}$ matrices que necesitamos es muy pequeña.

Puesto que la probabilidad de encontrar a una colección de matrices no singulares es uniforme, el valor esperado de tomar colecciones aleatorias, hasta encontrar una colección con las características que requerimos es: $\varepsilon =$(número total de colecciones)$\times$(probabilidad de una colección sea de bases distintas), donde el número total de colecciones es $\binom{(q^n-1)^n}{(s_{q,n})}$ (esto quiere decir, de todas la matrices posibles, seleccionar a $(s_{q,n})$ de ellas), por lo que:

$$\begin{aligned} \varepsilon &= \binom{(q^n-1)^n}{(s_{q,n})} \times \left(\frac{s_{q,n} \times (s_{q,n} - n!) \times \cdots (s_{q,n} - (s_{q,n} - 1)n!)}{\binom{(q^n-1)^n}{s_{q,n}}} \right) \\ &= s_{q,n} \times (s_{q,n} - n!) \times \cdots (s_{q,n} - (s_{q,n} - 1)n!), \end{aligned}$$

este último número claramente es muy grande y poco práctico, pues además debemos asegurarnos que tal colección sea de matrices no singulares (lo que implica una triangulación gaussiana por cada matriz) y debemos verificar que efectivamente las matrices son distintas salvo orden.

Entonces es claro por que esta no es una buena alternativa para la selección aleatoria de matroides representables.

Capítulo 4

El Sistema Propuesto

Brickell y Davenport proporcionan en [7] una caracterización de los ECS ideales: mostraron que dado un matroide representable sobre un campo finito, existe un ECS que es ideal con estructura de acceso igual a la colección de conjuntos dependientes del matroide. La caracterización presentada entonces no es efectiva en el sentido de que no se describe cómo desarrollar tales esquemas.

En esta tesis construimos un ECS ideal (como consecuencia del teorema principal de [7]) cuya estructura de acceso consiste de los conjuntos dependientes de un matroide representable.

El esquema general que proponemos en esta tesis se compone de dos fases. La primera consiste de inicializar los parámetros para la distribución de los fragmentos.

Figura 4.1: Esquema General: Fase Inicial

Para la segunda fase, existe el módulo para recuperar el secreto original (figura 4.2).

Primero, necesitamos los parámetros m, n y q del matroide representable que forma la estructura de acceso de un ECS ideal.

Figura 4.2: Esquema General del Sistema: Fase de Recuperación

En la sección 4.1 mostramos la construcción de ECS ideal elaborada por Brickell-Davenport y mostramos sus propiedades.

4.1 ECS Ideal mediante Gráficas y Matroides

La notación que ocupamos en esta sección es la introducida en la sección 2.2.4. Para un conjunto dado W, definimos el conjunto $W^{(k)} = \{U \subset W | card(U) = k\}$, que consiste de todos los subconjuntos de cardinalidad k de W. Así, $W^{(2)}$, por ejemplo, consiste de las parejas (desordenadas) de elementos en W.

Un *clan* sobre un conjunto no vacío C es una gráfica completa con conjunto de vértices C y conjunto de aristas $C^{(2)}$: Cualquier pareja forma una arista.

Dado el conjunto $\mathbb{F}_p^n$, supongamos que $\mathcal{V} = \{V_0, V_1, \ldots, V_{m-1}\}$ sea una partición de $\mathcal{V}$ en conjuntos no vacíos ajenos a pares, de cardinalidades respectivas $\nu_i = |V_i|$. La cardinalidad de V es entonces $n = \sum_{n=0}^{m-1} \nu_i$.

Sea $G_\mathcal{V}$ la gráfica con conjunto de vértices V y conjunto de aristas $A_\mathcal{V} = \bigcup_{i=0}^{m-1} V_i^{(2)}$. Entonces $G_\mathcal{V}$ es la unión, digamos disjunta, de m clanes. Así el número total de aristas en $G_\mathcal{V}$ es:

$$e(G_\mathcal{V}) = \sum_{i=0}^{m-1} \binom{\nu_i}{2}. \tag{4.1}$$

La gráfica complementaria $G_\mathcal{V}^c$ posee entonces $e(G_\mathcal{V}^c) = \binom{n}{2} - e(G_\mathcal{V})$ aristas.

Definamos ahora a la clase de subconjuntos de V que contienen a alguna arista en $G_\mathcal{V}$.

$$\mathcal{N}_\mathcal{V} = \{U \subset V | \exists a \in A_\mathcal{V} : a \subset U\}, \tag{4.2}$$

en otras palabras $\mathcal{N}_\mathcal{V} = cl(A_\mathcal{V})$. Esta clase está ordenada mediante la "inclusión de conjuntos". $\mathcal{N}_\mathcal{V}$ contiene como elemento al conjunto total V y sus elementos minimales son precisamente las aristas de $G_\mathcal{V}$. Ahora sea

$$\mathcal{M}_\mathcal{V} = \mathcal{P}(V) - \mathcal{N}_\mathcal{V}, \tag{4.3}$$

el complemento de $\mathcal{N}_\mathcal{V}$ respecto al conjunto potencia de V.

Proposición 4.1.1 *$\mathcal{M}_\mathcal{V}$ es un matroide.*

Demostración. Debemos demostrar que la colección $\mathcal{M}_\mathcal{V}$ satisface:

1. $\emptyset \in \mathcal{M}_{\mathcal{V}}$.
2. $B \in \mathcal{M}_{\mathcal{V}}\ ,\ C \subset B\ \Rightarrow\ C \in \mathcal{M}_{\mathcal{V}}$.
3. $B, C \in \mathcal{M}_{\mathcal{V}}\ ,\ |B| < |C|\ \Rightarrow\ \exists c \in C - B : B \cup \{c\} \in \mathcal{M}_{\mathcal{V}}$.

Las condiciones 1 y 2 se desprenden inmediatamente de las definiciones 4.2 y 4.3.

Para ver que vale la tercera, supongamos, por lo contrario, que hubiese dos conjuntos $B, C \in \mathcal{M}_{\mathcal{V}}$ tales que $|B| < |C|$ pero

$$\forall c \in C : B \cup \{c\} \notin \mathcal{M}_{\mathcal{V}},\ \text{ i.e.,}\ B \cup \{c\} \in \mathcal{N}_{\mathcal{V}}. \tag{4.4}$$

Como $B \cup \{c\}$ es un conjunto dependiente para todo $c \in C - B$ entonces necesariamente B debe estar en una componente de la partición, digamos V_i, y ha de contener 1 elemento. Por la condición 4.4, tenemos que $C - B \subset V_i$, lo cual implica que $C \subset V_i$.

Por lo tanto, C debe contener al menos 2 elementos en V_i, esto quiere decir que C es una arista. Esto contradice el hecho de que $C \in \mathcal{M}_{\mathcal{V}}$. □

Ahora bien, si partimos de un matroide representable entonces será posible realizar a sus conjuntos dependientes como la cerradura de parejas de elementos en una gráfica que es precisamente una unión de clanes. A partir de esta última, según vimos en la proposición 4.1.1 podemos construir un matroide. Resulta entonces que el matroide construido coincide con el original dado. Veremos esto más adelante con mayor detalle.

4.1.1 Estructura de Acceso

En esta sección vamos a considerar que el orden del campo es un número primo, es decir, la expresión $q = p^k$ implica que $k = 1$ (si acaso $q = p^k$ con $k > 1$, entonces las operaciones del campo se deben efectuar en la aritmética de polinomios módulo un polinomio irreducible).

Ahora bien, sea $\mathbb{F}_p^n$ el espacio vectorial de dimensión n sobre $\mathbb{F}_p$. Sea P el conjunto participantes de cardinalidad N y $\mathfrak{p}_0$ el distribuidor. Al partir de un matroide representable podemos suponer que existe una función inyectiva del conjunto de participantes P al espacio $\mathbb{F}_p^n - \{0\}$. Por consiguiente, en lo que sigue supondremos que el conjunto de participantes está incluido en $\mathbb{F}_p^n$ y que el matroide dado es de conjuntos l.i.. Por medio de la inclusión del conjunto de participantes en $\mathbb{F}_p^n$, los algoritmos al final del capítulo anterior se aplican sin más al conjunto de participantes.

Supondremos entonces que el matroide representable dado es $\mathcal{B}_{m,n,p}$ según aparece en la definición 2.2.10.

Puesto que deseamos fragmentar al espacio $\mathbb{F}_p^n$, primero debemos definir sobre él, una relación de equivalencia.

Sea $\mathfrak{D}_1^{(2)} = \{(x, x) | x \in \mathbb{F}_p^n - \{0\}\}$ la diagonal principal de $(\mathbb{F}_p^n - \{0\})^2$ y sea

$$\mathfrak{D}_2^{(2)} = \{(x, y) \in (\mathbb{F}_p^n - \{0\})^2 | \{x, y\} \text{ es un conjunto l.d. en } \mathcal{B}_{n,n,p}\}. \tag{4.5}$$

Definimos a la unión de los dos conjuntos anteriores, $\mathcal{E}_2 = \mathfrak{D}_1^{(2)} \cup \mathfrak{D}_2^{(2)}$. Podemos notar que esta definición es equivalente a decir que $x \sim y$ si y sólo si y es un múltiplo de x. En consecuencia, tenemos la siguiente proposición:

Proposición 4.1.2 *La relación $\sim$ definida por*

$$x \sim y \iff (x, y) \in \mathcal{E}_2$$

es una relación de equivalencia.

Demostración. Mostremos que $\sim$ es reflexiva, simétrica y transitiva. Por simplicidad consideremos la definición equivalente de $\sim$ (dos elementos están relacionados si y sólo si uno es múltiplo del otro).

- $\sim$ es *reflexiva.* Sea $x \in \mathbb{F}_p^n - \{0\}$. Entonces $x \sim x$, pues $x = 1 \cdot x$, donde 1 es la unidad multiplicativa de $\mathbb{F}_p$.
- $\sim$ es *simétrica.* Sean $\{x, y\} \in (\mathbb{F}_p^n - \{0\})^{(2)}$ y $x \neq y$. Si $x \sim y$ entonces $x = r \cdot y$. Como r es un elemento no cero en el campo $\mathbb{F}_p$, entonces existe $r^{-1} \in \mathbb{F}_p$. Multiplicando por la derecha a la expresión $x = r \cdot y$, tenemos $r^{-1} \cdot x = r^{-1} \cdot r \cdot y = y$, es decir $y = r^{-1} \cdot x$. En consecuencia $y \sim x$.
- $\sim$ es *transitiva.* Sean $\{x, y, z\} \subset \mathbb{F}_p^n - \{0\}$ tales que $x \sim y$ y $y \sim z$. Entonces existen elementos $r, r_1 \in \mathbb{F}_p$ no cero que satisfacen $x = r \cdot y$ y $y = r_1 \cdot z$. Multiplicando la expresión $y = r_1 \cdot z$ por r, tenemos $r \cdot y = r \cdot r_1 \cdot z$, es decir, $x = r \cdot y = r \cdot r_1 \cdot z = r_3 \cdot z$. Esto implica que $x \sim z$.

En consecuencia $\sim$ es una relación de equivalencia en $\mathbb{F}_p^n$. □

Observemos que dado un vector $x \in \mathbb{F}_p^n - \{0\}$, existen sólo $p - 1$ múltiplos no-nulos de x : x, $2x$, $3x$, …, $(p-2)x$, $(p-1)x$.

La relación $\sim$ induce una partición en el conjunto $\mathbb{F}_p^n - \{0\}$, digamos $V_0, V_1, \ldots, V_{m-1}$ donde cada clase de equivalencia V_i tiene $\nu_i = p - 1$ elementos (véase la figura 4.3). Esta partición también se traduce en el conjunto de participantes: el conjunto P queda particionado en clanes $V_0' = V_0|_P$, $V_1' = V_1|_P, \ldots, V_{m-1}' = V_{m-1}|_P$, donde $V_i|_p$ es la restricción del conjunto V_i a los elementos de P y cada V_i' tiene cardinalidad $\nu_i' \leq \nu_i$.

Entonces no es difícil notar que el número de clases de equivalencia en $\mathbb{F}_p^n$ es $m = \frac{p^n - 1}{p-1}$. Por lo tanto, G es una gráfica que es la unión disjunta de m clanes de cardinalidades respectivas ν_i y conjunto de aristas

$$A_G = \{R \in \mathbb{F}^{(2)} | \exists i : R \subset V_i\}. \tag{4.6}$$

Observación. Sea $\mathfrak{G}$ la gráfica complementaria de G. Entonces su conjunto de aristas es $A_{\mathfrak{G}} = \{\{x, y\} \subset \mathbb{F}^{(2)} | \{x, y\} \notin A_G\}$. Por la proposición 2.3.19 de la sección 2.3.3, hay un esquema ideal que realiza la estructura de acceso $\Gamma = cl(A_{\mathfrak{G}})$ sobre el conjunto $\mathbb{F}_p^n - \{0\}$.

La construcción de la partición del espacio $\mathbb{F}_p^n - \{0\}$ como una unión de clanes ($\bigcup_i V_i$) se muestra enseguida:

1. Se considera una pila L con todos los vectores en $\mathbb{F}_p^n - \{0\}$;
2. se inicializa una variable $i = 0$;

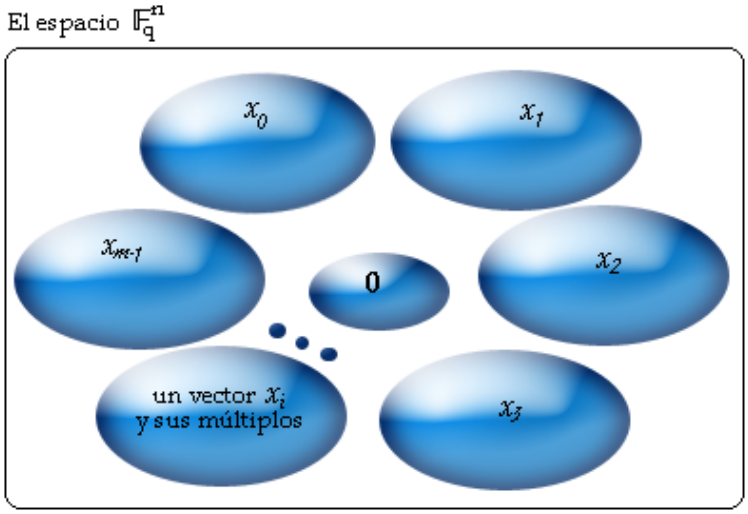


Figura 4.3: La partición del espacio $\mathbb{F}_p^n$

3. se quita el primer elemento x de la pila [1] L y se inserta en un clan vacío V_i;

4. se quitan a todos los elementos y de la pila L que estén relacionados con x (véase la relación 4.5 y el pie de página abajo) y se insertan en V_i;

5. se incrementa la variable i;

6. si L no está vacía, regresamos al paso 3.

7. Este algoritmo tiene como salida a $\bigcup_i V_i$ (la unión de los clanes).

El seudocódigo de este algoritmo se encuentra en el Apéndice 1. **Algoritmo 19. Partición**$[n, p]$.

A tal partición, le aplicamos la construcción bosquejada en la demostración de la proposición 2.3.20 y obtenemos un esquema de Shamir con umbral 2 en cada clan, cuya estructura de acceso Γ coincide (como vimos en la sección 2.3.3) con la cerradura de A_G. Vamos a denotar por Γ_0 a la colección de conjuntos que son minimales en Γ y, de hecho, Γ_0 no es otro que A_G.

Para recuperar el secreto s necesitamos que los participantes de un conjunto de la estructura de acceso Γ proporcionen sus fragmentos, es decir necesitamos que dos participantes que provengan de un mismo clan, proporcionen sus fragmentos.

Si se quisiera agrandar el tamaño del umbral a un valor, digamos $k \geq 2$ podemos generalizar la construcción previa. La relación (4.5) define una de equivalencia. Toda relación de equivalencia es, estrictamente, una colección, digamos $\mathcal{E}_2$ de parejas. Así la relación (4.5) se generaliza de manera recurrente. Supongamos ya definidos los conjuntos

[1] Si queremos encontrar la partición sobre el conjunto de participantes P, entonces debemos verificar que los vectores respectivos correspondan a participantes en P, pero podemos suponer, sin pérdida de generalidad, que el número de participantes es $N = p^n - 1$ (el 0 está reservado para el distribuidor).

$\mathfrak{D}_i^{(j)}$, con $1 \leq i \leq j < k$. Hagamos, para $i \leq k-1$:

$$x = (x_1, x_2, x_3, \ldots, x_k) \in \mathfrak{D}_i^{(k)} \Leftrightarrow \begin{array}{l} i \text{ coordenadas de } x \text{ forman un vector en } \mathfrak{D}_i^{(i)} \\ \text{y las restantes } k-i \text{ son repeticiones de ellas} \end{array}$$

y

$$(x_1, x_2, x_3, \ldots, x_k) \in \mathfrak{D}_k^{(k)} \Leftrightarrow \{x_1, \ldots, x_k\} \text{ es un conjunto l.d. minimal.} \tag{4.7}$$

Entonces, hagamos

$$(x_1, \ldots, x_k) \in \mathcal{E}_k \Leftrightarrow (x_1, \ldots, x_k) \in \bigcup_{i=1}^{k} \mathfrak{D}_i^{(k)}. \tag{4.8}$$

Para cada $x_1 \in \mathbb{F}_p^n$ sea

$$[x_1]_k = \{x_2 \in \mathbb{F}_p^n | \exists x_3, \ldots, x_k \in \mathbb{F}_p^n : (x_1, x_2, x_3, \ldots, x_k) \in \mathcal{E}_k\}.$$

Se puede ver que $([x_1]_k)_{x_1 \in \mathbb{F}_p^n}$ es una partición de $\mathbb{F}_p^n$ que lo divide como una unión disjunta de k-hipergráficas completas y a partir de ahí se repite la construcción presentada en el Algoritmo 19 de Partición en clanes. Aquí es necesario mencionar que las k-adas $(x_1, \ldots, x_k)$ que están $\mathfrak{D}_k^{(k)}$ según (4.7) son muy numerosas: Si se elige a los primeros vectores $x_1, \ldots, x_{k-1}$ de manera que ellos formen un conjunto l. i., entonces el k-ésimo puede elegirse como una combinación lineal de ellos que los involucre a todos, hay pues $(p-1)^{k-1}$ posibles selecciones del k-ésimo vector x_k. Así pues, el número de hiperaristas en $\mathfrak{D}_k^{(k)}$ es del orden de $(p-1)^{k-1} \prod_{j=0}^{k-2} \left(\frac{p^n - p^j}{j+1}\right)$. Enumerándolas, se puede enumerar también a todo el conjunto $\mathcal{E}_k$, y, de acuerdo con este conteo, se tendrá un análogo a la pila L en el Algoritmo 19.

Un conjunto l. d. minimal en $\mathbb{F}_p$ es aquel cuyos subconjuntos propios son l. i. (si A es un conjunto minimal entonces no puede contener propiamente a un subconjunto que sea l. d.), entonces la colección de conjuntos l. d. minimales son precisamente los circuitos de un matroide representable sobre $\mathbb{F}_p$.

La relación definida en la ecuación 4.7 generaliza a la relación de equivalencia definida en 2.3 sobre circuitos. De 4.7 se puede ver que el espacio $\mathbb{F}_p^n$ se partirá en una unión disjunta de conjuntos de manera tal que cada k-ada en uno de estos conjuntos estará en el conjunto $\mathcal{E}_k$ (en otras palabras, la relación que define a $\mathcal{E}_k$ es congruente con la partición). Consecuentemente podría realizarse una construcción del ECS del tipo de Brickell-Davenport, pero esta vez de umbral k.

4.2 Construcción del ECS

En las secciones 4.2.1 y 4.2.2 mostraremos las fases del sistema propuesto (la distribución de los fragmentos y la recuperación del secreto) tal como se delinea en las figuras 4.1 y 4.2.

Los resultados de las pruebas hechas a estos algoritmos se muestran en el capítulo 5.

4.2.1 La Partición del Secreto

Aquí hacemos la observación de que la estructura de acceso a utilizar es la que está constituida por la clausura de los clanes involucrados.

En la sección anterior vimos el algoritmo que efectúa la partición del conjunto $\mathbb{F}_p^n - \{0\}$ (bajo la relación $\sim$) en m conjuntos $V_0, V_1, \ldots, V_{m-1}$, donde pudimos notar que la cardinalidad de cada clan V_i es $\nu_i := |V_i| = p - 1$ y en consecuencia el número de clanes es $m = \frac{p^n-1}{p-1}$. También recordamos que la partición del conjunto $\mathbb{F}_p^n$ se traduce en una partición sobre el conjunto de participantes P, como la unión de los clanes V_i' que mencionamos anteriormente.

La gráfica $G = (P, A_G)$, con conjunto de aristas $A_G = \bigcup_i V_i'^{(2)}$, es unión disjunta de m clanes.

Algoritmo para la Partición del Secreto

En base a lo anterior, si queremos fragmentar un secreto s dado, primero a cada clan V_i, $0 \leq i < m$ (cada uno con ν_i elementos), le asociamos el número $y_i \in \mathbb{F}_{p_1} - \{0\}$, (donde $p_1 > p^n$ y $p_1 > s$ es un número primo, mismo que se puede seleccionar previamente). Observemos que por la elección de p_1, $s \in \mathbb{F}_{p_1}$.

Recordemos además, que cada participante $\mathfrak{p}_{i,j} \in V_i$ $(j = 1, \ldots, p-1)$ se ha identificado con un vector en $\mathbb{F}_p^n$, en consecuencia tiene asociado un número entero $x_{i,j} \in [\![0, p^n - 1]\!] \subset \mathbb{F}_{p_1}$, a saber el que resulta cuando el vector con el que se identifica, se lee como la representación de un entero en base p.

La regla de distribución que vimos en la proposición 2.3.20 se bosqueja a continuación:

Supongamos como antes que $s \in \mathbb{F}_{p_1}$ es el secreto que se desea fragmentar.

- Para cada $i = 0, \ldots, m-1$, se elige un elemento $y_i \in \mathbb{F}_{p_1} - \{0\}$.
- Para cada $i = 0, \ldots, m-1$, y cada j tal que $1 \leq j \leq p-1$, se realiza lo siguiente:
  - El distribuidor $\mathfrak{p}_0$ asocia al participante $\mathfrak{p}_{i,j} \in V_i$ el vector $(1, x_{i,j}) \in \mathbb{F}_{p_1}^2$ (observamos que los $x_{i,j} \in \mathbb{F}_{p_1}$ son distintos a pares por lo que esta asociación de participantes a vectores es inyectiva).
  - Al participante $\mathfrak{p}_{i,j}$ se le entrega como fragmento el número $s_{i,j} = (s, y_i) \cdot (1, x_{i,j}) = (s + y_i x_{i,j}) \; mod \, p_1$.

Con estas condiciones, si $\{\mathfrak{p}_{i,k}, \mathfrak{p}_{i,r}\} \in \Gamma$ (es decir, están en el mismo clan) proporcionan sus fragmentos, digamos $s_{i,k}$ y $s_{i,r}$, entonces la ordenada al origen de la recta que pasa por los puntos $(x_{i,k}, s_{i,k})$ y $(x_{i,r}, s_{i,r})$, resulta ser el secreto.

Observamos que la estructura de acceso es $\Gamma = cl(A_G)$ (A_G es el conjunto de aristas definido en la relación 4.6). El algoritmo explícito de la fragmentación se encuentra en el apéndice 1: **Algoritmo 20: Fragmentar**$[s, \bigcup_i V_i, m, \nu_i]$, en el cual, el primo p_1 se selecciona de tal forma que $p_1 > p^n$ y $p_1 > s$.

Complejidad del Algoritmo para la Partición del Secreto

Aquí utilizamos la misma notación que antes, el número asociado al participante $\mathfrak{p}_{i,j}$ lo denotado por $z_{i,j}$. Si N es el número de participantes, se debe satisfacer que $N < p^n$. El algoritmo de fragmentación requiere sólo un producto y una suma en el campo $\mathbb{F}_{p_1}$, (donde p_1 es un primo calculado tal que $p_1 > p^n$ y $p_1 > s$). Entonces, la complejidad de este algoritmo es lineal respecto al número de participantes, es decir, tiene complejidad $O(N)$.

4.2.2 La Recuperación del Secreto

Un subconjunto de participantes que esté es la estructura de acceso Γ puede recuperar el secreto, pues (como se mencionó anteriormente) con al menos dos elementos que estén relacionados, el secreto esta únicamente determinado.

Un subconjunto de participantes que no esté es la estructura de acceso Γ (es decir, que no estén en la cerradura de los clanes) no pueden recuperar el secreto.

Ésta es una noción limitada a 2 participantes, sin embargo, puede generalizarse a $k < p - 1$ participantes utilizando k-clanes, y la relación (4.7), da origen a k-clanes, en lugar de los 2-clanes que aquí utilizamos (debido a la relación de equivalencia que propusimos).

Algoritmo para la Recuperación del Secreto

Supongamos que dos participantes $\{\mathfrak{p}_{i,j}, \mathfrak{p}_{l,r}\}$ reúnen sus fragmentos, digamos $s_{i,j}$ y $s_{l,r}$ (si son más de dos participantes, debemos probar el siguiente algoritmo con todos los subconjuntos de dos elementos posibles).

Entonces:

- $\mathfrak{p}_0$ verifica si $\{\mathfrak{p}_{i,j}, \mathfrak{p}_{l,r}\}$ están en el mismo clan (en otras palabras, que los vectores en $\mathbb{F}_p^n$ asociados a los participantes $\mathfrak{p}_{i,j}$ y $\mathfrak{p}_{l,r}$ están relacionados, es decir son l.d.),

- si es así entonces sea $t = i = l$ el índice del clan V_t al que pertenecen los participantes. Entonces $\{\mathfrak{p}_{t,j}, \mathfrak{p}_{t,r}\}$ pueden recobrar el secreto (con sus respectivos fragmentos $s_{t,j}$ y $s_{t,r}$), utilizando el algoritmo de Shamir (ecuación 2.11). Por lo que tenemos que el secreto es:

$$s = c_1 s_{t,j} + c_2 s_{t,r},$$

donde:

$$c_1 = \frac{x_{t,r}}{x_{t,r} - x_{t,i}} \;\; y \;\; c_2 = \frac{x_{t,i}}{x_{t,i} - x_{t,r}},$$

recordemos que los números $x_{t,j}$ y $x_{t,r}$ son los números en $[\![1, p^n - 1]\!]$ asociados a los participantes $\mathfrak{p}_{t,j}$ y $\mathfrak{p}_{t,r}$ respectivamente.

El algoritmo de la recuperación del secreto se encuentra en el apéndice 1: **Algoritmo 21: Recuperación**$[A(\subset P), p_1, \Gamma_0]$ que se encuentra en el apéndice 1.

Complejidad del Algoritmo para la Recuperación del Secreto

Aquí hacemos referencia de recuperación del secreto: **Algoritmo 21**. El paso 1 del algoritmo depende del número de elementos de A, pues deseamos encontrar una pareja de participantes de A, que se encuentre en Γ_0, por lo que se han de probar a lo más $\binom{|A|}{2}$ conjuntos.

El peor caso es cuando $|A| = N$ (el número total de participantes), en cuyo caso la complejidad de la búsqueda de tal pareja (de tal forma que sean l.d.) es $O(\frac{N(N-1)}{2}) = O(N^2)$.

En los pasos 3 y 4 calculamos 4 sumas, 2 inversiones (módulo p_1) y 4 multiplicaciones, por los que la complejidad de estos dos pasos es $O(1)$ (constante).

Entonces, la complejidad de la recuperación del secreto depende básicamente del algoritmo de búsqueda de tal pareja de participantes. En consecuencia, la complejidad es $O(N^2)$.

4.2.3 Idealidad y Perfección

El ECS aquí propuesto es ideal y perfecto como consecuencia del teorema de Brickell-Davenport.

Idealidad

El esquema que proponemos es ideal:

- Si $\mathfrak{p}$ es un participante en el conjunto P, $\mathfrak{p}$ tiene asociado un número $x_{\mathfrak{p}} \in \mathbb{F}_{p_1}$ y además tiene asociado un fragmento $s_{\mathfrak{p}} \in \mathbb{F}_{p_1}$, por lo que la tasa de información para tal participante $\mathfrak{p}$ es:

$$\rho_{\mathfrak{p}} = max\left\{\frac{\log_2|s|}{\log_2|s_{\mathfrak{p}}|} : s \in \mathbb{F}_{p_1}\right\},$$

  podemos notar que éste número satisface $\rho_{\mathfrak{p}} \geq 1$, pues el dominio de secretos es el mismo que el dominio de fragmentos ($\mathbb{F}_{p_1}$) y s puede tomar cualquier valor en tal dominio.

- La tasa de información del esquema es:

$$\rho = min\{\rho_{\mathfrak{p}} : \mathfrak{p} \in P\}.$$

  Por la observación del punto anterior, podemos ver que $\rho = 1$.

Dado que para este esquema $\rho = 1$, el esquema es ideal.

Perfección

Supongamos que un conjunto de dos participantes $A = \{\mathfrak{p}_{i,j}, \mathfrak{p}_{l,r}\} \subset P$ proporcionan sus fragmentos, digamos $s_{i,j}$ y $s_{l,r}$, de tal forma que $A \notin \Gamma$ (esto quiere decir que los vectores

asociados a tales participantes no son l.d. y en consecuencia pertenecen a clanes disjuntos V_i y V_l). Entonces, tenemos dos ecuaciones con tres incógnitas (el secreto s, y_i y y_l):

$$\begin{aligned} s_{i,j} &= s + y_i\, x_{i,j} \; mod\, p_1 \\ s_{l,r} &= s + y_l\, x_{l,r} \; mod\, p_1 \end{aligned}$$

El sistema entonces no tiene solución única en la incógnita s, por lo que el ECS es perfecto.

4.2.4 Ejemplo del Sistema

- Supongamos que se tiene un número de participantes $N = 317$, digamos $P = \{\mathfrak{p}_1, \mathfrak{p}_2, \ldots, \mathfrak{p}_{317}\}$.
- Elegimos a $p = 7$ y $n = 3$ los parámetros para el espacio vectorial $\mathbb{F}_p^n = \mathbb{F}_7^3$, mismo que posee $343 (= 7^3)$ elementos. Entonces el conjunto de participantes se puede identificar con un conjunto de N vectores incluido en el espacio $\mathbb{F}_7^3$.
- El matroide $\mathcal{B}_{n,n,p} = \mathcal{B}_{3,3,7}$ posee un número de bases (es decir, de conjuntos independientes maximales) igual a

  $$\frac{(7^3-1)(7^3-7)(7^3-7^2)}{3!} = 5,630,688.$$

  (Un conjunto está en el matroide $\mathcal{B}_{3,3,7}$ cuando y sólo cuando está incluido en alguna base de $\mathcal{B}_{3,3,7}$.) Si, por ejemplo, tomamos $m = 2$ como cardinalidad de conjuntos independientes maximales, el matroide $\mathcal{B}_{m,m,p} = \mathcal{B}_{2,2,7}$, posee un número de bases igual a:

  $$\frac{(7^2-1)(7^2-7)}{2!} = 1,008,$$

  por lo que un matroide de la forma $\mathcal{B}_{2,3,7}$ posee también ese mismo número de bases ($\mathcal{B}_{2,3,7}$ es, de hecho, isomorfo a $\mathcal{B}_{2,2,7}$).
- La relación 2.2que define a matroides de la forma $\mathcal{B}_{m,n,p}$ es relativa a un subespacio fijo V en $\mathbb{F}_p^n$. En el ejemplo que aquí estamos desarrollando tenemos tantas posibilidades como haya subespacios de dimensión $m = 2$ en el de dimensión $n = 3$. Cuando se genera aleatoriamente un matroide representable, éste será de la forma $\mathcal{B}_{2,3,7}$. La primera parte del algoritmo da esta selección del matroide.
- Ahora, hemos de identificar al conjunto de N participantes con un subconjunto de $\mathbb{F}_7^3$. Aquí se tiene que el número de posibilidades es, precisamente: $N!\binom{7^3}{N}$, el cual es un entero del orden de 10^{695}. En particular, tomemos una correspondencia muy sencilla. Podemos poner en correspondencia natural a los participantes con elementos de $\mathbb{F}_7^3$ como sigue: $\mathfrak{p}_1 \leftrightarrow 1 \leftrightarrow (0,0,1) \in \mathbb{F}_7^3$; $\mathfrak{p}_2 \leftrightarrow 2 \leftrightarrow (0,0,2) \in \mathbb{F}_7^3$; $\ldots$; $\mathfrak{p}_{317} \leftrightarrow 317 \leftrightarrow (6,3,2) \in \mathbb{F}_7^3$.
- Supongamos también que se quiere fragmentar el secreto $s = 257$. Elijamos, siguiendo la construcción descrita, al primo p_1 como 347, pues éste satisface $347 > 343 = 7^3$ y $347 > 257$. La partición descrita en la construcción, fracciona al espacio $\mathbb{F}_7^3 - \{0\}$ en $\frac{7^3-1}{7-1} = 57$ clanes, cada uno de ellos con $7 - 1 = 6$ elementos.

- Debido a que hay 57 clanes, para ilustrar el proceso de asignación de fragmentos, tomaremos en particular, la clase de equivalencia (o clan) del elemento $(2,3,2) \in \mathbb{F}_7^3$. Este (digamos que es el k-ésimo), consiste de los 6 elementos:

$$\begin{aligned} V_k &= \{(2,3,2),(4,6,4),(6,2,6),(1,5,1),(3,1,3),(5,4,5)\} \\ &\approx \{121,242,314,85,157,278\} \\ &\approx \{\mathfrak{p}_{121},\mathfrak{p}_{242},\mathfrak{p}_{314},\mathfrak{p}_{85},\mathfrak{p}_{157},\mathfrak{p}_{278}\} \end{aligned}$$

  Supongamos que $y_k = 54 \in \mathbb{F}_{347}$ es el número asociado al clan V_k entonces el algoritmo de fragmentación para cada elemento en el clan V_k produce:

$$\begin{aligned} \text{Al participante } \mathfrak{p}_{121} \text{ se asigna} &: s_{121} = 257 + (54)(121)\, mod\, 347 = 198 \\ \text{Al participante } \mathfrak{p}_{242} \text{ se asigna} &: s_{242} = 257 + (54)(242)\, mod\, 347 = 139 \\ \text{Al participante } \mathfrak{p}_{314} \text{ se asigna} &: s_{314} = 257 + (54)(314)\, mod\, 347 = 210 \\ \text{Al participante } \mathfrak{p}_{85} \text{ se asigna} &: s_{85} = 257 + (54)(85)\, mod\, 347 = 336 \\ \text{Al participante } \mathfrak{p}_{157} \text{ se asigna} &: s_{157} = 257 + (54)(157)\, mod\, 347 = 60 \\ \text{Al participante } \mathfrak{p}_{278} \text{ se asigna} &: s_{278} = 257 + (54)(278)\, mod\, 347 = 1 \end{aligned}$$

  Consideremos también la clase de equivalencia (o clan) del elemento $(0,5,3) \in \mathbb{F}_7^3$. El clan (digamos l) consiste de los 6 elementos:

$$\begin{aligned} V_k &= \{(0,5,3),(0,3,6),(0,1,2),(0,6,5),(0,4,1),(0,2,4)\} \\ &\approx \{38,28,9,47,29,18\} \\ &\approx \{\mathfrak{p}_{38},\mathfrak{p}_{28},\mathfrak{p}_{9},\mathfrak{p}_{47},\mathfrak{p}_{29},\mathfrak{p}_{18}\} \end{aligned}$$

  Supongamos que $y_l = 100 \in \mathbb{F}_{347}$ es el número asociado al clan V_l entonces el algoritmo de fragmentación para cada elemento en el clan V_l produce:

$$\begin{aligned} \text{Al participante } \mathfrak{p}_{38} \text{ se asigna} &: s_{38} = 257 + (100)(38)\, mod\, 347 = 240 \\ \text{Al participante } \mathfrak{p}_{28} \text{ se asigna} &: s_{28} = 257 + (100)(28)\, mod\, 347 = 281 \\ \text{Al participante } \mathfrak{p}_{9} \text{ se asigna} &: s_{9} = 257 + (100)(9)\, mod\, 347 = 116 \\ \text{Al participante } \mathfrak{p}_{47} \text{ se asigna} &: s_{47} = 257 + (100)(47)\, mod\, 347 = 99 \\ \text{Al participante } \mathfrak{p}_{29} \text{ se asigna} &: s_{29} = 257 + (100)(29)\, mod\, 347 = 34 \\ \text{Al participante } \mathfrak{p}_{18} \text{ se asigna} &: s_{18} = 257 + (100)(18)\, mod\, 347 = 322 \end{aligned}$$

- En cuanto a la recuperación del secreto, supongamos que los participantes $\{\mathfrak{p}_{47},\mathfrak{p}_{85},\mathfrak{p}_{18}\}$ proporcionan sus fragmentos $\{s_{47} = 99, s_{85} = 336, s_{18} = 322\}$. Los vectores en $\mathbb{F}_7^3$ asociados a tales participantes son: $(0,6,5) \sim \mathfrak{p}_{47}$, $(1,5,1) \sim \mathfrak{p}_{85}$ y $(0,2,4) \sim \mathfrak{p}_{18}$. Los vectores asociados a los participantes $\mathfrak{p}_{18}$ y $\mathfrak{p}_{47}$ sí se encuentran relacionados, mientras que alguno de estos con el vector asociado al participante $\mathfrak{p}_{85}$ forman un conjunto l.i., entonces el secreto está dado por la expresión:

$$s = c_1 s_{18} + c_2 s_{47},$$

donde:

$$c_1 = \frac{47}{47-18} mod\ 347 = 217 \quad y \quad c_2 = \frac{18}{18-47}\ mod\ 347 = 131,$$

sustituyendo estos valores en la expresión para s obtenemos:

$$s = (217)(322) + (131)(99)\ mod\ 347 = 257,$$

puesto que este número es igual a s, hemos recuperado el secreto original.

Por otra parte si utilizamos a los participantes $\mathfrak{p}_{85}$ y $\mathfrak{p}_{47}$ (que no se encuentran relacionados) y procedemos de la misma manera que antes, quisiéramos que el secreto estuviera dado por una expresión:

$$s' = c_1 s_{85} + c_2 s_{47},$$

donde:

$$c_1 = \frac{47}{47-85} mod\ 347 = 154 \quad y \quad c_2 = \frac{85}{85-47}\ mod\ 347 = 194,$$

y haciendo las sustituciones:

$$s' = (154)(336) + (194)(99)\ mod\ 347 = 162,$$

por lo que es claro que $s' = 162$ es distinto del secreto original $s = 257$.

- En resumen, los primeros cuatro puntos anteriores permiten identificar al conjunto de participantes con un subconjunto del espacio $\mathbb{F}_7^3$ y mediante una selección aleatoria en este espacio de un matroide de la forma $B_{2,3,7}$, con la identificación hecha, obtenemos un matroide representable en el conjunto de participantes isomorfo al seleccionado. Los puntos siguientes en este ejemplo construyen un ECS sobre $\mathbb{F}_7^3$ el cual se transporta directamente al conjunto de participantes vía las identificaciones establecidas.

Capítulo 5

Resultados y Discusión

En esta sección mostramos las pruebas de desempeño de los algoritmos presentados.

La implementación de los programas necesarios para la selección aleatoria de matroides representables, requiere de diversas operaciones complejas como la triangulación gaussiana de matrices o la multiplicación de matrices por lo que utilizar el lenguaje C para su implementación tiene las siguientes ventajas: el código generado es compacto, corresponde casi de manera literal a la formulación de los algoritmos, es legible, sumamente rápido (debido a que C es un lenguaje de nivel medio), transportable y puede, además, ejecutarse en cualquier máquina y bajo cualquier sistema operativo. Vale la pena mencionar que, sin embargo, la selección aleatoria de matroides representables conlleva el listado de todas sus bases, y tal lista es de longitud $O(q^{n^2})$, donde q y n son el orden del campo finito y la dimensión del espacio vectorial respectivamente. Así que, a pesar de las ventajas de C, poco se puede lograr en cuanto a eficiencia.

Por otro lado, la implementación de la fragmentación y la recuperación de secretos requieren de operaciones convencionales en campos finitos (sumas, multiplicaciones y en particular, para la recuperación del secreto se requieren de dos inversiones en el campo). Los desarrolladores de `Java` lo han dotado de una gran variedad de "paquetes" (*packages*) con funciones primitivas de géneros variados. Algunos de éstos son de gran utilidad en criptografía: paquetes que proporcionan aritmética entera de grandes números, operaciones simbólicas en estructuras abstractas, funciones de *hashing*, de manejo y administración de llaves, etc. Debido a esto, la implementación de esta parte fue elaborada en `Java`. Su "librería" de criptografía permite, por ejemplo, encontrar números primos de tamaño arbitrario (para la fragmentación se selecciona un número primo que sea mayor que el número de participantes y mayor que el secreto). `Java` también ofrece portabilidad, por lo que es posible correr los programas en diversas plataformas. Así, en la programación de los algoritmos pudimos concentrarnos en los detalles propios de ellos y pasar por alto la implementación de las funciones primitivas. Los programas en `Java` de nuestra implementación pueden transcribirse a C sin más, siempre que se cuente con las funciones primitivas en C ya implementadas (localización de primos, aritmética entera de grandes números, y aritmética en campos finitos). Naturalmente, debido a que `Java` es un lenguaje de alto nivel ocupa, ya corriendo, tiempos mayores que los que ocuparía C, acaso siempre en una razón constante.

La computadora utilizada para nuestras pruebas posee las siguientes características:

Procesador Pentium 4 a 1.60 GHZ, Memoria RAM de 256 MB, Disco Duro con 80 GB y la plataforma Windows XP.

En seguida listamos los algoritmos y sus respectivas pruebas. Se explica el proceso de evaluación del algoritmo y el número de corridas hechas. Los tiempos reportados en general estarán en milisegundos (ms), a menos que se indique lo contrario, y el valor de n comprenderá entre 1 y 7.

- **Algoritmo 8. ComoNumero**$[\mathbf{x}, n, q]$

  Este procedimiento realiza una mera conversión de vectores a números enteros y, como ya vimos, es lineal respecto a la dimensión n de los vectores. El parámetro p es irrelevante en las operaciones involucradas. En las siguientes tablas observamos, la dependencia lineal respecto a n y el carácter constante del tiempo respecto a p. Aquí seleccionamos un vector x aleatoriamente por cada corrida. Tomamos el tiempo promedio de 10^6 corridas.

n	p	p^n	Tiempo (ms)
3	2	8	0.00001098901098901098
4	2	16	0.00016483516483516483
5	2	32	0.00016483516483516483
6	2	64	0.00021978021978021977
7	2	128	0.00027472527472527474

n	p	p^n	Tiempo (ms)
1	3	3	0.00000068725468424858
2	3	9	0.00005494505494505494
3	3	27	0.00010989010989010988
4	3	81	0.00016483516483516483
5	3	243	0.00016483516483516483
6	3	729	0.00021978021978021977
7	3	2187	0.00021978021978021977

n	p	p^n	Tiempo (ms)
2	5	25	0.00010989010989010988
3	5	125	0.00010989010989010988
4	5	625	0.00016483516483516483
5	5	3125	0.00016483516483516483
6	5	15625	0.00016483516483516483
7	5	78125	0.00021978021978021977

n	p	p^n	Tiempo (ms)
2	7	49	0.00010989010989010988
3	7	343	0.00010989010989010988
4	7	2401	0.00016483516483516483
5	7	16807	0.00021978021978021977
6	7	117649	0.00021978021978021977
7	7	823543	0.00027472527472527474

- **Algoritmo 9. ComoVector**$[a]$

  Este procedimiento realiza la conversión de un número a vector, en las siguientes tablas, también observamos que existe dependencia lineal respecto al número n. Aquí consideramos números aleatorios en el intervalo $[\![0, p^n - 1]\!]$. Tomamos el tiempo promedio de 10^6 corridas.

n	p	p^n	Tiempo (ms)
3	2	8	0.00009890109890109899
4	2	16	0.00015384615384615397
5	2	32	0.00009890109890109899
6	2	64	0.00014087912087120894
7	2	128	0.00015384615384615397

n	p	p^n	Tiempo (ms)
2	3	9	0.00054945054945054949
3	3	27	0.00060439560439560446
4	3	81	0.00060439560439560446
5	3	243	0.00072151512364895412
6	3	729	0.00072314612257438213
7	3	2187	0.00078164015175638248

n	p	p^n	Tiempo (ms)
2	5	25	0.00082417582417582424
3	5	125	0.00082417582417582424
4	5	625	0.00082417582417582424
5	5	3125	0.00087912087912087910
6	5	15625	0.00087912087912087910
7	5	78125	0.00087912087912087910

n	p	p^n	Tiempo (ms)
2	7	49	0.00082417582417582424
3	7	343	0.00082417582417582424
4	7	2401	0.00082417582417582424
5	7	16807	0.00087912087912087910
6	7	117649	0.00087912087912087910
7	7	823543	0.00087912087912087910

- **Algoritmo 10. VectorSiguiente**[x]

  En este algoritmo notamos que existe dependencia del tiempo respecto al parámetro n. Aquí consideramos al vector x, inicializado en $(1, 0, \ldots, 0)$ y sobre el mismo escribimos al vector siguiente (si el "siguiente vector" es ultimo, x se reinicia nuevamente a $(1, 0, \ldots, 0)$). Tomamos el tiempo promedio de 10^7 corridas.

n	p	p^n	Tiempo (ms)
2	2	4	0.00006043956043956044
3	2	8	0.00006043956043956044
4	2	16	0.00006043956043956044
5	2	32	0.00006043956043956044
6	2	64	0.00006043956043956044
7	2	128	0.00006593406593406593

n	p	p^n	Tiempo (ms)
2	3	9	0.00006593406593406593
3	3	27	0.00007142857142857143
4	3	81	0.00006593406593406593
5	3	243	0.00006593406593406593
6	3	729	0.00005098901098901098
7	3	2187	0.00006593406593406593

n	p	p^n	Tiempo (ms)
2	5	25	0.00006593406593406593
3	5	125	0.00006593406593406593
4	5	625	0.00006593406593406593
5	5	3125	0.00007142857142857143
6	5	15625	0.00007142857142857143
7	5	78125	0.00008593134658769870

n	p	p^n	Tiempo (ms)
2	7	49	0.00007142857142857143
3	7	343	0.00007692307692307692
4	7	2401	0.00008241758241758242
5	7	16087	0.00008241758241758242
6	7	117649	0.00008241758241758242
7	7	823543	0.00008912514632457429

- **Algoritmo 11. InsercionDeColumna**[M, x]

  Este algoritmo sólo depende del parámetro n, y trata básicamente de crear un nodo y poner ahí al vector x. Aquí consideramos una matriz seleccionada aleatoriamente. Una corrida estaba constituida de la inserción aleatoria de un vector x. Debido a la simplicidad de este procedimiento, sólo colocamos una pequeña tabla para observar el rango de tiempo. Tomamos el tiempo promedio de 10^5 corridas.

n	Tiempo (ms)
2	0.000549456872873890054 9
3	0.000573872349450549450 5
4	0.000678278572450549441 0
5	0.000628623794505494417 7
6	0.000836850549441069859 9
7	0.001549879268717848715 9

- **Algoritmo 12. TriangulacionGaussiana**[M]

  Aquí podemos observar un crecimiento del tiempo respecto al tamaño n ($O(n^3)$). Podemos observar que la complejidad de este algoritmo depende del parámetro n. Una buena elección de p y n hacen que éste algoritmo sea eficiente. Aquí consideramos una matriz M seleccionada aleatoriamente por cada corrida. Tomamos el tiempo promedio de 10^6 corridas.

n	p	p^n	Tiempo (ms)
2	2	4	0.0011538461538461539
3	2	8	0.0015384615384615385
4	2	16	0.0019780219780219781
5	2	32	0.0032967032967032969
6	2	64	0.0041208791208791213
7	2	128	0.0047802197802197801

n	p	p^n	Tiempo (ms)
2	3	9	0.0018131868131856813
3	3	27	0.0024175824175824178
4	3	81	0.0030769230769230709
5	3	243	0.0037362637362637631
6	3	729	0.0043956043956043598
7	3	2187	0.0052197802197802198

n	p	p^n	Tiempo (ms)
2	5	25	0.0017582417582417582
3	5	125	0.0024175824175824178
4	5	625	0.0030219780219780219
5	5	3125	0.0037362637362637363
6	5	15625	0.0045604395604395602
7	5	78125	0.0052747252747252746

n	p	p^n	Tiempo (ms)
2	7	49	0.0030769230769230709000
3	7	343	0.0039560439560439560000
4	7	2401	0.0053846153846153503000
5	7	16807	0.0066483516483516487100
6	7	117649	0.0067582417582417582100
7	7	823543	0.0092857142857142864800

- **Algoritmo 13. RevisionLI**$[M, \mathbf{x}]$

  La operación más fuerte en este algoritmo es la (única) triangulación gaussiana de una matriz que se efectúa. Aquí consideramos una matriz M y un vector x que seleccionamos aleatoriamente, por cada corrida. Tomamos el tiempo promedio de 10^5 corridas.

n	p	p^n	Tiempo (ms)
2	2	4	0.00032967032967032966500
3	2	8	0.00049450549450549452500
4	2	16	0.00060439560439560446900
5	2	32	0.00082417582417582424500
6	2	64	0.00104395604395604402000
7	2	128	0.00120879120879120894000

n	p	p^n	Tiempo (ms)
2	3	9	0.00043956043956043955
3	3	27	0.00071428571428571430
4	3	81	0.00087912087912087910
5	3	243	0.00109890109890109899
6	3	729	0.00137362637362637363
7	3	2187	0.00153846153846153855

n	p	p^n	Tiempo (ms)
2	5	25	0.00038461538461538463
3	5	125	0.00054945054945054949
4	5	625	0.00065934065934065933
5	5	3125	0.00087912087912087910
6	5	15625	0.00109890109890109899
7	5	78125	0.00126373626373626369

n	p	p^n	Tiempo (ms)
2	7	49	0.00054945054945054949
3	7	343	0.00054945054945054949
4	7	2401	0.00082417582417582424
5	7	16807	0.00104395604395604402
6	7	117649	0.00131868131868131866
7	7	823543	0.00148351648351648358

Vemos aquí, que los procedimientos enlistados arriba habían sido relativamente rápidos por ser elementales. Los siguientes algoritmos están involucrados en el recuento exhaustivo de las bases del matroide de conjuntos l.i. Veremos que sus tiempos se incrementan grandemente.

- **Algoritmo 14. ListaDeLIs**$[M]$

  La operación más fuerte en este algoritmo es hacer una triangulación gaussiana por cada vector en $\mathbb{F}_p^n - \{0\}$. Entonces, los tiempos aquí crecen en un factor n^3 por el número p^n. Aquí consideramos una matriz M seleccionada aleatoriamente y tomamos el tiempo promedio de 10 corridas en segundos (s).

n	p	p^n	Tiempo (s)
3	2	8	0.0021978021978021977
4	2	16	0.0027472527472527474748
5	2	32	0.0027472527472527474748
6	2	64	0.0027472527472527474748
7	2	128	0.0038461538461538463 7

n	p	p^n	Tiempo (s)
2	3	9	0.002747252747252747 4800
3	3	27	0.002747252747252747 4800
4	3	81	0.002747252747252747 4800
5	3	243	0.004945054945054945 2500
6	3	729	0.014835164835164835 8000
7	3	2187	0.038461538461538462 6000

n	p	p^n	Tiempo (s)
2	5	25	0.274725274725274748
3	5	125	0.329670329670329665
4	5	625	0.714285714285714302
5	5	3125	3.131868131868131840
6	5	15625	17.692307692307693400
7	5	78125	101.373626373626379000

n	p	p^n	Tiempo (s)
2	7	49	0.604395604395604469 00
3	7	343	0.439560439560439553 00
4	7	2401	1.868131868131868160 00
5	7	16807	15.219780219780220800 00
6	7	117649	130.219780219780219000 0
7	7	823543	1054.945054945505500 00

- **Algoritmo 15.ListadoMNS**$[q, n]$

  Aquí deseamos encontrar todas las matrices no singulares como se describe en la sección 3.2. En este algoritmo elaboramos un total de $p^n - 1$ multiplicado por la ec. 3.1, de triangulaciones gaussianas de una matriz. Claramente los tiempos crecen exponencialmente en el parámetro p^n. Tomamos el tiempo de 1 corrida.

n	p	p^n	Tiempo (s)
1	3	3	0.00000126373626373626
2	3	9	0.00000759340659340659
3	3	27	0.10989010989010988800

n	p	p^n	Tiempo (s)
1	5	5	0.00000274725274725274
2	5	25	0.05494505494505494410
3	5	125	5.61845631564517655714

n	p	p^n	Tiempo (s)
1	7	7	0.00000065934065934065
2	7	49	0.06871505494505494 41
3	7	343	9.04395604395604469 00

n	p	p^n	Tiempo (s)
1	11	11	0.00000879120879120879
2	11	121	6.05494505494505494410

Éste algoritmo es necesario para encontrar el matroide $B_{m,n,p}$, pues se debe hacer el listado de las matrices no singulares en F_p^m.

- **Algoritmo 16. MNSAleatoria**$[q, n]$

  En este algoritmo deseamos determinar, siguiendo la estrategia del algoritmo anterior, una matriz no singular $n \times n$ con entradas en $\mathbb{F}_p$. El número triangulaciones gaussianas realizadas esta dado por la ecuación 3.1. Entonces, los tiempos crecen exponencialmente respecto al número p^n. Tomamos el tipo promedio de 10^5 corridas.

n	p	p^n	Tiempo (s)
3	2	8	0.00329670329670329665
4	2	16	0.00439560439560439553
5	2	32	0.00549450549450549497
6	2	64	0.00604395604395604469
7	2	128	0.00549450549450549497

n	p	p^n	Tiempo (s)
2	3	9	0.002197802197802197700
3	3	27	0.002747252747252747480
4	3	81	0.003846153846153846370
5	3	243	0.004395604395604395530
6	3	729	0.006043956043956044690
7	3	2187	0.007692307692307692740

n	p	p^n	Tiempo (s)
2	5	25	0.001648351648351648320
3	5	125	0.003296703296703296650
4	5	625	0.003846153846153846370
5	5	3125	0.006593406593406593300
6	5	15625	0.009890109890109890500
7	5	78125	0.010659878494654

n	p	p^n	Tiempo (s)
2	7	49	0.060439560439560446900
3	7	343	0.054945054945054949700
4	7	2401	0.060439560439560446900
5	7	16807	0.060439560439560446900
6	7	117649	0.060439560439560446900
7	7	823543	0.060439560439560446900

- **Algoritmo 7. MatroideRepAleatorio**$[m, n, p]$

  En este procedimiento realizamos un **ListadoMNS**$[q, m]$ (por esta razón podemos variar a n con los valores 3 y 4). Cada una de las matrices obtenidas de tal listado, es multiplicada por una matriz M de tamaño $n \times m$. Los resultados muestran que el tiempo crece de forma exponencial. El algoritmo **MatroideRepAleatorio**$[q, m, n]$ cuando $m = n$ equivale a **ListadoMNS**$[q, n]$.

m	n	p	p^m	Tiempo (s)
1	3	3	3	0.10989865384615384404
2	3	3	9	0.22004990109890109567
3	3	3	27	0.66110975274725279159
1	4	3	3	0.10989865384615384404
2	4	3	9	0.27499495604395606667
3	4	3	27	0.66110975274725279159

m	n	p	p^m	Tiempo (s)
1	3	5	5	0.10992994505494505282
2	3	5	25	0.33041208791208793385
3	3	5	125	6.94548241454627543692
1	4	5	5	0.10992994505494505282
2	4	5	25	0.27546703296703296285
3	4	5	125	7.23119670026056124592

m	n	p	p^m	Tiempo (s)
1	3	7	7	0.110478434065934063
2	3	7	49	0.398412307692307686
3	3	7	343	12.19914629120879191
1	4	7	7	0.110478434065934063
2	4	7	49	0.453357362637362658
3	4	7	343	13.18815728021978141

m	n	p	p^m	Tiempo (s)
1	3	11	11	0.1099835164835164814
2	3	11	121	6.5531736263736263921
1	4	11	11	0.1109983516483516481
2	4	11	121	6.6630637362637363361

Aquí concluimos los procedimientos relativos a la enumeración de bases en el matroide. Los siguientes algoritmos están involucrados con la generación del ECS. Sus tiempos son polinomiales en p y en n. En consecuencia, ilustramos sus crecimientos de manera gráfica (la escala de valores, vertical en las gráficas, es lineal, y su extremo superior de órdenes de 10^{-4} a 10^{0} segundos).

- **Algoritmo 19. Partición**$[n, p]$. En este procedimiento sólo calculamos operaciones en $\mathbb{F}_p$ y hacemos eliminaciones de un arreglo lineal, por lo que el tiempo crece linealmente respecto a p^n. Hicimos pruebas para los diferentes $1 \leq n \leq 7$ y $p = 3, 5, 7, 11$, mostrado en la figura 5.1.

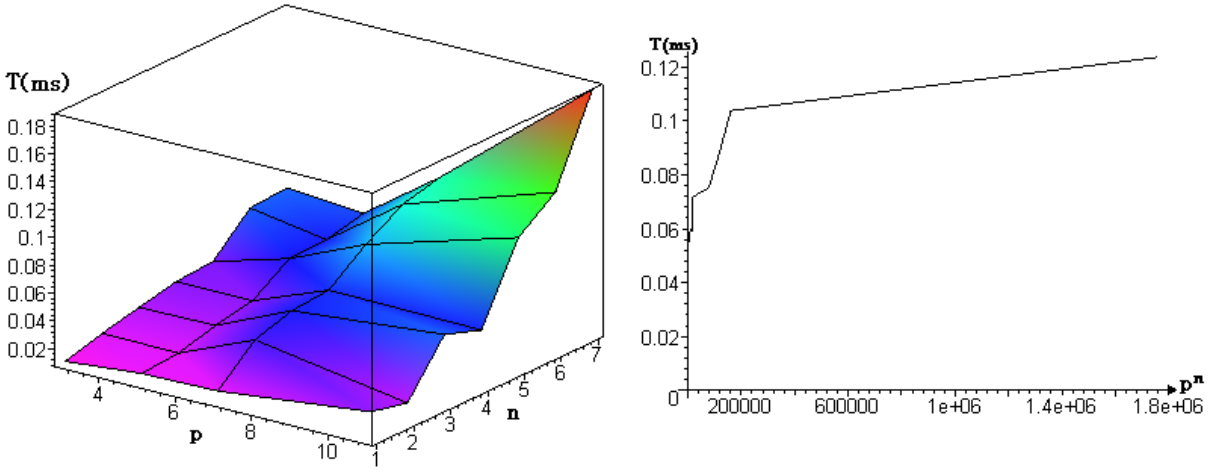


Figura 5.1: Resultados de la partición del espacio $\mathbb{F}_p^n$

- **Algoritmo 20. Fragmentación**$[s, \cup_i V_i, m, \nu_i]$. En este procedimiento sólo calculamos operaciones en $\mathbb{F}_p$, por lo que el tiempo crece de forma lineal respecto a p^n que es el número de participantes. Aquí hicimos pruebas para compartir el secreto $s = 987654321$ con el valor primo $p_1 = 987654323$ calculado tal que siempre satisface $p_1 > s$ y $p_1 > p^n$. Hicimos pruebas para los diferentes $1 \leq n \leq 7$ y $p = 3, 5, 7, 11$, mostrado en la figura 5.2.

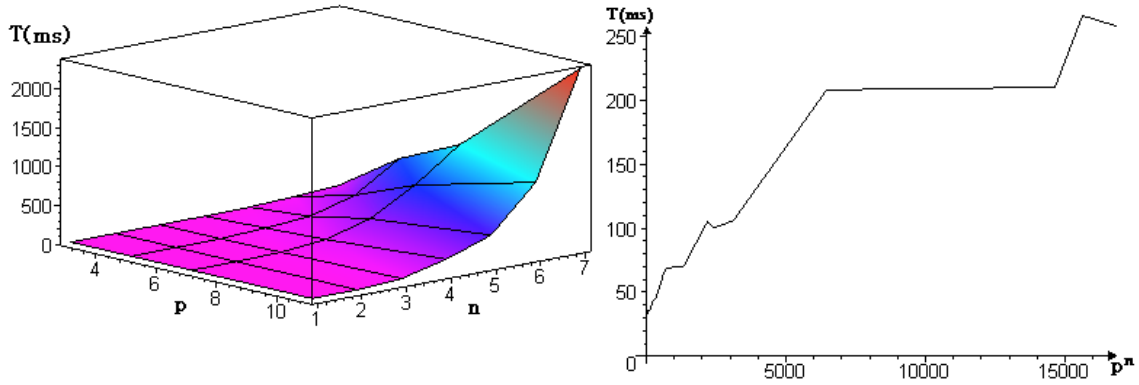


Figura 5.2: Resultados de la fragmentación del secreto

- **Algoritmo 21. Recuperación**$[A(\subset V), p_1, \Gamma_0]$

  Este procedimiento sólo efectuamos operaciones en $\mathbb{F}_p^n$, entonces el tiempo crece linealmente respecto a p^n. El primo considerado es el mismo del algoritmo anterior, $p_1 = 987654323$. Hicimos pruebas para los diferentes $1 \leq n \leq 7$ y $p = 3, 5, 7, 11$, mostrado en la figura 5.3.

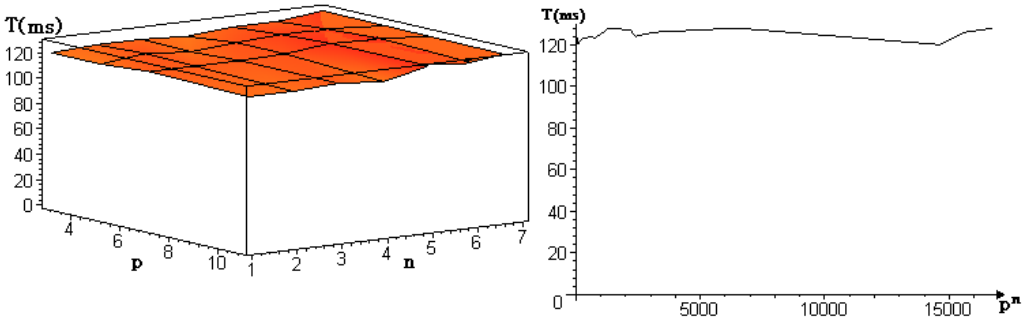


Figura 5.3: Resultados de la recuperación del secreto

Capítulo 6

Conclusiones y Trabajo a Futuro

En esta tesis ideamos un sistema para la selección aleatoria de matroides representables. El seleccionar un tal matroide tiene como consecuencia lógica que se puede identificar al conjunto de participantes con un subconjunto de un espacio vectorial y a los circuitos en el matroide, con conjuntos l. d. minimales para realizar a éstos como conjuntos de acceso de un ECS ideal. Una vez fijada tal identificación se procede a construir el ECS siguiendo el teorema de Brickell-Davenport. Tal secuencialidad es de tipo lógico. Desde el punto de vista computacional el orden en el que se genere el matroide o se construya el ECS sobre el espacio vectorial sobre un campo finito apropiado, es indiferente. Lo que liga a ambas construcciones es la identificación de participantes y vectores y de circuitos con conjuntos minimales l.d.

Mostramos pruebas de la implementación del sistema realizado, y observamos algunos problemas de eficiencia.

El énfasis de este trabajo está dado en la parte analítica. Las enumeraciones que realiza nuestra implementación permiten verificar que los métodos de conteo son correctos, y sirven sólo para el análisis. Por otro lado, la construcción de los ECS que realizamos es eficiente, y ésta bien puede incorporarse a un ulterior desarrollo.

La implementación de la selección de un matroide representable fue hecha en C, y la fragmentación y la recuperación del secreto fueron implementadas en `Java`.

Concluimos lo siguiente:

- El estado del arte. Actualmente se han desarrollado diversos esquemas. Esta variedad siempre es dependiente de las aplicaciones del esquema.

- La selección aleatoria de matroides. Debido al rápido crecimiento de los tiempos requeridos para hacer recuentos exhaustivos de conjuntos l.i. es muy conveniente que la dimensión del espacio vectorial tratado se mantenga de un tamaño lo más pequeño posible (del orden de unidades). Así mismo, el orden del campo es relevante para fines de desempeño. Un orden que es primo, permite realizar las operaciones en los enteros módulo ese primo. En cambio, un orden que es potencia de un primo implica una aritmética de tipo polinomial reducida un polinomio irreducible que permita realizar eficientemente las reducciones. En la implementación de los algoritmos en este trabajo sólo hemos considerado órdenes primos.

- La partición. El número de elementos en los clanes es uno menos que el orden del campo, cuando éste es primo. En tal caso el primo, obvia y necesariamente tiene que ser más grande que el umbral para recuperar secretos.

- La seguridad. Aquí confiamos que los conjuntos de acceso están conformados por participantes honestos. Sin embargo siempre es posible aumentar de tamaño los umbrales aunque el costo en complejidad crece rápidamente. El principal interés es aumentar la confiabilidad en el ECS y con esto se plantea la típica disyuntiva entre costo y seguridad.

- La validez de los programas. No es posible admitir que los programas implementados son totalmente efectivos, debido a esto, los programas fueron probados con parámetros más usuales y con entradas aleatorias. La variación de los tiempos obtenidos nos confirma que el tiempo de ejecución es mejor o peor para las diversas entradas de los programas.

- La contribución. Esta tesis se encargó de analizar a los matroides representables y la forma de obtenerlos aleatoriamente. Con matroides representables, también se pueden crear códigos lineales. Por otra parte proporcionamos un ECS ideal con aritmética en campos finitos mediante el uso de los conjuntos dependientes de un matroide representable.

- Las limitaciones. En el ECS ideal, el número de participantes, para un mejor tratamiento con nuestro sistema, debe ser cercano por debajo a una potencia de un número primo. La diferencia entre el número actual y la potencia cercana introduce participantes ficticios. Sin embargo, conjuntos de acceso sin éstos, legales desde el punto de vista de nuestra construcción, son legales desde el punto de vista del matroide que se haya generado con ellos.

- Las aplicaciones. Los esquemas de compartición de secretos son utilizados en diversos sistemas en los que se requiera fragmentar la responsabilidad de acceso. Para un sistema en particular existen diversas opciones para un esquema de Compartición de secretos. El sistema que aquí proponemos puede cambiar de acuerdo a las necesidades de un sistema en particular.

Sugerencias para trabajos futuros

Debido a la importancia de la Compartición de secretos, son muchas las Sugerencias que pueden hacerse a partir de esta tesis. El recomendable trabajo a futuro consiste de:

- Selección aleatoria de matroides. En la implementación realizada para la selección del matroide representable, sólo consideramos campos $\mathbb{F}_p$, con p un número primo. Es posible tomar campos de la forma $\mathbb{F}_q$ con $q = p^k$ y $k > 1$ mediante un mero cambio de la aritmética de campos, considerando en este último caso a la de polinomios módulo un irreducible. Se tendría que considerar entonces operaciones sobre polinomios con coeficientes en $\mathbb{F}_p$ módulo un polinomio irreducible. Queda todavía por hacer las operaciones con polinomios y la localización de polinomios irreducibles de manera automática.

- Optimizar los algoritmos. Encontrar alternativas para optimizar a los algoritmos aquí propuestos para la selección aleatoria de matroides representables. Sobre todo para evitar almacenar casos previos en los procedimientos de *backtracking* (véase nuestros comentarios al final de capítulo 3), lo cual de acuerdo con nuestro análisis lo consideramos altamente plausible.

- Desarrollar otros ECS ideales. Como la construcción de espacio vectorial de Brickell no propone una correspondencia fija de participantes a sus vectores asociados, es posible intentar otra construcción del ECS ideal, de tal forma que éste esquema posea como estructura de acceso a los conjuntos dependientes de un matroide representable

- Aumentar el umbral. Como el mínimo número de personas para recuperar el secreto es 2 y este número podría considerarse poco seguro, es posible aumentar el umbral a un valor $k > 2$ utilizando hipergráficas y k-clanes lo cual implica aumentos de órdenes y características de campos (véase nuestros comentarios al final de capítulo 2 y de la sección 4.1, en particular). Aquí ha de procederse tal como se bosquejó al final de la subsección 4.1.1. Por la explosión combinatoria estimada ahí para el número de hiperaristas, la implementación que se haga ha de ser muy cuidadosa en el manejo de pilas y de los controles de los procedimientos de *backtracking* involucrados.

Apéndice 1. Algoritmos

A continuación se enlistan los 12 principales algoritmos para elaborar la selección aleatoria, como se describieron en la sección 3.2.

Algoritmo 8 ComoNumero$[\mathbf{x}, n, q]$

Require: $n \in \mathbb{N},\ q = p^k,\ p$ primo y $x \in \mathbb{F}_q^n$.
Ensure: $z = N(x) = \sum_{j=0}^{n-1} x_j\ q^{n-(j+1)}$.

Algoritmo 9 ComoVector$[a]$

Require: $a \in [\![0, q^n - 1]\!]$.
Ensure: $\mathbf{x} \in \mathbb{F}_q^n$ tal que $a = N(\mathbf{x})$.

Algoritmo 10 VectorSiguiente$[\mathbf{x}]$

Require: Un vector $\mathbf{x} \in \mathbb{F}_q^n$.
Ensure: El vector siguiente $\mathbf{y} \in \mathbb{F}_q^n$ o un indicativo de que $\mathbf{x}$ era el ultimo.

1: $a \leftarrow$ ComoNumero$[\mathbf{x}]$;
2: **if** $a < q^n - 1$ **then**
3: $\quad \mathbf{y} \leftarrow$ ComoVector$[a + 1]$;
4: **else**
5: $\quad \mathbf{y} \leftarrow \emptyset$;
6: **end if**

Algoritmo 11 InsercionDeColumna$[M, \mathbf{x}]$

Require: Una matriz $M \in \mathbb{F}_q^{n \times j}$ y
un vector $\mathbf{x} \in \mathbb{F}_q^n$.
Ensure: La matriz en $\mathbb{F}_q^{n \times (j+1)}$, que coincide con
M al añadir $\mathbf{x}$ como su ultima columna.

Algoritmo 12 TriangulacionGaussiana$[M]$

Require: Una matriz $M \in \mathbb{F}_q^{m \times n}$.
Ensure: La matriz triangular superior obtenida por reducción gaussiana a partir de M.

Algoritmo 13 RevisionLI$[M, \mathbf{x}]$

Require: Una matriz $M \in \mathbb{F}_q^{n \times j}$ y un vector $\mathbf{x} \in \mathbb{F}_q^n$.
Ensure: Un valor 1 si $\mathbf{x}$ junto a las columnas de M forma un conjunto l. i.; 0 en otro caso.

1: $M_1 \leftarrow$ InsercionDeColumna$[M, \mathbf{x}]$;
2: $U \leftarrow$ TriangulacionGaussiana$[M_1]$;
3: Si todos los elementos de la diagonal de U son no-nulos, dése como resultado 1, en otro caso 0;

Algoritmo 14 ListaDeLIs$[M]$

Require: Una matriz $M \in \mathbb{F}_q^{n \times j}$.
Ensure: La lista de vectores $L \subset \mathbb{F}_q^n$ tal que $\forall \mathbf{x} \in L,\ M$ junto con $\mathbf{x}$ es l.i.

1: Inicialmente $L \leftarrow \emptyset$;
2: $\mathbf{x} \leftarrow$ ComoVector[1];
3: **while** $\mathbf{x} \neq \emptyset$ **do**
4: $\quad s \leftarrow$ revisionLI$[M, x]$;
5: $\quad$ **if** $s = 1$ **then**
6: $\quad\quad L \leftarrow$ UnirVector$[\mathbf{x}, L]$;
7: $\quad$ **end if**
8: $\quad \mathbf{x} \leftarrow$ VectorSiguiente$[\mathbf{x}]$;
9: **end while**

Algoritmo 15 ListadoMNS$[q, n]$

Require: $n \in \mathbb{N},\ q = p^k,\ p$ primo.
Ensure: La lista L de las $s(q, n)$ matrices no singulares en $M \in \mathbb{F}_q^{n \times m}$.
(se utiliza una pila P que consiste a su vez de n pilas $p_0, \ldots, p_{n-1}$)

1: $\mathbf{x} \leftarrow$ ComoVector[1], $M \leftarrow [\mathbf{x}]$;
2: $p_0 \leftarrow$ ListaDeLIs$[M]$, $P \leftarrow [p_0]$;
3: $L \leftarrow \emptyset,\ \ j \leftarrow 1$;
4: **while** $P \neq \emptyset$ **do**
5: **while** $j < n - 1$ **do**
6: $p_{j-1} \leftarrow$ Pop$[P]$;
7: $\mathbf{x}_j \leftarrow$ Pop$[p_{j-1}]$;
8: Push$[P, p_{j-1}]$;
9: InsercionDeColumna$[M, \mathbf{x}_j]$;
10: $p_j \leftarrow$ ListaDeLIs$[M]$;
11: Push$[P, p_j]$; $j++$;
12: **end while**
13: **if** $j = n - 1$ **then**
14: p_{n-1}Pop$[P]$;
15: $\forall \mathbf{x} \in p_{n-1}$, Unir a L la matriz obtenida de: InsercionDeColumna$[M, \mathbf{x}]$;
16: $p_{n-1} \leftarrow \emptyset$;
17: **end if**
18: **while** $p_{j-1} = \emptyset$ **do**
19: Pop$[P]$;
20: $j--$;
21: **end while**
22: **end while**
23: $L \leftarrow$ QuitaRepeticion$[L]$;
24: Regresar L.

Algoritmo 16 MNSAleatoria$[q, n]$

Require: $n \in \mathbb{N},\ q = p^k,\ p$ primo.
Ensure: Una matriz no-singular $M \in \mathbb{F}_q^{n \times n}$.

1: $M \leftarrow \emptyset;\ j \leftarrow 0;\ P \leftarrow \mathbb{F}_q^n \setminus \{\mathbf{0}\}$;
2: $lp \leftarrow$ longitud$[P]$;
3: **while** $j < n$ **do**
4: $i \leftarrow$ AleatorioEntre$[1, lp]$;
5: $\mathbf{x} \leftarrow$ (i-esima entrada de P);
6: $M \leftarrow$ InsercionDeColumna$[M, \mathbf{x}]$;
7: $P \leftarrow$ ListaDeLIs$[M]$;
8: $lp \leftarrow$ longitud$[P]$;
9: $j++$;
10: **end while**
11: Regresa L.

Algoritmo 17 BaseComoNumeros$[M]$

Require: Una matriz no-singular $M \in \mathbb{F}_q^{n\times n}$.
Ensure: Un arreglo ordenado de enteros $A_M \subset [\![0, q^n - 1]\!]$ que representa la base dada por las columnas de M.

$A_M \leftarrow \emptyset$; $j \leftarrow 1$;
for $j \leq n$ **do**
  $\mathbf{x}_j \leftarrow$ (la $j-$ésima columna de M);
  $a_j \leftarrow$ ComoNumero$[\mathbf{x_j}]$;
  Añádase a_j a A_M;
  $j++$;
end for
Ordenar$[A_M]$ (de menor a mayor);
Dése A_M como resultado.

Algoritmo 18 ListaBasesComoNumeros$[q, n]$

Require: $n \in \mathbb{N}$, $q = p^k$, p primo.
Ensure: La lista de bases de $\mathbb{F}_q^n$ vistas como subconjuntos de $[\![0, q^n - 1]\!]$.

$LNum \leftarrow []$;
for $\forall M \in L$ **do**
  $c(M) \leftarrow$ BaseComoNumeros$[M]$;
  Añádase $c(M)$ a $LNum$;
end for
Ordenar$LNum$ (de manera lexicográfica);
$N \leftarrow$ longitud$[LNum]$;
Sea $B \leftarrow []$;
for $i = 1$ to N **do**
  añádase $LNum[i]$ a B;
  $i \leftarrow i + n!$;
end for
Regresar B.

Algoritmo 19 Partición$[n, p]$

Require: Los parámetros n, p del espacio vectorial $\mathbb{F}_q^n$.
Ensure: La partición $\cup_i V_i$ del espacio $\mathbb{F}_q^n$.

1: Sea $[\![0, p^n - 1]\!]$ la colección de números asociados a los vectores en $\mathbb{F}_p^n$.
2: $L \leftarrow [\![1, p^n - 1]\!]$;
3: $k \leftarrow 0$;
4: **while** $L \neq \emptyset$ **do**
5: El clan $V_k \leftarrow \emptyset$;
6: Tomar un elemento $x \in L$;
7: $V_k \leftarrow V_k \cup \{x\}$;
8: $L \leftarrow L - \{x\}$;
9: Buscar a todos los elementos $y \in L$ que son l.d. con x, añadirlos a V_k y eliminarlos de L;
10: $k++$;
11: **end while**
12: Regresar la colección $\cup_i V_i$.

Algoritmo 20 Fragmentación$[s, \cup_i V_i, m, \nu_i]$

Require: s el secreto, $\cup_i V_i$ es la partición del conjunto del espacio $\mathbb{F}_p^n$, m el número de clanes y ν_i el número de elementos en cada clan V_i.
Ensure: Un primo p_1 y los fragmentos s_i para cada participante de $\mathfrak{p}_i \in \cup_i V_i$.

1: Elegir un primo p_1 que satisface $p_1 > \sum_i \nu_i$ y $p_1 > s$;
2: Se seleccionan m números distintos $y_i \in \mathbb{F}_{p_1} - \{0\}$, uno para cada clan V_i;
3: **for** $i = 0$ a $m - 1$ **do**
4: **for** $\forall\, \mathfrak{p}_{i,j} \in V_i$ (cada participante en el clan V_i) **do**
5: Asignamos al (participante $\mathfrak{p}_{i,j}$) $\leftarrow (s + y_i \cdot x_{i,j}) \, mod \, p_1$ ($x_{i,j} \in \mathbb{F}_{p_1}$ es el número asociado al participante $\mathfrak{p}_{i,j}$);
6: **end for**
7: **end for**

Algoritmo 21 Recuperación$[A(\subset P), p_1, \Gamma_0, q, n]$

Require: Un subconjunto A de participantes que reúnen sus fragmentos y el primo p_1.
Ensure: El secreto s.

1: Buscamos algún conjunto $\{\mathfrak{p}_{i,j_1}, \mathfrak{p}_{i,j_2}\}$ contenido en A que esté en Γ_0 (es decir que sus vectores asociados estén relacionados, es decir, que sean l.d.;
2: **if** Existe tal pareja $\mathfrak{p}_{i,j_1}$, $\mathfrak{p}_{i,j_2}$, con fragmentos respectivos s_{i,j_1}, s_{i,j_2} **then**
3: Calculamos $c_1 = \frac{x_{i,j2}}{x_{i,j_2} - x_{i,j_1}}$ y $c_2 = \frac{x_{i,j_1}}{x_{i,j_1} - x_{i,j_2}}$ ($x_{i,j} \in \mathbb{F}_{p_1}$ es el número asociado al participante $\mathfrak{p}_{i,j}$);
4: Regresa: $s = c_1 \cdot s_{j_1} + c_2 \cdot s_{j_2}$;
5: **else**
6: Regresa: "No es posible recuperar el secreto";
7: **end if**

Apéndice 2. Composición del CD

El disco compacto incluido en esta tesis, contiene los trabajos desarrollados en el proceso de la elaboración de la misma. En seguida detallamos el contenido:

- En la carpeta con nombre *presentaciones* colocamos:

  1. la presentación elaborada el seminario de tesis que muestra a grandes rasgos la composición inicial de esta tesis.
  2. la presentación de la ponencia ofrecida en el *Sexto Coloquio Nacional de Códigos, Criptografía y Áreas Relacionadas,* realizado del 16 al 18 de Junio de 2004 en la Ciudad de México. En este evento mostramos los resultados obtenidos de la selección aleatoria de matroides representables y la dificultad de elegir números apropiados q y n para el espacio vectorial $\mathbb{F}_q^n$.

- En la carpeta con nombre *articulo* colocamos el artículo aceptado por el CIE-2004 con el título *Random Generation of Representable Matroid* mismo que fue aceptado para su publicación.

- El la carpeta con nombre *tesis* colocamos el documento PDF de la esta tesis.

- El la carpeta con nombre *programas* colocamos

  1. la carpeta *source*, que contiene todos los códigos fuente (en C y `java`).
  2. la carpeta *include*, que contiene el archivo módulo**.**h necesario para las operaciones modulares.
  3. la carpeta *exec*, donde se encuentran los ejecutables para Windows.
  4. la carpeta *exam*, donde se encuentran los algunos ejemplos de prueba (programas ejecutables).
  5. la carpeta *info*, que contiene un archivo *leame.txt* que es el manual de usuario.

Bibliografía

[1] Béguin, P., Cresti, A. Gneral Short Computational Secret Sharing Schemes *Advances in Cryptology - EUROCRYPT 95*, volume 921 of LNCS, pp. 194-208, 1995.

[2] Beimel, A. , Chor, B. Universally Ideal Secret Sharing Schemes. *Lecture Notes in Computer Science*, 740, pp. 183-195, 1993.

[3] Bertoni, G., Breveglieri, L., Fragneto, P. Efficient Finite Field Digit-Serial Multiplier Architecture for Cryptography Applications. *Design, Automation and Test in Europe Conference and Exhibition 2001*, IEEE Computer Society, 2001.

[4] Blakley, R., Safeguarding cryptographic keys, *Proceedings of AFIPS 1979 National Computer Conference*, vol.48, N. Y., 1979, pp. 313-317.

[5] Blundo, C., De Santis, A. Graph Descompositions and Secret Sharing Schemes. *Research of D.R. Stinson*, 2003.

[6] Brickell, E. F., Some Ideal Secret Sharing Schemes *J. Combin. Math. Combin. Comput*, 9: 105-113, 1998.

[7] Brickell, E. F., Davenport, D. M., On the Classification of Ideal Secret Sharing. *J. of Cryptology*, 4: 123-134, 1991.

[8] Diestel, R., *Graph Theory* (Electronic Edition 2000). Springer-Verlag, New York, 2000.

[9] Gennaro, R., Micali, S., Verifiable Secret Sharing and Secury Computation. *J. Combin. Math. Combin. Comput*, 10: 168-182, 1998.

[10] Herstein, I. N. *Algebra moderna*. Editorial Trillas, 1970.

[11] Hoffman, K., Kunze, R., *Algebra Lineal*. Prentice-Hall Hispanoamericana, 1973.

[12] Hong-Jian, Lai. *Introduction to Combinatorial Optimization in Matroids*. Buscar su editorial, 2004.

[13] Frías, M., Nieto, J., Carrillo, Ma. Clanes y Matroides. *Mosaicos Matemáticos*, No. 11, pp. 25-32, 2003.

[14] Malkhi, D., *An advance course in computer and network security*. Lecture Notes, 2002, Disponible en: `http://www.cs.huji.ac.il/~ns/SS.doc`.

[15] Mollin, R., *an Introducion to Cryptography.*Chapman & Hall/CRC, 2001.

[16] N.Tharani Rajan, A write-up on Mathematical Basics for Sharing a Secret. *Cryptography and Network Security,* Department of Computer Science and Engineering, Indian Institute of Technology, 2001.

[17] Oxley, J. G., *Matroid Theory.* Oxford University Press, 1992.

[18] Padró, C., Saenz, G., Tasa de información de los esquemas para compartir secretos con estructura homogénea de rango 3. *Criptología y Seguridad de la información*, 171-179, 2000, Ed. Ra-Mo. España.

[19] Ruiz-Hernández, G., Morales-Luna, G., Computational experiments on the analysis of ideal sharing secrets schemes, *DISC: Seguridad 2000.* UNAM.

[20] Shamir, A., How to share a secret, *Communications of the ACM*, vol. 22, no. 1, 1979, pp. 612-613.

[21] Trappe, W., *Introduction to Cryptography*, Prentice Hall, 2002.

[22] Vinod, V., Narayanan, Arvind, Srinathan K., Pandu Rangan C. and Kim Kwangjo. On the Power of Computational Secret Sharing. *Indian Institute of Technology, Madras.*

[23] Welsh D. J. A., *Matroid Theory.* Academic Press, 1976.

[24] Schneier, B. *Applied Cryptography.* Protocol, Algorithms and Source Code in C. Wiley, Second Ed., 1996.

Los abajo firmantes, integrantes del jurado para el examen de grado que sustentará la **Srita. Leonor Vázquez González**, declaramos que hemos revisado la tesis titulada:

Métodos Computacionales para Diversos Esquemas de Compartición de Secretos

Y consideramos que cumple con los requisitos para obtener el Grado de Maestría en Ciencias en la especialidad de Ingeniería Eléctrica opción Computación.

Atentamente,

Dr. Guillermo Morales Luna _____

Dr. Arturo Díaz Pérez _____

Dr. Francisco Rodríguez Henríquez _____