CAPÍTULO 1

Marco teórico

1.1 Historia del desarrollo de la criptografía

La criptografía surge desde los inicios de la humanidad y fue creada para mantener la privacidad de la información que se transmite, garantizando que alguien externo o ajeno no pueda leer el contenido del mensaje. Presentamos aquí un poco de la historia del desarrollo de la criptografía:

1.1.1 Criptografía antigua

Aunque los historiadores afirman que la criptografía está presente en todas las civilizaciones de la antigüedad, es conveniente resaltar que todos los ejemplos documentados que presentan son puntuales, ya que ninguna de estas civilizaciones utilizó de forma común la criptografía. Ningún imperio de aquella época se sirvió de dicha ciencia para enviar sus correspondencias confidenciales, sólo en contadas ocasiones hacían uso de ella. El uso regular de la criptografía comienza en la Edad Media, con los árabes, y en Europa durante el Renacimiento. En la antigüedad las pocas muestras de criptografía son muy simples, sin embargo es de vital importancia comenzar por dichas muestras, no sólo por curiosidad histórica, sino porque en ellas se encuentra la base de la criptografía que vendrá después.

El ocultamiento de la información en las primitivas civilizaciones.

Cuestiones militares, religiosas y comerciales impulsaron desde tiempos remotos el uso de escrituras secretas. Las primitivas civilizaciones que tuvieron relevancia en la historia de la criptografía fueron las civilizaciones: egipcia, mesopotámica, china e india.

CIVILIZACIÓN EGIPCIA. Escritura jeroglífica: aparece después de los 3000 A.C. y consiste en una escritura fonética que incluye ocasionalmente semagramas. Desaparece en el siglo IV D.C. prohibida por el Cristianismo y se sustituye por la escritura copta.

CIVILIZACION MESOPOTÁMICA: La escritura cuneiforme aparece aproximadamente en el año 3300 A.C. Los escribas de la antigua Mesopotamia, al igual que sus colegas egipcios, también cambiaban en ocasiones los signos cuneiformes de su escritura por otros con el fin de alterar la misma. Sin embargo y a diferencia de los egipcios, los escribas mesopotámicos si tuvieron intención de ocultar el significado de la escritura.

CIVILIZACION CHINA: Usaban la esteganografía de escritura en seda o papiro envuelto en cera. Otra forma de llevar a cabo la ocultación de un mensaje mediante el método de esteganografía es ocultar el contenido del mismo en un canal de información, pero en paridad.

CIVILIZACIÓN INDIA: Hubo un desarrollo temprano del cifrado. Artha-Sastra (Libro del Estado) que contiene recomendaciones para los espías. Lolita-Vistara (Vida de Buda). Arthasastra, 300 A.C. es una criptografía para embajadores. En el libro del Kamasutra. El número 45 de la lista es: mlecchitavikalpa, el arte de la escritura secreta, para ayudar a las mujeres a ocultar los detalles de sus relaciones amorosas. Todo esto será la base de una de las descripciones más antiguas de codificación llevadas a cabo más tarde por el erudito Brahmin Vatsyayana.

La escitala Espartana (siglo V a.c)

El método que consistía en tomar una vara (escítala) a la cual se le enroscaba una cinta de cuero y luego se escribía de forma longitudinal. Por último, se desenrollaba la cinta con letras sin sentido y enviaba con el mensajero.

El cifrador del César (siglo I a.c)

Al igual que los espartanos, los romanos crearon su propio sistema de encriptación llamado el cifrador del César (Siglo I A.C), el cual consistía en el sustituir cada letra por tres posiciones más allá en el alfabeto. Implícitamente utilizaban el método que hoy llamamos congruencia, donde le asignaban un valor a cada letra del abecedario ($A = 00$, $B = 01$,$Z = 26$), esta transformación criptográfica en términos de congruencia queda expresada como:

$$C \equiv M + 3 \text{ mód } 26$$

Donde, M corresponde a la letra del mensaje original y C, corresponde a la letra

del mensaje cifrado.

1.1.2 Criptografía en Europa

La criptografía en Europa data de la edad media, el primer libro europeo que describe el uso de la criptografía fue escrito en el siglo XIII por el monje franciscano Roger Bacon, titulado La Epístola sobre las obras de arte secretas y la nulidad de la magia. En el año 1379 Gabriele de Lavinde de Parma escribió el primer manual sobre criptografía. En 1563 el físico Giambattista Della Porta crea un sistema con la particularidad de cifrar bloques de dos en dos letras. Muchos consideran Della Porta como el "Padre de la criptografía moderna".
El desarrollo de los cifrados polialfabéticos empezó con Leon Battista Alberti, que en 1568 publicó un manuscrito describiendo un disco de cifrado que define múltiples sustituciones. En 1586 Blasie de Vigenére generalizó el criptosistema de Julio Cesar utilizando todos los corrimientos posibles. En los años 1600, se comenzaron a dar soluciones a los criptosistemas existentes. En 1624 Augustus II. Duque Alemán, escribió el libro Cryptomenytices et cryptographiae, libri IX, en el que se dedicaba a la solución de varios criptosistemas. En 1663 en Roma, el jesuita Athanasius Kircher escribió el libro Polygraphia nova et universalis, el que consiste en una colección de criptosistemas usados en la época.
En el siglo XIX, Kerckhoffs estableció los principios de la criptografía moderna. Los algoritmos modernos usan una clave para controlar el cifrado y descifrado de los mensajes. Se acepta, la denominada hipótesis de Kerckhoffs, que establece que la seguridad del cifrado debe residir, exclusivamente, en el secreto de la clave y no en el del mecanismo de cifrado.
Un año importante para la criptografía fue el de 1976, cuando W. Diffie y M. Hellman crean el concepto de Criptosistema de clave pública, es decir, un sistema donde la clave de cifrado se puede encontrar en un directorio público de usuarios; sin embargo, la clave de descifrado es diferente y no se obtiene fácilmente de la primera. Poco más tarde, en 1978 se da a conocer el criptosistema de clave pública más seguro y usado hasta la fecha, el RSA (sistema que se estudiara a detalle posteriormente). Sus inventores R. L. Rivest, A. Shamir y L. Adleman del MIT proponen la función de un sólo sentido que utiliza el exponente módulo un número entero n, producto de dos números primos y que tiene como seguridad la dificultad de factorizar a un número n de entre 100 y 200 dígitos. La necesidad de romper este criptosistema desarrolla la teoría de factorizar números grandes. Otro sistema que se ha mantenido hasta hoy, es el propuesto por T. ElGamal en 1984, que basa su seguridad en el problema del logaritmo discreto.

1.2 Conceptos matemáticos

Para entrar de lleno en el funcionamiento de los criptosistemas es necesario recordar algunos conceptos matemáticos que nos ayudan a cifrar y descifrar mensajes.

1.2.1 Divisibilidad

Definición 1. *Sea $a, b \in \mathbb{Z}$. Se dice que a divide a b o que b es divisible por a sí y solo si existe $k \in \mathbb{Z}$ tal que:*

$$ka = b$$

- *Se anota $a \mid b$ para indicar la divisibilidad.*
- *Cuando no hay divisibilidad se anota $a \nmid b$*

Proposición 1. *a, b y $c \in \mathbb{Z}$, tenemos:*

1. $a \mid b \wedge b \mid c \Rightarrow a \mid c$.
2. $a \mid b \Rightarrow ac \mid bc$, *con* $c \neq 0$.
3. $a \mid b \wedge b \mid a \Rightarrow a = \pm b$.
4. $a \mid \pm 1 \Leftrightarrow a = \pm 1$.
5. $c \mid a \wedge c \mid b \Rightarrow c \mid (ax + by)$, *para todo* $x, y \in \mathbb{Z}$.

1.2.2 Máximo común divisor (MCD)

Definición 2. *: Sea $a, b \in \mathbb{Z}$, ambos no nulos. El máximo común divisor de a y b es un entero positivo d tal que:*

1. $d/a \wedge d/b$
2. *Si* $k/a \wedge k/b \Rightarrow k/d$

Es decir, d es el mayor de los divisores comunes entre a y b. Se anota como mcd (a,b)

1.2.3 Aritmética Modular

Algunos conceptos que se deben entender para poder comprender la Aritmética Modular son:

- Relación de equivalencia:
  En teoría de conjuntos y álgebra, la noción de relación de equivalencia (que será definida más abajo en la investigación) sobre un conjunto permite establecer una relación entre los elementos del conjunto que comparten cierta característica o propiedad. Esto permite reagrupar dichos elementos en clases de equivalencia, es decir, ≪paquetes≫ de elementos similares.

- Raíz primitiva módulo n: Dado un número natural n, decimos que a es una raíz primitiva módulo n, si a genera como grupo a $\mathbb{Z}_n^*$, es decir, si $\forall b \in \mathbb{Z}_n^*$ existe $k \in \mathbb{Z}$ tal que $a^k \equiv b \mod n$. Aqui $\mathbb{Z}_n^*$ denota los elementos invertibles módn.
  Dado que el orden de $\mathbb{Z}_n^*$ es función phi de euler (n), una raíz primitiva es un elemento con ese orden.

  Nota: Se dice que una clase [a] en $\mathbb{Z}_n$ es invertible si a tiene una inversa (multiplicativa) (mod N); o sea, existe un entero b tal que ab es congruente con 1 (mod N).

Definición 3. *Sea m un entero positivo, donde $a, b \in \mathbb{Z}$. Se define*

$$a \equiv b \text{ mód } m \Leftrightarrow m \mid (a - b)$$

Proposición 2. .

(a) $a \equiv b$ mód $m \Leftrightarrow$ *a y b dejan el mismo resto cuando son divisibles por m.*

(b) La relación de congruencia mod n, es una relación de equivalencia en $\mathbb{Z}$ donde Relación de equivalencia sobre un conjunto permite establecer una relación entre los elementos del conjunto que comparten cierta característica o propiedad.

Observación 1. *Sea $a \in \mathbb{Z}$, su clase de equivalencia en* mód *m es el conjunto definido por: $\overline{a} = \overline{x}$ si $x \equiv a$* mód *m*

¿Cuántas clases de equivalencia existen?
Hay m clases : $[\overline{0}, \overline{1}, ...\overline{m-1}]$ y $0, 1, ...m-1$ se denomina conjunto completo de representantes.

Proposición 3. *Sea $m \geq 1$ un entero fijo:*

1. *Si $a \equiv b$* mód *m y $c \equiv d$* mód *m. Entonces $a + c \equiv b + d$* mód *m*
2. *Si $a \equiv b$* mód *m y $c \equiv d$* mód *m. Entonces $ac \equiv bd$* mód *m.*

Primos relativos

Definición 4. *Sea $a \wedge b$ enteros no nulos. Se dice que $a \wedge b$ son primos relativos, si y solo si $mcd(a, b) = 1$*

Ejemplo 1. $mcd(73, 25) = 1$

Corolario 1. *$a \wedge b$ son primos relativos, si y solo si existen $x_0, y_0 \in \mathbb{Z}$ tal que $ax_0 + by_0 = 1$*

Se pueden obtener primos relativos usando:

Proposición 4. *Si $d = (a, b) \Rightarrow (a/d, b/d) = 1$*

1.3 El Problema del Logaritmo Discreto

Definición 5. *Logaritmo discreto*
Si g es una raíz primitiva módulo p, y h un elemento distinto de cero de $\mathbb{F}_p$, el problema del logaritmo discreto es el problema de encontrar un x tal que $g^x \equiv h \bmod p$ El número x tal que $h = g^x \bmod p$ se llama Logaritmo Discreto de h módulo p con base g y lo denotamos por $log_g(h)$.

Proposición 5. *Sea p un primo y g una raíz primitiva módulo p. Así:*

1. *$log_g(ab) \equiv log_g(a) + log_g(b) \bmod p - 1$ para todo $a, b \in \mathbb{F}_p^*$.*

2. *$log_g(a^k) \equiv k\ lg_g(a) \bmod p - 1$ para todo $a \in \mathbb{F}_p^*$ y $k \in \mathbb{Z}$.*

Ejemplo 2.

Calcular el siguiente logaritmo discreto.
$Log_3\ 7$ para $p = 17$, donde $g = 3$ es una raíz primitiva.
$3^k \equiv 7$ mód 17
$k = 11$
Así $3^{11} \equiv 7$ mód17

Definición 6. *Sea G un grupo abeliano finito y sea $g \in G$ de orden n. Dado $a \in < g > G \subseteq G$, se define el logaritmo discreto de a en base g como el entero $k, 0 \leq k \leq n - 1$, tal que:*
$g^k = a$
Se dice también que k es el índice de a en base g. El Problema del Logaritmo Discreto consiste en , dados g y a, calcular k.

Algoritmo para calcular potencias: Sean a, k, m enteros positivos, entonces a^k mód m se puede calcular con el algoritmo siguiente

1. Calcule la expansión binaria de k
   $k = k_0 + k_1 2 + k_2 2^2 \ldots + k_t 2^t$ con $k_0, k_1, \ldots, k_t \in \{0, 1\}$ y $k_t \neq 0$

2. Calcular las potencias

$$\begin{array}{lll} a_0 & \equiv a & \text{mód} m \\ a_1 & \equiv a_0^2 \equiv a^2 & \text{mód} m \\ a_2 & \equiv a_1^2 \equiv a^{2^2} & \text{mód} m \\ \vdots & \vdots & \vdots \\ a_t & \equiv a_{t-1}^2 \equiv a^{2^t} & \text{mód} m \end{array}$$

3. Cacular a^k mód n usando la formula

$$\begin{array}{rcll} a^k & = & a^{k_0 + k_1 2 + k_2 2^2 \ldots + k_t 2^t} & \\ & = & a^{k_0} \cdot (a^2)^{k_1} \cdot (a^{2^2})^{k_2} \cdots (a^{2^t})^{k_t} & \\ & \equiv & a_0^{k_0} \cdot a_1^{k_1} \cdot a_2^{k_2} \cdots a_t^{k_t} & \text{mód } m \end{array}$$

Ejemplo 3.

Calcule

$$3^{151} \text{ mód } 1000.$$

$$\begin{array}{rcl} 151 & = & 75 \cdot 2 + 1 \\ 75 & = & 37 \cdot 2 + 1, \\ 37 & = & 18 \cdot 2 + 1, \\ 18 & = & 9 \cdot 2 + 0 \\ 9 & = & 4 \cdot 2 + 1 \\ 4 & = & 2 \cdot 2 + 0 \\ 2 & = & 1 \cdot 2 + 0 \\ 1 & = & 0 \cdot 2 + 1 \end{array}$$

$151_{10} = 10010111_2$ tenemos que $151 = 1 + 2 + 2^2 + 2^4 + 2^7$

$$\begin{array}{rcl} 3^{151} & = & 3 \cdot 3^2 \cdot 3^{2^2} \cdot 3^{2^4} \cdot 3^{2^7} \\ & \equiv & 3 \cdot 9 \cdot 81 \cdot 721 \cdot 961 \text{ mód } 1000 \\ & \equiv & 747 \text{ mód } 1000 \end{array}$$

i	3^{2^i} mód 1000
0	3
1	9
2	81
3	561
4	721
5	841
6	281
7	961

CAPÍTULO 2

Tipos de cifrados

Existen varios métodos de cifrados diferentes y cada uno de ellos ha sido desarrollado teniendo en cuenta diferentes necesidades de seguridad y protección. Los dos tipos principales de cifrado de datos son el cifrado asimétrico y el simétrico.

2.1 Métodos de cifrado asimétrico

El cifrado asimétrico cifra y descifra los datos mediante dos claves criptográficas asimétricas independientes. Estas dos claves se conocen como clave pública y clave privada.

Métodos más comunes de cifrado asimétrico:

- RSA: recibe su nombre de las iniciales de los informáticos Ron Rivest, Adi Shamir y Leonard Adleman. Es un algoritmo popular que se utiliza para cifrar datos con una clave pública y descifrarlos con una clave privada para transmitir los datos de forma segura.

- ElGamal: Es un algoritmo de criptografía asimétrica basado en la idea de Diffie-Hellman y que funciona de una forma parecida a este algoritmo discreto.

2.2 Métodos de cifrado simétrico

El cifrado simétrico es un tipo de cifrado en el que solo se utiliza una clave simétrica secreta para cifrar el texto sin formato y descifrar el texto cifrado.

Métodos comunes de cifrado simétrico:

- Estándares de cifrado de datos (DES): es un algoritmo de cifrado por bloques de bajo nivel que convierte texto sin formato en bloques de 64 bits y los convierte en texto cifrado utilizando claves de 48 bits.

- Triple DES: ejecuta el cifrado DES tres veces diferentes en las que se cifran, se descifran y luego se cifran los datos nuevamente.

- Estándar de cifrado avanzado (AES): a menudo se conoce como el criterio de referencia para el cifrado de datos y se utiliza en todo el mundo como el estándar del gobierno de Estados Unidos.

¿Por qué es importante el tipo de cifrado?

Los métodos de cifrado varían según la cantidad de datos que pueden manejar a la vez y qué tipo de clave necesitan para su descifrado. Algunos cifrados se hackean más fácilmente que otros. Si bien algunas empresas o personas eligen el tipo de cifrado de acuerdo con los estándares dictados por las regulaciones legales o industriales, otros pueden simplemente elegir su tipo en función de sus preferencias personales.

CAPÍTULO 3

Criptosistemas más comunes basados en el logaritmo discreto

3.1 Diffie - Hellman

El nombre se debe a sus creadores Whitfield Diffie y Martin Hellman. Creado en 1976, es uno de los protocolos de intercambio de claves más antiguos.
Este algoritmo permite crear una clave secreta entre dos equipos informáticos que nunca han tenido contacto previo, a través de un canal inseguro, y mediante el envío de solo dos mensajes.

La limitación más seria de Diffie-Hellman en su forma básica o pura es la falta de autenticación. Las comunicaciones que usan Diffie-Hellman por sí solas son vulnerables a ataques de hombre en el medio. Idealmente, Diffie-Hellman debería usarse junto con un método de autenticación reconocido, como las firmas digitales, para verificar las identidades de los usuarios a través del medio de comunicación público. Diffie -Hellman es muy adecuado para su uso en la comunicación de datos, pero se usa con menos frecuencia para datos almacenados o archivados durante largos periodos de tiempo.

Ejemplo 4. *Alicia y Benito deben ponerse de acuerdo en una clave, pero no disponen de un canal seguro para intercambiar la clave. Para esto acuerdan (en público) un n° primo p y una base g, idealmente una raíz primitiva de p.*
Supongamos que eligen $p = 23$ y $g = 5$.
- Alicia elige una clave secreta $a = 6$, calcula g^a (mód p) y da a conocer el resultado a Benito. $5^6 \text{ mód } 23 = 8$
- Benito hace lo mismo, elige una clave secreta secreta $b = 15$, calcula g^b mód p

y comunica el resultado a Alicia (en público). 5^{15} *(mód 23) = 19.*
- Alicia calcula (g^b mód p, g^a mód p.
19^6 *mód 23 = 2.*
- Benito calcula $(g^a)^b$ mód p.
8^{15} mód $23 = 2$.
Por lo tanto, la clave que comparten en común es 2.
Ejemplo extraído de una clase de criptografía por Pablo De Nápoli, 2014).

3.2 Criptosistema ElGamal

Es un protocolo de criptografía asimétrica inventado por Taher Elgamal en 1984 y construido a partir del problema de logaritmo discreto. Es uno de los criptosistemas de clave pública más utilizados en la actualidad.

- **Generación de claves**

  1. $\mathcal{A}$ genera un número primo grande p.
  2. $\mathcal{A}$ elige un generador g de $\mathbb{F}_p^*$
  3. $\mathcal{A}$ genera un número aleatorio, con $1 < a < p-1$ y calcula $A = g^a$ mód p
  4. La clave pública de $\mathcal{A}$ es (p, g, A) y su clave privada es a.

- **Cifrado de mensajes**

  1. $\mathcal{B}$ obtiene la clave pública de $\mathcal{A}$ (p, g, A)
  2. $\mathcal{B}$ representa el mensaje M como un elemento de $\mathbb{F}_p^*$; es decir, como un entero m en el intervalo $[1, p-1]$.
  3. $\mathcal{B}$ genera un número aleatorio k, $1 < k < p-2$, calcula $c_1 = g^k$ mód p, y $c_2 = m \cdot A^k$ mód p y envía el criptograma $c = (c_1, c_2)$

- **Descifrado de mensajes**

  1. $\mathcal{A}$ utiliza su clave privada, a, para calcular $c_1^{-a} = c_1^{p-1-a}$ mód p.
  2. $\mathcal{A}$ obtiene m multiplicando $c_1^{-a} \cdot c_2$ mód p.

CAPÍTULO 4

Algoritmos genéricos y de colisión

4.1 El algoritmo paso pequeño paso gigante

Sea G un grupo y sea $g \in G$ un elemento de orden $N > 2$. El siguiente algoritmo resuelve el problema del logaritmo discreto $g^x = h$

1. Sea $n = 1 + \lfloor \sqrt{N} \rfloor$, entonces en particular,$n > \sqrt{N}$.

2. Crear dos listas,
   Lista 1: e, g, g^2, g^3, ... , g^n,
   Lista 2: h, hg^{-n}, hg^{-2n}, hg^{-3n}, ... , hg^{-n^2},

3. Encontrar una igualdad entre las dos listas, $g^i = hg^{-jn}$.

4. Entonces $x = i + jn$ es una solución de $g^x = h$.

Ejemplo 5.

Resolver $57 = 3^y$ mód 113. $n = \lfloor 112 \rfloor + 1 = 11$

j	3^j mód 113
0	1
1	3
2	9
3	27
4	81
5	17
6	51

i	$57 \cdot 58^i$ mód 113
0	57
1	29
2	100
3	37
4	112
5	55
6	26
7	39
8	2
9	3

$57 \cdot 58^9 = 57 \cdot 3^{-11 \cdot 9} = 3 = 3^1$ y $57 = 3^{100}$.

Por lo tanto, $x = 100$ dado que ; $n = 11$, $i = 9$ y $j = 11$.

$x = j + i \cdot n$

$x = 11 + 9 \cdot 11$

$x = 100$

4.2 Algoritmo de Pollard Rho

El algoritmo de Pollard fue introducido por John M. Pollard en 1978 y es utilizado para resolver el Problema de logaritmo discreto. Consiste en construir una secuencia pseudoaleatoria de elementos de G en la que existan dos términos iguales y, a partir de esos términos, calcular el logaritmo discreto. Se trata del algoritmo genérico más eficaz y, por ello, además de estudiar el método origina, se presentan algunas variantes del mismo. A continuación, se expone el método original en detalle.

En primer lugar, se toma una partición de G en tres conjuntos de la ecuación.aproximadamente el mismo tamaño s_1 , s_2 y s_3 con la única condición de que 1 no pertenezca a s_2. La secuencia estará dada por:

$$x_0 = 1; x_i + 1 = \begin{cases} x_i a & \text{si } x_i \in s_1 \\ x_i^2 & \text{si } x_i \in s_2 \\ x_i g & \text{si } x_i \in s_3 \end{cases}$$

Si 1 está en s_2, entonces la secuencia es constante igual a 1. Esto explica la condición requerida para s_2 a la hora de elegir la partición de G.

Como $a = g^k$, se tiene que $x_i = g^{a_1} a^{b_1}$ y los enteros a_i y b_i pueden calcularse independientemente del valor de x_i de forma recurrente:

$$a_0 = 0; a_i + 1 = \begin{cases} a_i & \text{si } x_i \in s_1 \\ 2a_i & \text{si } x_i \in s_2 \\ a_i + 1 & \text{si } x_i \in s_3 \end{cases}$$

$$b_0 = 0; b_i + 1 = \begin{cases} b_i & \text{si } x_i \in s_1 \\ 2b_i & \text{si } x_i \in s_2 \\ b_i & \text{si } x_i \in s_3 \end{cases}$$

Cuando se encuentran dos enteros i, j tales que $x_i = x_j$, se verifica que:
$g_i^a a_i^b = g_j^a a_j^b$
esto es
$g_i^a g^k b_i = g_j^a g^k b_j$
y, por lo tanto
$g_i^a - a_j = g^k(b_j - b_i)$
de donde se obtiene:
$a_i - a_j \equiv k(b_j - b_i)(\mod n)$
Denotando $u = a_i$ y $v = (b_j - b_i$ se tiene que $g^u = g^{kv}$. Es claro que, si v = 0, el algoritmono puede calcular el logaritmo discreto. En tal caso, se toma otra semilla, esto es, otro valor para x_0 y se repite el proceso. Si $v \neq 0$ se aplica el algoritmo extendido de euclides para resolver la ecuación. Sea $d = (n, v)$, entonces:

$d = vs + nt$
como

$$g^{us} = g^{kvs} = g^{k(d-nt)} = g^{kd}$$

Se tiene que $us \equiv kd(\mod n)$
y, por lo tanto
$us - wn = kd$
Puesto que $d \mid n, d \mid us$ y, finalmente, se obtiene que:

$$k = (us - wn)/d$$

donde w es un entero entre 0 y d que puede obtenerse por búsqueda exhaustiva.

Ejemplo 6.

(El siguiente ejemplo es extraído de un trabajo de licenciatura en Matemáticas en España el año 2013 por J. Santamaria Fernández).
El entero $p = 809$ es primo y $g = 89$ tiene orden 101 en $\mathbb{F}^*_{809}$. El elemento 618 pertenece al subgrupo generado por g. Se desea calcular el logaritmo en base 89 de 618 en $\mathbb{F}^*_{809}$. En primer lugar, se toma a la siguiente partición del grupo $\mathbb{F}^*_{809}$.
$s_1 = x \in \mathbb{F}^*_{809} : x \equiv 1$ mód 3
$s_2 = x \in \mathbb{F}^*_{809} : x \equiv 0$ mód 3
$s_3 = x \in \mathbb{F}^*_{809} : x \equiv 2$ mód 3
En este momento se genera la secuencia aleatoria aplicando la función definida en la descripción del método y se busca una colisión en la misma.

i	(x_i, a_i, b_i)
1	$(618, 0, 1)$
2	$(76, 0, 2)$
3	$(46, 0, 3)$
4	$(113, 0, 4)$
5	$(349, 1, 1)$
6	$(488, 1, 5)$
7	$(555, 2, 5)$

i	(x_i, a_i, b_i)
8	$(605, 4, 10)$
9	$(451, 5, 10)$
10	$(422, 5, 11)$
11	$(344, 6, 11)$
12	$(683, 7, 11)$
13	$(112, 8, 11)$
14	$(451, 8, 12)$

Se tiene que $x_9 = x_{14} = 451$. Por lo tanto, el logaritmo discreto de 618 en base 89 es:
$k = (8 - 5)(10 - 12)^{-1} = 3 \cdot 99^{-1} = 3 \cdot 50 = 49 \mod 101$

CAPÍTULO 5

Magma

Magma es un paquete de software grande y bien soportado diseñado para cálculos en álgebra, teoría de números, geometría algebraica y combinatoria algebraica. Proporciona un entorno matemáticamente riguroso para definir y trabajar con estructuras como grupos, anillos, campos, módulos, álgebras, esquemas, curvas, gráficos, diseños, códigos y muchos otros. Magma también es compatible con una serie de bases de datos diseñadas para ayudar a la investigación computacional en aquellas áreas de las matemáticas que son de naturaleza algebraica.

5.1 Cuerpos finitos y Logaritmo Discreto

- Crear el cuerpo $\mathbb{F}_q$ con $q = p^n$ donde p es un primo.

```
> FiniteField(7);
Finite field of size 7
> FiniteField(7^2);
Finite field of size 7^2
> FiniteField(7,2);
Finite field of size 7^2
> GaloisField(7);
Finite field of size 7
> GF(7);
Finite field of size 7
```

- Calcular el logaritmo discreto en base b de un elemento no cero x del cuerpo F.

```
> F:=GF(127);
> b:=F!3;
> x:=F!8;
> Log(b,x);
90
```

- Calcular el logaritmo discreto de un elemento no cero x del cuerpo F donde la base es un generador (elemento primitivo) de F que se obtiene usando PrimitiveElement.

```
> F:=GF(127);
> PrimitiveElement(F);
3
> x:=F!8;
> Log(x);
90
```

5.2 Definir funciones

Para definir una función la estructura básica es la siguiente:

nombre:=function(entrada)
return salida **;**
end function;

Ejemplo 7.

A continuación con fines didáticos daremos algunas funciones simplificadas: Una versión simplificada para cifrar con ElGamal sobre un cuerpo finito $\mathbb{F}_p$:

```
> ElGamalEncrypt:=function(m,p,g,A)
> k:=Random([2..p-3]);
> return [g^k mod p, m*A^k mod p];
> end function;
```

Una versión simplificada para descifrar con ElGamal sobre un cuerpo finito $\mathbb{F}_p$:

```
> ElGamalDecrypt:=function(c,p,g,a)
> return Modexp(c[1],-a,p)*c[2] mod p ;
> end function;
```

Ejemplo 8.

Paso de bebé y paso de gigante:

```
> BabyGiant:=function(h,g,p)
> baby:=[1];
> giant:=[h];
> n:=Floor(Sqrt(p-1))+1;
> for i:=1 to n do baby[i+1]:=baby[i]*g mod p; end for;
> gn:=Modexp(g, -n, p);
> for i:=1 to n do giant[i+1]:=giant[i]*gn mod p; end for;
> for d in Set(baby) meet Set(giant) do
> i:=Index(baby,d)-1;
> j:=Index(giant,d)-1;
> x:=i+n*j;
> end for;
> return x;
> end function;
```

Ejemplo 9.

"Pollard Rho"

El entero p = 809 es primo y g = 89 tiene orden 101 en $\mathbb{F}*_{809}$. El elemento 618 pertenece al subgrupo generado por g. Se desea calcular el logaritmo en base 89 de 618 en $\mathbb{F}^*_{809}$. En primer lugar, se toma a la siguiente partición del grupo $\mathbb{F}^*_{809}$

```
> p:=809;
> g:=89;
> 89 ∧ 101 mod 809;
> a:=618;
> x0:=1;
> alpha0:=0;
> beta0:=1;
> x:=1;
> alpha:=0;
> beta:=1;
> X:=[ ];
> for i := 1 to 14 do
```

```
>  if (x mod 3 ) eq 1 then x:=a*x mod p;
>  elif (x mod 3) eq 0
> then x:=x^2 mod p; else x:=x*g mod p;
> end if;
> X:=X cat [x];
> end for;
> print "x";
> A:=[];
> print "alpha ";
> for i:= 1 to 13 do x:=X[i];
> if (x mod 3) eq 1 then alpha:=alpha mod p;
>  elif (x mod 3) eq 0 then alpha:=2*alpha mod p;
> else alpha:=alpha+1 mod p; end if;
> A:=A cat [alpha]; end for;
> print "beta";
> B:=[];
>  for i:= 1 to 13 do x:=X[i];
>  if (x mod 3) eq 1 then beta:=beta+1 mod p;
> elif (x mod 3) eq 0 then beta:=2*beta mod p;
> else beta:=beta mod p; end if;
> B:=B cat [beta]; end for;
```

Resolución

```
> X;
 [ 618, 76, 46, 113, 349, 488, 555, 605, 451, 422, 344, 683, 112, 451 ]
> A;
[ 0, 0, 0, 1, 1, 2, 4, 5, 5, 6, 7, 8, 8 ]
> B;
[ 2, 3, 4, 4, 5, 5, 10, 10, 11, 11, 11, 11, 12 ]
```

Se tiene que $x_9 = x_{14} = 451$. Por lo tanto, el logaritmo discreto de 618 en base 89 es:
$k = (8-5)(10-12)^{-1} = 3 \cdot 99^{-1} = 3 \cdot 50 = 49$ mód 101

CAPÍTULO 6

Conclusión

Es importante aclarar que la criptografía es un método utilizado para la protección de información y las comunicaciones mediante el uso de códigos que permite que solo a aquellos a quienes está destinada la información puedan leerla y procesarla. En la antigüedad la criptografía era utilizada para la guerra, donde los soldados enviaban mensajes encriptados en algún material (papel, madera, etc.) a sus aliados y que pudieran ser leídos por sus oponentes. Algunos de los métodos utilizados en la antigüedad son: la escítala espartana, el cifrador de César, entre otros.

Hoy en día existen muchos algoritmos que se utilizan para cifrar mensajes y/o información, que son utilizados por los bancos, el poder judicial, etc. Algoritmos como Diffie-Helman, El Gamal y otros algoritmos genéricos y de colisión como paso Bebé Paso Gigante, Pollard Rho, entre otros. Para poder resolver dichos algoritmos es necesario manejar algunos conceptos tales como: Aritmética modular, divisibilidad, logaritmo discreto, etc.

Se escogió el software llamado magma para resolver estos algoritmos, ya que, fue diseñado para cálculos en álgebra, teoría de números, geometría algebraica y combinatoria algebraica. En dicho software se puede calcular logaritmo discreto, definir funciones, cifrar y descifrar el algoritmo ElGamal, Pollard Rho y Paso de bebé y paso de gigante (como se muestra en los ejemplos realizados). Dicho software nos permitió crear cuerpos finitos, logaritmo discreto y su resolución, así como también permitió resolver algoritmos de colisión como Pollard rho y Paso de bebé y paso de gigante que involucran el logaritmo discreto.

Bibliografía

[1] Ivo Basso, Edgardo Riquelme: Apuntes básicos de criptografía, 2017.

[2] Ivo Basso, *Apuntes de Aritmética*, UBB,2007.

[3] Hoffstein, J. Pipher, J.H. Silverman, An introduction to mathematical cryptography Springer, New York, 2008.

[4] Jorge Blasco, Antecedentes y perspectivas de estudio en estudio de la Criptografía, 2009.

[5] Jennifer Santamaría, el algoritmo discreto y sus aplicaciones en criptografía, 2013.

[6] Luis Suaréz, Criptografía y seguridad,2005.

[7] Joppe W. Bos, Alina Dudeanu y Dimitar Jetchev,Collision bounds for the additive Pollard Rho alagotithm for solving discrete logarithms,2014.

[8] Richard Mollin, An Introduction to Cryptography, CRC Press, 2007.

[9] Emilio Lluis, Teoría de grupos, 2014.

[10] Pablo Galán, Héctor García y Mateo Pinzón, Criptografía desde su origen hasta la actualidad.

[11] Joaquín Oyarzún, algebra y geometría, capítulo 3 aritmética modular.

[12] Victoria Peréz, Logaritmo discreto, 2010.

[13] Felipe Zaldivar, Introducción a la teoría de números. Fondo de Cultura Económica 2012.