INSTITUTO POTOSINO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA, A.C.

POSGRADO EN CONTROL Y SISTEMAS DINÁMICOS

Protocolo algebraico de establecimiento de una llave pública basado en trenzas

Tesis que presenta

David Iván Hernández Granados

Para obtener el grado de

Maestro en Control y Sistemas Dinámicos

Director de la Tesis:

Dr. Hugo Cabrera Ibarra

San Luis Potosí, S.L.P., 01 de agosto de 2018

IPICYT

Constancia de aprobación de la tesis

La tesis ***"Protocolo algebraico de establecimiento de una llave pública basado en trenzas"*** presentada para obtener el Grado de Maestro en Control y Sistemas Dinámicos, fue elaborada por **David Iván Hernández Granados** y aprobada el **primero de agosto del dos mil dieciocho** por los suscritos, designados por el Colegio de Profesores de la División de Matemáticas Aplicadas del Instituto Potosino de Investigación Científica y Tecnológica, A.C.

Dr. Hugo Cabrera Ibarra
Director de la tesis

Dr. Juan Gonzalo Barajas Ramírez
Jurado en el Examen

Dr. David Antonio Lizárraga Navarro
Jurado en el Examen

Dr. Eric Campos Cantón
Jurado en el Examen

Créditos Institucionales

Esta tesis fue elaborada en la División de Matemáticas Aplicadas del Instituto Potosino de Investigación Científica y Tecnológica, A.C., bajo la dirección del Dr. Hugo Cabrera Ibarra.

Durante la realización del trabajo el autor recibió una beca académica del Consejo Nacional de Ciencia y Tecnología 619730 y del Instituto Potosino de Investigación Científica y Tecnológica, A. C.

IPICYT

Instituto Potosino de Investigación Científica y Tecnológica, A.C.

Acta de Examen de Grado

El Secretario Académico del Instituto Potosino de Investigación Científica y Tecnológica, A.C., certifica que en el Acta 030 del Libro Primero de Actas de Exámenes de Grado del Programa de Maestría en Control y Sistemas Dinámicos está asentado lo siguiente:

En la ciudad de San Luis Potosí a los 1 días del mes de agosto del año 2018, se reunió a las 12:30 horas en las instalaciones del Instituto Potosino de Investigación Científica y Tecnológica, A.C., el Jurado integrado por:

Dr. David Antonio Lizárraga Navarro	**Presidente**	**IPICYT**
Dr. Hugo Cabrera Ibarra	**Secretario**	**IPICYT**
Dr. Eric Campos Cantón	**Sinodal**	**IPICYT**
Dr. Juan Gonzalo Barajas Ramírez	**Sinodal**	**IPICYT**

a fin de efectuar el examen, que para obtener el Grado de:

MAESTRO EN CONTROL Y SISTEMAS DINÁMICOS

sustentó el C.

David Iván Hernández Granados

sobre la Tesis intitulada:

Protocolo algebraico de establecimiento de una llave pública basado en trenzas

que se desarrolló bajo la dirección de

Dr. Hugo Cabrera Ibarra

El Jurado, después de deliberar, determinó

APROBARLO

Dándose por terminado el acto a las 14:30 horas, procediendo a la firma del Acta los integrantes del Jurado. Dando fe el Secretario Académico del Instituto.

A petición del interesado y para los fines que al mismo convengan, se extiende el presente documento en la ciudad de San Luis Potosí, S.L.P., México, a los 1 días del mes de agosto de 2018.

Dr. Horacio Flores Zúñiga
Secretario Académico

Mtra. Ivonne Lizette Cuevas Vélez
Jefa del Departamento del Posgrado

INSTITUTO POTOSINO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA, A.C.
IPICYT
SECRETARIA ACADEMICA

Agradecimientos

• A mí asesor el *Dr. Hugo Cabrera Ibarra*, por brindarme su apoyo total, su buen ejemplo y por invertir en mí no sólo su conocimiento sino también algo de su tiempo. Especialmente, agradecer su confianza y la oportunidad que me brindo para iniciarme en el ámbito de la ciencia y su divulgación.

• A *mis dos familias*, fuente de apoyo constante e incondicional. Especialmente, agradecer a mis padres ya que, sin su ejemplo, esfuerzo, comprensión y amor, no hubiera tenido la oportunidad de ser una persona de bien y mucho menos tener una formación académica.

• A *mi esposa*, por su apoyo incondicional, paciencia y amor. Especialmente, agradecer sus palabras de aliento, sus retos, su buen ejemplo y su dedicación, lo cual me motiva siempre a seguir adelante y a esforzarme un poco más día a día con el fin de estar a su nivel.

• Por último y no menos importante. Quiero agradecer *a los doctores*: Juan Gonzalo Barajas Ramírez, Eric Campos Cantón y David Antonio Lizárraga Navarro, por sus comentarios, conocimientos y disponibilidad. Permitiéndome con ello madurar mi conocimiento y visualizar otras perspectivas.

Contenido

Lista de figuras

Resumen

Dado que la finalidad de la criptología es ocultar y trasmitir información, usualmente usando un canal público, es importante establecer un procedimiento que cumpla con dichos propósitos de manera segura y eficiente, sin olvidar que debe resolver principalmente el problema de la confidencialidad.

Por ello en esta tesis, siguiendo el trabajo de Anshel y colaboradores en [1] y de Ki Hyoung Ko y colaboradores en [3], se desarrolló un protocolo algebraico para establecer una llave privada por medio del grupo no conmutativo de las 3-trenzas, buscando con ello trasmitir nuestra información por un canal público. Además, en dicho desarrollo se propuso también un procedimiento denominado ML_2, basado en el uso del mapeo logístico, con el fin de generar una cadena de 0's y 1's, con la cual se encripte o desencripte una cadena de 0's y 1's asociada a un texto plano, buscando con ello el ocultar o recuperar nuestra información respectivamente. Por último, se implementó este protocolo en el software MATLAB® y además se optó por desarrollar una interfaz gráfica, para con ello permitir a los usuarios una interacción cómoda y visual con dicho protocolo, se incluyó también el pseudocódigo correspondiente para su implementación en cualquier lenguaje de programación.

Palabras clave: Criptografía, cifrado en flujo, criptografía de llave pública, 3-trenzas, mapeo logístico.

Abstract

The purpose of cryptology is hide and transmit information, usually using a public channel. For this reason, it is important to establish a procedure that accomplishes these purposes safely and efficiently, without forgetting that it must mainly solve the problem of the confidentiality.

Therefore, in this thesis, following the work of Anshel *et al*. in [1] and the work of Ki Hyoung Ko *et al*. in [3], an algebraic protocol was developed to establish a private key through the non-commutative group of the 3-braids, with the propose to transmit our information through a public channel. Besides, in this development a procedure called ML_2 was proposed, based on the use of logistic map, to generate a chain of 0's and 1's, with which can to encrypt or decrypt a chain of 0's and 1's associated with a plain text, with the propose to hide or recover our information respectively. Finally, this protocol was implemented in the MATLAB ® software and also it was decided to develop a graphical interface with the propose to permit to users a comfortable and visual interaction with this protocol, additionally the corresponding pseudocode was include for its implementation in any programming language.

Key Words: Cryptography, stream cipher, public-key cryptography, 3-braids, logistic map.

Capítulo 1

Introducción

Las nuevas tecnologías han impuesto cambios en las formas como interactuamos con nuestro medio, un ejemplo de ello son las transacciones monetarias, si bien nos aportan facilidades también nos inquietan, dado que no tenemos la certeza de que nuestro dinero y nuestros datos están bien resguardados en este nuevo mundo virtual. Con este fin, nos interesa encontrar una forma con la que podamos establecer quien accede a nuestra información.

La criptografía ha establecido procesos con el fin de ocultar información desde tiempos antiguos, quizás el proceso más antiguo conocido para ocultar información es el de los egipcios, los cuales codificaban cierta información en jeroglíficos. Buscando extender este fin a nuestro tiempo se busca incluir en la criptografía clásica un canal público y con ello generar un código más eficiente y seguro, requiriendo para ello de una llave confidencial para encriptar y de una llave pública para desencriptar. Los esquemas de encriptación clásicos a lo largo del tiempo se han vuelto vulnerables a diversos ataques, dado que se basan en aritmética simple y son cifradores simétricos, los cuales son algoritmos que utilizan la misma llave tanto para cifrar como para descifrar un mensaje, especialmente el uso de las computadoras ha permitido que dichos esquemas se descifren fácilmente.

En los últimos años se han realizado diversas investigaciones con el fin de encontrar esquemas de encriptación más seguros y eficientes basados en grupos no abelianos, semigrupos o anillos [1, 2] y donde la seguridad se base en la dificultad de resolver en ellos problemas característicos de las diferentes estructuras algebraicas. Especialmente, los grupos de interés incluyen al grupo de las n-trenzas, a los grupos lineales y a los grupos modulares [3, 4, 5]. Dada la dificultad de resolver el problema de la palabra en un grupo, esté podría utilizarse como recurso para la criptografía de clave pública. Por ello, nuestro objetivo es el implementar un esquema de encriptación, él cual se base en el uso del grupo no conmutativo de las 3-trenzas, la representación de un texto plano en una cadena de 0's y 1's y la facilidad para encriptar o desencriptar un texto plano sólo si se conoce la llave secreta, esto, siguiendo los trabajos de Anshel y colaboradores [1] y de Ki Hyoung Ko y colaboradores [3].

Así, el presente trabajo busca implementar un protocolo algebraico para el establecimiento de una llave pública, el cual se base en el uso del grupo no conmutativo de las 3-trenzas y en la dificultad de resolver en él el problema de la palabra. Para ello:

En el capítulo 2 se expondrán algunos detalles y conceptos importantes que nos servirán como base para entender este trabajo, como lo son algunos conceptos básicos utilizados en la criptología. Se introducirán también, algunos conceptos y ejemplos con el fin de definir el grupo no conmutativo de las 3-trenzas. Después, en el capítulo 3 siguiendo el trabajo de Anshel y colaboradores [1] y de Ki Hyoung Ko y colaboradores [3] se implementará un protocolo de llave pública, al cual denominaremos TC, proponiendo para ello utilizar el grupo no conmutativo de las 3-trenzas con el fin de generar una llave privada y de distribuir una llave pública simétrica. También, se establecerá el procedimiento ML_2, basado en el mapeo logístico, con la finalidad de generar una cadena de 0's y 1's para encriptar y desencriptar información. Además, en el capítulo 4 el protocolo TC será implementado en una interfaz gráfica mediante el software MATLAB®, con la finalidad de permitir al usuario interactuar fácilmente con dicho protocolo. Luego, en el capítulo 5 se establecerán las conclusiones referentes a este trabajo. Por último, en el Apéndice se mostrará el pseudocódigo referente al algoritmo para establecer el protocolo TC, con la finalidad de que se pueda implementar en diversos lenguajes de programación.

Capítulo 2

Preliminares

En el presente capítulo se expondrán algunos conceptos que nos servirán como base para el desarrollo de esta tesis.

2.1. Criptología

Actualmente el uso de Internet está ganando terreno e imponiendo cambios en las formas tradicionales en las que interactuamos con nuestro medio. Especialmente hemos adoptado esta tecnología con el fin de ser más eficientes al realizar nuestras tareas cotidianas, por ejemplo, realizar compras y pagos online a través de PAYPAL, hacer transferencias bancarias a través de medios electrónicos como SPEI, solicitar servicios de transporte e inclusive reservar un lugar en nuestro restaurante favorito. Esta comodidad nos beneficia, pero al mismo tiempo nos inquieta, dado que no tenemos la certeza de que nuestra información y nuestro dinero estén seguros dentro de este nuevo mundo digital, por ejemplo, el reciente robo de datos confidenciales a Facebook o el reciente ataque cibernético que sufrieron algunas instituciones financieras mexicanas que afectó su proceso de conexión con SPEI. Una forma de lidiar con esta inquietud es buscar un mecanismo que oculte nuestra información y nos permita compartirla sólo con personas a quienes autoricemos.

En toda comunicación requerimos de un emisor, un receptor y un mensaje: el **emisor** es la persona o dispositivo que trasmitirá algún tipo de información, el **receptor** es la persona o dispositivo que recibirá la información enviada por el emisor y el **mensaje** es la información que nos interesa compartir, teniendo en mente que dicho mensaje será enviado por un medio donde cualquier persona puede leerlo, dicho medio lo denominaremos un **canal público**. Así, con el fin de ocultar información, se requiere de un procedimiento que permita sólo al emisor y al receptor acceder a ella. Es en este punto que nos encontramos con una buena herramienta: La criptología.

El término **criptología** proviene de los vocablos griegos *kryptós* y *logós* que significan "oculto" y "estudio", respectivamente, así es que la criptología tiene como finalidad estudiar distintas formas de ocultar información [6]. Además, al momento de transmitir información, se busca resolver los siguientes problemas [7] :

- La **confidencialidad**: Que sólo las personas autorizadas deban tener acceso a la información.
- La **integridad**: Que se pueda verificar si la información transmitida no ha sido alterada.
- La **autenticidad**: Que el emisor y el receptor realmente sean quienes dicen ser.
- El **no rechazo**: Que si el emisor envía un mensaje, este no pueda negar ante terceros, que envió dicha información. Del mismo modo, que el receptor, una vez que reciba el mensaje, no pueda negar que recibió la información.
- La **disponibilidad**: Que el mensaje deba estar disponible para cualquier entidad que lo requiera, ya sea una persona o un dispositivo.

El estudio de la criptología se enfoca principalmente en dos vertientes [8]:

La **criptografía** cuya finalidad es ocultar información, dicha información es ocultada a través de algunos de los siguientes algoritmos:

- Simétricos, también conocidos como algoritmos de **llave privada**: Cuya efectividad se basa en compartir una llave secreta. Hasta 1976 la criptografía estaba basada en estos algoritmos principalmente.
- Asimétricos, también conocidos como algoritmos de **llave pública**: Cuya efectividad se basa en generar una llave secreta y compartir una llave pública. La criptografía después de 1976 tuvo un cambio debido a estos nuevos algoritmos, los cuales fueron introducidos por Whitfield Diffie and Martin Hellman en [9].

Y su contraparte, el **criptoanálisis** cuya finalidad es romper los esquemas de encriptación para descifrar información oculta.

Hasta el momento sólo hemos indagado un poco sobre la criptología, pero ¿Cómo se puede aplicar para ocultar información? La siguiente explicación nos dará una mejor idea al respecto:

Si el emisor quiere utilizar un algoritmo simétrico para ocultar su información, requerirá de una clave, la cual denominaremos **llave**. Por otra parte, el receptor necesitará recibir el mensaje, conocer el algoritmo que se utilizó para ocultar el mensaje y la llave, esto con el fin de acceder al mensaje, ver Figura 2.1.

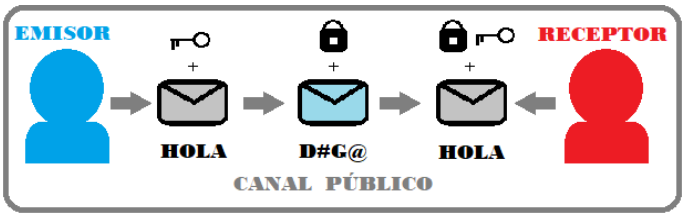


Figura 2.1: *Ejemplo de intercambio de información, caso simétrico.*

Si en su defecto, el emisor quiere utilizar un algoritmo asimétrico para ocultar su información, requerirá de una llave privada y de una llave que enviará por un canal público, la cual denominaremos **llave pública**. Por otra parte, el receptor necesitará recibir el mensaje, conocer el algoritmo que se utilizó para ocultar el mensaje y utilizar tanto su llave privada como la llave pública recibida, esto con el fin de acceder al mensaje, ver Figura 2.2.

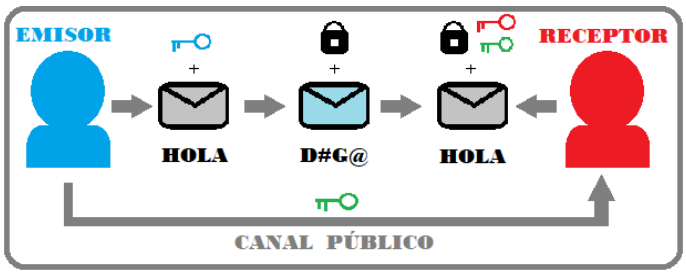


Figura 2.2: *Ejemplo de intercambio de información, caso asimétrico.*

En ambos casos, cuando el emisor oculte información mediante alguno de los dos algoritmos diremos que está **encriptando**, mientras que, cuando el receptor acceda al mensaje diremos que está **desencriptando**.

2.2. El grupo de las 3-trenzas

Ki Hyoung Ko y colaboradores en [3] muestran que el grupo de trenzas es un buen recurso para enriquecer la criptografía. Dado que el grupo de trenzas tiene muchos problemas matemáticamente difíciles de resolver, que involucran a la topología, la combinatoria y a la teoría de grupos; como por ejemplo: El **problema de la palabra**, dada una estructura algebraica generada de forma finita, el problema es determinar si dos palabras en los generadores representan al mismo elemento. O el problema de la conjugación, dada una estructura algebraica, el problema es determinar si dos palabras en los generadores representan un elemento conjugado. Los cuales pueden ser utilizados para diseñar algoritmos criptográficos cuya seguridad se base en la dificultad de resolverlos.

Para definir el grupo de trenzas son necesarios los conceptos de grupo y grupo abeliano.

Definición 2.2.1. *[10]: Un **grupo** (*G,•*) es un conjunto* G*, junto con una operación binaria*[1] *"•" en* G *que satisface los siguientes axiomas:*

(i) La operación binaria • es **asociativa**, es decir se satisface:

$(a \bullet b) \bullet c = a \bullet (b \bullet c)$ para toda a, b, c en G.

(ii) Existe un elemento e en G tal que:

$e \bullet x = x \bullet e = x$ para todas las x en G.

Este elemento e es un **elemento identidad** para • en G.

(iii) Para cada a en G existe un elemento a' en G tal que:

$a' \bullet a = a \bullet a' = e$

El elemento a' es un **inverso** de a para • en G.

Definición 2.2.2. *[10]: Un **grupo abeliano** es un grupo* G *cuya operación binaria "•" es conmutativa, es decir se satisface:*

$$a \bullet b = b \bullet a \quad \text{para toda } a \text{ y } b \text{ en G.}$$

Ejemplo 1. Si definimos la operación binaria "•" en $\mathbb{Q}^+$, números racionales positivos, tal que: a • b = $\frac{ab}{2}$, tendremos que $\mathbb{Q}^+$ con dicha operación es un grupo. A saber, la operación es asociativa, donde e=2 es un elemento identidad y $a' = \frac{4}{a}$ es un inverso de a.

Ejemplo 2. El conjunto $\mathbb{Z}^+$, números enteros positivos, con la operación binaria suma "+" no es un grupo. Dado que no existe un elemento identidad para la suma en $\mathbb{Z}^+$.

[1] Es una regla que asigna a cada par ordenado de elementos de un conjunto, algún elemento del mismo conjunto.

Ejemplo 3. Si consideramos el conjunto finito G $= \{e, a, b\}$ y la operación binaria "$\star$" que se describe en la siguiente tabla:

$\star$	e	a	b
e	e	a	b
a	a	b	e
b	b	e	a

Para operar dos elementos de la tabla, se elige un elemento de la primer columna y un elemento de la primer fila, para obtener el resultado de la operación, buscamos el elemento donde se intersecan la columna con la fila. Por ejemplo: $a \star b = e$ mientras que $b \star b = a$.

Analizando dicha tabla tendremos que G será un grupo abeliano. Es decir, la operación es asociativa y abeliana, donde e es un elemento identidad y a es el inverso de b y viceversa.

Ejemplo 4. Si consideramos el conjunto finito de simetrías del triángulo equilátero D_3 y la operación binaria "$\circ$" que se describe en la siguiente tabla:

$\circ$	e	a	a^2	b	ab	a^2b
e	e	a	a^2	b	ab	a^2b
a	a	a^2	e	ab	a^2b	b
a^2	a^2	e	a	a^2b	b	ab
b	b	a^2b	ab	e	a^2	a
ab	ab	b	a^2b	a	e	a^2
a^2b	a^2b	ab	b	a^2	a	e

Por lo cual, D_3 es un grupo no abeliano. Notar que $a \circ b = ab$ mientras que $b \circ a = a^2b$. Por lo tanto, se tiene que $a \circ b \neq b \circ a$.

Ejemplo 5. El conjunto $\mathbb{R}$ con la operación suma "+" es un grupo abeliano. Dado que, la operación es asociativa y abeliana, donde e=0 es un elemento identidad y $a' = -a$ es un inverso de a.

También es importante definir el concepto de subgrupo y subgrupo generado.

Definición 2.2.3. *Si* H *es un subconjunto de un grupo* G *cerrado bajo la operación del grupo de* G *y si* H *es él mismo un grupo bajo esta operación inducida, entonces* H *es un* ***subgrupo*** *de* G.

Ejemplo 6. Si consideramos los conjunto $\mathbb{Z}$ y $\mathbb{R}$, ambos con la operación suma "+", se tiene que $\mathbb{Z}$ es un subgrupo de $\mathbb{R}$. Ya que, el subconjunto $\mathbb{Z}$ es cerrado bajo la operación "+" y además con dicha operación es un grupo.

Ejemplo 7. Considerando el Ejemplo 3 y el subconjunto H $= \{e, a\}$ se tiene que H no será un subgrupo de G. Dado que, H no es cerrado bajo "$\star$". A saber, $a \star a = b$ y $b \notin$ H.

Definición 2.2.4. *[10]: Sea* G *un grupo y* $a_i \in$ G *para* $i \in I$. *El menor subgrupo de* G *que contiene* $\{a_i\}$ *es el* ***subgrupo generado*** *por* $\{a_i\}$, *al cual denotamos* $\langle a_i \rangle$. *Es decir,* $\langle a_i \rangle$ *está conformado por todos los elementos que pueden ser expresados como la operación de elementos* $\{a_i\}$ *y de sus inversos.*

Ejemplo 8. Considerando el Ejemplo 4 estableceremos $\langle b \rangle$, $\langle a \rangle$ y $\langle ab, b \rangle$.

$$\langle b \rangle = \{e, b\} \qquad \langle a \rangle = \{e, a, a^2\} \qquad \langle ab, b \rangle = \{e, a, a^2, b, ab, a^2 b\}$$

Además, es importante definir el concepto de trenza, el cual fue introducido a principios de 1930 por E. Artin con el fin de aplicarlo en el estudio de la teoría de nudos, para ello utilizaremos como referencia [11].

Para definir lo que es una trenza primero debemos imaginar un cubo C, y marcar en él n puntos tanto en su parte lateral izquierda, $A_1, A_2, ..., A_n$, como derecha, $A'_1, A'_2, ..., A'_n$; dichos puntos pueden ser colocados arbitrariamente, sin embargo, para facilitar la exposición utilizaremos las siguientes convenciones:

Primero, debemos definir el conjunto C en $\mathbb{R}^3$ como:

$$C = \{(x, y, z) | 0 \leq x, y, z \leq 1\}$$

Y escogeremos los puntos $A_1, A_2, ..., A_n$ y los $A'_1, A'_2, ..., A'_n$ de la siguiente forma:

$$A_1 = \left(\tfrac{1}{2}, 0, \tfrac{1}{1}\right), ..., A_n = \left(\tfrac{1}{2}, 0, \tfrac{1}{n}\right) \quad \text{y} \quad A'_1 = \left(\tfrac{1}{2}, 1, \tfrac{1}{1}\right), ..., A'_n = \left(\tfrac{1}{2}, 1, \tfrac{1}{n}\right)$$

Dichas convenciones están representadas en la Figura 2.3.

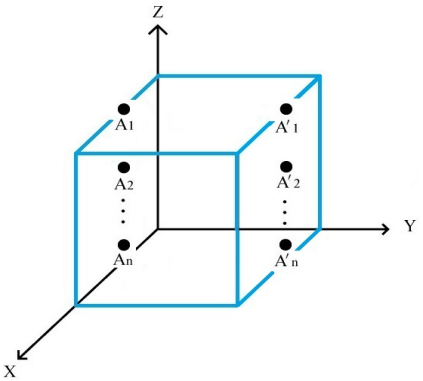


Figura 2.3: *Los puntos* $A_1, A_2, ..., A_n$ *y* $A'_1, A'_2, ..., A'_n$ *en el cubo* C.

Después, uniremos los puntos $A_1, A_2, ..., A_n$ con los puntos $A'_1, A'_2, ..., A'_n$ mediante n curvas poligonales en C, de tal forma que dichas curvas no se intersecten. Denominaremos **cuerdas** a dichas curvas poligonales.

Por último, imaginemos un plano arbitrario E, el cual divida en dos partes al cubo C y que sea perpendicular a la base del cubo. Entonces si E interseca cada cuerda en un único punto, se dirá que estas n cuerdas en el cubo C son una **n-trenza**. Es de nuestro interés trabajar con la proyección de una n-trenza sobre el plano YZ, a tal proyección se le denominará **diagrama regular** si dicho diagrama no posee puntos de tangencia, puntos triples y en los puntos dobles se marca el segmento que pasa por debajo de otro con una línea discontinua. Además, los puntos dobles serán denominados **cruces**.

Ejemplo 9. En la Figura 2.4 podemos observar dos diagramas que no son regulares. El primero posee poca información de como se unen las dos cuerdas que pasan por debajo de la cuerda central, mientras que, el segundo en el cruce no se marca que segmento pasa por debajo.

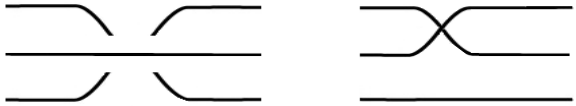


Figura 2.4: *Dos diagramas no regulares.*

Ejemplo 10. En la Figura 2.5 podemos observar una 3-trenza dentro del cubo C y su respectivo diagrama regular, de aquí en adelante sólo utilizaremos el diagrama regular de la n-trenza.

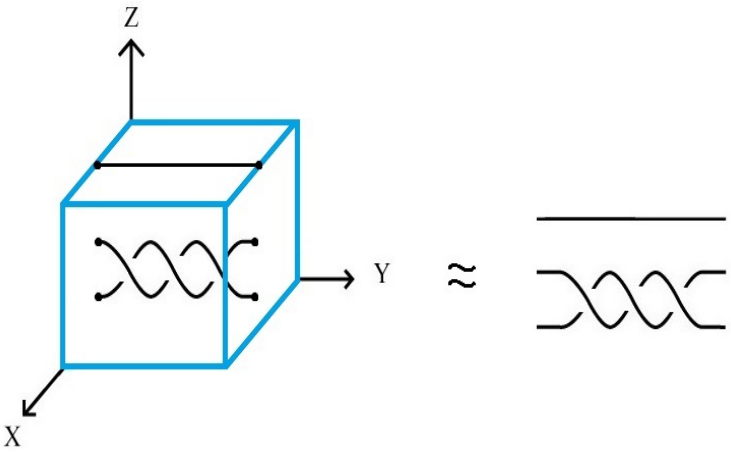


Figura 2.5: *Una 3-trenza y su diagrama regular.*

Ejemplo 11. En la Figura 2.6 podemos observar 3 cuerdas que forman una 3-trenza.

Figura 2.6: *Es una 3-trenza.*

Ejemplo 12. La Figura 2.7 muestra 4 cuerdas que no forman una 4-trenza, dado que, cada A_i debe unirse a un A'_i y en este caso A_1 esta unido con A_3 y A'_2 con A'_3.

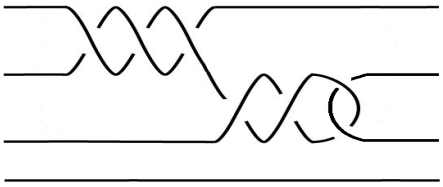


Figura 2.7: *No es una 4-trenza.*

Es de nuestro interés trabajar con las 3-trenzas, denotadas como $\mathcal{T}(a_1, a_2, ..., a_n)$ con $n \in \mathbb{N}$ y dependiendo del valor de n tendremos distintas 3-trenzas. La Figura 2.8 muestra el diagrama de una 3-trenza.

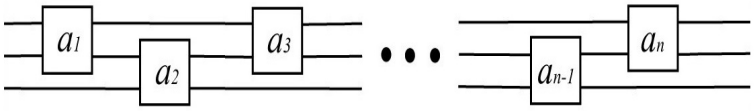


Figura 2.8: *Una 3-trenza.*

En la Figura 2.8 los a_i hacen referencia a $|a_i|$ cruces tomando en cuenta la nomenclatura de la Figura 2.9. En cada cruce nos fijaremos en la primer cuerda que intervendrá en él, si dicha cuerda pasa por arriba de otra la marcaremos con una línea continua, en caso contrario, la marcaremos con una línea sesgada.

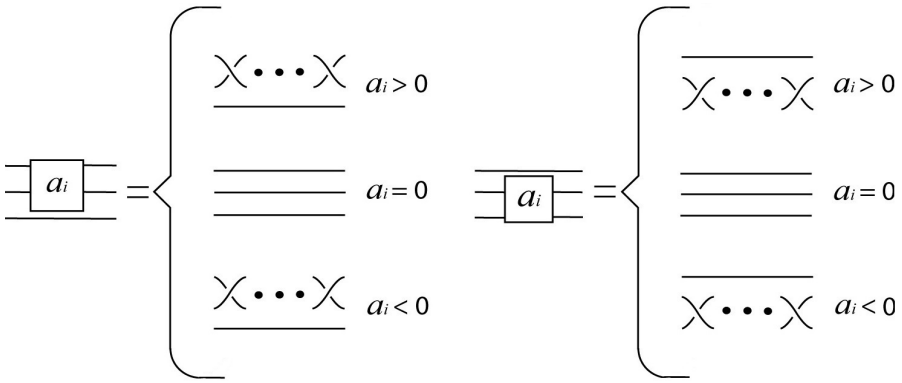


Figura 2.9: *Nomenclatura de los cruces.*

Ejemplo 13. En la Figura 2.10 podemos observar la 3-trenza $\mathcal{T}(-1,-3,1)$.

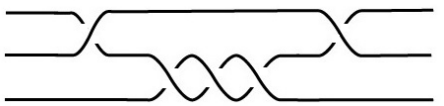
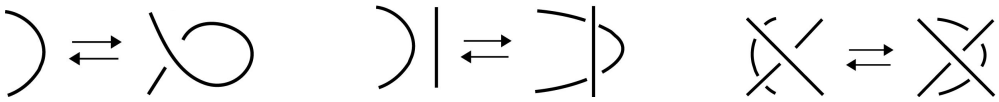


Figura 2.10: *3-trenza* $\mathcal{T}(-1,-3,1)$.

Imaginemos ahora que tenemos dos 3-trenzas $\mathcal{T}_1$ y $\mathcal{T}_2$ una pregunta interesante es ¿Cómo determinar si la 3-trenza $\mathcal{T}_1$ es equivalente a la 3-trenza $\mathcal{T}_2$? En general, un problema difícil de resolver es el determinar cuándo dos diagramas regulares son equivalentes, o en su defecto diferentes.

Definición 2.2.5. *Si se puede deformar un diagrama regular D en uno D′ mediante un número finito de* ***movidas de Reidemeister*** *locales Ω_1, Ω_2 y Ω_3, mostradas en la Figura 2.11. Entonces diremos que D y D′ son* ***equivalentes*** *y lo denotaremos como $D \cong D'$.*



(a) Movida Ω_1. (b) Movida Ω_2. (c) Movida Ω_3.

Figura 2.11: *Movidas de Reidemeister.*

Ejemplo 14. En la Figura 2.12 podemos observar dos 3-trenzas que son equivalentes. Note que, se puede deformar una en la otra usando la movida Ω_2.

Figura 2.12: *3-trenzas equivalentes.*

Así, las movidas de Reidemeister nos permiten determinar cuándo dos diagramas son equivalentes, pero no establecen el número de movidas necesarias para transformar uno en el otro. Por ello, es necesario el uso de un invariante que nos dé más información para poder distinguirlos. Un **invariante** es una función de un conjunto a otro cuyos valores dependen sólo de las clases de equivalencia. La cual asocia un número, una propiedad o un polinomio, de tal forma que se preserve bajo las movidas de Reidemeister. Quizás, el invariante de trenzas más simple es el que le asocia a una trenza su matriz permutación.

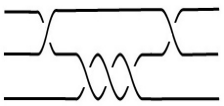
Para definirlo, imaginemos una n-trenza $\mathcal{T}(a_1, a_2, ..., a_n)$ cuyas n cuerdas estén conectadas de la siguiente forma: A_1 con A_4', A_2 con A_1', A_3 con A_n', $\cdots$, A_n con A_2'. Entonces podemos asociar a dicha n-trenza su **matriz permutación**, la cual denotaremos como ρ:

$$\rho(\mathcal{T}(a_1, a_2, ..., a_n)) = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ 4 & 1 & n & \cdots & 2 \end{pmatrix}$$

Donde en el primer renglón se colocan los puntos iniciales de cada cuerda de la n-trenza y en el segundo renglón se colocan sus puntos finales.

Así, si dos matrices permutación asociadas a dos n-trenzas son diferentes entonces dichas n-trenzas no son equivalentes.

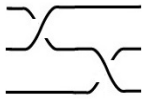
Ejemplo 15. Asociaremos su matriz permutación a la siguiente 3-trenza:



$\mathcal{T}(-1,-3,1)$

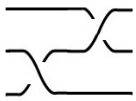
$$\rho(\mathcal{T}(-1,-3,1)) = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

Ejemplo 16. Veamos que la 3-trenza $\mathcal{T}(-1,-1)$ no es equivalente a la 3-trenza $\mathcal{T}(0,-1,-1)$. Las 3-trenzas $\mathcal{T}(-1,-1)$ y $\mathcal{T}(0,-1,-1)$ no son equivalentes, dado que sus matrices permutación asociadas son diferentes.



$\mathcal{T}(-1,-1)$

$$\rho(\mathcal{T}(-1,-1)) = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$



$\mathcal{T}(0,-1,-1)$

$$\rho(\mathcal{T}(0,-1,-1)) = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

Así, $\rho(\mathcal{T}(-1,-1))$ es diferente a $\rho(\mathcal{T}(0,-1,-1))$. Y por lo tanto, dichas 3-trenzas no son equivalentes.

Para dotar a las 3-trenzas con una estructura de grupo es necesario establecer una operación binaria. Para ello, definiremos la **suma** de 3-trenzas a la cual representaremos con el signo "+". Dicha suma consiste en unir dos trenzas, uniendo las cuerdas finales de una con las cuerdas iniciales de la otra, con lo cual obtendremos una nueva trenza, ver Ejemplo 17. Además, podemos mover dichas cuerdas libremente con la condición de no romper las cuerdas y de mantener fijos sus extremos. Por conveniencia en la nomenclatura siempre se inicia con cruces entre las cuerdas uno y dos, de no existir cruces, usaremos una pareja de números para denotar a la 3-trenza, el primero de ellos será un cero, mientras que, el segundo de ellos corresponderá al numero de cruces entre las cuerdas dos y tres, ver Ejemplo 18.

Ejemplo 17. En la Figura 2.13 podemos observar como la 3-trenza $\mathcal{T}(-1,-3)$ puede ser representada, respetando el orden, como la suma de las trenzas: $\mathcal{T}(-1)$ y $\mathcal{T}(0,-3)$.

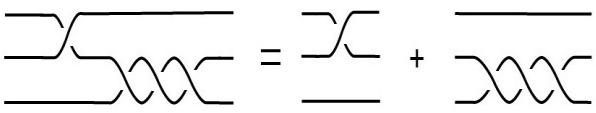


Figura 2.13: *Suma de 3-trenzas.*

Ejemplo 18. En la Figura 2.14 podemos observar una 3-trenza que no inicia con cruces entre las cuerdas uno y dos. Para denotar a esta 3-trenza usaremos la pareja de números $(0,-3)$. Donde, el primero de ellos será un 0, dado que la trenza no posee cruces entre las cuerdas uno y dos, mientras que, el segundo de ellos será un -3, dado que dicha trenza posee tres cruces negativos entre las cuerdas dos y tres.

Figura 2.14: *La 3-trenza* $\mathcal{T}(0,-3)$.

Con los conceptos definidos en esta sección, veremos que las 3-trenzas forman un grupo no conmutativo, el cual denotamos como $\mathcal{G}$ y cuya operación binaria es la suma. A saber:

(i) La operación binaria suma es asociativa.

Ejemplo 19. En la Figura 2.15 se muestra un ejemplo de que la suma es asociativa. Así:

$$(\mathcal{T}(-1)+\mathcal{T}(0,-3))+\mathcal{T}(1)=\mathcal{T}(-1,-3,1)=\mathcal{T}(-1)+(\mathcal{T}(0,-3)+\mathcal{T}(1))$$

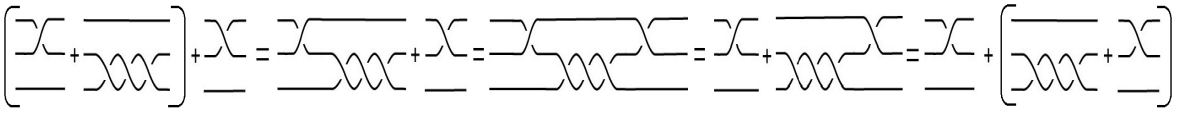


Figura 2.15: *La 3-trenza* $\mathcal{T}(-1,-3,1)$.

(ii) En la Figura 2.16 se muestra el elemento identidad para la suma en $\mathcal{G}$.

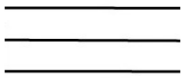


Figura 2.16: *La 3-trenza identidad* $\mathcal{T}(0)$.

Ejemplo 20. En la Figura 2.17 se muestra la suma de la 3-trenza identidad $\mathcal{T}(0)$ con la 3-trenza $\mathcal{T}(1)$. Podemos deformar la 3-trenza resultante en la 3-trenza $\mathcal{T}(1)$.

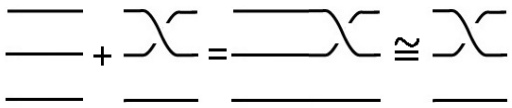


Figura 2.17: *Suma de la 3-trenza identidad.*

(iii) Para cada 3-trenza en $\mathcal{G}$ existe una 3-trenza inversa tal que al sumarlas obtenemos la 3-trenza identidad. Si a una 3-trenza le cambiamos sus cruces, a saber, el cruce que pasa por arriba es cambiado por un cruce que pasa por abajo y viceversa, podemos obtener su 3-trenza inversa.

Ejemplo 21. En la Figura 2.18 se muestra la suma de la 3-trenza $\mathcal{T}(-1)$ con su inversa $\mathcal{T}(1)$, obteniendo así la 3-trenza identidad $\mathcal{T}(0)$. Podemos deformar la 3-trenza resultante en la 3-trenza identidad aplicando una movida de Reidemeister Ω_2.

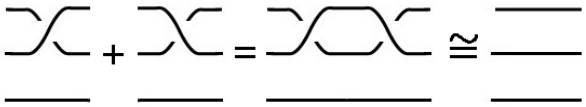


Figura 2.18: *Suma de una 3-trenza con su inversa .*

(iv) El grupo $\mathcal{G}$ no es un grupo abeliano.

Ejemplo 22. La Figura 2.19 muestra una suma de 3-trenzas que no cumple ser conmutativa. En el Ejemplo 16 se mostró que las 3-trenzas, $\mathcal{T}(-1,-1)$ y $\mathcal{T}(0,-1,-1)$, no son equivalentes dado que sus matrices permutación asociadas son diferentes.

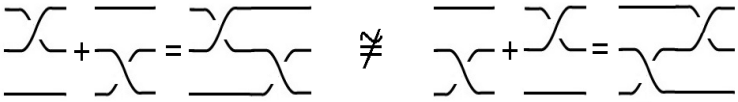


Figura 2.19: *La suma de 3-trenzas no es conmutativa.*

Con el fin de poder utilizar a las 3-trenzas como un recurso para la criptografía moderna es necesario representarlas de tal forma que se puedan implementar fácilmente en un software para computadora. Un concepto importante para ello es el de **fracción continua**, con el cual, a una sucesión finita de números $[a_1, a_2, \ldots, a_n]$, se le puede asignar la siguiente fracción:

$$a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots + \frac{1}{a_n}}}$$

Pudiéndose simplificar para obtener una fracción de la forma p entre q, donde p y q sean números enteros, primos relativos y con q distinto de cero. Denotaremos al numerador de la fracción continua simplificada como $N[a_1, a_2, \ldots, a_n]$ y al denominador como $D[a_1, a_2, \ldots, a_n]$. Obteniendo así las siguientes equivalencias:

$$[a_1, a_2, \ldots, a_n] = a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots + \frac{1}{a_n}}} = \frac{p}{q} = \frac{N[a_1, a_2 ..., a_n]}{D[a_1, a_2, ..., a_n]}$$

Ejemplo 23. A la sucesión $[2,1,2]$ le asociaremos su fracción continua.

$$[2,1,2] = 2 + \cfrac{1}{1 + \frac{1}{2}} = \frac{8}{3}$$

Luego, se tiene que el numerador de la fracción continua $[2,1,2]$ es 8, mientras que, su denominador es 3. Así, $N[2,1,2] = 8$ y $D[2,1,2] = 3$.

El siguiente lema, el cual es una versión modificada del Lema 2.7 en [12], nos permite asociar a cada 3-trenza una matriz, utilizando el concepto de fracción continua.

Lema 2.2.1. *[12] A la 3-trenza $\mathcal{T}(a_1,...,a_n)$ se le puede asignar la matriz $\mathcal{M}(\mathcal{T}(a_1,...,a_n))$, la cual será invariante bajo las movidas de Reidemeister Ω_2 y Ω_3, de la siguiente forma:*

- *Si n es impar:*

$$\mathcal{M}(\mathcal{T}(a_1,...,a_n)) = \begin{pmatrix} D[a_1,...,a_n] & \frac{-1}{i}D[a_1,...,a_{n-1}] \\ \frac{1}{i}N[a_1,...,a_n] & N[a_1,...,a_{n-1}] \end{pmatrix}$$

- *Si n es par:*

$$\mathcal{M}(\mathcal{T}(a_1,...,a_n)) = \begin{pmatrix} D[a_1,...,a_{n-1}] & \frac{-1}{i}D[a_1,...,a_n] \\ \frac{1}{i}N[a_1,...,a_{n-1}] & N[a_1,...,a_n] \end{pmatrix}$$

Además, el Lema 2.2.2, el cual es una versión modificada del Lema 2.3 en [12] establece una propiedad importante que satisface la matriz $\mathcal{M}(\mathcal{T}(a_1,...,a_n))$.

Lema 2.2.2. *[12] Dados dos diagramas de 3-trenzas $\mathcal{T}_1$ y $\mathcal{T}_2$ se tiene que:*

$$\mathcal{M}(\mathcal{T}_1 + \mathcal{T}_2) = \mathcal{M}(\mathcal{T}_1)\mathcal{M}(\mathcal{T}_2)$$

Una vez determinada la matriz asociada a una 3-trenza, la modificaremos con el fin de obtener una matriz donde cada una de sus entradas sean números reales, para ello multiplicaremos la entrada $\mathcal{M}_{12}$ por $-i$ y la entrada $\mathcal{M}_{21}$ por i. A la matriz resultante la denotaremos como $\mathcal{M}'$, la cual, también será invariante bajo las movidas de Reidemeister Ω_2 y Ω_3. Así, dada la 3-trenza $\mathcal{T}(a_1,...,a_n)$, se le puede asignar la matriz $\mathcal{M}'(\mathcal{T}(a_1,...,a_n))$ de la siguiente forma:

- Si n es impar:

$$\mathcal{M}'(\mathcal{T}(a_1,...,a_n)) = \begin{pmatrix} D[a_1,...,a_n] & D[a_1,...,a_{n-1}] \\ N[a_1,...,a_n] & N[a_1,...,a_{n-1}] \end{pmatrix}$$

- Si n es par:

$$\mathcal{M}'(\mathcal{T}(a_1,...,a_n)) = \begin{pmatrix} D[a_1,...,a_{n-1}] & D[a_1,...,a_n] \\ N[a_1,...,a_{n-1}] & N[a_1,...,a_n] \end{pmatrix}$$

Ejemplo 24. Asociaremos su matriz $\mathcal{M}'$ a la 3-trenza $\mathcal{T}(-1,-3,1)$.

$$\mathcal{M}'(\mathcal{T}(-1,-3,1))= \begin{pmatrix} D[-1,-3,1] & D[-1,-3] \\ N[-1,-3,1] & N[-1,-3] \end{pmatrix} = \begin{pmatrix} -2 & -3 \\ 3 & 4 \end{pmatrix}$$

Donde $[-1,-3,1] = -1 + \dfrac{1}{-3+\frac{1}{1}} = \dfrac{3}{-2}$ y $[-1,-3] = -1 + \dfrac{1}{-3} = \dfrac{4}{-3}$

Dada la importancia que tiene la llave para poder encriptar o desencriptar un mensaje, es necesario buscar un mecanismo con el cual podamos compartir de forma segura dicha llave y con ello salvaguardar nuestra información. Para ello, en el siguiente capítulo implementaremos un protocolo basado en el grupo no conmutativo de las 3-trenzas, con el cual podamos compartir de forma segura dicha llave a pesar de utilizar un canal público.

Capítulo 3

Protocolo algebraico para el establecimiento de una llave pública

Los algoritmos que se utilizan en la criptografía necesitan de una llave con la cual se proteja la confidencialidad de la información. Por ello, es importante el cómo se genera la llave, cómo se distribuye y así como también el establecer un tiempo para el cual dicha llave será válida. Así, se necesita de un método que permita al emisor y al receptor establecer una llave simétrica para encriptar y desencriptar su información, pero sin haberla establecido previamente, ha dicho método se le denomina: *algoritmo, o protocolo, de llave pública*.

Anshel y colaboradores proponen en [1] un protocolo algebraico para el establecimiento de una llave pública, el cual consiste en un quinteto ($\mathbf{U}$, $\mathbf{V}$, β, γ_1, γ_2) donde $\mathbf{U}$ y $\mathbf{V}$ son monoides[1], y tres funciones:

$$\beta : \mathbf{U} \text{ x } \mathbf{U} \longrightarrow \mathbf{V} \quad , \quad \gamma_i : \mathbf{U} \text{ x } \mathbf{V} \longrightarrow \mathbf{V} \quad (i = 1,2)$$

Donde dichas funciones satisfacen:

(i) Para todo elemento $x, y_1, y_2 \in \mathbf{U}$: $\beta(x,(y_1 \cdot y_2)) = \beta(x,y_1) \cdot \beta(x,y_2)$

(ii) Para todo elemento $x, y \in \mathbf{U}$: $\gamma_1(x,\beta(y,x)) = \gamma_2(y,\beta(x,y))$

(iii) Suponga que $y_1, y_2, \cdots, y_k \in \mathbf{U}$ y $\beta(x,y_1), \beta(x,y_2), \cdots, \beta(x,y_k)$ son públicamente conocidos para algún elemento secreto $x \in \mathbf{U}$. Entonces, en general, no es posible calcular fácilmente el elemento x.

Sea el emisor, el usuario A, y el receptor, el usuario B. Entonces, se les asignan públicamente los submonoides $S_A, T_B \subseteq \mathbf{U}$, respectivamente. Supongamos que S_A y T_B son generados, respectivamente, por los elementos:

$$\{s_1, \cdots, s_m\} \qquad \{t_1, \cdots, t_n\}$$

[1] Un **monoide** es un conjunto junto con una operación binaria que satisface: ser asociativa y poseer un elemento identidad.

Para iniciar el protocolo, el usuario A debe elegir un elemento secreto a en S_A y el usuario B debe elegir un elemento secreto b en T_B y respectivamente transmitirse los elementos:

$$\beta(a,t_i) \quad con\ i = 1,\cdots,n \quad , \quad \beta(b,s_i) \quad con\ i = 1,\cdots,m.$$

Por la propiedad (iii), se tiene que a pesar de que la transmisión sea mediante un canal público, los elementos secretos a y b estarán seguros. Mientras que por la propiedad (i), los usuarios A y B pueden calcular respectivamente los siguientes elementos:

$$\beta(b,a)\ \ y\ \ \gamma_1(a,\beta(b,a)) \quad , \quad \beta(a,b)\ \ y\ \ \gamma_2(b,\beta(a,b)).$$

Utilizando la propiedad (ii), se puede establecer una llave en común k dado que:

$$k = \gamma_1(a,\beta(b,a)) = \gamma_2(b,\beta(a,b)).$$

Además, muestran un ejemplo donde establecen un protocolo basado en la teoría de grupos, con **U = V** como un grupo no conmutativo, el cual denotan por G, y a los usuarios A y B se les asignan públicamente los subgrupos:

$$S_A = \langle s_1, s_2, \cdots, s_m \rangle \quad , \quad T_B = \langle t_1, t_2, \cdots, t_n \rangle.$$

Donde S_A es el subgrupo generado por los elementos $s_1,\cdots,s_m$, mientras que, T_B es el subgrupo generado por los elementos $t_1,\cdots,t_n$.

Se elige a β : G x G $\longrightarrow$ G como la conjugación:

$$\beta(x,y) = x^{-1}yx. \qquad (F_1)$$

Y las funciones γ_1 y γ_2 como:

$$\gamma_1(u,v) = u^{-1}v \quad , \quad \gamma_2(u,v) = v^{-1}u. \qquad (F_2)$$

Después, los usuarios A y B, escogen su elemento secreto $a \in S_A$ y $b \in T_B$, respectivamente, luego para iniciar el protocolo, los usuarios A y B deben calcular, reescribir y transmitir, respectivamente, su colección de elementos:

$$a^{-1}t_1a, a^{-1}t_2a, \cdots, a^{-1}t_na \quad , \quad b^{-1}s_1b, b^{-1}s_2b, \cdots, b^{-1}s_mb.$$

Aunque un adversario pueda ver la transmisión pública de los elementos, no podrá determinar a o b a menos que él pueda resolver un sistema de ecuaciones conjugadas basadas en el grupo G. Recordar que el conjugado del producto de dos elementos es el producto de la conjugación de cada uno de ellos, propiedad (i).

Luego, los usuarios A y B pueden ahora calcular mediante la propiedad (i), respectivamente, el elemento:

$$\beta(b,a) = b^{-1}ab \quad , \quad \beta(a,b) = a^{-1}ba.$$

En el paso (V) del Ejemplo 32 se mostrará en concreto como se realizan dichos cálculos.

Por último, para obtener una llave en común k. Los usuarios A y B deben calcular, respectivamente, los elementos:

$$k_1 = \gamma_1(a,\beta(b,a)) \quad , \quad k_2 = \gamma_2(b,\beta(a,b)).$$

Así, $k = k_1 = k_2$, se obtiene dicha llave en común dado que:

$\gamma_1(a,\beta(b,a)) = a^{-1}\beta(b,a) = a^{-1}b^{-1}ab.$
$\gamma_2(b,\beta(a,b)) = \beta(a,b)^{-1}b = (a^{-1}ba)^{-1}b = a^{-1}b^{-1}ab.$

En la siguiente sección, usaremos esta información para generar una cadena de 0's y 1's y con ella establecer un proceso para encriptar o desencriptar nuestra información.

3.1. El mapeo logístico en criptografía

La teoría del caos ha venido a ser una buena alternativa en los procesos de cifrado de información, por ejemplo, los cifrados en flujo, y con ello ha intentado resolver uno de los problemas a los cuales se enfrenta la criptografía: La confidencialidad. Dicha teoría, analiza el comportamiento de sistemas dinámicos no lineales, especialmente se enfoca en las ecuaciones diferenciales ó en ecuaciones en diferencias. Es de nuestro interés analizar uno de los sistemas iterados más sencillo que produce caos: El mapeo logístico.

El **mapeo logístico** trata de describir el crecimiento de una población y se popularizo en 1976 en el libro publicado por el biólogo Robert May [13], en el cual exhibe que este mapeo tiene un amplio rango de comportamientos a medida que la tasa de crecimiento varía. Dicho mapeo se describe mediante la siguiente ecuación en diferencias:

$$X_{n+1} = \mu X_n(1 - X_n). \tag{3.1}$$

Donde μ representa la tasa de crecimiento, generalmente con $\mu \in [0,4]$, dado que en este intervalo el mapeo logístico permanece dentro del intervalo cerrado $[0,1]$ y X_n representa el número de individuos en la n-ésima generación de dicha población. Para iterar dicho mapeo necesitamos una condición inicial $X_0 \in [0,1]$, la cual evaluaremos en la ecuación (3.1). Así, tendremos:

$$X_0, \quad X_1 = \mu X_0(1 - X_0), \quad X_2 = \mu X_1(1 - X_1), \quad \cdots \quad , \quad X_n = \mu X_{n-1}(1 - X_{n-1}).$$

Ahora, dada una llave en común k, nos interesa utilizarla en un algoritmo criptográfico que pueda encriptar y desencriptar un texto utilizando el comportamiento caótico del mapeo logístico, puede indagar sobre el comportamiento caótico del mapeo logístico en [14].

Ahora bien, tomando en cuenta que una computadora trabaja con el uso de la lógica binaria, es decir, con los bits 0 y 1. Es conveniente traducir el texto que deseamos encriptar, al cual denominaremos **texto plano**, a una cadena en este lenguaje, que en lo consecuente nos referiremos a ella como una **cadena binaria**. Así, cada letra del texto a encriptar será asociada a una cadena de 8 bits con la norma ISO 8859-1, obteniendo con ello una cadena binaria asociada al texto plano.

La **norma ISO 8859-1** establece la codificación del alfabeto latino, el cual incluye los acentos diacríticos y algunos símbolos especiales necesarios para la escritura en el idioma español. Esta codificación se caracteriza por poseer en sus primeros lugares, la codificación ASCII (128 caracteres, utilizando 7 bits para representar cada carácter) e incluye otros 128 caracteres para la codificación del alfabeto latino, utilizando así 8 bits para representar cada carácter.

En la Figura 3.1 podemos observar una tabla que muestra la codificación ASCII. Mientras que en la Figura 3.2 podemos observar una tabla que muestra la cadena binaria en 8 bits asociada a cada carácter del alfabeto latino. Los símbolos para este alfabeto están en la columna 1.

Binario	Representación	Binario	Representación	Binario	Representación
0010 0000	espacio ( )	0100 0000	@	0110 0000	`
0010 0001	!	0100 0001	A	0110 0001	a
0010 0010	"	0100 0010	B	0110 0010	b
0010 0011	#	0100 0011	C	0110 0011	c
0010 0100	$	0100 0100	D	0110 0100	d
0010 0101	%	0100 0101	E	0110 0101	e
0010 0110	&	0100 0110	F	0110 0110	f
0010 0111	'	0100 0111	G	0110 0111	g
0010 1000	(	0100 1000	H	0110 1000	h
0010 1001	)	0100 1001	I	0110 1001	i
0010 1010	*	0100 1010	J	0110 1010	j
0010 1011	+	0100 1011	K	0110 1011	k
0010 1100	,	0100 1100	L	0110 1100	l
0010 1101	-	0100 1101	M	0110 1101	m
0010 1110	.	0100 1110	N	0110 1110	n
0010 1111	/	0100 1111	O	0110 1111	o
0011 0000	0	0101 0000	P	0111 0000	p
0011 0001	1	0101 0001	Q	0111 0001	q
0011 0010	2	0101 0010	R	0111 0010	r
0011 0011	3	0101 0011	S	0111 0011	s
0011 0100	4	0101 0100	T	0111 0100	t
0011 0101	5	0101 0101	U	0111 0101	u
0011 0110	6	0101 0110	V	0111 0110	v
0011 0111	7	0101 0111	W	0111 0111	w
0011 1000	8	0101 1000	X	0111 1000	x
0011 1001	9	0101 1001	Y	0111 1001	y
0011 1010	:	0101 1010	Z	0111 1010	z
0011 1011	;	0101 1011	[	0111 1011	{
0011 1100	<	0101 1100	\	0111 1100	|
0011 1101	=	0101 1101	]	0111 1101	}
0011 1110	>	0101 1110	^	0111 1110	~
0011 1111	?	0101 1111	_		

Figura 3.1: *Codificación ASCII.*

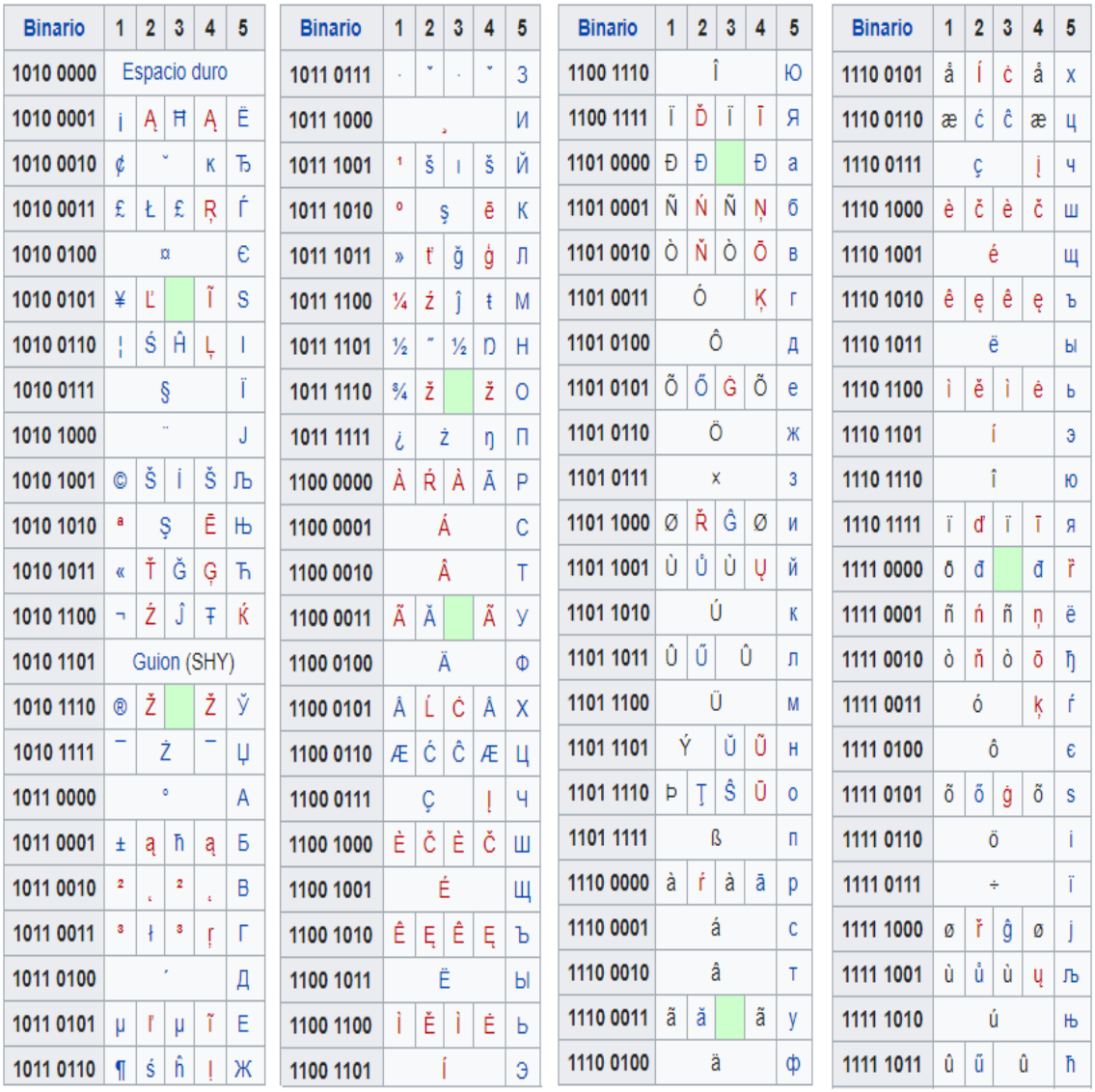


Figura 3.2: *Codificación por la Norma ISO 8859.*

Ejemplo 25. Usaremos el texto plano *Código* y lo traduciremos a una cadena binaria utilizando la norma ISO 8859-1. Para ello, usaremos las tablas de las Figuras 3.1 y 3.2.

Texto plano:	*C*	*ó*	*d*	*i*	*g*	*o*
Cadena binaria:	01000011	11110011	01100100	01101001	01100111	01101111

Note que la longitud del texto plano es de 6 caracteres (letras o símbolos) mientras que su cadena asociada es de 48 bits. Si queremos recuperar el texto plano dividiremos la cadena asociada en cadenas de 8 bits y les asociaremos su carácter correspondiente siguiendo la norma ISO 8859-1.

NOTA: si a una cadena de 8 bits no se le puede asociar un carácter con la norma ISO 8859-1, entonces se le asigna el símbolo □.

Usando la suma módulo dos podremos sumar dos cadenas binarias. A saber:

$$\begin{aligned} 0+0 &= 0 \\ 0+1 &= 1 \\ 1+0 &= 1 \\ 1+1 &= 0 \end{aligned}$$

Ejemplo 26. Sumaremos las siguientes dos cadenas binarias: 01010101 y 01101110

$$\begin{array}{rcccccccc} & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ + & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ \hline & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{array}$$

Para poder utilizar el mapeo logístico como recurso para la criptografía, se requiere que con este mapeo podamos generar con una condición inicial una cadena binaria de la misma longitud que la cadena asociada al texto plano con el fin de poder sumar ambas cadenas.

Para poder utilizar el mapeo logístico como generador de una cadena binaria, utilizaremos el siguiente procedimiento, al cual denotaremos ML_1 :

(1) Se establecerá el texto plano que se desea encriptar y su cadena binaria asociada.

(2) Se establecerá una condición inicial X_0.

(3) Se fijará el valor de μ en 4 y se iterará n-veces el mapeo logístico. Donde el valor de n dependerá de la longitud de la cadena binaria asociada al texto plano. Generando así, las iteraciones: $X_1, X_2, \cdots, X_n$.

(4) Se generará una cadena binaria, asignando un 1 si la iteración X_i es mayor o igual a 0.5 y en caso contrario asignando un 0.

(5) Se sumará la cadena binaria obtenida en el paso (4) con la cadena binaria asociada al texto plano del paso (1), para con ello obtener una cadena binaria encriptada.

(6) Si se desea desencriptar la cadena binaria encriptada, se deberán repetir los pasos (2), (3) y (4) y sumar la cadena obtenida con la cadena binaria encriptada del paso (5).

(7) Se traducirá la cadena binaria obtenida en el paso (6) a texto, es decir, se recuperará el texto plano. Para ello, dicha cadena será separada en cadenas de 8 bits y usando la norma ISO 8859-1 se recuperará el texto.

Ejemplo 27. Veremos en concreto como se lleva acabo el procedimiento ML_1.

Se establece el texto plano y su cadena binaria asociada C_0.

Texto plano: Código

C_0: 010000111111001101100100011010010110011101101111

Se establece la condición inicial $X_0 = 0.123456789$

Se itera 48 veces el mapeo logístico, dado que esta es la longitud de la cadena binaria asociada al texto plano: Código.

Siguiendo el paso (4), se genera la cadena binaria C_1.

A la iteración $X_1 = 0.432860841$ se le asigna un 0.
A la iteración $X_2 = 0.981969333$ se le asigna un 1.
$\vdots$
A la iteración $X_{48} = 0.855763385$ se le asigna un 1.

C_1: 010011111000010101110011011111110001011110010111

Sumamos C_1 con C_0 para obtener la cadena binaria encriptada C_2 .

$$\begin{array}{rl} & 010011111000010101110011011111110001011110010111 \\ + & 010000111111001101100100011010010110011101101111 \\ \hline & 000011000111011000010111000101100111000011111000 \end{array}$$

C_2: 000011000111011000010111000101100111000011111000

Siguiendo el paso (6), se genera la cadena binaria C_3 la cual sumamos con C_2 para obtener la cadena binaria original C_0.

$$\begin{array}{rl} & 010011111000010101110011011111110001011110010111 \\ + & 000011000111011000010111000101100111000011111000 \\ \hline & 010000111111001101100100011010010110011101101111 \end{array}$$

C_0: 010000111111001101100100011010010110011101101111

Siguiendo el paso (7), se recupera el texto plano: Código

Así, utilizando el procedimiento ML_1 podemos encriptar y desencriptar nuestra información.

Ahora, nos interesa usar este procedimiento para que dos usuarios puedan establecer entre ellos una comunicación segura, es decir, que puedan acceder al texto plano sólo si conocen la condición inicial X_0. Pero ¿Tendremos la certeza de que sólo el emisor y el receptor podrán acceder a esta información? Para tratar de resolver esta incógnita intentaremos una de las técnicas más usadas para romper un algoritmo de cifrado: El ataque por fuerza bruta.

Un **ataque de fuerza bruta** es aquel procedimiento que a partir del conocimiento del algoritmo con el que se encripto un texto plano se busca recuperarlo probando con diversas combinaciones posibles.

Imaginemos ahora que dos usuarios se están comunicando por medio de Internet a través de mensajes cifrados con el procedimiento expuesto anteriormente. Y dado que Internet es un canal público, en algún momento de esa conversación interceptamos una de las cadenas binarias trasmitidas. Conociendo esta cadena es de nuestro interés recuperar el texto plano.

Ejemplo 28. Interceptamos la siguiente cadena binaria:

C_b: 101000110101111110000000001110111001101010011111110011000

Luego, dado que conocemos que dicha cadena fue encriptada con el procedimiento ML_1, intentaremos generar diferentes cadenas binarias con dicho procedimiento y sumarlas a la cadena binaria interceptada, esperando así recuperar un texto plano legible.

Ejemplo 29. Probaremos con tres cadenas binarias diferentes, provenientes de tres condiciones iniciales diferentes:

- Cd_1: 110110011101101001111011111010100001101000110001110110000

Sumaremos C_b con Cd_1:

$$\begin{array}{rl} & 101000110101111110000000001110111001101010011111110011000 \\ + & 110110011101101001111011111010100001101000110001110110000 \\ \hline & 011110101000010111111011110100011000000010101110010000000 \end{array}$$

Con la cual recuperamos el siguiente texto plano: z □ *û* Ñ □ ®@

- Cd_2: 010111000101110001010000111011010000000011000100100000011

Sumaremos C_b con Cd_2:

$$\begin{array}{rl} & 101000110101111110000000001110111001101010011111110011000 \\ + & 010111000101110001010000111011010000000011000100100000011 \\ \hline & 111111110000001111010000110101101001101001011011000110 11 \end{array}$$

Con la cual recuperamos el siguiente texto plano: *ÿ* □ Ð Ö □ ⊏ □

- Cd_3: 110000001010110011100100010100101111110111110000111010 11

Sumaremos C_b con Cd_3:

$$\begin{array}{rl} & 101000110101111110000000001110111001101010011111110011000 \\ + & 110000001010110011100100010100101111110111110000111010 11 \\ \hline & 011000111111001101100100011010010110011101101111011100 11 \end{array}$$

Con la cual recuperamos el siguiente texto plano: códigos

El número de cadenas binarias con las que podemos intentar este procedimiento es realmente grande y no con todas ellas lograremos nuestro objetivo. A saber, tenemos 2^{8L} posibles cadenas binarias, siendo L la longitud del texto, de tal forma que hacer esta búsqueda a mano es una tarea titánica. Por ello, se implementó un algoritmo en un software para computadora con el fin de llevar a cabo esta búsqueda exhaustiva, teniendo como única información la

cadena binaria cifrada interceptada y cuyo objetivo es recuperar fácilmente el texto plano. Pero, dicha implementación mostró que:

(a) Si se generan diversas cadenas binarias y se suman a la cadena binaria cifrada interceptada, al trascurrir el tiempo se encuentran todas las letras del abecedario, así como diversos números y símbolos. Así que dicha búsqueda era muy lenta y al final no conducía a nada, ya que se podía encontrar cualquier texto plano. Por ello, se optó por orientar dicha búsqueda.

(b) Si se intercepta una cadena binaria encriptada, de la cual se intuye una palabra de siete letras que es *altamente probable* que aparezca en el texto plano, se puede modificar el algoritmo anterior para que realice una búsqueda exhaustiva pero ahora a través del intervalo cerrado [0 , 0.5] con el fin de recuperar el texto plano. La búsqueda se lleva a cabo sólo en este intervalo aprovechando la simetría que presenta el mapeo logístico, el algoritmo genera mil condiciones iniciales y con ellas genera mil cadenas binarias. Luego, suma cada cadena binaria a la cadena binaria encriptada interceptada y busca la similitud entre estas cadenas y la palabra clave. Por último, el algoritmo empieza su búsqueda en la similitud de los primeros 3 bits, al encontrarla dicho algoritmo reajusta el intervalo de búsqueda para aumentar la similitud entre bits, repitiendo este proceso hasta encontrar la similitud en todos los bits que conforman la cadena binaria asociada a la palabra probable. Al final, se logra visualizar el texto plano.

Cabe señalar, que se podrá visualizar sólo el texto plano a partir de la palabra clave encontrada. Pero, con el texto plano obtenido se puede intuir una nueva palabra clave e iniciar nuevamente la búsqueda, con el fin de visualizar más partes del texto plano, realizando iterativamente este proceso se puede visualizar el texto plano completo.

Buscando robustecer el procedimiento ML_1 ante el anterior ataque de fuerza bruta. Se estableció otro procedimiento, similar al procedimiento ML_1, pero ahora aumentando 10 veces más la longitud de la cadena binaria generada con los pasos del (1) al (4) del procedimiento ML_1 y de dicha cadena se utilizaron sólo los bits ubicados en una posición múltiplo de diez, generando así una nueva cadena binaria con la misma longitud que la de la cadena binaria asociada al texto plano. Luego, como en el procedimiento anterior, se realizó otra búsqueda exhaustiva pero ahora adaptada a este otro procedimiento. Al final, no se logra visualizar el texto plano.

Así, implementaremos un nuevo procedimiento para encriptar y desencriptar un texto plano basándonos en el procedimiento anterior, del cual sólo aumentaremos la longitud de la cadena binaria y de la cual utilizaremos sólo los bits ubicados en una posición múltiplo de diez, al cual denominaremos ML_2.

Procedimiento ML_2.

El emisor:

(1) Establece el texto plano que se desea transmitir y su cadena binaria asociada.

(2) Establece una condición inicial X_0, la cual será la llave en común que permitirá sólo al receptor y al emisor acceder a dicha información.

(3) Fijando el valor de μ en 4, itera n-veces el mapeo logístico. Donde el valor de n deberá ser igual a diez veces la longitud de la cadena binaria asociada al texto plano. Generando así, las iteraciones: $X_1, X_2, \cdots, X_n$.

(4) Se generará una cadena binaria, asignando un 1 si la iteración X_i es mayor o igual a 0.5 y en caso contrario asignando un 0.

(5) De la cadena binaria obtenida en el paso (4) se genera una cadena con la misma longitud que la asociada al texto plano. Tomando de dicha cadena binaria todos los bits ubicados en las posiciones que sean múltiplos de diez.

(6) Suma la cadena binaria obtenida en el paso (5) con la cadena binaria asociada al texto plano obtenida del paso (1), para con ello obtener la cadena binaria encriptada.

(7) Envía la cadena binaria encriptada al receptor.

El receptor:

(8) Recibe la cadena binaria encriptada del emisor.

(9) Deberá conocer la condición inicial X_0 y con ella repetir los pasos del (3) al (5) del emisor, para con ello generar una cadena binaria con la misma longitud que la asociada al texto plano.

(10) Sumará la cadena binaria del paso (9) con la cadena binaria encriptada recibida del paso (8), para con ello generar la cadena binaria desencriptada.

(11) Traducirá la cadena binaria obtenida en el paso (10) a texto, es decir, recuperará el texto plano. Para ello, dicha cadena binaria la separará en segmentos de 8 bits y usará la norma ISO 8859-1 para recuperar el texto plano.

Ejemplo 30. Veremos en concreto como se lleva acabo el procedimiento ML_2.

El emisor:

Establece el texto plano y su cadena binaria asociada C_0.

Texto plano: Código

C_0: 010000111111001101100100011010010110011101101111

Establece la condición inicial $X_0 = 0.123456789$

Siguiendo el paso (3) itera 480 veces el mapeo logístico, dado que está es diez veces la longitud de la cadena asociada al texto plano: Código.

Siguiendo el paso (4) genera la cadena binaria C_1, cuya longitud será de 480 bits.

A la iteración $X_1 = 0.432860840999238$ se le asigna un 0.
A la iteración $X_2 = 0.981969333314682$ se le asigna un 1.
$\vdots$

A la iteración $X_{479} = 0.024888998508306$ se le asigna un 0.
A la iteración $X_{480} = 0.097078145046240$ se le asigna un 0.

C_1: 0100111110000101011100110111111$\cdots$0110001010001100

Siguiendo el paso (5) utiliza C_1 para generar la nueva cadena binaria C_2, cuya longitud será de 48 bits. Se resaltaron los bits ubicados en las posiciones múltiplos de diez.

C_1: 010011111**0**0010101111**0**0111011111**1**$\cdots$011000**1**010001100**0**

C_2: 011110011000010100010110000010011000101000101100

Notar que C_2 tiene la misma longitud que la cadena binaria asociada al texto plano C_0.

Sumando C_2 con C_1 obtiene la cadena binaria encriptada C_3.

$$\begin{array}{rl} & 011110011000010100010110000010011000101000101100 \\ + & 010000111111001101100100011010010110011101101111 \\ \hline & 001110100111011001110010011000001110110101000011 \end{array}$$

C_3: 001110100111011001110010011000001110110101000011

Envía la cadena binaria encriptada C_3 al receptor.

El receptor:

Recibe la cadena binaria encriptada C_3 del emisor.

Siguiendo el paso (9) genera la cadena binaria C_4.

C_4: 011110011000010100010110000010011000101000101100

Sumando C_4 con C_3 obtiene la cadena binaria desencriptada, es decir, la cadena original C_0.

$$\begin{array}{rl} & 011110011000010100010110000010011000101000101100 \\ + & 001110100111011001110010011000001110110101000011 \\ \hline & 010000111111001101100100011010010110011101101111 \end{array}$$

C_0: 010000111111001101100100011010010110011101101111

Siguiendo el paso (11), se recupera el texto plano: Código

Cabe resaltar que se implementó un ataque mediante fuerza bruta en contra del procedimiento ML_2, el cual, consistió en adecuar el ataque de fuerza bruta realizado al procedimiento ML_1, con el fin de encontrar el texto plano teniendo sólo la cadena encriptada. Dicho ataque resultó fallido dado que no se logró obtener el texto plano, por ello, podemos decir que el procedimiento ML_2 es más seguro que el procedimiento ML_1, al menos ante el mismo ataque mediante fuerza bruta.

Así, con el grupo no conmutativo de las 3-trenzas, el protocolo anterior y el procedimiento ML_2 implementaremos en la siguiente sección un protocolo cuya finalidad sea el establecer una llave pública.

3.2. Protocolo basado en trenzas

Con el fin de implementar un protocolo algebraico para el establecimiento de una llave pública, que involucre al mapeo logístico y al grupo no conmutativo de las 3-trenzas, es necesario tener presentes los conceptos y definiciones establecidas en el Capítulo 2. Con estas definiciones en mente, estableceremos el siguiente protocolo, al cual denominaremos TC. Además, dicho protocolo se implementará en una interfaz gráfica en el Capítulo 4.

Usaremos las funciones β, γ_1 y γ_2 definidas en (F_1) y (F_2).

(I) Definiremos U = V como el grupo no conmutativo de la 3-trenzas, $\mathcal{G}$. Para poder iniciar el protocolo estableceremos $\mathcal{S}_i$ 3-trenzas, con $i = 1, \cdots, m$, para el usuario A y $\mathcal{T}_j$ 3-trenzas, con $j = 1, \cdots, n$, para el usuario B. A las cuales les asociaremos su matriz $\mathcal{M}'$ con el procedimiento establecido en la Sección 2.2. Para con ello establecer, respectivamente, los subgrupos públicos generados S_A y T_B.

$$S_A = \langle \mathcal{M}'(\mathcal{S}_1), \cdots, \mathcal{M}'(\mathcal{S}_m) \rangle \ , \ \ T_B = \langle \mathcal{M}'(\mathcal{T}_1), \cdots, \mathcal{M}'(\mathcal{T}_n) \rangle .$$

(II) Ambos usuarios deberán generar su elemento secreto, *llave privada*. Para ello, cada usuario seleccionará algunos números enteros dentro de un cierto intervalo, $[1, m]$ para el usuario A y $[1, n]$ para el usuario B, cada número estará previamente asociado sólo a un generador del subgrupo S_A y T_B respectivamente. Después de seleccionar dichos números, se multiplicarán las matrices asociadas a cada uno de ellos, para con ello generar su elemento secreto. Obteniendo así, el elemento secreto $a \in S_A$ para el usuario A y el elemento secreto $b \in T_B$ para el usuario B.

(III) Dada la importancia que tiene la llave privada, para encriptar y desencriptar la información entre los dos usuarios, es importante el establecer conjuntamente un tiempo T, en minutos, mediante el cual dicho protocolo reconocerá esa llave como válida. Después de que el tiempo expire, el protocolo deberá reiniciarse.

(IV) Ambos usuarios deben calcular y transmitir públicamente sólo los resultados de operar su llave privada con los elementos generadores del subgrupo T_B para el usuario A y S_A para el usuario B. A saber:

$$a^{-1}t_1a, \cdots, a^{-1}t_na \quad y \quad b^{-1}s_1b, \cdots, b^{-1}s_mb.$$

(V) Ambos usuarios calculan usando la propiedad (i), respectivamente, el elemento:

$$\beta(b, a) \ , \ \ \beta(a, b).$$

(VI) Ambos usuarios deben obtener la llave en común k calculando, respectivamente, el elemento:

$$k_1 = \gamma_1(a, \beta(b,a)) \ , \ \ k_2 = \gamma_2(b, \beta(a,b)).$$

Obteniendo así, $k = k_1 = k_2$.

Siguiendo el trabajo de Anshel y colaboradores realizado en [1] hemos establecido una llave en común k. Así, es necesario establecer también un procedimiento para poder generar una cadena binaria con la finalidad de encriptar y desencriptar una cadena binaria asociada a un texto plano. Para ello, utilizaremos el procedimiento ML_2 descrito en la Sección 3.1.

(VII) Ambos usuarios tendrán la llave en común k, será una matriz de 2x2 con elementos en $\mathbb{Z}$. De la cual, se deben operar sus entradas para generar una condición inicial $X_0 \in [0,1]$, la cual será usada en el procedimiento ML_2 para generar una cadena binaria, con el siguiente procedimiento:

(a) Se calcula el valor absoluto $|(k_{11}+k_{22})(k_{12}+k_{21})|$.

(b) Del número obtenido en el paso (a) se seleccionan sólo los primeros 15 dígitos. En caso de que dicho número sea menor, se deberá repetir dicho número con el fin de extenderlo a un número de 15 dígitos.

Ejemplo 31. Si $|(k_{11}+k_{22})(k_{12}+k_{21})| = 11235$, debemos repetir tres veces dicho número para formar con ello un número de 15 dígitos. Así, dicho número será 112351123511235.

(c) El número obtenido en el paso (b) se multiplica por 10^{-15} para obtener así la condición inicial X_0.

(VIII) El usuario A establecerá y enviará su cadena encriptada binaria siguiendo los pasos del (1) al (6) del procedimiento ML_2, establecido en la Sección 3.1.

(IX) El usuario B recibirá la cadena encriptada del paso (VIII) y siguiendo los pasos del (9) al (11) del procedimiento ML_2, establecido en la Sección 3.1, recuperará el texto plano enviado por el usuario A.

NOTA$_1$: Si el usuario B desea responder al usuario A, deberá repetir sólo el paso (VIII) y consecuentemente, el usuario A deberá repetir sólo el paso (IX) para recuperar el texto plano.

NOTA$_2$: Si se excede el tiempo establecido para la validez de la llave, dicho protocolo se reiniciará. Implicando que los usuarios inicien de nuevo el protocolo TC.

Ejemplo 32. Veremos concretamente como se desarrolla el protocolo TC, para ello usaremos las funciones β, γ_1 y γ_2 anteriormente definidas en (F_1) y (F_2).

Para iniciar el protocolo debemos seguir el paso (I) y así establecer cinco 3-trenzas para cada usuario, cabe señalar, que no necesariamente se debe seleccionar el mismo número de 3-trenzas para cada usuario. Además, a cada 3-trenza se le asocia su matriz $\mathcal{M}'$ con el procedimiento establecido en la Sección 2.2.

Al usuario A se le asignan las siguientes cinco 3-trenzas:

$\mathcal{S}_1 = \mathcal{T}(3,1,-3)$, $\mathcal{S}_2 = \mathcal{T}(-3,1,2)$, $\mathcal{S}_3 = \mathcal{T}(1,2,-2)$, $\mathcal{S}_4 = \mathcal{T}(2,-1,3)$ y $\mathcal{S}_5 = \mathcal{T}(3,1,2)$

A las cuales se les asocia su matriz $\mathcal{M}'$:

$$\mathcal{M}'(\mathcal{S}_1)=\begin{pmatrix} -2 & 1 \\ -9 & 4 \end{pmatrix} \qquad \mathcal{M}'(\mathcal{S}_2)=\begin{pmatrix} 3 & 1 \\ -7 & -2 \end{pmatrix} \qquad \mathcal{M}'(\mathcal{S}_3)=\begin{pmatrix} -3 & 2 \\ -5 & 3 \end{pmatrix}$$

$$\mathcal{M}'(\mathcal{S}_4)=\begin{pmatrix} -2 & -1 \\ -1 & -1 \end{pmatrix} \qquad \mathcal{M}'(\mathcal{S}_5)=\begin{pmatrix} 3 & 1 \\ 11 & 4 \end{pmatrix}$$

Estableciendo así el subgrupo generado S_A, el cual será público.

$$S_A = \langle \begin{pmatrix} -2 & 1 \\ -9 & 4 \end{pmatrix}, \begin{pmatrix} 3 & 1 \\ -7 & -2 \end{pmatrix}, \begin{pmatrix} -3 & 2 \\ -5 & 3 \end{pmatrix}, \begin{pmatrix} -2 & -1 \\ -1 & -1 \end{pmatrix}, \begin{pmatrix} 3 & 1 \\ 11 & 4 \end{pmatrix} \rangle$$

Al usuario B se le asignan las siguientes cinco 3-trenzas:

$$\mathcal{T}_1 = \mathcal{T}(1,-3,-2),\ \mathcal{T}_2 = \mathcal{T}(-2,1,-3),\ \mathcal{T}_3 = \mathcal{T}(4,2,-1),\ \mathcal{T}_4 = \mathcal{T}(-2,1,1) \text{ y}$$
$$\mathcal{T}_5 = \mathcal{T}(1,3,-1)$$

A las cuales se les asocia su matriz $\mathcal{M}'$:

$$\mathcal{M}'(\mathcal{T}_1)=\begin{pmatrix} 7 & -3 \\ 5 & -2 \end{pmatrix} \qquad \mathcal{M}'(\mathcal{T}_2)=\begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix} \qquad \mathcal{M}'(\mathcal{T}_3) =\begin{pmatrix} -1 & 2 \\ -5 & 9 \end{pmatrix}$$

$$\mathcal{M}'(\mathcal{T}_4) =\begin{pmatrix} 2 & 1 \\ -3 & -1 \end{pmatrix} \qquad \mathcal{M}'(\mathcal{T}_5)=\begin{pmatrix} -2 & 3 \\ -3 & 4 \end{pmatrix}$$

Estableciendo así el subgrupo generado T_B, el cual será público.

$$T_B = \langle \begin{pmatrix} 7 & -3 \\ 5 & -2 \end{pmatrix}, \begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} -1 & 2 \\ -5 & 9 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ -3 & -1 \end{pmatrix}, \begin{pmatrix} -2 & 3 \\ -3 & 4 \end{pmatrix} \rangle$$

Siguiendo el paso (II) cada usuario selecciona 5 números enteros, entre el 1 y el 5, para con ello generar su elemento secreto. A saber, el elemento secreto $a \in S_A$ para el usuario A y el elemento secreto $b \in T_B$ para el usuario B.

El usuario A selecciona los siguientes cinco números: 1, 2, 3, 4, y 5. Cabe señalar, que previo a la selección de los cinco números, al número 1 se le asoció la matriz $\mathcal{M}'(\mathcal{S}_1)$, al 2 la matriz $\mathcal{M}'(\mathcal{S}_2)$, $\cdots$, y al 5 la matriz $\mathcal{M}'(\mathcal{S}_5)$. Así, para calcular su llave secreta deberá multiplicar las matrices asociadas a sus cinco números seleccionados. Es decir, $a = \mathcal{M}'(\mathcal{S}_1)\mathcal{M}'(\mathcal{S}_2)\mathcal{M}'(\mathcal{S}_3)\mathcal{M}'(\mathcal{S}_4)\mathcal{M}'(\mathcal{S}_5)$.

$$a = \begin{pmatrix} -2 & 1 \\ -9 & 4 \end{pmatrix}\begin{pmatrix} 3 & 1 \\ -7 & -2 \end{pmatrix}\begin{pmatrix} -3 & 2 \\ -5 & 3 \end{pmatrix}\begin{pmatrix} -2 & -1 \\ -1 & -1 \end{pmatrix}\begin{pmatrix} 3 & 1 \\ 11 & 4 \end{pmatrix}$$

Por lo tanto, el usuario A genera su llave privada:

$$a = \begin{pmatrix} -471 & -164 \\ -1996 & -695 \end{pmatrix}$$

El usuario B selecciona los siguientes cinco números: 5, 4, 3, 2, y 1. Cabe señalar, que previo a la selección de los cinco números, al número 1 se le asocio la matriz $\mathcal{M}'(\mathcal{T}_1)$, al 2 la matriz $\mathcal{M}'(\mathcal{T}_2)$, $\cdots$, y al 5 la matriz $\mathcal{M}'(\mathcal{T}_5)$. Así, para calcular su llave secreta deberá multiplicar las matrices asociadas a sus cinco números seleccionados. Es decir, $b = \mathcal{M}'(\mathcal{T}_5)\mathcal{M}'(\mathcal{T}_4)\mathcal{M}'(\mathcal{T}_3)\mathcal{M}'(\mathcal{T}_2)\mathcal{M}'(\mathcal{T}_1)$.

$$b = \begin{pmatrix} -2 & 3 \\ -3 & 4 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ -3 & -1 \end{pmatrix}\begin{pmatrix} -1 & 2 \\ -5 & 9 \end{pmatrix}\begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 7 & -3 \\ 5 & -2 \end{pmatrix}$$

Por lo tanto, el usuario B genera su llave privada:

$$b = \begin{pmatrix} -484 & 223 \\ -675 & 311 \end{pmatrix}$$

A saber, no necesariamente se debe asociar un número a cada matriz de la misma forma para cada usuario, usamos sólo cinco números dado que sólo ocuparemos los cinco elementos generadores de cada subgrupo, además, los cinco números seleccionados no necesariamente deben estar en orden y no existe impedimento para poderlos repetir. En este ejemplo, por simplicidad usamos 1, 2, 3, 4, 5 para el usuario A y 5, 4, 3, 2, 1 para el usuario B.

Siguiendo el paso (III) ambos usuarios establecen T=20.

Siguiendo el paso (IV) ambos usuarios calculan y transmiten públicamente sólo los resultados de las siguientes multiplicaciones de elementos:

El usuario A, debe calcular $a^{-1}t_1a$, $a^{-1}t_2a$, $\cdots$, $a^{-1}t_5a$:

Donde $a = \begin{pmatrix} -471 & -164 \\ -1996 & -695 \end{pmatrix}$ y $a^{-1} = \begin{pmatrix} -695 & 164 \\ 1996 & -471 \end{pmatrix}$

$$a^{-1}\begin{pmatrix} 7 & -3 \\ 5 & -2 \end{pmatrix}a = \begin{pmatrix} -1601777 & -557735 \\ 4600209 & 1601782 \end{pmatrix} = A_1$$

$$a^{-1}\begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix}a = \begin{pmatrix} 982630 & 342149 \\ -2822059 & -982633 \end{pmatrix} = A_2$$

$$a^{-1}\begin{pmatrix} -1 & 2 \\ -5 & 9 \end{pmatrix}a = \begin{pmatrix} -112781 & -39270 \\ 323923 & 112789 \end{pmatrix} = A_3$$

$$a^{-1}\begin{pmatrix} 2 & 1 \\ -3 & -1 \end{pmatrix} a = \begin{pmatrix} 2600986 & 905653 \\ -7469887 & -2600985 \end{pmatrix} = A_4$$

$$a^{-1}\begin{pmatrix} -2 & 3 \\ -3 & 4 \end{pmatrix} a = \begin{pmatrix} 2429326 & 845883 \\ -6976875 & -2429324 \end{pmatrix} = A_5$$

Luego, debe trasmitir $\{A_1, A_2, A_3, A_4, A_5\}$ por un canal público al usuario B.

El usuario B, debe calcular $b^{-1}s_1b$, $b^{-1}s_2b$, $\cdots$, $b^{-1}s_5b$:

Donde $b = \begin{pmatrix} -484 & 223 \\ -675 & 311 \end{pmatrix}$ y $b^{-1} = \begin{pmatrix} 311 & -223 \\ 675 & -484 \end{pmatrix}$

$$b^{-1}\begin{pmatrix} -2 & 1 \\ -9 & 4 \end{pmatrix} b = \begin{pmatrix} -278165 & 128164 \\ -603729 & 278167 \end{pmatrix} = B_1$$

$$b^{-1}\begin{pmatrix} 3 & 1 \\ -7 & -2 \end{pmatrix} b = \begin{pmatrix} -1718071 & 791589 \\ -3728917 & 1718072 \end{pmatrix} = B_2$$

$$b^{-1}\begin{pmatrix} -3 & 2 \\ -5 & 3 \end{pmatrix} b = \begin{pmatrix} -56363 & 25969 \\ -122330 & 56363 \end{pmatrix} = B_3$$

$$b^{-1}\begin{pmatrix} -2 & -1 \\ -1 & -1 \end{pmatrix} b = \begin{pmatrix} 252516 & -116345 \\ 548069 & -252519 \end{pmatrix} = B_4$$

$$b^{-1}\begin{pmatrix} 3 & 1 \\ 11 & 4 \end{pmatrix} b = \begin{pmatrix} 1127855 & -519651 \\ 2447891 & -1127848 \end{pmatrix} = B_5$$

Luego, debe trasmitir $\{B_1, B_2, B_3, B_4, B_5\}$ por un canal público al usuario A.

Note que, usamos sólo $t_1, t_2, \cdots, t_5$ ya que son los cinco elementos generadores del subgrupo T_B, así mismo, usamos sólo $s_1, s_2, \cdots, s_5$ ya que son los cinco elementos generadores del subgrupo S_A.

Siguiendo el paso (V) ambos usuarios calculan β, la propiedad (i) establece que:

$$\beta(x,(y_1 \cdot y_2)) = \beta(x,y_1) \cdot \beta(x,y_2)$$

El usuario A, debe calcular $\beta(b,a)$, una vez recibidas las matrices $\{B_1, B_2, B_3, B_4, B_5\}$:

Para calcular $\beta(b,a)$ el usuario A debe usar las matrices asociadas a los números que utilizó para establecer su llave privada, recordar que el usuario A seleccionó los números: 1, 2, 3, 4 y 5, y que su elemento secreto $a = \mathcal{M}'(\mathcal{S}_1)\mathcal{M}'(\mathcal{S}_2)\mathcal{M}'(\mathcal{S}_3)\mathcal{M}'(\mathcal{S}_4)\mathcal{M}'(\mathcal{S}_5)$.

Se calcula:

$\beta(b, \mathcal{M}'(\mathcal{S}_1)) = b^{-1}\mathcal{M}'(\mathcal{S}_1)b = B_1$

$\beta(b, \mathcal{M}'(\mathcal{S}_2)) = b^{-1}\mathcal{M}'(\mathcal{S}_2)b = B_2$

$\beta(b, \mathcal{M}'(\mathcal{S}_3)) = b^{-1}\mathcal{M}'(\mathcal{S}_3)b = B_3$

$\beta(b, \mathcal{M}'(\mathcal{S}_4)) = b^{-1}\mathcal{M}'(\mathcal{S}_4)b = B_4$

$\beta(b, \mathcal{M}'(\mathcal{S}_5)) = b^{-1}\mathcal{M}'(\mathcal{S}_5)b = B_5$

Luego:

$\beta(b,a) = \beta(b, \mathcal{M}'(\mathcal{S}_1)\mathcal{M}'(\mathcal{S}_2)\mathcal{M}'(\mathcal{S}_3)\mathcal{M}'(\mathcal{S}_4)\mathcal{M}'(\mathcal{S}_5))$

Por la propiedad (i), se tiene que:

$\beta(b,a) = \beta(b, \mathcal{M}'(\mathcal{S}_1))\beta(b, \mathcal{M}'(\mathcal{S}_2))\beta(b, \mathcal{M}'(\mathcal{S}_3))\beta(b, \mathcal{M}'(\mathcal{S}_4))\beta(b, \mathcal{M}'(\mathcal{S}_5))$

Por lo tanto:

$$\beta(b,a) = B_1B_2B_3B_4B_5 = \begin{pmatrix} -214722643 & 98931912 \\ -466033276 & 214721477 \end{pmatrix}$$

El usuario B, debe calcular $\beta(a,b)$, una vez recibidas las matrices $\{A_1, A_2, A_3, A_4, A_5\}$:

Para calcular $\beta(a,b)$ el usuario B debe usar las matrices asociadas a los números que utilizó para establecer su llave privada, recordar que el usuario B seleccionó los números: 5, 4, 3, 2 y 1, y que su elemento secreto $b = \mathcal{M}'(\mathcal{T}_5)\mathcal{M}'(\mathcal{T}_4)\mathcal{M}'(\mathcal{T}_3)\mathcal{M}'(\mathcal{T}_2)\mathcal{M}'(\mathcal{T}_1)$.

Se calcula:

$\beta(a, \mathcal{M}'(\mathcal{T}_1)) = a^{-1}\mathcal{M}'(\mathcal{T}_1)a = A_1$

$\beta(a, \mathcal{M}'(\mathcal{T}_2)) = a^{-1}\mathcal{M}'(\mathcal{T}_2)a = A_2$

$\beta(a, \mathcal{M}'(\mathcal{T}_3)) = a^{-1}\mathcal{M}'(\mathcal{T}_3)a = A_3$

$\beta(a, \mathcal{M}'(\mathcal{T}_4)) = a^{-1}\mathcal{M}'(\mathcal{T}_4)a = A_4$

$\beta(a, \mathcal{M}'(\mathcal{T}_5)) = a^{-1}\mathcal{M}'(\mathcal{T}_5)a = A_5$

Luego:

$\beta(a,b) = \beta(a, \mathcal{M}'(\mathcal{T}_5)\mathcal{M}'(\mathcal{T}_4)\mathcal{M}'(\mathcal{T}_3)\mathcal{M}'(\mathcal{T}_2)\mathcal{M}'(\mathcal{T}_1))$

Por la propiedad (i), se tiene que:

$\beta(a,b) = \beta(a, \mathcal{M}'(\mathcal{T}_5))\beta(a, \mathcal{M}'(\mathcal{T}_4))\beta(a, \mathcal{M}'(\mathcal{T}_3))\beta(a, \mathcal{M}'(\mathcal{T}_2))\beta(a, \mathcal{M}'(\mathcal{T}_1))$

Por lo tanto:

$$\beta(a,b) = A_5A_4A_3A_2A_1 = \begin{pmatrix} 101250796 & 35255275 \\ -290786023 & -101250969 \end{pmatrix}$$

Siguiendo el paso (VI) Ambos usuarios obtienen la llave en común k:

El usuario A, debe calcular $k_1 = \gamma_1(a, \beta(b,a))$

$$k_1 = \gamma_1(a,\beta(b,a)) = a^{-1}\beta(b,a) = \begin{pmatrix} -695 & 164 \\ 1996 & -471 \end{pmatrix}\begin{pmatrix} -214722643 & 98931912 \\ -466033276 & 214721477 \end{pmatrix}$$

Obteniendo así:

$$k_1 = \begin{pmatrix} 72802779621 & -33543356612 \\ -209084722432 & 96334280685 \end{pmatrix}$$

El usuario B, debe calcular $k_2 = \gamma_2(b, \beta(a,b))$

$$k_2 = \gamma_2(b,\beta(a,b)) = \beta(a,b)^{-1}b = \begin{pmatrix} -101250969 & -35255275 \\ 290786023 & 101250796 \end{pmatrix}\begin{pmatrix} -484 & 223 \\ -675 & 311 \end{pmatrix}$$

Obteniendo así:

$$k_2 = \begin{pmatrix} 72802779621 & -33543356612 \\ -209084722432 & 96334280685 \end{pmatrix}$$

Así, $k_1 = k_2$. Por lo tanto, estableceremos una llave en común a la cual sólo denotaremos como k. Donde k es la matriz asociada a la 3-trenza:

$\mathcal{T}$(-2, -1, -6, -1, -4, -4, -1, -1, -4, -1, -6, -1, -1, -1, -1, -2, -4, -4, -1, -6, -1, -3, -2, -1, -1, -6, -1, -5, -2)

Siguiendo el paso (VII) ambos usuarios generan una condición inicial X_0 dada la llave en común k:

(a) Calculan $|(k_{11}+k_{22})(k_{12}+k_{21})|$:
$|(72802779621+96334280685)(-33543356612-209084722432)| =$
$|-4103740003719396\text{x}10^7| = 4103740003719396\text{x}10^7$

(b) Seleccionan del número anterior sólo los primeros 15 dígitos: 410374000371939

(c) Multiplican el número anterior por 10^{-15} para obtener la condición inicial X_0:
$(410374000371939)(10^{-15}) = 0.410374000371939 = X_0$

Así, ambos usuarios obtienen la condición inicial:

$$X_0 = 0.410374000371939$$

Siguiendo el paso (VIII) el usuario A genera y envía su cadena binaria encriptada .

El usuario A, primero establece el texto plano *Instituto Potosino de Investigación Científica y Tecnológica A.C.* Luego, genera su cadena binaria encriptada $110100011\cdots011110111$, utilizando la condición inicial X_0 y siguiendo los pasos del (1) al (6) del procedimiento ML_2 establecido en el Sección 3.1. Por último, envía la cadena binaria encriptada al usuario B.

Siguiendo el paso (IX) el usuario B recupera el texto plano.

El usuario B, recibe la cadena binaria encriptada generada en el paso (VIII). Luego, utilizando la condición inicial X_0 y siguiendo los pasos del (9) al (11) del procedimiento ML_2 establecido en el Sección 3.1, recupera el texto plano enviado por el usuario A.

Siguiendo este procedimiento, podemos implementar dicho protocolo en un software con el fin de poder hacerlo accesible para los usuarios. Por ello, implementaremos en el siguiente capítulo una interfaz gráfica en MATLAB® R2016a.

Capítulo 4

Implementación del protocolo en MATLAB®

En este capítulo se muestra la implementación del protocolo TC, presentado en la Sección 3.2, en una interfaz gráfica, GUI, por las siglas en inglés de Graphical User Interface, del software MATLAB®. Proporcionando al usuario un entorno visual que le permita interactuar fácilmente con dicho protocolo. MATLAB® permite controlar el diseño, las propiedades y el comportamiento de los componentes que conforman la GUI, en la cual, se pueden incorporar botones, paneles, iconos, menús y cuadros de diálogos. Además, nos permite combinar el diseño, la programación y los cálculos matemáticos, especialmente con matrices.

Usaremos como base el uso de la GUI, en la cual implementamos el protocolo TC. La Figura 4.1 muestra la ventana de la GUI. Además, usaremos la *ventana de comandos* de MATLAB® para generar la información necesaria para poder llevar a cabo dicho protocolo. El pseudocódigo correspondiente para su implementación será mostrado en el Apéndice.

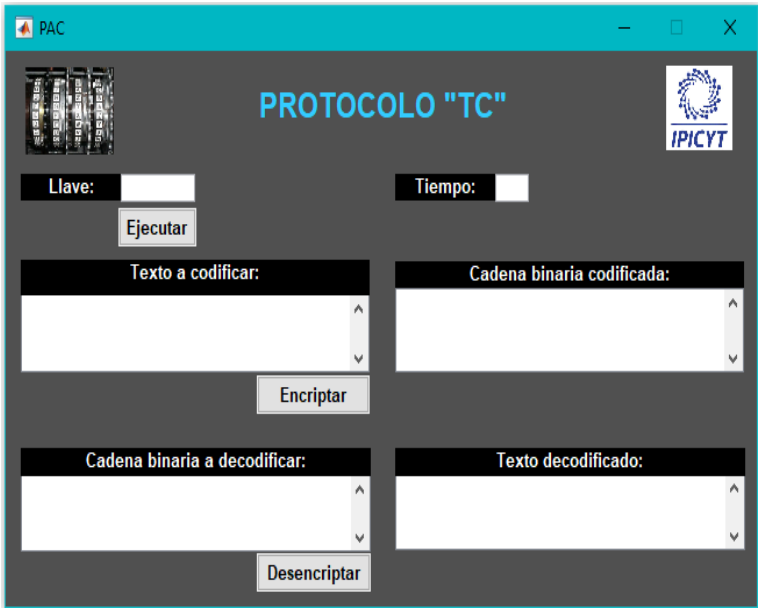


Figura 4.1: *Ventana de la GUI.*

Recordar que para llevar acabo el protocolo TC se requiere que cada usuario:

Utilice las funciones : $\beta(x,y) = x^{-1}yx \quad , \quad \gamma_1(x,y) = x^{-1}y \quad , \quad \gamma_2(x,y) = y^{-1}x$

(I) Establezca un subgrupo público generado, T_A y S_B respectivamente.

(II) Elija su elemento secreto, $a \in T_A$ y $b \in S_B$ respectivamente.

(III) Acuerde con el otro usuario un tiempo de validez para la llave.

(IV) Calcule y transmita, respectivamente: $a^{-1}s_1a, \cdots, a^{-1}s_na \quad , \quad b^{-1}t_1b, \cdots, b^{-1}t_mb$

(V) Calcule, respectivamente: $\beta(b,a) = b^{-1}ab \quad , \quad \beta(a,b) = a^{-1}ba$

(VI) Calcule la llave en común k, respectivamente: $k_1 = \gamma_1(a, \beta(b,a)) \ , \ k_2 = \gamma_2(b, \beta(a,b))$

(VII) Genere una condición inicial X_0.

Para que un usuario pueda encriptar un texto plano necesita:

(VIII) Establecer y enviar su cadena binaria encriptada, siguiendo los pasos del (1) al (6) del procedimiento ML_2

Para que un usuario pueda desencriptar un texto plano necesita:

(IX) Recibir una cadena binaria encriptada y seguir los pasos del (9) al (11) del procedimiento ML_2, para recuperar el texto plano.

Cabe señalar que, la persona que inicie dicho protocolo para transmitir un mensaje deberá fungir como el usuario A, mientras que, el usuario que recibe el mensaje deberá fungir como el usuario B.

Para poder iniciar la GUI se requiere que cada usuario tenga en sus documentos la carpeta PROTOCOLO TC, en la cual encontrarán sólo dos carpetas: Usuario A y Usuario B, cuyo contenido se muestra en la Figura 4.2 respectivamente.

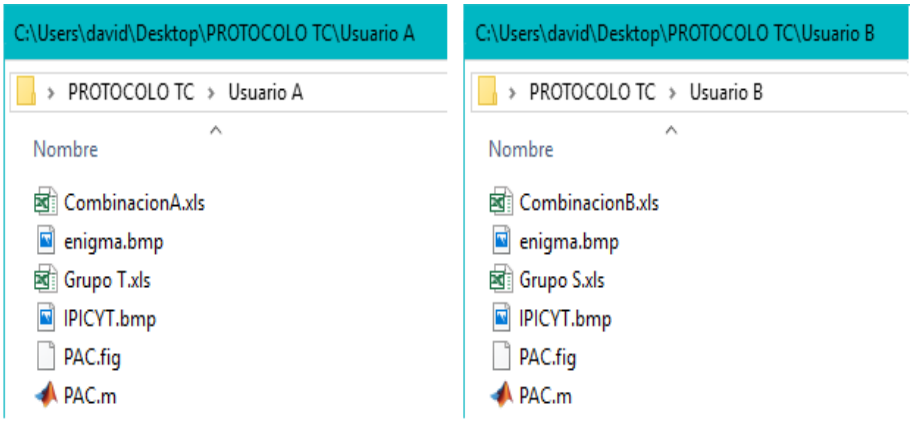


Figura 4.2: *Carpetas de ambos usuarios.*

De los cuales:

Los archivos “CombinacionA.xls” y “CombinacionB.xls” son creados en Excel usando el paso (IV). Es decir, para esta implementación, contienen los resultados de las siguientes multiplicaciones, respectivamente: $a^{-1}s_1a,\cdots,a^{-1}s_{10}a$ y $b^{-1}t_1b,\cdots,b^{-1}t_{10}b$.

Los archivos “enigma.bmp” e “IPICYT.bmp” son imágenes en el formato mapa de bits.

Los archivos “Grupo T.xls” y “Grupo S.xls” son creados en Excel usando el paso (I). Es decir, contienen la información que genera al subgrupo T_A y S_B respectivamente. Para esta implementación, cada subgrupo estará conformado por cinco elementos generadores y sus inversas.

El archivo “PAC.fig” contiene la información referente al diseño de la GUI-PAC.

El archivo “PAC.m” contiene el programa que se ejecutará en la GUI-PAC.

De dichos archivos deberemos ejecutar el archivo “PAC.m” mediante el software MATLAB® lo cual mostrará la GUI-PAC en pantalla. En la cual habrá dos grupos de elementos, ver Figura 4.3. Los recuadros en blanco, denotados por las letras A, B, D, E, G y H, y los botones, denotados por las letras C, F e I.

Figura 4.3: *Imagen que muestra las partes que conforman la GUI-PAC.*

4.1. Uso de la GUI

Para usar la GUI, en esta implementación, cada usuario:

[1] Debe ingresar en el recuadro A cinco números enteros, los cuales no necesariamente deben estar ordenados y además se pueden repetir, entre el número uno y el siete. De los diez elementos generadores que conforman cada subgrupo, cinco matrices y sus inversa, se seleccionarán sólo siete de ellos para generar la *llave privada*. Establecerá dicha llave usando el paso (II).

Dado que utilizamos multiplicaciones de matrices, dichas multiplicaciones aumentan rápidamente la longitud de cada componente en la matriz resultante. Por ello, para poder representar dichas matrices resultantes en el software MATLAB® establecimos sólo siete elementos generadores para con ellos establecer la llave privada.

[2] Debe ingresar en el recuadro B el tiempo T durante el cual la llave será válida. Establecerá dicho tiempo usando el paso (III).

[3] Una vez completados los recuadros A y B debe dar clic en el botón Ejecutar, botón C, para con ello generar los archivos Grupo T.xls y CombinacionA.xls para el usuario A y los archivos Grupo S.xls y CombinacionB.xls para el usuario B. Se establecerán dichos archivos usando los pasos (I), (II) y (IV). Además, se calculará la llave en común y con ella se establecerá la condición inicial X_0. Se establecerá la llave en común usando los pasos (V) y (VI), mientras que, para establecer la condición inicial se usará el paso (VII). Luego, los usuarios deberán dejar de usar un momento la GUI y utilizar la *ventana de comandos*, la cual mostrará la siguiente información en pantalla:

Usuario A:

```
Espere un momento mientras se genera el archivo Grupo T.xls
```

Usuario B:

```
Espere un momento mientras se genera el archivo Grupo S.xls
```

Una vez que generado el archivo, la *ventana de comandos* mostrará la siguiente información:

Usuario A:

```
Envíe el archivo Grupo T.xls al usuario B y después teclee 1 para continuar:
```

Usuario B:

```
Envíe el archivo Grupo S.xls al usuario A y después teclee 1 para continuar:
```

Una vez que se envíe dicho archivo se debe teclear el número 1. Con lo cual, la *ventana de comandos* mostrará la siguiente información:

Usuario A:

```
Una vez recibido el archivo Grupo S.xls teclee 2 para continuar:
```

Usuario B:

```
Una vez recibido el archivo Grupo T.xls teclee 2 para continuar:
```

Una vez que se reciba dicho archivo se debe teclear el número 2. Lo cual, mostrará en la *ventana de comandos* la siguiente información:

Usuario A:

```
Espere un momento mientras se genera el archivo CombinacionA.xls
```

Usuario B:

```
Espere un momento mientras se genera el archivo CombinacionB.xls
```

Una vez generado el archivo, la *ventana de comandos* mostrará la siguiente información:

Usuario A:

```
Envíe el archivo CombinacionA.xls al usuario B y después teclee 1 para conti-
nuar:
```

Usuario B:

```
Envíe el archivo CombinacionB.xls al usuario A y después teclee 1 para conti-
nuar:
```

Una vez que se envíe dicho archivo se debe teclear el número 1. Con lo cual, la *ventana de comandos* mostrará la siguiente información:

Usuario A:

```
Una vez recibido el archivo CombinacionB.xls teclee 2 para continuar:
```

Usuario B:

```
Una vez recibido el archivo CombinacionA.xls teclee 2 para continuar:
```

Una vez que se reciba dicho archivo se debe teclear el número 2. Lo cual, mostrará en la *ventana de comandos* de ambos usuarios la siguiente información:

```
Puede volver a utilizar la GUI
=======================================================
Hora en la que se reiniciará la GUI: xx
=======================================================
```

Una vez visualizada esta información se podrá volver a utilizar la GUI, tenga en cuenta que la hora "xx" indica la hora en que el protocolo TC se reiniciará. Si esto llegará a ocurrir la *ventana de comandos* de ambos usuarios mostrará el siguiente mensaje:

```
=======================================================
El programa se reiniciará por su seguridad
=======================================================
```

Con lo cual se borrará toda la información de la *ventana de comandos* y la GUI se reiniciará.

[4] Debe volver a utilizar la GUI.

• Para encriptar: una vez completado el recuadro D con un texto plano, debe dar clic en el botón F, lo cual encriptará la cadena binaria asociada al texto plano. Además, se mostrará la cadena binaria encriptada del paso (VIII) en el recuadro E. Esta cadena binaria deberá ser enviada por un canal público, por ejemplo correo electrónico o WhatsApp, al otro usuario.

• Para desencriptar: una vez completado el recuadro G con una cadena binaria encriptada, debe dar clic en el botón I, lo cual desencriptará la cadena binaria encriptada obteniendo así la cadena binaria asociada a un texto plano. Además, el recuadro H mostrará el texto plano obtenido en el paso (IX).

Ejemplo 33. Veremos en concreto cómo se llevó acabo el uso de la GUI entre dos usuarios.

Siguiendo el paso [1] el usuario A seleccionó los números: 1, 2, 3, 4 y 5, mientras que el usuario B seleccionó los números: 5, 4, 3, 2 y 1. La Figura 4.4 muestra la GUI del usuario A con los 5 números que ingresó, mientras que, la Figura 4.5 muestra la GUI del usuario B con los 5 números que ingresó. Cabe señalar que se debe dejar un espacio entre cada número ingresado.

Figura 4.4: *Llave privada usuario A.*

Figura 4.5: *Llave privada usuario B.*

Siguiendo el paso [2] ambos usuarios acordaron T=20 como tiempo de validez para la llave privada. La Figura 4.6 muestra la GUI del usuario A con los 20 minutos establecidos.

Figura 4.6: *Tiempo de validez.*

El usuario A siguiendo el paso [3], generó y envió los archivos Grupo T.xls y CombinacionA.xls. Además, recibió los archivos Grupo S.xls y CombinacionB.xls. Si siguió los pasos indicados, su *ventana de comandos* mostrará la siguiente información:

```
Espere un momento mientras se genera el archivo Grupo T.xls
Envíe el archivo Grupo T.xls al usuario B y después teclee 1 para continuar:1
Una vez recibido el archivo Grupo S.xls teclee 2 para continuar:2
Espere un momento mientras se genera el archivo CombinacionA.xls
Envíe el archivo CombinacionA.xls al usuario B y después teclee 1 para conti-
nuar:1
Una vez recibido el archivo CombinacionB.xls teclee 2 para continuar:2
Puede volver a utilizar la GUI
=====================================================
Hora en la que se reiniciará la GUI: 10:20 am
=====================================================
```

El usuario B siguiendo el paso [3], generó y envió los archivos: Grupo S.xls y CombinacionB.xls. Además, recibió los archivos: Grupo T.xls y CombinacionA.xls. Si siguió los pasos indicados, su *ventana de comandos* mostrará la siguiente información:

```
Espere un momento mientras se genera el archivo Grupo S.xls
Envíe el archivo Grupo S.xls al usuario A y después teclee 1 para continuar:1
Una vez recibido el archivo Grupo T.xls teclee 2 para continuar:2
Espere un momento mientras se genera el archivo CombinacionB.xls
Envíe el archivo CombinacionB.xls al usuario A y después teclee 1 para conti-
nuar:1
Una vez recibido el archivo CombinacionA.xls teclee 2 para continuar:2
Puede volver a utilizar la GUI
=====================================================
Hora en la que se reiniciará la GUI: 10:20 am
=====================================================
```

Usando el paso [4]

• Para encriptar: el usuario A volvió a la GUI y en ella introdujo el texto plano: Instituto Potosino de Investigación Científica y Tecnológica A.C. Dio clic en el botón encriptar y con ello generó una cadena binaria encriptada, misma que envió al usuario B por medio de un canal público. La Figura 4.7 muestra su GUI al completar este proceso.

• Para desencriptar: el usuario B volvió a la GUI y en ella introdujo la cadena binaria encriptada recibida. Dio clic en el botón desencriptar y con ello generó una cadena binaria desencriptada, la cual se tradujo a texto plano. La Figura 4.8 muestra su GUI al completar este proceso.

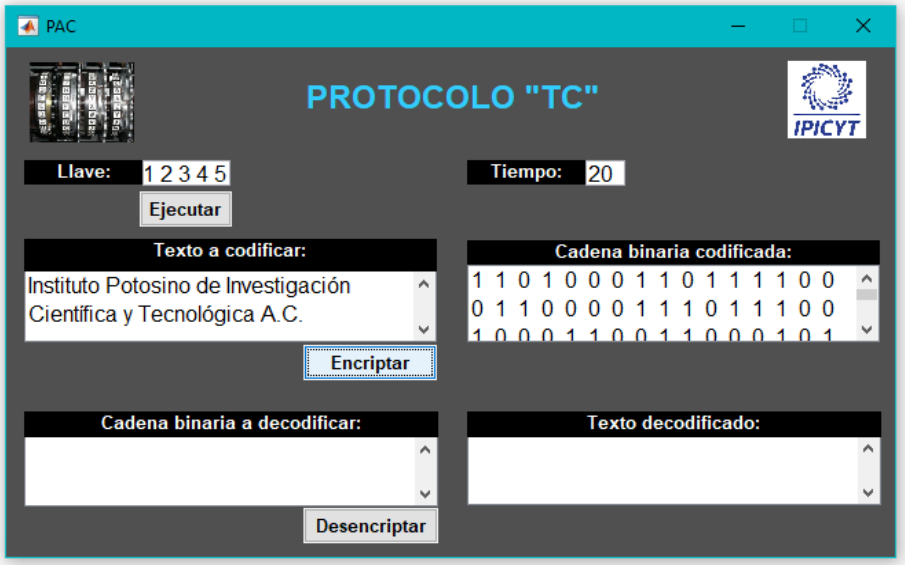


Figura 4.7: *Encriptar con la GUI.*

Figura 4.8: *Desencriptar con la GUI.*

Recordar que ambos usuarios podrán intercambiar información utilizando la GUI, siempre y cuando, no se exceda el tiempo establecido T. Si esto ocurre, tanto la *ventana de comandos* como la GUI se reiniciarán.

Capítulo 5

Conclusiones

La criptología tiene como finalidad transmitir información de manera segura, por lo cual, es importante establecer un procedimiento que lleve a cabo dicho propósito. Así, dada la importancia de trasmitir información por un canal público, siguiendo el trabajo de Anshel y colaboradores en [1], se estableció el protocolo TC cuya finalidad es la gestión y distribución de las llaves, utilizando para ello, el grupo no conmutativo de las 3-trenzas para generar una llave en común entre dos usuarios una vez recibida la llave pública. Siguiendo el trabajo de Ki Hyoung Ko y colaboradores en [3], se estableció el grupo no conmutativo de las 3-trenzas debido a la dificultad de resolver en él el problema de la palabra. La llave en común será introducida como condición inicial en el procedimiento ML_2.

Además, se utilizó el mapeo logístico para generar una cadena binaria para encriptar o desencriptar una cadena binaria asociada a un texto plano. Por lo cual, se propuso el procedimiento ML_2, con el fin de robustecer el procedimiento ML_1 ante un ataque de fuerza bruta, para establecer así una nueva cadena para encriptar o desencriptar. El ataque de fuerza bruta consistió en, interceptar una cadena binaria encriptada de la cual se intuye una palabra de siete letras que es *altamente probable* que aparezca en el texto plano. Implementar un algoritmo que realizará una búsqueda exhaustiva mediante la generación de mil condiciones iniciales y el procedimiento ML_1, para con ello, generar mil cadenas binarias. Luego, sumar cada cadena binaria a la cadena binaria encriptada interceptada y buscar la similitud entre estas cadenas y la palabra probable, logrando con ello recuperar el texto plano. De la misma forma, una vez establecido el procedimiento ML_2 se intentó el mismo ataque por fuerza bruta y en este caso no se logró recuperar el texto plano.

Por último, dada la necesidad de hacer accesible el protocolo TC a los usuarios, se implementó en una GUI, permitiendo a los usuarios establecer una comunicación, a través de un canal público, mediante un entorno visual que le permita interactuar fácilmente con dicho protocolo y el software MATLAB®. Incluyendo su respectivo pseudocódigo en el Apéndice con la finalidad de que pueda ser implementado en otro software, por ejemplo, en Python, dado que es un lenguaje interpretador, que posee una licencia de código abierto y además es multiplataforma.

Apéndice

Dado que se creó una GUI basándonos en el procedimiento establecido por el protocolo TC, en el cual cada paso fue denotado por un número romano, en este Apéndice mostraremos su pseudocódigo con la finalidad de que pueda ser implementado en diversos lenguajes de programación.

Descripción de las funciones involucradas en esta implementación:

- $\beta(x,y) = x^{-1}yx$ $\quad$ • $\gamma_1(x,y) = x^{-1}y$ $\quad$ • $\gamma_2(x,y) = y^{-1}x$
- $\delta(x)$ toma los primeros 15 dígitos del número x.
- $ML_2(X_0)$ genera con la condición inicial X_0 una cadena binaria mediante el procedimiento ML_2 de la Sección 3.1.

IMPLEMENTACIÓN DEL PROTOCOLO TC

CON (I) ASOCIAN A CADA 3-TRENZA SU MATRIZ $\mathcal{M}'$

▷ Usuario A

para $i \leftarrow 1$ hasta n **hacer**
  $T_i \leftarrow$ 3-TRENZA i.
  $L_i \leftarrow length(T_i)$
  para $N \leftarrow 1$ hasta L_i **hacer**
    si N es par **entonces**
      $A_i[1:2, 2N-1:2N] \leftarrow [1\ ,\ -T_i(N)/\sqrt{-1}\ ;\ 0\ ,\ 1]$
    si no
      $A_i[1:2, 2N-1:2N] \leftarrow [1\ ,\ 0\ ;\ T_i(N)/\sqrt{-1}\ ,\ 1]$
    fin si
    para $N \leftarrow 2$ hasta L_i **hacer**
      $M'T_i \leftarrow A_i[1:2, 1:2]A_i[1:2, 2N-1:2N]$
    fin para
  fin para
  $M'T_i[2,1] \leftarrow M'T_i[2,1](\sqrt{-1})$
  $M'T_i[1,2] \leftarrow M'T_i[1,2](-\sqrt{-1})$
devolver $M'T_i$
fin para
Exporta y envía los $M'T_i$

▷ Usuario B

para $i \leftarrow 1$ hasta m **hacer**
    $S_i \leftarrow$ 3-TRENZA i.
    $LL_i \leftarrow length(S_i)$
    para $N \leftarrow 1$ hasta L_i **hacer**
        si N es par **entonces**
            $B_i[1:2, 2N-1:2N] \leftarrow [1\ ,\ -S_i(N)/\sqrt{-1}\ ;\ 0\ ,\ 1]$
        si no
            $B_i[1:2, 2N-1:2N] \leftarrow [1\ ,\ 0\ ;\ S_i(N)/\sqrt{-1}\ ,\ 1]$
        fin si
        para $N \leftarrow 2$ hasta L_i **hacer**
            $M'S_i \leftarrow B_i[1:2, 1:2]B_i[1:2, 2N-1:2N]$
        fin para
    fin para
    $M'S_i[2,1] \leftarrow M'S_i[2,1](\sqrt{-1})$
    $M'S_i[1,2] \leftarrow M'S_i[1,2](-\sqrt{-1})$
devolver $M'S_i$
fin para
Exporta y envía los $M'S_i$

CON (II) GENERAN SU LLAVE SECRETA

▷ Usuario A

para $i \leftarrow 1$ hasta j **hacer**
    $MT_i \leftarrow$ algún $M'T$
fin para
$ESA \leftarrow (MT_1) \cdots (MT_j)$

▷ Usuario B

para $i \leftarrow 1$ hasta k **hacer**
    $MS_i \leftarrow$ algún $M'S$
fin para
$ESB \leftarrow (MS_1) \cdots (MS_k)$

CON (III) REINICIAN PROTOCOLO TC

▷ Usuario A y B

$T \leftarrow$ tiempo de validez
$TT \leftarrow$ tiempo transcurrido
si $TT > T$ **entonces**
    reiniciar protocolo TC
fin si

CON (IV) GENERAN COMBINACIÓN

▷ Usuario A
Recibe MS_i
para $i \leftarrow 1$ hasta m **hacer**
    $COMA_i \leftarrow (ESA)^{-1} MS_i (ESA)$
fin para
Exporta y envía $COMA_i$
▷ Usuario B
Recibe MT_i
para $i \leftarrow 1$ hasta n **hacer**
    $COMB_i \leftarrow (ESB)^{-1} MT_i (ESB)$
fin para
Exporta y envía $COMB_i$

CON (V) CALCULAN β

▷ Usuario A
Recibe $COMB_i$
$BA \leftarrow \beta(b,a)$

▷ Usuario B
Recibe $COMA_i$
$AB \leftarrow \beta(a,b)$

CON (VI) GENERAN LA LLAVE EN COMÚN

▷ Usuario A
$K \leftarrow \gamma_1(ESA, BA)$

▷ Usuario B
$K \leftarrow \gamma_2(ESB, AB)$

CON (VII) GENERAN LA CONDICIÓN INICIAL

▷ Usuario A o B
$ci \leftarrow (K[2,1] + K[1,2])(K[1,1] + K[2,2])$
$CI \leftarrow (10^{-15})\delta(ci)$

CON (VIII) ENCRIPTAN

▷ Usuario A o B
$CC \leftarrow ML_2(CI)$
$TB \leftarrow$ Texto plano en representación binaria
$CE \leftarrow TB + CC$
Envía CE

CON (IX) DESENCRIPTAN

▷ Usuario A o B

Recibe CE

$CC \leftarrow ML_2(CI)$

$TB \leftarrow CE + CC$

Traduce TB a Texto plano

TERMINA IMPLEMENTACIÓN DEL PROTOCOLO TC

Bibliografía

[1] I. Anshel and M. Anshel and D. Goldfeld, *An algebraic method for public-key cryptography*, Mathematical Research Letters 6, pp 287-291, 1999.

[2] G. Baumslag and Y. Brukhov and B. Fine and G. Rosenberger, *Encryption methods using formal power series rings*, 2018.

[3] K. H. Ko and S. J. Lee and J. W. Han and J. Kang and C. Park, *New Public-Key Cryptosystem Using Braid Groups*, Advances in Cryptology-CRYPTO 2000. Lecture Notes in Computer Science, vol. 1880, Springer-Berlin Heidelberg, pp 166-183, 2000.

[4] G. Baumslag and B. Fine and X. Xu, *Cryptosystems using linear groups*, Appl. Algebra Eng., Commun. Comput. 17, pp 205-217, 2006.

[5] G. Baumslag and B. Fine and X. Xu, *A proposed public key cryptosystem using the modular groups*, Contemp. Math. 421, Amer. Math. Soc., pp 34-43, 2006.

[6] R. A. Mollin, An introduction to cryptography, Second Edition, Chapman and Hall/CRC, 2007.

[7] B. Schneier, *Applied Cryptography*, Second Edition: Protocols, Algorithms, and Source Code in C, Wiley Computer Publishing, John Wiley and Sons, Inc., 1996.

[8] C. Paar and J. Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, Springer-Verlag Berlin Heidelberg, 2010.

[9] W. Diffie and M. Hellman, *New directions in cryptography*, IEEE Trans. Inf. Theory 22, pp. 644-654, 1976.

[10] J. B. Fraleigh, Álgebra abstracta: Primer curso, Tercera edición, Addison-Wesley Iberoamérica S.A., 1988.

[11] K. Murasugi, Knot theory and its applications, Birkäuser-Boston Basel Berlin, 1996.

[12] H. Cabrera-Ibarra and D. A. Lizárraga-Navarro, *Braid solutions to the action of the Gin enzyme*, Journal of Knot Theory and Its Ramifications, vol. 19, pp 1051-1074, 2010.

[13] R. M. May, Theoretical Ecology: principles and applications, Blackwell Scientific Ltd, 1976.

[14] K. T. Alligood and T. D. Sauer and J. A. Yorke, CHAOS: An Introduction to Dynamical Systems, Springer-Verlag New York, 1996.