MÁSTER EN MATEMÁTICAS

TRABAJO FIN DE MÁSTER:

UN ESQUEMA DE MANEJO DE CLAVES EN AMBIENTES DISTRIBUIDOS BASADO EN ACCIONES DE SEMIGRUPOS

Autor: Mohamed Baouch

Directores:
Blas Torrecillas Jover
Juan Antonio López Ramos

MÁSTER EN MATEMÁTICAS

TRABAJO FIN DE MÁSTER:

UN ESQUEMA DE MANEJO DE CLAVES EN AMBIENTES DISTRIBUIDOS BASADO EN ACCIONES DE SEMIGRUPOS

Memoria presentada para optar
al Máster en Matemáticas
por Mohamed Baouch

Universidad de Almería, diciembre de 2016.

Índice general

Capítulo 1

Introducción

Desde que el ser humano ha necesitado comunicarse con los demás, por un canal inseguro, sin que algunos de sus mensajes fueran conocidos por otras personas distintas a los destinatarios, surgió la idea de crear sistemas de cifrado (criptosistemas), de forma que un mensaje original (llamado también texto inicial o texto en claro) después de un proceso de transformaciones, lo que llamamos mensaje cifrado, solo pudiera ser leído por los receptores.

La ciencia encargada de estudiar estos sistemas criptográficos o criptosistemas, se denomina *Criptología*. Esta ciencia abarca dos disciplinas con objetivos totalmente opuestos. La primera de ellas es la *criptografía*, la cual se dedica a la creación de dichos criptosistemas y posteriormente su implementación. Por lo tanto, su propósito es asegurar la seguridad y la confidencialidad de la comunicación.

Dependiendo del número de claves que utilicen estos sistemas criptográficos para cifrar o descifrar la información, podemos dividir la criptografía en dos grandes familias: La *criptografía de clave privada* o *simétrica* y la *criptografía de clave pública* o *asimétrica*. En la primera familia están los algoritmos que usan una única clave conocida entre usuarios que comparten el sistema, de forma que sólo se puede obtener el texto descifrado conociendo dicha clave privada. Por lo tanto, el emisor y el receptor deben haber acordado dicha clave de forma confidencial antes de intercambiar información (Figura 1.1). El ejemplo más clásico de la criptografía simétrica es el cifrado de *César*, que usaba en sus fines militares. Este cifrado consiste en desplazar las letras del

alfabeto un número determinado de posiciones hacia la derecha o hacia la izquierda.

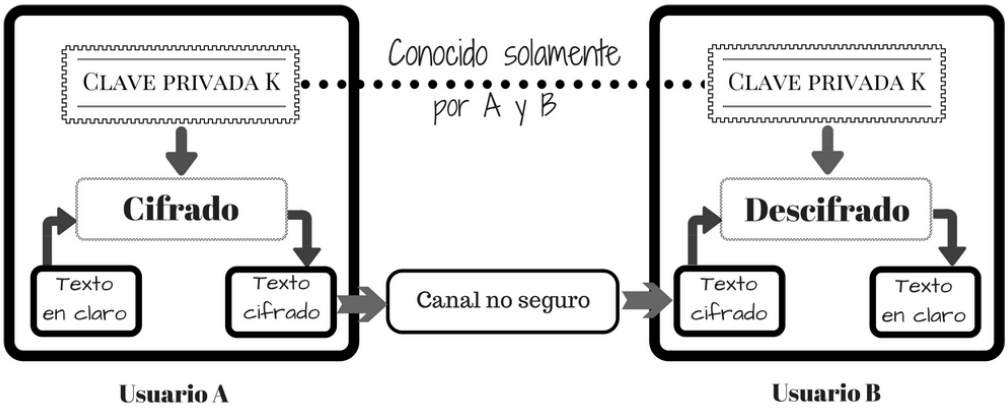

Figura 1.1: Criptografía de clave privada.

A diferencia de la familia anterior, en la criptografía de clave pública o criptografía asimétrica, los usuarios poseen dos claves distintas, la primera de ellas es pública y distribuida libremente entre los demás usuarios, se utiliza para cifrar el mensaje. La otra clave es privada, por lo que es conocida solamente por el receptor para poder descifrar y obtener el mensaje en claro. La seguridad de esta técnica reside en la complejidad o dificultad computacional a la hora de calcular la clave privada conociendo la clave pública sin tener acceso a cierta información que posee el receptor (Figura 1.2). El *RSA* desarrollado por *Rivest, Shamir* y *Adleman* en 1978 [20] es uno de los sistemas de clave pública más usados en la actualidad para la transmisión de información a través de canales inseguros. La seguridad de este algoritmo reside en el problema de la factorización de números enteros junto con distintas versiones basadas en el cálculo del logaritmo discreto sobre un cuerpo finito, [8], y sobre el grupo de puntos de una curva elíptica, [14] y [18].

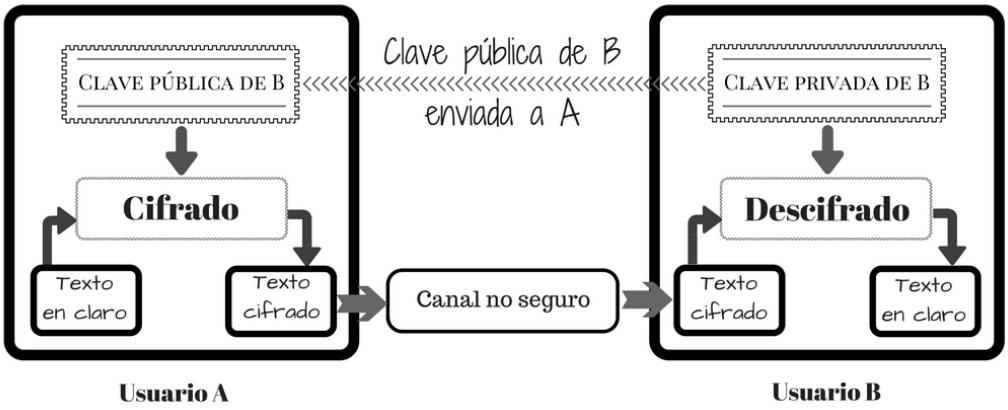

Figura 1.2: Criptografía de clave pública.

La criptografía de clave pública tiene dos aplicaciones fundamentales, como son la firma digital, que acompaña archivos digitales, y el intercambio de claves. Nosotros nos centraremos en esta última.

La otra disciplina de la criptología es el *criptoanálisis*, que como hemos mencionado anteriormente, tiene un objetivo totalmente opuesto a la criptografía. Su función es encontrar 'agujeros' o debilidades en un sistema criptográfico para poder 'romperlo' sin tener acceso a la información secreta requerida, y así interferir en el proceso de comunicación. La estrategia usada por el criptoanalista depende de la naturaleza del esquema de cifrado y de la información disponible. Los ataques criptoanalíticos se pueden clasificar de muchas maneras, pero nosotros nos centraremos en su clasificación según la forma de actuar del atacante. Las técnicas utilizadas por los criptoanalistas se clasifican en dos categorías, llamadas *ataques pasivos* y *ataques activos*.

Los ataques pasivos son aquellos en los que el atacante se dedica a la escucha y la vigilancia de la transmisión de datos (Figura 1.3). En otras palabras, el propósito del atacante es tratar de obtener la información que se encuentra en tránsito. El término 'pasivo' se debe a que el atacante no intenta, en ningún momento durante la vigilancia, realizar alguna modificación en los datos. De hecho, por esta razón, los atacantes pasivos son más difíciles de detectar. Por lo tanto el enfoque general que se suele seguir para hacer frente

a los atacantes pasivos, es el uso de mecanismos de prevención, más que de detección.

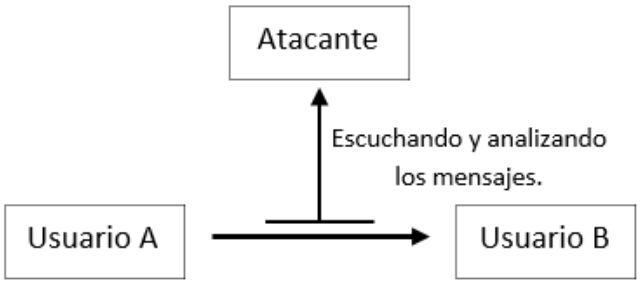

Figura 1.3: Ataque pasivo.

Existen dos tipos de ataques pasivos: *ataques de obtención de información* y *análisis de tráfico*. En el primer tipo, el atacante se dedica exclusivamente a la captura de los datos, sin el conocimiento de los usuarios del sistema, que contengan información confidencial. Cifrar todos los datos puede ser una solución para protegerse de este tipo de ataques. En el segundo tipo, el atacante intenta analizar los mensajes cifrados con el fin de encontrar patrones, es decir, trata de averiguar similitudes en los datos para llegar a obtener alguna pista acerca de la comunicación que se está llevando a cabo, y así descifrar los mensajes, o mejor aún, conseguir la clave utilizada en el sistema.

A diferencia de los ataques pasivos, los atacantes activos tratan de alterar, cambiar o modificar los datos. Se trata de un ataque directo a los usuarios del sistema. En este tipo, el atacante puede modificar la información o los datos durante la transmisión, o puede crear datos falsos y enviarlos al destinatario.

Los ataques activos se pueden clasificar en tres categorías:

- Suplantación.
- Repetición (*Man-In-The-Middle*).
- Interrupción.

- **Suplantación:** En esta categoría, están los atacantes que intentan hacerse pasar por un usuario del sistema. Supongamos que dos usuarios A y B se están comunicando a través de un canal inseguro (Figura 1.4). En este caso, el atacante se pondría en contacto con B reemplazando al usuario A (sin el conocimiento de A y B).

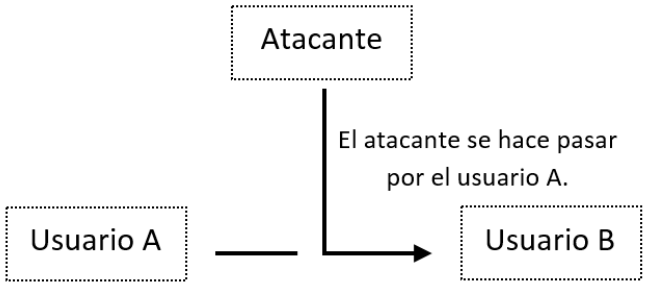

Figura 1.4: Ataque activo: Suplantación.

- **Repetición** *(Man-In-The-Middle)*: (Figura 1.5) Este tipo de ataques pueden ser tanto pasivos como activos. En el caso pasivo, el atacante captura los datos que se están transmitiendo, hace una copia de dicha información y después se la envía al receptor sin ser detectado. Sin embargo, en el caso activo, los datos capturados son modificados antes de ser enviados al destinatario. De esta manera, el atacante obtiene acceso y la capacidad de hacer cualquier acción que un usuario autorizado pueda hacer en el sistema.

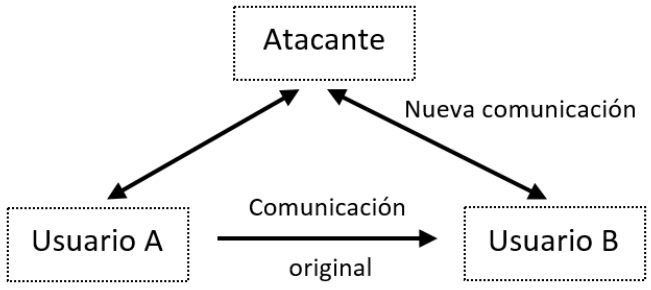

Figura 1.5: Ataque activo: *Man-In-The-Middle*.

- **Interrupción** *(Denial Of Service)*: Los atacantes intentan evitar que los usuarios del sistema tengan acceso a ciertos servicios. Por ejemplo, un usuario no autorizado podría enviar demasiadas solicitudes de acceso a un servidor utilizando los identificadores de usuario al azar, una tras otra en rápida sucesión, a fin de 'inundar' el sistema y negar a otros usuarios del uso de la red.

Ejemplo 1.1. *EL ejemplo más clásico de los ataques activos es el Man-In-The-Middle contra el protocolo de intercambio de claves en grupo de Diffie-Hellman [7]. Este ataque (Figura 1.6) funciona de la siguiente manera:*

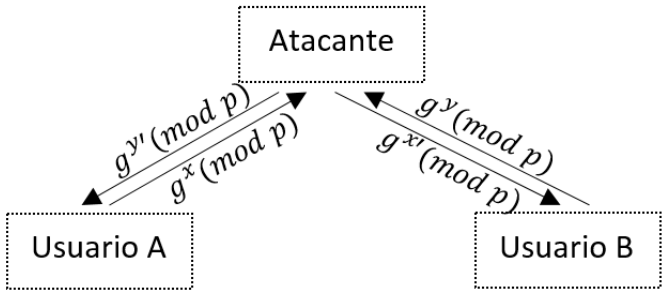

Figura 1.6: *Man-In-The-Middle* en *Diffie-Hellman*.

Los usuarios A y B tienen sus claves privadas x e y, respectivamente. El atacante M se introduce a sí mismo en la trayectoria de la comunicación entre los usuarios A y B (Sin el conocimiento de estos.):

1. *M genera las claves x' e y' (para suplir las claves privadas de A y B).*
2. *M intercepta la clave pública de A, $g^x \pmod p$, la reemplaza por $g^{x'} \pmod p$ y la envía al usuario B.*
3. *M, al mismo tiempo, intercepta también la clave pública del usuario B, $g^y \pmod p$, la reemplaza por $g^{y'} \pmod p$ y se le manda al usuario A.*

Entonces, el usuario A se comunicará con el usuario B con la clave $K_A = g^{xy'} \pmod p$, el usuario B con A usando la clave $K_B = g^{xy'} \pmod p$ y el atacante será capaz de obtener las dos claves. De este

modo, cuando el usuario A envía un mensaje cifrado (con su clave K_A) al usuario B, el atacante M lo intercepta, lo descifra, lo cifra con la clave de B (K_B) y se lo manda al usuario B. De la misma manera, descifra los mensajes de B (para A) y los cifra con la clave de A. Los usuarios A y B creen que se están comunicando de forma segura, mientras M lee todo el tráfico de mensajes.

Sin embargo, los nuevos modos de comunicación que actualmente se están extendiendo rápidamente, sobre todo por el uso y gran auge que está teniendo la denominada "Internet de las Cosas", o "Internet of Things", IOT, el modelo de comunicación está cambiando, de una "comunicación punto a punto", a comuniciones en grupo, en el que la confidencialidad y la seguridad, siguen siendo básicos. Existen distintos modelos de comunicación en grupo dependiendo de la existencia o no de una figura o figuras que juegan un papel más importante que el resto de comunicantes como es el caso de los esquemas centralizados o los que dividen a los usuarios en grupos y, otros, los que quizás se ajustan más al modelo que da base a la Internet de las Cosas, en los que todos los nodos juegan un papel similar y que se conocen como esquemas distribuidos, en el que la clave que compartirá el grupo de comunicantes se construye de modo colaborativo. Y es precisamente éste el problema sobre el que nos centramos en este trabajo. En [16], los autores tratan, en un contexto genérico, un conjunto de protocolos que extienden de forma natural, a la comunicación en grupos de usuarios, el intercambio de claves de *Diffie-Hellman*. Sin embargo, el número de rondas y mensajes necesarios para iniciar éstos depende del número de usuarios y, dependiendo de dicho número, éste puede volverse demasiado grande.

Nuestro objetivo principal en esta memoria consiste en el estudio y extensión a un contexto más general de un protocolo bastante eficiente en cuanto al número de rondas utilizadas, tan solo dos, introducido por *Burmester* y *Desmedt* en [4], donde las operaciones se llevan a cabo en un grupo cíclico, y que extenderemos seguidamente también para un caso más general siguiendo la idea de [16]. Llevamos a cabo además un estudio de su seguridad y se propone un ataque activo exitoso contra el protocolo original introducido en [4].

El contenido de esta memoria es, por tanto, como sigue. En el segundo capítulo, teniendo en cuenta que el protocolo que estudiaremos es una ex-

tensión del protocolo propuesto por *Diffie* y *Hellamn* ([7]) para dos partes, haremos una recopilación de los protocolos más conocidos de intercambio de claves en grupo de tipo distribuido que extienden dicho protocolo, haciendo una breve explicación del funcionamiento de cada uno de ellos, el número de rondas que se necesitan para llevarse a cabo, el número de mensajes que se intercambian los usuarios del sistema, además de la seguridad que presentan contra ataques de tipo pasivo.

El tercer capítulo se divide en dos partes. En la primera parte, recordaremos el protocolo original de *Burmester* y *Desmedt* introducido sobre grupos en [4]. En la segunda parte, veremos que el protocolo mencionado, puede extenderse a cualquier acción de un semigrupo abeliano sobre un conjunto con estructura de grupo. Veremos algunos ejemplos de aplicación de la extensión del protocolo que proponemos tanto en el caso conmutativo como en el caso no conmutativo.

En el cuarto capítulo, estudiaremos la seguridad del protocolo que hemos propuesto contra posibles ataques pasivos basándonos en los problemas relacionados con el protocolo de intercambio de dos partes de *Diffie-Hellman*, ya que nuestro protocolo es una generalización de éste.

En el quinto capítulo, presentaremos un ataque activo contra el protocolo original introducido en [4], donde utilizaremos una modificación del ataque de tipo de repetición o *Man-In-The-Middle*. El ataque que proponemos consiste en suplantar un miembro del sistema, deteniendo los mensajes enviados por éste, obtener la clave K del grupo en colaboración de los demás usuarios, y finalmente enviarle los valores necesarios al usuario que ha sido reemplazado para que éste tenga la misma clave del sistema. También veremos que este ataque se puede utilizar en el caso del protocolo que generalizamos en el capítulo cuatro.

Finalizamos la memoria exponiendo los principales resultados y problemas que se nos plantean para futuras investigaciones.

Capítulo 2

Protocolos de intercambio de clave en grupo

Durante muchos años, como ya hemos mencionado anteriormente, el problema central de la Criptografía fue el encontrar un modo de transmitir un secreto a través de un canal público inseguro, problema que finalmente resolvieron *Diffie* y *Hellman* en su artículo fundacional [7] dando lugar al posterior nacimiento de la Criptografía de clave pública. El conocido como protocolo de *Diffie-Hellman* proporciona un modo de que dos comunicantes se pongan de acuerdo en una clave común que puede ser usada para una posterior comunicación confidencial.

De este modo, la mayoría de los protocolos de intercambio de clave en grupo que podemos calificar como distribuidos y en los que estamos interesados, son extensiones basadas en el protocolo de *Diffie-Hellman* propuesto, como hemos indicado anteriormente, para asegurar la comunicación entre 2 partes (*2-party Diffie-Hellman*).

Dado que a lo largo de este apartado vamos a recordar algunos de los protocolos de manejo de claves en grupo de tipo distribuido, vamos a recordar dicho protocolo fundacional.

El protocolo de *Diffie-Hellman* está basado en la dificultad computacional para resolver el problema del logaritmo discreto. Es un protocolo que permite a dos participantes, A y B, crear una clave común con el siguiente algoritmo:

1. Los dos participantes eligen, juntos y públicamente, un primo p suficientemente grande y un generador g del grupo multiplicativo $\mathbb{Z}_p^*$.

2. Cada participante genera de forma segura un número entero aleatorio, que será su clave privada. Sean $a, b \leq p-1$ las claves privadas de A y B, respectivamente.

3. En este paso, cada participante calcula su clave pública. En el caso de A su clave será $g^a \pmod p$ y en el caso del participante B, será $g^b \pmod p$.

4. Finalmente, los dos participantes se intercambian las claves públicas y cada uno de ellos, usando su clave privada y la pública del otro, obtiene la clave $K = g^{ab} \pmod p$.

Este protocolo es seguro contra atacantes pasivos, ya que estos conociendo las claves públicas g^a y g^b no serán capaces de calcular g^{ab} siempre que el primo sea lo suficientemente grande, para que el logaritmo discreto correspondiente sea un problema difícil de resolver desde el punto de vista computacional.

Existen muchos intentos de extender el protocolo de *Diffie-Hellman* para un grupo de más de dos participantes. Estas extensiones difieren principalmente en la clave calculada, el número de rondas de comunicación necesarias para establecer la clave y en la naturaleza de los cálculos realizados.

El protocolo de *Ingemarsson, Tang y Wong* [9] es uno de los primeros intentos de extensión del protocolo de *Diffie* y *Hellman* a un contexto de un grupo de dos o más participantes. En este protocolo, los miembros del grupo tienen que estar sincronizados en un ciclo de manera que el miembro U_i se comunica con el miembro U_{i+1}, para $i = 1, \ldots, n$ y U_n se comunica con el miembro U_1. Se asume que los participantes U_i, $i = 1, \ldots, n$, se han puesto de acuerdo previamente sobre un grupo multiplicativo $\mathbb{Z}_p$ donde p es primo, y un generador g del grupo. Los miembros del sistema calculan la clave K del protocolo después de $n-1$ rondas de la siguiente manera:

1. En la *Ronda 1*, cada miembro U_i, $i = 1, \ldots, n$, elige un entero aleatorio $r_i \in \mathbb{Z}_p$, calcula g^{r_i} y se lo envía al miembro U_{i+1}.

2. En la *Ronda m*, $m = 2, \ldots, n-1$ cada miembro calcula $g^{\prod\{r_j : j \in [i-m, i]\}}$, usando lo recibido en la ronda anterior y teniendo en cuenta que los índices de los miembros son módulo n.

Por lo que, al final del protocolo, es decir, después de $n-1$ rondas, todos los los usuarios U_i, $i = 1, \ldots, n$, obtendrán la misma clave

$$K := g^{r_1 r_2 \cdots r_n}.$$

Cabe mencionar, que en este protocolo, es preciso volver a empezar de nuevo en caso de añadir o suprimir un miembro del sistema.

Ejemplo 2.1. *Supongamos que tres usuarios U_1, U_2 y U_3 quieren usar el protocolo anterior. Para ello, eligen en primer lugar un grupo multiplicativo $\mathbb{Z}_p$, un generador g de éste. Dado que el número de usuarios es $n = 3$, entonces obtendrán la misma clave después de $n-1 = 2$ rondas de la siguiente manera:*

- *En la primera ronda, los usuarios U_1, U_2 y U_3 eligen sus claves privadas r_1, r_2 y $r_3 \in \mathbb{Z}_p$, respectivamente. Cada uno de ellos calcula su clave pública que serán g^{r_1}, g^{r_2} y g^{r_3}. El usuario U_1 comparte g^{r_1} con U_2, U_2 comparte g^{r_2} con U_3 y U_3 comparte g^{r_3} con U_1.*
- *En la segunda ronda, cada usuario U_i, $i = 1, 2, 3$, usa su clave privada y lo recibido del usuario anterior en la ronda 1 para calcular $g^{r_i r_{i+1}}$ y compartirlo con el usuario U_{i+1}. De esta manera, al final de esta ronda, el usuario U_1 recibirá $g^{r_2 r_3}$, U_2 recibirá $g^{r_1 r_3}$ y U_3 recibirá $g^{r_1 r_2}$. Usando la clave privada de nuevo, cada usuario U_i, $i = 1, 2, 3$ obtendrá la misma clave $K = g^{r_1 r_2 r_3}$ del protocolo.*

En 1996, *Steiner, Tsudik y Waidner* [24] propusieron otras tres extensiones del protocolo de *Diffie-Hellman* reagrupados bajo el nombre de *Generic Group Diffie Hellman Agreement (GDH)*. Estos tres protocolos GDH.1, GDH.2 y GDH.3 son mejoras del protocolo de *Ingemarsson, Tang y Wong*, y de hecho utilizan la misma idea y los usuarios calculan la misma clave. La diferencia reside en el número de mensajes que recibe cada miembro en cada ronda del protocolo y en el número de rondas necesarias.

El protocolo GDH.1 se lleva a cabo en dos fases, una fase de flujo ascendente y otra fase de flujo descendente. Cada miembro U_i, $i = 1, \ldots, n$ escoge su clave privada $r_i \in \mathbb{Z}_p$ y comienzan el protocolo:

- *Fase de flujo ascendente.* La fase de flujo ascendente la empieza el primer miembro del sistema, U_1. Esta fase permite agrupar todas las

contribuciones de todos los miembros del grupo. El miembro U_1, usa su clave privada y comienza compartiendo $\{g^{r_1}\}$ con el miembro U_2. Después, cada usuario U_i recibe del anterior U_{i-1} un conjunto de $i-1$ valores: $\{g^{r_1}, g^{r_1 r_2}, \ldots, g^{r_1 r_2 \cdots r_{i-1}}\}$, usa su clave privada elevando el último término a r_i, lo añade al conjunto que se convierte en $\{g^{r_1}, g^{r_1 r_2}, \ldots, g^{r_1 r_2 \cdots r_{i-1}}, g^{r_1 r_2 \cdots r_{i-1} r_i}\}$ y lo comparte con el usuario posterior, es decir, con U_{i+1}. En resumen, cada usuario U_i, $i = 1, \ldots, n-1$ comparte

$$\bigcup_{j=1}^{i} \{g^{\prod_{t=1}^{j} r_t}\}$$

con U_{i+1}. La fase de flujo ascendente termina en el miembro U_n, el cual inicia la fase de flujo descendente.

- *Fase de flujo descendente.* En la fase de flujo descendente, cada miembro U_i efectúa i cálculos, uno para obtener la clave del grupo K e $i-1$ para obtener los valores necesarios para que los demás miembros puedan calcular K. El usuario U_n comienza elevando todos los términos del conjunto que ha recibido, a r_n, su clave privada. El último valor del conjunto es la clave del protocolo $K = g^{r_1 r_2 \cdots r_n}$, retira la clave K, es decir, el último término, añade en primer lugar (del conjunto) el término g^{r_n} y comparte el conjunto resultante con U_{n-1}. Después, cada miembro U_i eleva a r_i todos los valores del conjunto, retira el último valor (la clave del grupo), añade el valor g^{r_i} en primera posición y envía el conjunto a U_{i-1}.

El protocolo GDH.1 tiene por tanto $2(n-1)$ rondas y cada miembro U_i, con $i < n$, efectúa $i+1$ cálculos para obtener la clave del protocolo, mientras que el usuario U_n efectúa n cálculos.

Ejemplo 2.2. *Supongamos que tres usuarios U_i, $i = 1, 2, 3$, usan el protocolo GDH.1 para establecer la clave K. Para ello, en primer lugar, los usuarios U_1, U_2 y U_3 eligen las claves privadas r_1, r_2 y r_3 y llevan a cabo el protocolo de la siguiente manera:*

- *En la fase ascendente, el usuario U_1 usa su clave privada y comienza compartiendo $\{g^{r_1}\}$ con el usuario U_2. U_2 recibe el conjunto de U_1, usa su clave privada para obtener el conjunto $\{g^{r_1}, g^{r_1 r_2}\}$ y lo comparte con U_3. El usuario U_3 inicia la fase de flujo descendente.*

- *En esta fase, comienza el miembro U_3 elevando los términos del conjunto recibido a su clave privada r_3, obteniendo de esta manera el conjunto $\{g^{r_1r_3}, g^{r_1r_2r_3}\}$. El último término del conjunto es la clave del sistema: $K = g^{r_1r_2r_3}$, retira dicho valor y coloca g^{r_3} en la primera posición compartiendo con el usuario U_2 el conjunto $\{g^{r_3}, g^{r_1r_3}\}$. U_2 recibe el conjunto, usa su clave privada, obtiene el conjunto $\{g^{r_2r_3}, g^{r_1r_2r_3}\}$, retira el último valor que coincide con la clave del grupo, añade el término g^{r_2} en la primera posición del conjunto y envía $\{g^{r_2}, g^{r_2r_3}\}$ al usuario U_1. El usuario U_1 eleva el último término de este conjunto a su clave y privada y obtiene la misma clave que los usuarios U_1 y U_2: $K = g^{r_1r_2r_3}$.*

El protocolo GDH.2 es una extensión del protocolo GDH.1 con el fin de de disminuir el número de rondas necesarias para obtener la clave K del grupo: n rondas en lugar de $2(n-1)$. Para llevar a cabo GDH.2, como en el protocolo anterior, los usuarios U_i, $i = 1, \ldots, n$, han de elegir sus claves privadas r_i. Este protocolo necesita dos fases, una fase de flujo ascendente parecida a la del protocolo GDH.1 y otra fase en la que el último miembro juega un papel importante en el que comparte el conjunto que él obtiene con todos los demás usuarios del sistema:

- *Fase de flujo ascendente.* La fase de flujo ascendente, como en el caso anterior, se utiliza para agrupar las contribuciones de los participantes. La diferencia es que, durante esta fase, cada miembro U_i,$i = 1, \ldots, n-1$ (empieza el primero U_1), utilizando todas las posibilidades, debe calcular i valores, cada uno de ellos con $i-1$ exponentes, y un valor más con i exponentes. A este último valor, se le denomina *valor cardinal*. Esto es, para $i = 1, \ldots, n-1$, cada usuario comparte

$$\bigcup_{j=1}^{i} \{g^{\prod r_t | t \in [1,i] \wedge t \neq j}\} \cup \{g^{r_1 \ldots r_i}\}$$

  con U_{i+1}.

- *Broadcast.* Cuando el flujo ascendente llega al miembro U_n, éste será el primero en calcular la clave K del protocolo, que coincide con el valor cardinal que él obtiene, eleva el resto de los valores del conjunto a r_n y comparte el conjunto resultante con los demás usuarios. Cada miembro U_i entonces eleva a su clave privada r_i el valor del conjunto que le corresponde obteniendo de esta manera la clave del grupo K.

Ejemplo 2.3. *Supongamos que cuatros usuarios U_i, $i = 1, 2, 3, 4$, desean utilizar el protocolo GDH.2 para obtener una clave en grupo. Para ello, los usuarios U_1, U_2, U_3 y U_4 escogen las claves privadas r_1, r_2, r_3 y r_4 respectivamente y empiezan la fase de flujo ascendente:*

- *Empieza U_1 la fase compartiendo g^{r_1} con U_2. El miembro U_2 recibe el mensaje de U_1 y calcula el conjunto de la siguiente manera: Dado que en este caso $t \in [1, 2]$ y por tanto $j = 1, 2$, la primera posibilidad es $t = 2$ (de esta forma $t \neq 1$) por lo que el primer elemento del conjunto será $\{g^{r_2}\}$ y si $t = 1$ entonces el segundo elemento será g^{r_1}, añadimos el valor cardinal de U_2, $g^{r_1 r_2}$, y obtenemos el conjunto que será compartido con U_3: $\{g^{r_2}, g^{r_1}, g^{r_1 r_2}\}$. En el caso del usuario U_3 tenemos que $t \in [1, 3]$ y $j = 1, 2, 3$, por lo que empezamos por $j = 1$, luego $t = 2, 3$ y obtenemos el primer valor del conjunto (que será compartido con U_4), $g^{r_2 r_3}$, si $j = 2$, entonces el segundo valor es $g^{r_1 r_3}$ y el penúltimo valor es cuando $j = 3$ por lo que $t = 1, 2$, de donde se tiene $g^{r_1 r_2}$, añadiendo el valor cardinal de U_3, $g^{r_1 r_2 r_3}$ se obtiene el conjunto que será compartido con U_4: $\{g^{r_2 r_3}, g^{r_1 r_3}, g^{r_1 r_2}, g^{r_1 r_2 r_3}\}$.*
- *Dado que U_4 es el último usuario, éste será el primero en obtener la clave del grupo y también el encargado de compartir su conjunto con los demás usuarios del sistema. El miembro U_4 recibe el conjunto $\{g^{r_2 r_3}, g^{r_1 r_3}, g^{r_1 r_2}, g^{r_1 r_2 r_3}\}$ de U_3, eleva todos los valores a su clave privada r_4 y obtiene $\{g^{r_2 r_3 r_4}, g^{r_1 r_3 r_4}, g^{r_1 r_2 r_4}, g^{r_1 r_2 r_3 r_4}\}$. El último valor del conjunto anterior, es la clave K del grupo, por lo que U_4 la retira y comparte el conjunto $\{g^{r_2 r_3 r_4}, g^{r_1 r_3 r_4}, g^{r_1 r_2 r_4}\}$ con U_1, U_2 y U_3. Observe que lo necesario, para cada uno de estos usuarios, está ordenado en el conjunto, es decir, lo necesario para U_1 es el primer valor, para U_2 el segundo valor y para U_3 el tercer valor del conjunto. Por lo que, estos usarán cada uno de ellos su clave privada y obtendrán la misma clave que U_4: $K = g^{r_1 r_2 r_3 r_4}$.*

En los dos protocolos GDH.1 y GDH.2, cada miembro U_i debe efectuar $i+1$ cálculos, a excepción del último miembro U_n que lleva a cabo n cálculos, por lo que el coste computacional a la hora de implementar estos protocolos crecerá cuanto más grande sea el número de miembros del sistema. El principal objetivo del último protocolo GDH.3 es de reducir el número de cálculos efectuados por cada miembro. En efecto, cada miembro, excepto el penúltimo

y el último (U_{n-1} y U_n) con un rol especial, tendrán que hacer solamente cuatro cálculos. En primer lugar, todos los usuarios U_i, $i = 1, \ldots, n$, eligen sus claves privadas r_i. El protocolo consiste en cuatro fases, una fase de flujo ascendente, una de *Broadcast*, una de respuesta y otra de *Broadcast*:

- *Fase de flujo ascendente.* En esta fase, se reunirán las contribuciones de todos los usuarios, menos U_{n-1} y U_n, de una manera similar que en el protocolo GDH.1, es decir, para $i = 1, \ldots, n-2$, cada miembro U_i comparte
$$g^{\prod(r_t|t\in[1,i])}$$
con el usuario U_{i+1}.

- *Broadcast.* En esta fase, el usuario U_{n-1} recibe del miembro anterior (U_{n-2}), $g^{r_1 \cdots r_{n-2}}$, usa su clave privada y comparte con todos los miembros del sistema
$$g^{\prod(r_t|t\in[1,n-1])} = g^{r_1 \cdots r_{n-1}}.$$
Observe que en esta fase el usuario U_n ya puede calcular la clave del grupo $K = g^{r_1 \cdots r_n}$ después de recibir el mensaje de U_{n-1}.

- *Respuesta.* En esta fase, todos los usuarios U_i, excepto el último U_n, retiran su contribución del valor recibido de U_{n-1} en la fase anterior, es decir, cada usuario U_i, $i = 1, \ldots n-1$ eleva $g^{r_1 \cdots r_{n-1}}$ a r_i^{-1} y envía el resultado al miembro U_n.

- *Broadcast.* En esta última fase, el usuario U_n reúne todas las contribuciones recibidas en la tercera fase, las eleva a su clave privada r_n y las comparte con todos los usuarios. De esta manera, cada miembro U_i del sistema recibirá $g^{r_1 \cdots r_{i-1} r_{i+1} \cdots r_n}$ y usando r_i obtendrá la clave del grupo $K = g^{r_1 \cdots r_n}$.

Por lo tanto, en el protocolo GDH.3 los miembros necesitan $n+1$ rondas para obtener la misma clave de grupo. Cada usuario U_i, $i = 1, \ldots n-2$, efectúan cuatro cálculos, mientras que el miembro U_{n-1} dos cálculos y el último miembro U_n efectúa n operaciones.

Ejemplo 2.4. *Supongamos que cuatro miembro U_i, $i = 1, 2, 3, 4$, quieren usar el protocolo GDH.3 para establecer una clave en grupo. Cada usuario U_i elige una clave privada r_i y calculan la clave K llevando a cabo las siguientes fases:*

- *En la primera fase de flujo ascendente, empieza el usuario U_1, usa su clave privada, calcula g^{r_1} y lo comparte con U_2. U_2 recibe el mensaje de U_1 y con su clave privada calcula $g^{r_1 r_2}$ y se lo envía al usuario U_3.*

- *En esta fase, el miembro U_3, calcula usando las contribuciones de U_1 y U_2 además de su clave privada, $g^{r_1 r_2 r_3}$ y lo comparte con todos los usuarios del sistema. El miembro U_4, antes de empezar la siguiente fase, ya puede calcular con su clave privada la clave del grupo: $K = g^{r_1 r_2 r_3 r_4}$.*

- *En esta etapa, después de que los usuarios U_i, $i = 1, 2$, reciban el mensaje compartido en la fase anterior, estos además de U_3 deshacen sus contribuciones y comparten el resultado con el último usuario U_4, es decir, U_1 compartirá con U_4: $(g^{r_1 r_2 r_3})^{r_1^{-1}} = g^{r_2 r_3}$, U_2 compartirá: $(g^{r_1 r_2 r_3})^{r_2^{-1}} = g^{r_1 r_3}$ y U_3: $(g^{r_1 r_2 r_3})^{r_3^{-1}} = g^{r_1 r_2}$. Por lo que U_4 recibirá $\{g^{r_2 r_3}, g^{r_1 r_3}, g^{r_1 r_2}\}$.*

- *En la última fase, el usuario U_4 después de recibir los mensajes de los demás usuarios, eleva estos valores a su clave privada: $\{g^{r_2 r_3 r_4}, g^{r_1 r_3 r_4}, g^{r_1 r_2 r_4}\}$ y de esta manera cada usuario U_i, $i = 1, 2, 3$, usa de nuevo, cada uno de ellos, su clave privada y calculan la clave $K = g^{r_1 r_2 r_3 r_4}$.*

Para evitar la ejecución completa de nuevo del protocolo después de suprimir o añadir un miembro, que como hemos ya mencionado anteriormente, era el inconveniente del protocolo de *Ingemarsson, Tang y Wong* [9]; *Steiner, Tsudik* y *Waidner*, desarrollaron un protocolo de intercambio de claves en grupo dinámico llamado *CLIQUES* [23]. *CLIQUES* consiste en un protocolo de intercambio de claves inicial en grupo (*Initial Key Agreement, IKA*), que coincide con el protocolo GDH.2 de [24], y operaciones de un protocolo de intercambio auxiliar (*Auxiliary Key Agreement, AKA*). Estas operaciones son diferentes protocolos que sirven para afrontar distintas situaciones que lleven a cambiar el número de miembros del grupo que han participado en el protocolo de intercambio inicial, como por ejemplo:

- *La adición de un miembro.* El protocolo para llevar a cabo esta operación se aprovecha del papel que juega el usuario U_n en el protocolo de intercambio inicial (GDH.2), asumiendo que se queda con el contenido de los mensajes de la fase de flujo ascendente recibidos en la ronda $n - 1$.

- *La exclusión de un miembro.* Como en el caso anterior, el último usuario repite de nuevo la última ronda generando otra clave privada r'_n, de esta manera el miembro excluido no será capaz de obtener la nueva clave.

Además, *CLIQUES* ofrece dos protocolos eficientes para la adición de múltiplos miembros, y propone varias maneras para la fusión de grupos. Esta última operación se puede considerar como un caso especial de la adición de múltiples miembros, aunque también existe la posibilidad de empezar el protocolo *IKA* (GDH.2) de nuevo. Otra propuesta para la fusión de dos grupos G_1 y G_2 por ejemplo, es usar las correspondientes claves K_1 y K_2 y obtener una nueva clave $K := g^{K_1 K_2}$, aunque esta manera no es segura contra ataques del tipo *Man-In-The-Middle*.

Dos años más tarde, *Steiner, Tsudik* y *Waidner* extendieron *CLIQUES* con otro protocolo inicial de intercambio en grupo llamado *IKA.2* en [25] que se corresponde con el protocolo GDH.3 de [24], y protocolos para refrescar la clave del grupo en caso de algún cambio en éste. Estos protocolos (para añadir un miembro, excluir un miembro, etc.) funcionan de la misma manera que los de *IKA.1* (GDH.2), es decir, un miembro del grupo, que posee los contenidos del mensaje compartido en la última ronda en el protocolo inicial, genera una nueva clave privada y repite esta ronda usando los nuevos valores.

Otra generalización del protocolo de *Diffie-Hellman* es el protocolo de *Steer, Strawczynski, Diffie* y *Wiener* para asegurar los sistemas de teleconferencia de audio ([22]). Todas las operaciones de este protocolo se llevan a cabo en un grupo cíclico G de orden p primo con un generador g, y la clave del protocolo que se obtiene para n usuarios es de la siguiente forma:

$$K = g^{r_n g^{r_{n-1} g^{\cdots g^{r_3 g^{r_1 r_2}}}}}$$

donde r_i son las claves privadas de los usuarios U_i, $i = 1, \ldots, n$, respectivamente. Este protocolo consiste en dos fases con $n-1$ rondas, en la primera fase cada usuario U_i, $i = 1, \ldots, n$ calcula $z_i = g^{r_i}$ y lo comparte con todos los demás usuarios. Por lo que los usuarios U_1 y U_2 serán capaces de calcular la clave del grupo K al terminar la primera fase, suponiendo que $l_2 = l_1 = g^{r_1 r_2}$ y calculando de forma recursiva $l_j = (z_j)^{l_{j-1}}$, $j = 3, \ldots, n$, dicha clave vendrá dada por $K = l_n$. En la segunda fase, hay $n-2$ rondas.

Cada ronda $i = 1, \ldots, n-2$, el cálculo de la clave, por cada usuario U_i, $i = 3, \ldots, n$ está basado en la siguientes variables recursivas:

$$\begin{array}{rcl} \gamma_1 & = & z_1 \qquad = g^{r_1} \\ k_j & = & (\gamma_{j-1})^{r_j} \\ \gamma_j & = & g^{k_j}. \end{array}$$

Después de recibir γ_{i-1} de U_{i-1}, U_i hallará k_i y γ_i, y calculará $l_j = (z_j)^{lj-1}$, $j = i+1, \ldots, n$ como en la fase 1. La clave del protocolo entonces será $K = l_n$ (el usuario U_n obtendrá la clave haciendo el cálculo $K = \gamma_{n-1}^{r_n}$). El total de mensajes compartidos entre los usuarios es $2(n-1)$.

El protocolo de *Steer, Strawczynski, Diffie* y *Wiener* ha sido mejorado en 2001 por *Kim, Perrig* y *Tsudik* [11]. El protocolo mejorado reduce las $n-1$ rondas del protocolo anterior a solamente dos rondas, aunque manteniendo el mismo número de mensajes compartidos.

En 1998, *Becker* y *Wille* [3], introdujeron un nuevo protocolo llamado *Octopus*, el cual consiste en dividir un grupo de n usuarios en cuatro subgrupos (no necesariamente del mismo tamaño). Cada subgrupo elige un representante U_i, $i = 1, 2, 3, 4$, el cual se encargará del control. Sean $\{P_i | i \in I_{U_1}\}$, $\{P_i | i \in I_{U_2}\}$, $\{P_i | i \in I_{U_3}\}$ y $\{P_i | i \in I_{U_4}\}$ los cuatros subgrupos. Los miembros de cada subgrupo establecen una clave K_i con su representante, usando el protocolo de *Diffie-Hellman*, y los representantes de cada subgrupo intercambian las claves K_i de sus respectivos subgrupos entre ellos, para calcular la clave del sistema de la siguiente manera: U_1 y U_2 intercambian sus claves y calculan $g^{K_1 K_2}$. De la misma manera, U_3 y U_4 obtienen $g^{K_3 K_4}$. Posteriormente, U_1 y U_3 intercambian las claves $g^{K_1 K_2}$ y $g^{K_3 K_4}$ obteniendo la clave del sistema $K = g^{g^{K_1 K_2} g^{K_3 K_4}}$. De igual modo, U_2 y U_4 obtienen K. Finalmente, cada representante comparte la información necesaria con cada miembro de su subgrupo, para que todos los P_i puedan calcular la misma clave K. Por ejemplo, para un grupo de 8 usuarios, la clave del sistema será

$$K = g^{g^{g^{r_1 r_2} g^{r3r4}} g^{g^{r5r6} g^{r7r8}}}.$$

Este protocolo, requiere $2n-4$ intercambios de mensajes y 4 rondas para obtener la clave K. En el mismo artículo [3], *Becker* y *Wille*, generalizan la idea del protocolo para el caso en el que $n = 2^d$, con $d \in \mathbb{N}$, proponiendo otro

protocolo llamado *Hypercube*, el cual necesita d rondas para llevarse a cabo. Los autores demostraron la seguridad de los dos protocolos contra ataques pasivos bajo las condiciones del problema de *Diffie-Hellman*.

La idea de *Becker* y *Wille* utilizada en el protocolo *Hypercube* fue mejorada por *Perrig* un año más tarde en [19], con un protocolo que necesita solamente $log_2 n$ rondas para obtener la clave del sistema para n usuarios. En [12], *Kim*, *Perrig* y *Tsudik* extendieron el protocolo de *Perrig* [19] desarrollando técnicas para afrontar los cambios en el sistema (adición de miembros, supresión, etc.).

Todas las generalizaciones del protocolo de *Diffie-Hellman* vistas hasta ahora, necesitan de al menos dos rondas para completarse. Existen algunos protocolos que se llevan a cabo en una sola ronda, pero no todos garantizan la seguridad. El protocolo de *Joux* [10] es el único protocolo de intercambio de claves en grupo, que puede ejecutarse en una única ronda, conocido hasta ahora, garantizando la seguridad contra ataques pasivos. Sin embargo, este protocolo está diseñado solamente para ser utilizado por tres usuarios. La seguridad del protocolo de *Joux* está basado en el problema de *Diffie-Hellman* sobre curvas elípticas.

El protocolo de intercambio de claves en grupo, propuesto por *Joux*, ha sido extendido por varios autores (por ejemplo en [15] y [2]).

Dado que solamente hemos estudiado protocolos sin autentificación, ninguno de estos puede proporcionar seguridad contra ataques activos. Sin embargo, la mayoría de los protocolos garantizan seguridad contra atacantes pasivos bajo las condiciones del problema de *Diffie-Hellman*. *Steiner*, *Tsudik* y *Waidner* probaron en [24] que el problema decisional de *Diffie-Hellman* para n usuarios es seguro si el problema original de *Diffie-Hellman* es seguro. No obstante, no existen demostraciones de seguridad para el protocolo de *Steer, Strawczynski, Diffie* y *Wiener* ([22]) y tampoco para el protocolo de *Perrig* ([19]). *Becker* y *Wille* ([3]), también demostraron que la seguridad del problema de *Diffie-Hellman* implica la seguridad del protocolo *Octopus*.

Inspirados por la extensión del intercambio de *Diffie-Hellman* llevada a cabo en [17] para el caso de acciones de semigrupos, en [16], los autores han llevado a cabo una extensión de los protocolos de intercambio de clave que

fundamentan CLIQUES para el caso de otras estructuras más generales que incluyen ambientes no conmutativos y que nos sirven como inspiración para los resultados que presentamos en los siguientes capítulos.

Capítulo 3

Un protocolo de intercambio de clave basado en acciones sobre un grupo

El objetivo de este capítulo es la introducción de un protocolo de manejo de claves para un conjunto de usuarios, que está basado en la acción de un semigrupo. Como ya se ha comentado anteriormente, la motivación del mismo viene dada, por un lado, por el protocolo introducido por *Burmester* y *Desmedt* en [4] y, por otro, por los criptosistemas de clave pública introducidos por *Maze*, *Monico* y *Rosenthal* en [17]. Comenzaremos pues, recordando el protocolo original introducido sobre grupos en [4].

3.1. El protocolo original para grupos

El protocolo que a continuación recordamos hace uso de técnicas asimétricas y está basado en la dificultad de resolver el problema computacional de *Diffie-Hellman* en $\mathbb{Z}_p$:

Definición 3.1. ***El problema computacional de Diffie-Hellman*** *Sean un primo p, un generador g de $\mathbb{Z}_p$ y dos enteros $x, y \in \mathbb{Z}_p$. El problema Computacional de Diffie-Hellman o CDH consiste en encontrar $x^{\log_g y}$ (mód p), es decir, si $x = g^a$ (mód p) e $y = g^b$ (mód p), encontrar g^{ab} (mód p).*

Los usuarios del sistema eligen un grupo multiplicativo $\mathbb{Z}_p$, con p primo

suficientemente grande y un elemento $g \in \mathbb{Z}_p$ de orden q. Los parámetros (p, g y q) son públicos y accesibles a todas las partes.

Supongamos ahora que n usuarios, U_i con $1 \leq i \leq n$, desean generar una clave común y de forma colaborativa. Entonces dichos usuarios siguen los siguientes pasos:

- **Ronda 1.** Cada usuario U_i, $i = 1, 2, \ldots, n$, elige un elemento aleatorio $r_i \in \mathbb{Z}_q$, que constituirá su clave privada, calcula, usando la misma, $z_i := g^{r_i} \pmod p$, su clave pública, y la comparte con los usuarios U_{i-1} y U_{i+1}). Hay que tener en cuenta que los índices son tomados en ciclo (Figura 3.1), es decir, en el caso del usuario U_1, el anterior es U_n, y para U_n el posterior es U_1.

- **Ronda 2.** Cada usuario U_i, $i = 1, 2, \ldots, n$, tras recibir las claves públicas z_{i+1} y z_{i-1} de los usuarios U_{i+1} y U_{i-1}, respectivamente, usa su clave privada r_i y calcula $X_i := ((z_{i+1})(z_{i-1})^{-1})^{r_i} \pmod p$. Los X_i son hechos públicos.

- **Construcción de la clave** Cada usuario U_i, $i = 1, 2, \ldots, n$, calcula la clave:
$$K_i := (z_{i-1})^{nr_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i-2} \pmod p.$$

El siguiente resultado, que aparece en [4], asegura el buen funcionamiento del protocolo.

Lema 3.2. *El protocolo* 1 *es de intercambio de clave en grupo*[1]*, esto es, si todos los usuarios siguen el protocolo descrito, todos obtendrán la misma clave que viene dada por:*

$$K \equiv g^{r_1r_2+r_2r_3+\ldots+r_nr_1} \pmod p.$$

En la siguiente sección veremos un resultado y cuya demostración, en un contexto más general, asegura, como hemos dicho antes el buen funcionamiento del protocolo en este caso. Los autores afirman en [4] que este resultado se puede extender a cualquier grupo finito de orden un primo p.

[1]La clave calculada por los cuatro usuarios de la Figura 3.1, en este caso, será $K \equiv g^{r_1r_2+r_2r_3+r_3r_4+r_4r_1} \pmod p$.

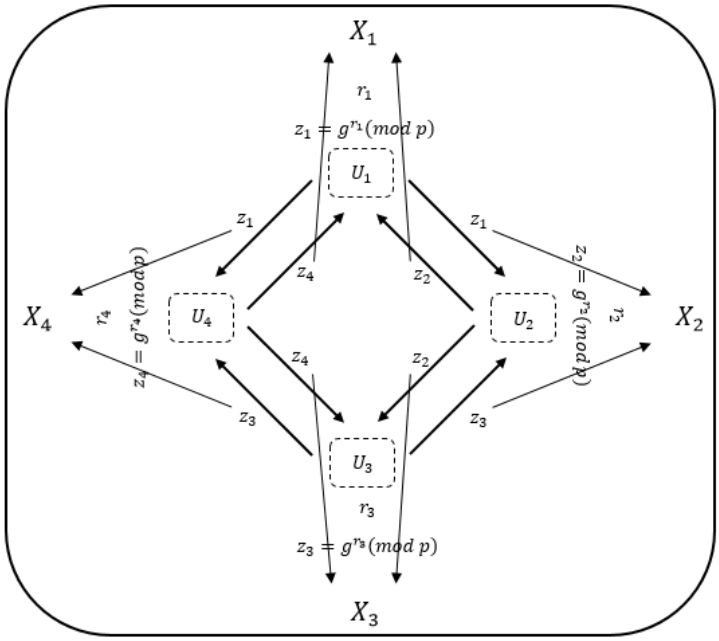

Figura 3.1: El protocolo de *Burmester* y *Desmedt* para 4 usuarios.

3.2. Un protocolo basado en acciones sobre un grupo

Veamos como el protocolo anterior puede extenderse a cualquier acción de un semigrupo abeliano sobre un conjunto con estructura de grupo. De este modo, el protocolo anterior puede verse como un caso particular en el que el semigrupo multiplicativo $\mathbb{Z}$ actúa sobre un grupo finito.

En nuestro contexto general, los usuarios U_i, $i = 1, \ldots, n$, deben ponerse de acuerdo sobre un semigrupo abeliano S, un conjunto finito X con estructura de grupo, y una acción de S sobre X, $\phi : S \times X \to X$, tal que $\phi(s, xx') = \phi(s, x)\phi(s, x')$, donde $x, x' \in X$ y $s \in S$ y además que $\phi(s, e) = e$, siendo s cualquier elemento de S y e el elemento neutro de X. Recordemos también que para todo $x \in X$ y $s, s' \in S$ tenemos que $\phi(ss', x) = \phi(s, \phi(s', x))$. Los pasos que definen la construcción de forma colaborativa de nuestro protocolo basado en la acción ϕ son los siguientes:

- **Ronda 1.** Cada usuario U_i, $i = 1, \ldots, n$, escoge un elemento privado

$s_i \in S$ y comparte $z_i = \phi(s_i, x)$.

- **Ronda 2.** Cada usuario U_i, $i = 1, \ldots, n$, calcula $X_i = \phi(s_i, z_{i+1} z_{i-1}^{-1})$ y lo comparte con los demás usuarios.
- **Recuperación de la clave** Cada usuario U_i, $i = 1, \ldots, n$, calcula

$$K_i = \phi(s_i, z_{i-1})^n X_i^{n-1} X_{i+1}^{n-2} \ldots X_{i-2}$$

A continuación se muestran algunos casos de aplicación del protocolo anterior mediante ejemplos de acciones que podemos utilizar.

Ejemplos. 1. Sea S el semigrupo $(\mathbb{Z}, \cdot)$ de los enteros, un grupo cíclico G, la acción $\phi : \mathbb{Z} \times G \to G$ definida por $\phi(s, g) = g^s$ con $g \in G$ y $s \in \mathbb{Z}$. Para el caso en el que $G = \mathbb{Z}_p$, con p primo, es la acción introducida en [4] en el protocolo original y puede observarse claramente que $\phi(s, g)\phi(s, g') = \phi(s, gg')$ para cualesquiera $g, g' \in \mathbb{Z}_p$ y $s \in \mathbb{Z}$. La clave pública de cada U_i es $z_i = g^{s_i}$, tal y como ya se indicó anteriormente y la clave calculada por el usuario U_i, $i = 1, \ldots, n$, al final del protocolo en dicho caso original es

$$K_i = (z_{i-1})^{n s_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i-2} \quad (\text{mód } p).$$

2. Los autores en [4] indican que el protocolo original puede extenderse a cualquier grupo abeliano, lo cual, a su vez, es una consecuencia de nuestra propia definición del protocolo basado en una acción puesto que cualquier grupo abeliano tiene una estructura de $\mathbb{Z}$-módulo inducida por la acción $\phi : \mathbb{Z} \times G \to G$ definida por $\phi(n, g) = g * \overset{(n)}{\cdots} \cdots * g$.

   Como caso particular de esta situación, sea E el conjunto de puntos de una curva elíptica. Podemos considerar E como un $\mathbb{Z}$-módulo, tal y como antes hemos indicado

$$\begin{array}{rccl} \phi: & \mathbb{Z} \times E & \to & E \\ & (d, P) & \to & dP = \phi(d, P) \end{array}$$

   con $d \in \mathbb{Z}$ y $P \in E$. En este caso, la clave privada de cada usuario U_i es d_i y la correspondiente clave pública es $z_i = \phi(d_i, P) = d_i P$. La clave común obtenida tras la ejecución del protocolo es el punto de la curva E dado por:

$$K_i = n r_i(z_{i-1}) + (n-1) X_i + (n-2) X_{i+1} \cdots X_{i-2}.$$

3. Otro ejemplo que podemos utilizar para este protocolo, es la acción estudiada en [5] que da lugar a diversos criptosistemas de clave pública, pero que aquí podemos utilizar para obtener un ejemplo en un contexto no conmutativo. Consideramos el anillo:

$$E_p^{(m)} = \left\{ [a_{ij}] \in \mathrm{Mat}_{m\times m}(\mathbb{Z}) \mid a_{ij} \in \mathbb{Z}_{p^i} \text{ si } i \leq j, \text{ y } a_{ij} \in p^{i-j}\mathbb{Z}_{p^j} \text{ si } i > j \right\}$$

con las operaciones de suma y multiplicación definidas de la siguiente manera:

$$\begin{aligned} \left[a_{ij}\right] + \left[b_{ij}\right] &= \left[(a_{ij} + b_{ij}) \text{ mód } p^i\right], \\ \left[a_{ij}\right] \cdot \left[b_{ij}\right] &= \left[\left(\sum_{k=1}^{m} a_{ik}b_{kj}\right) \text{ mód } p^i\right]. \end{aligned}$$

donde $\mathrm{Mat}_{m\times m}(\mathbb{Z})$ denota el conjunto de las matrices de m columnas y m filas con coeficientes en $\mathbb{Z}$, y $p^r\mathbb{Z}_{p^s}$ es el conjunto $\{p^r u \mid u \in \mathbb{Z}_{p^s}\}$ donde r y s son enteros positivos. Este anillo es claramente no conmutativo y su producto definen una acción del semigrupo multiplicativo $E_p^{(m)}$ sobre el conjunto $\mathbb{Z}_p \times \mathbb{Z}_{p^2} \times \cdots \times \mathbb{Z}_{p^m}$:

$$\begin{array}{rccccc} \phi: & E_p^{(m)} & \times & (\mathbb{Z}_p \times \mathbb{Z}_{p^2} \times \cdots \times \mathbb{Z}_{p^m}) & \to & \mathbb{Z}_p \times \mathbb{Z}_{p^2} \times \cdots \times \mathbb{Z}_{p^m} \\ & (A & , & b) & \to & Ab = \phi(A,b). \end{array}$$

Sin embargo, para asegurar que nuestro protocolo funcione, necesitamos que los elementos del semigrupo conmuten. Esto se puede conseguir, considerando elementos del semigrupo $E_p^{(m)}$ que son de la forma $\sum_{i=0}^{r} C_i M^i$, tal que para todo $i = 0, \ldots, r$, C_i está en el centro de $E_p^{(m)}$ que viene dado por (Véase [6]):

$$Z(E_p^{(m)}) = \left\{ [a_{ij}] \in E_p^{(m)} | a_{ii} = \sum_{j=0}^{i-1} p^j u_j, \text{con } u_j \in \mathbb{Z}_p \text{ y } a_{ij} = 0 \text{ si } i \neq j \right\},$$

y $M \in E_p^{(m)}$ es un elemento público tal que el conjunto de sus potencias sea suficientemente grande. En otras palabras, usaremos el subsemigrupo $Z[M]$ de $E_p^{(m)}$, donde Z es el centro de $E_p^{(m)}$.

En el siguiente lema veremos que, efectivamente, el protocolo anterior es un protocolo de grupo de intercambio de claves, esto es, que todos los miembros del sistema calcularán la misma clave.

Lema 3.2.1. *Si todos los U_i $i = 1, \ldots, n$ siguen los pasos del protocolo descrito, obtendrán la misma clave K que viene dada por:*

$$K = \phi(s_1 s_2, x)\phi(s_2 s_3, x) \ldots \phi(s_n s_1, x).$$

Demostración. Vamos a demostrar el lema por inducción sobre n. Comenzamos considerando el caso $n = 3$. En este caso, conocidas las claves públicas, los usuarios U_1, U_2 y U_3 calculan:

$$\begin{array}{lll} U_1: & z_1 = \phi(s_1, x) & X_1 = \phi(s_1, (z_2 z_3^{-1})) \\ U_2: & z_2 = \phi(s_2, x) & X_2 = \phi(s_2, (z_3 z_1^{-1})) \\ U_3: & z_3 = \phi(s_3, x) & X_3 = \phi(s_3, (z_1 z_2^{-1})) \end{array}$$

y obtienen sus claves K_i sabiendo que

$$K_i = \phi(s_i, z_{i-1})^n X_i^{n-1} X_{i+1}^{n-2} \ldots X_{i-2}$$

Las claves construídas por cada uno de ellos son entonces las siguientes:

- La clave obtenida por U_1 es:

$$\begin{aligned} K_1 &= \phi(s_1, z_3)^3 X_1^2 X_2 \\ &= \phi(s_1, z_3)^3 \phi(s_1, \phi(s_2, x)\phi(s_3, x)^{-1})^2 \phi(s_2, \phi(s_3, x)\phi(s_1, x)^{-1}) \\ &= \phi(s_1, \phi(s_3, x))^3 \phi(s_1, \phi(s_2, x)\phi(s_3, x)^{-1})^2 \phi(s_2, \phi(s_3, x)\phi(s_1, x)^{-1}) \\ &= \phi(s_1 s_3, x)^3 (\phi(s_1 s_2, x)^2 (\phi(s_1 s_3, x)^{-1}))^2 \phi(s_2 s_3, x)\phi(s_1 s_2, x)^{-1} \\ &= \phi(s_1 s_2, x)\phi(s_2 s_3, x)\phi(s_3 s_1, x). \end{aligned}$$

- La clave obtenida por U_2:

$$\begin{aligned} K_2 &= \phi(s_2, z_1)^3 X_2^2 X_3 \\ &= \phi(s_2, z_1)^3 \phi(s_2, \phi(s_3, x)\phi(s_1, x)^{-1})^2 \phi(s_3, \phi(s_1, x)\phi(s_2, x)^{-1}) \\ &= \phi(s_2, \phi(s_1, x))^3 \phi(s_2, \phi(s_3, x)\phi(s_1, x)^{-1})^2 \phi(s_3, \phi(s_1, x)\phi(s_2, x)^{-1}) \\ &= \phi(s_2 s_1, x)^3 (\phi(s_3 s_2, x)\phi(s_2 s_1, x)^{-1})^2 \phi(s_3 s_1, x)\phi(s_2 s_3, x)^{-1} \\ &= \phi(s_1 s_2, x)\phi(s_2 s_3, x)\phi(s_3 s_1, x). \end{aligned}$$

- La clave que obtiene U_3:

$$\begin{aligned} K_3 &= \phi(s_3,z_2)^3X_3^2X_1 \\ &= \phi(s_3,z_2)^3\phi(s_3,\phi(s_1,x)\phi(s_2,x)^{-1})^2\phi(s_1,\phi(s_2,x)\phi(s_3,x)^{-1}) \\ &= \phi(s_3,\phi(s_2,x))^3\phi(s_3,\phi(s_1,x)\phi(s_2,x)^{-1})^2\phi(s_1,\phi(s_2,x)\phi(s_3,x)^{-1}) \\ &= \phi(s_3s_2,x)^3(\phi(s_1s_3,x)\phi(s_3s_2,x)^{-1})^2\phi(s_1s_2,x)\phi(s_3s_1,x)^{-1} \\ &= \phi(s_1s_2,x)\phi(s_2s_3,x)\phi(s_3s_1,x). \end{aligned}$$

En este caso hemos aplicado que $\phi(s,x^{-1}) = \phi(s,x)^{-1}$ para cualquiera $r \in S$ y $x \in X$, que es consecuencia de las dos propiedades satisfechas por la acción ϕ:

$$\begin{aligned} e &= \phi(s,e) \\ &= \phi(s,xx^{-1}) \\ &= \phi(s,x)\phi(s,x^{-1}) \end{aligned}$$

Lo cual significa que $\phi(s,x^{-1})$ es la inversa de $\phi(s,x)$, esto es, $\phi(s,x^{-1}) = \phi(s,x)^{-1}$.

De este modo, el resultado es cierto para $n = 3$. Supongamos cierta la hipótesis para n, esto es, los n usuarios siguen el protocolo, es decir, calculan:

$$\begin{array}{rcrcl;{\quad}rcl} U_1 & : & z_1 & = & \phi(s_1,x) & ; & \overline{X}_1 & = & \phi(s_1,(z_2z_n^{-1})) \\ U_2 & : & z_2 & = & \phi(s_2,x) & ; & X_2 & = & \phi(s_2,(z_3z_1^{-1})) \\ \vdots & & & & & & & & \\ U_{n-1} & : & z_{n-1} & = & \phi(s_{n-1},x) & ; & X_{n-1} & = & \phi(s_{n-1},(z_nz_{n-2}^{-1})) \\ U_n & : & z_n & = & \phi(s_n,x) & ; & \overline{X}_n & = & \phi(s_n,(z_1z_{n-1}^{-1})) \end{array}$$

y obtienen la misma clave (observemos que los únicos X_i distintos a los que se calcularán para $n+1$ usuarios serán X_1 y X_n, de ahí la notación de éstos). Lo demostraremos para el primer usuario (U_1), ya que para los demás será de forma similar. Nuestra hipótesis de inducción es entonces:

$$\phi(s_1,z_n)^n\overline{X}_1^{n-1}X_2^{n-2}\ldots X_{n-1} = K = \phi(s_1s_2,x)\phi(s_2s_3,x)\ldots\phi(s_ns_1,x). \quad (1)$$

Vamos a demostrar que la clave construída por $n+1$ usuarios también será la misma. De este modo, cada usuario calcula:

$$\begin{array}{rcrcl;{;}rcl}
U_1 & : & z_1 & = & \phi(s_1,x) & ; & X_1 & = & \phi(s_1,(z_2z_{n+1}^{-1})) \\
U_2 & : & z_2 & = & \phi(s_2,x) & ; & X_2 & = & \phi(s_2,(z_3z_1^{-1})) \\
\vdots & & & & & & & & \\
U_{n-1} & : & z_{n-1} & = & \phi(s_{n-1},x) & ; & X_{n-1} & = & \phi(s_{n-1},(z_nz_{n-2}^{-1})) \\
U_n & : & z_n & = & \phi(s_n,x) & ; & X_n & = & \phi(s_n,(z_{n+1}z_{n-1}^{-1})) \\
U_{n+1} & : & z_{n+1} & = & \phi(s_{n+1},x) & ; & X_{n+1} & = & \phi(s_{n+1},(z_1z_n^{-1}))
\end{array}$$

Entonces la clave obtenida por U_1 es:

$$\phi(s_1,z_{n+1})^{n+1}X_1^nX_2^{n-1}\dots X_{n-1}^2X_n. \qquad (2)$$

De la hipótesis de inducción (1) se tiene que:

$$X_2^{n-2}X_3^{n-3}\dots X_{n-1} = K(\phi(s_1,z_n)^n)^{-1}(\overline{X_1}^{n-1})^{-1},$$

sustituyendo en (2) se tiene que:

$$\phi(s_1,z_{n+1})^{n+1}X_1^n\phi(s_1s_2,x)\phi(s_2s_3,x)\dots\phi(s_ns_1,x)(\phi(s_1,z_n)^n)^{-1}(\overline{X_1}^{n-1})^{-1}X_2$$

$$\begin{aligned}
X_3\dots X_{n-1}X_n &= \phi(s_1,z_{n+1})^{n+1}\phi(s_1,(z_2z_{n+1}^{-1}))^n\phi(s_1s_2,x)\phi(s_2s_3,x) \\
&\quad \dots\phi(s_{n-1}s_n,x)\phi(s_ns_1,x)(\phi(s_1,z_n)^n)^{-1} \\
&\quad (\phi(s_1,(z_2z_n^{-1}))^{n-1})^{-1}\phi(s_2,(z_3z_1^{-1}))\phi(s_3,(z_4z_2^{-1}))\dots \\
&\quad \phi(s_{n-1},(z_nz_{n-2}^{-1}))\phi(s_n,(z_{n+1}z_{n-1}^{-1})) \\
&= \phi(s_1s_{n+1},x)^{n+1}(\phi(s_1s_2,x)\phi(s_1s_{n+1})^{-1})^n\phi(s_1s_2,x) \\
&\quad \phi(s_2s_3,x)\dots\phi(s_{n-1}s_n,x)\phi(s_ns_1,x)(\phi(s_1s_n,x)^n)^{-1} \\
&\quad ((\phi(s_1s_2,x)\phi(s_1s_n,x)^{-1})^{n-1})^{-1}\phi(s_2s_3,x)\phi(s_2s_1,x)^{-1} \\
&\quad \phi(s_3s_4,x)\phi(s_3s_2,x)^{-1}\dots\phi(s_{n-1}s_n,x)\phi(s_{n-1}s_{n-2},x)^{-1} \\
&\quad \phi(s_ns_{n+1},x)\phi(s_ns_{n-1},x)^{-1} \\
&= \phi(s_1s_{n+1},x)\phi(s_1s_2,x)\phi(s_2s_3,x) \\
&\quad \dots\phi(s_ns_1,x)^0\phi(s_{n-1}s_n,x)\phi(s_ns_{n+1},x) \\
&= \phi(s_1s_2,x)\phi(s_2s_3,x)\dots\phi(s_{n-1}s_n,x)\phi(s_ns_{n+1},x)\phi(s_{n+1}s_1,x).
\end{aligned}$$

■

A continuación exponemos un ejemplo de aplicación del protocolo en el caso del grupo de puntos de una curva elíptica. La corrección del protocolo en el caso general de un grupo abeliano es consecuencia del lema que acabamos de demostrar.

Ejemplo. Supongamos que un grupo de cuatro usuarios elige el cuerpo finito $\mathbb{F}_p$ con $p = 83$, la curva elíptica $E : y^2 = x^3 + 36x + 82$ sobre dicho cuerpo y una acción del cuerpo sobre el conjunto de puntos de la curva elíptica dada por $\phi(d_i, P) = d_iP$ siendo $P = (10, 23)$ un punto de $E(\mathbb{F}_{83})$. Cada usuario U_i, $1 \leq i \leq 4$, escoge su propio d_i y procede de la siguiente forma:

- U_1 elige $d_1 = 13$ y calcula $z_1 = \phi(d_1, P) = d_1P = 13 \cdot (10, 23) = (55, 14)$.
- U_2 elige $d_2 = 21$ y calcula $z_2 = \phi(d_2, P) = d_2P = 21 \cdot (10, 23) = (76, 63)$.
- U_3 elige $d_3 = 47$ y calcula $z_3 = \phi(d_3, P) = d_3P = 47 \cdot (10, 23) = (22, 20)$.
- U_4 elige $d_4 = 52$ y calcula $z_4 = \phi(d_4, P) = d_4P = 52 \cdot (10, 23) = (25, 67)$.

El siguiente paso es obtener los X_i por parte de cada usuario U_i:

- $X_1 = \phi(d_1, z_2 - z_4) = d_1(z_2 - z_4) = 13 \cdot ((76, 63) + (25, -67)) = (31, 69)$.
- $X_2 = \phi(d_2, z_3 - z_1) = d_2(z_3 - z_1) = 21 \cdot ((22, 20) + (55, -14)) = (56, 29)$.
- $X_3 = \phi(d_3, z_4 - z_2) = d_3(z_4 - z_2) = 47 \cdot ((25, 67) + (76, -63)) = (10, 60)$.
- $X_4 = \phi(d_4, z_1 - z_3) = d_4(z_1 - z_3) = 52 \cdot ((55, 14) + (22, -20)) = (28, 47)$.

Cada usuario calcula su clave:

$$\begin{aligned} K_1 &= 4d_1z_4 + 3X_1 + 2X_2 + X_3 \\ &= 4 \cdot 13 \cdot (25, 67) + 3 \cdot (31, 69) + 2 \cdot (56, 29) + (10, 60) \\ &= (51, 21) \end{aligned}$$

$$\begin{aligned} K_2 &= 4d_2z_1 + 3X_2 + 2X_3 + X_4 \\ &= 4 \cdot 21 \cdot (55, 14) + 3 \cdot (56, 29) + 2 \cdot (10, 60) + (28, 47) \\ &= (51, 21) \end{aligned}$$

$$\begin{aligned} K_3 &= 4d_3z_2 + 3X_3 + 2X_4 + X_1 \\ &= 4 \cdot 47 \cdot (76, 63) + 3 \cdot (10, 60) + 2 \cdot (28, 47) + (31, 69) \\ &= (51, 21) \end{aligned}$$

$$\begin{aligned} K_4 &= 4d_4z_3 + 3X_4 + 2X_1 + X_2 \\ &= 4 \cdot 52 \cdot (22, 20) + 3 \cdot (28, 47) + 2 \cdot (31, 69) + (56, 29) \\ &= (51, 21) \end{aligned}$$

Observamos pues, que efectivamente los cuatro usuarios obtendrán la misma clave $K = (51, 21)$.

El ejemplo que se expone a continuación extiende el uso del protocolo que introducimos en este trabajo a un ambiente no conmutativo, más concretamente, al caso de un módulo sobre un anillo no conmutativo, cuya estructura viene inducida por la acción definida anteriormente.

Ejemplo. Sean cuatro usuario que desean utilizar el protocolo usando la acción del semigrupo multiplicativo $E_p^{(m)}$ sobre el conjunto $\mathbb{Z}_p \times \mathbb{Z}_{p^2} \times \cdots \times \mathbb{Z}_{p^m}$. Supongamos que los usuarios eligen el primo $p = 7$ y el parámetro $m = 3$. Para llevar a cabo el protocolo, los usuarios acuerdan la matriz $M \in E_7^{(3)}$, sea

$$M = \begin{pmatrix} 5 & 4 & 3 \\ 21 & 35 & 40 \\ 147 & 245 & 253 \end{pmatrix},$$

cada usuario U_i, $i = 1, 2, 3, 4$, para conseguir que $r_i r_j = r_j r_i$, con $i, j = 1, 2, 3, 4$ y $i \neq j$, calcula su clave privada r_i de la siguiente manera:

- Para U_1: El usuario U_1 escoge las matrices C_i, con $i = 0, 1, 2$, del centro de $E_7^{(3)}$ ($Z(E_7^{(3)})$):

$$C_0 = \begin{pmatrix} 7^0 \cdot 1 & 0 & 0 \\ 0 & 7^0 \cdot 1 + 7^1 \cdot 3 & 0 \\ 0 & 0 & 7^0 \cdot 1 + 7^1 \cdot 3 + 7^2 \cdot 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 22 & 0 \\ 0 & 0 & 267 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 7^0 \cdot 2 & 0 & 0 \\ 0 & 7^0 \cdot 2 + 7^1 \cdot 5 & 0 \\ 0 & 0 & 7^0 \cdot 2 + 7^1 \cdot 5 + 7^2 \cdot 4 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 37 & 0 \\ 0 & 0 & 233 \end{pmatrix},$$

$$C_2 = \begin{pmatrix} 7^0 \cdot 3 & 0 & 0 \\ 0 & 7^0 \cdot 3 + 7^1 \cdot 1 & 0 \\ 0 & 0 & 7^0 \cdot 3 + 7^1 \cdot 1 + 7^2 \cdot 6 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 304 \end{pmatrix}$$

y calcula el polinomio: $f_1(X) = C_0 + C_1 X + C_2 X^2$. De esta manera, la clave privada de U_1 será:

$$s_1 = f_1(M) = \begin{pmatrix} 2 & 5 & 1 \\ 14 & 1 & 4 \\ 196 & 245 & 76 \end{pmatrix}.$$

- Para U_2: El usuario U_2 escoge las matrices C_i, con $i = 0, 1, 2, 3$, del centro de $E_7^{(3)}$ ($Z(E_7^{(3)})$):

$$C_0 = \begin{pmatrix} 7^0 \cdot 1 & 0 & 0 \\ 0 & 7^0 \cdot 1 + 7^1 \cdot 3 & 0 \\ 0 & 0 & 7^0 \cdot 1 + 7^1 \cdot 3 + 7^2 \cdot 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 22 & 0 \\ 0 & 0 & 218 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 7^0 \cdot 4 & 0 & 0 \\ 0 & 7^0 \cdot 4 + 7^1 \cdot 5 & 0 \\ 0 & 0 & 7^0 \cdot 4 + 7^1 \cdot 5 + 7^2 \cdot 1 \end{pmatrix} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 39 & 0 \\ 0 & 0 & 88 \end{pmatrix},$$

$$C_2 = \begin{pmatrix} 7^0 \cdot 0 & 0 & 0 \\ 0 & 7^0 \cdot 0 + 7^1 \cdot 0 & 0 \\ 0 & 0 & 7^0 \cdot 0 + 7^1 \cdot 0 + 7^2 \cdot 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} 7^0 \cdot 3 & 0 & 0 \\ 0 & 7^0 \cdot 3 + 7^1 \cdot 1 & 0 \\ 0 & 0 & 7^0 \cdot 3 + 7^1 \cdot 1 + 7^2 \cdot 6 \end{pmatrix} = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 34 & 0 \\ 0 & 0 & 279 \end{pmatrix}$$

y halla el polinomio: $f_2(X) = C_0 + C_1X + C_2X^2 + C_3X^3$. De esta manera, la clave privada de U_2 será:

$$s_2 = f_2(M) = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 36 & 15 \\ 147 & 294 & 53 \end{pmatrix}.$$

- Para U_3: El usuario U_3 elige las matrices C_i, con $i = 0, 1, 2$, del centro de $E_7^{(3)}$ ($Z(E_7^{(3)})$):

$$C_0 = \begin{pmatrix} 7^0 \cdot 5 & 0 & 0 \\ 0 & 7^0 \cdot 5 + 7^1 \cdot 4 & 0 \\ 0 & 0 & 7^0 \cdot 5 + 7^1 \cdot 4 + 7^2 \cdot 2 \end{pmatrix} = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 33 & 0 \\ 0 & 0 & 131 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 7^0 \cdot 0 & 0 & 0 \\ 0 & 7^0 \cdot 0 + 7^1 \cdot 0 & 0 \\ 0 & 0 & 7^0 \cdot 0 + 7^1 \cdot 0 + 7^2 \cdot 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$C_2 = \begin{pmatrix} 7^0 \cdot 0 & 0 & 0 \\ 0 & 7^0 \cdot 0 + 7^1 \cdot 3 & 0 \\ 0 & 0 & 7^0 \cdot 0 + 7^1 \cdot 3 + 7^2 \cdot 6 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 21 & 0 \\ 0 & 0 & 315 \end{pmatrix}$$

y obtiene el polinomio: $f_3(X) = C_0 + C_1X + C_2X^2$. De esta manera, la clave privada de U_3 será:

$$s_3 = f_3(M) = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 33 & 7 \\ 0 & 0 & 54 \end{pmatrix}.$$

- Para U_4: El usuario U_4 escoge las matrices C_i, con $i = 0, 1, 2, 3$, del centro de $E_7^{(3)}$ ($Z(E_7^{(3)})$):

$$C_0 = \begin{pmatrix} 7^0 \cdot 4 & 0 & 0 \\ 0 & 7^0 \cdot 4 + 7^1 \cdot 2 & 0 \\ 0 & 0 & 7^0 \cdot 4 + 7^1 \cdot 2 + 7^2 \cdot 6 \end{pmatrix} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 18 & 0 \\ 0 & 0 & 312 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 7^0 \cdot 3 & 0 & 0 \\ 0 & 7^0 \cdot 3 + 7^1 \cdot 1 & 0 \\ 0 & 0 & 7^0 \cdot 3 + 7^1 \cdot 1 + 7^2 \cdot 5 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 255 \end{pmatrix},$$

$$C_2 = \begin{pmatrix} 7^0 \cdot 0 & 0 & 0 \\ 0 & 7^0 \cdot 0 + 7^1 \cdot 1 & 0 \\ 0 & 0 & 7^0 \cdot 0 + 7^1 \cdot 1 + 7^2 \cdot 4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 203 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} 7^0 \cdot 6 & 0 & 0 \\ 0 & 7^0 \cdot 6 + 7^1 \cdot 2 & 0 \\ 0 & 0 & 7^0 \cdot 6 + 7^1 \cdot 2 + 7^2 \cdot 1 \end{pmatrix} = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 69 \end{pmatrix}$$

y calcula el polinomio: $f_4(X) = C_0 + C_1X + C_2X^2 + C_3X^3$. De esta manera, la clave privada de U_4 será:

$$s_4 = f_4(M) = \begin{pmatrix} 6 & 3 & 6 \\ 28 & 46 & 45 \\ 0 & 49 & 6 \end{pmatrix}.$$

Teniendo las claves privadas, sea

$$x = \begin{pmatrix} 4 \\ 25 \\ 325 \end{pmatrix} \in \mathbb{Z}_7 \times \mathbb{Z}_{7^2} \times \mathbb{Z}_{7^3},$$

las claves públicas de cada usuario son:

- La clave pública de U_1:

$$z_1 = \phi(s_1, x) = \begin{pmatrix} 2 & 5 & 1 \\ 14 & 1 & 4 \\ 196 & 245 & 76 \end{pmatrix} \begin{pmatrix} 4 \\ 25 \\ 325 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \\ 53 \end{pmatrix}.$$

- La clave pública de U_2:

$$z_2 = \phi(s_2, x) = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 36 & 15 \\ 147 & 294 & 53 \end{pmatrix} \begin{pmatrix} 4 \\ 25 \\ 325 \end{pmatrix} = \begin{pmatrix} 3 \\ 42 \\ 124 \end{pmatrix}.$$

- La clave pública de U_3:

$$z_3 = \phi(s_3, x) = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 33 & 7 \\ 0 & 0 & 54 \end{pmatrix} \begin{pmatrix} 4 \\ 25 \\ 325 \end{pmatrix} = \begin{pmatrix} 6 \\ 13 \\ 57 \end{pmatrix}.$$

- La clave pública de U_4:

$$z_4 = \phi(s_4, x) = \begin{pmatrix} 6 & 3 & 6 \\ 28 & 46 & 45 \\ 0 & 49 & 6 \end{pmatrix} \begin{pmatrix} 4 \\ 25 \\ 325 \end{pmatrix} = \begin{pmatrix} 5 \\ 11 \\ 88 \end{pmatrix}.$$

Conociendo las claves públicas, cada usuario U_i procede a calcular el correspondiente X_i:

- El usuario U_1:

$$X_1 = \phi(s_1, z_2 - z_4) = \begin{pmatrix} 2 & 5 & 1 \\ 14 & 1 & 4 \\ 196 & 245 & 76 \end{pmatrix} \left(\begin{pmatrix} 3 \\ 42 \\ 124 \end{pmatrix} - \begin{pmatrix} 5 \\ 11 \\ 88 \end{pmatrix} \right) = \begin{pmatrix} 5 \\ 0 \\ 335 \end{pmatrix}$$

- El usuario U_2:

$$X_2 = \phi(s_2, z_3 - z_1) = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 36 & 15 \\ 147 & 294 & 53 \end{pmatrix} \left(\begin{pmatrix} 6 \\ 13 \\ 57 \end{pmatrix} - \begin{pmatrix} 3 \\ 9 \\ 53 \end{pmatrix} \right) = \begin{pmatrix} 4 \\ 8 \\ 114 \end{pmatrix}$$

- El usuario U_3:

$$X_3 = \phi(s_3, z_4 - z_2) = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 33 & 7 \\ 0 & 0 & 54 \end{pmatrix} \left(\begin{pmatrix} 5 \\ 11 \\ 88 \end{pmatrix} - \begin{pmatrix} 3 \\ 42 \\ 124 \end{pmatrix} \right) = \begin{pmatrix} 3 \\ 48 \\ 114 \end{pmatrix}$$

- El usuario U_4:

$$X_4 = \phi(s_4, z_1 - z_3) = \begin{pmatrix} 6 & 3 & 6 \\ 28 & 46 & 45 \\ 0 & 49 & 6 \end{pmatrix} \left(\begin{pmatrix} 3 \\ 9 \\ 53 \end{pmatrix} - \begin{pmatrix} 6 \\ 13 \\ 57 \end{pmatrix} \right) = \begin{pmatrix} 2 \\ 42 \\ 123 \end{pmatrix}$$

Por lo tanto, la clave del protocolo calculada por cada usuario es:

- La clave de U_1:

$$\begin{aligned} K_1 &= 4s_1z_4 + 3X_1 + 2X_2 + X_3 \\ &= 4 \cdot \begin{pmatrix} 2 & 5 & 1 \\ 14 & 1 & 4 \\ 196 & 245 & 76 \end{pmatrix} \begin{pmatrix} 5 \\ 11 \\ 88 \end{pmatrix} + 3 \cdot \begin{pmatrix} 5 \\ 0 \\ 335 \end{pmatrix} + 2 \cdot \begin{pmatrix} 4 \\ 8 \\ 114 \end{pmatrix} + \begin{pmatrix} 3 \\ 48 \\ 114 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 32 \\ 267 \end{pmatrix}. \end{aligned}$$

- La clave de U_2:

$$\begin{aligned} K_2 &= 4s_2z_1 + 3X_2 + 2X_3 + X_4 \\ &= 4 \cdot \begin{pmatrix} 1 & 0 & 2 \\ 0 & 36 & 15 \\ 147 & 294 & 53 \end{pmatrix} \begin{pmatrix} 3 \\ 9 \\ 53 \end{pmatrix} + 3 \cdot \begin{pmatrix} 4 \\ 8 \\ 114 \end{pmatrix} + 2 \cdot \begin{pmatrix} 3 \\ 48 \\ 114 \end{pmatrix} + \begin{pmatrix} 2 \\ 42 \\ 123 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 32 \\ 267 \end{pmatrix}. \end{aligned}$$

- La clave de U_3:

$$\begin{aligned} K_3 &= 4s_3z_2 + 3X_3 + 2X_4 + X_1 \\ &= 4 \cdot \begin{pmatrix} 5 & 0 & 0 \\ 0 & 33 & 7 \\ 0 & 0 & 54 \end{pmatrix} \begin{pmatrix} 3 \\ 42 \\ 124 \end{pmatrix} + 3 \cdot \begin{pmatrix} 3 \\ 48 \\ 114 \end{pmatrix} + 2 \cdot \begin{pmatrix} 2 \\ 42 \\ 123 \end{pmatrix} + \begin{pmatrix} 5 \\ 0 \\ 335 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 32 \\ 267 \end{pmatrix}. \end{aligned}$$

- La clave de U_4:

$$\begin{aligned} K_4 &= 4s_4z_3 + 3X_4 + 2X_1 + X_2 \\ &= 4 \cdot \begin{pmatrix} 6 & 3 & 6 \\ 28 & 46 & 45 \\ 0 & 49 & 6 \end{pmatrix} \begin{pmatrix} 6 \\ 13 \\ 57 \end{pmatrix} + 3 \cdot \begin{pmatrix} 2 \\ 42 \\ 123 \end{pmatrix} + 2 \cdot \begin{pmatrix} 5 \\ 0 \\ 335 \end{pmatrix} + \begin{pmatrix} 4 \\ 8 \\ 114 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 32 \\ 267 \end{pmatrix}. \end{aligned}$$

De esta forma, los cuatro usuarios del sistema obtienen la misma clave

$$K = \begin{pmatrix} 1 \\ 32 \\ 267 \end{pmatrix}.$$

Capítulo 4

Seguridad del protocolo

El propósito de este apartado es el estudio de la seguridad del protocolo que hemos propuesto contra posibles ataques pasivos. Como ya hemos mencionado anteriormente, el protocolo que *Burmester* y *Desmedt* introdujeron en [4] es una generalización del protocolo de intercambio de dos partes de *Diffie-Hellman*, [7] , lo que implica que la seguridad del protocolo que estamos estudiando dependerá de problemas relacionados con dicho intercambio de claves .

En su ánimo de introducir criptosistemas de clave pública basados en acciones de grupo, en [17] los autores introducen el problema de la acción del semigrupo.

Definición 4.0.1. *Dados un semigrupo S, un conjunto finito X, $\phi : S \times X \to X$ una acción de S sobre X, $x, x' \in X$, el problema de la acción del semigrupo, SAP (Semigroup Action Problem), consiste en encontrar el elemento $s \in S$, de manera que $x' = \phi(s, x)$.*

El elemento s de la definición anterior se denomina logaritmo discreto de x' de base x y se denota por $\log_x x'$.

Relacionado con este problema, cuando los autores en [17] definen el intercambio de *Diffie-Hellman* asociado a una acción de semigrupo, introducen también el problema de *Diffie-Hellman* asociado a dicha acción del siguiente modo y que nosotros, para distinguirlo de otro problema también asociado al intercambio de claves, llamaremos problema computacional de *Diffie-Hellman* asociado a la correspondiente acción.

Definición 4.0.2. *Dados un semigrupo S, un conjunto finito X, $\phi : S \times X \to X$ una acción de S sobre X, un elemento $x \in X$ y la pareja de valores $\phi(a, x)$*

y $\phi(b, x)$ con $a, b \in S$, el problema computacional de Diffie-Hellman asociado a la acción del semigrupo, CDH-SAP, consiste en encontrar $\phi(ab, x)$.

El segundo problema que podemos asociar a un intercambio de clave de *Diffie-Hellman* basado en una acción de un semigrupo es el siguiente.

Definición 4.0.3. *Dados un semigrupo S, un conjunto finito X, $\phi : S \times X \to X$ una acción de S sobre X, un elemento $x \in X$, $\phi(a, x)$, $\phi(b, x)$ y $\phi(c, x)$, el problema decisional de Diffie-Hellman asociado a la acción del semigrupo,* DDH-SAP, consiste en decidir si $\phi(c, x) = \phi(ba, x)$.

A continuación veremos la relación de estos problemas con la privacidad y el secretismo teniendo en cuenta las siguientes definiciones.

Definición 4.0.4. *Se dice que un protocolo de intercambio de claves garantiza* ***privacidad*** *si dicha clave es imposible de obtener de un modo computacional por parte de un criptoanalista $\mathcal{A}$.*

Definición 4.0.5. *Se dice que un protocolo de intercambio de claves garantiza* ***secretismo*** *si un criptoanalista $\mathcal{A}$ es incapaz de distinguir la clave utilizada en una cadena de bits aleatoria.*

En los siguientes resultados veremos las condiciones bajo las cuales el protocolo de intercambio de claves que hemos introducido es seguro contra ataques pasivos. Para ello vamos a considerar un anillo A actuando sobre un grupo abeliano G dotando a éste con estructura de A-módulo.

Teorema 4.0.6. *Sea x, el elemento público de G, y sea el número de participantes en el protocolo, n par, coprimo con el cardinal, m, del subgrupo cíclico generado por x. Entonces el protocolo garantiza privacidad si y sólo si el problema computacional de Diffie-Hellman asociado a la acción,* CDH-SAP*, es intratable.*

Demostración.

$\Rightarrow$) Si el problema CDH-SAP es tratable, entonces dadas las claves públicas de cada usuario $z_i = \phi(s_i, x)$, $i = 1, \ldots, n$, podemos calcular los valores $\phi(s_1 s_2, x)$, $\phi(s_2 s_3, x)$,..., $\phi(s_{n-1} s_n, x)$, y $\phi(s_n s_1, x)$ y por tanto obtenemos la clave compartida puesto que

$$K = \phi(s_1 s_2, x)\phi(s_2 s_3, x) \ldots \phi(s_{n-1} s_n, x)\phi(s_n s_1, x)$$

$\Leftarrow$) Demostraremos esta implicación también por contradicción. Supongamos que el protocolo no garantiza privacidad, esto es, existe un algoritmo $\mathcal{A}$ que es capaz de recuperar la clave que calcula uno de los usuarios en tiempo polinomial. Sea ésta, sin pérdida de generalidad, la correspondiente a U_1, K_1.

Consideremos entonces un anillo A, un grupo abeliano G, $\phi : A \times G \to G$ una acción de A sobre G, un elemento $x \in G$ y sean $z_1 = \phi(a, x)$ y $z_n = \phi(b, x)$ con $a, b \in A$. Nuestro objetivo es resolver el problema CDH-SAP, es decir, calcular $\phi(ab, x)$. Para ello tomamos al azar los enteros b_i con $i \in [2, n-1]$ y calculamos:

$$z_i := z_{i-2}\phi(b_i, x) \qquad (1)$$

para $i \in [2, n-1]$

Podemos obtener los X_i de la siguiente manera:

$$\begin{aligned}
X_1 := \phi(b_2, z_1) &= \phi(b_2, \phi(a, x)) \\
&= \phi(b_2 a, x) \\
&= \phi(a, \phi(b_2, x)) \\
&= \phi(a, z_2 z_n^{-1}) \\
\\
X_2 := \phi(b_3, z_2) &= \phi(b_3, z_n \phi(b_2, x)) \\
&= \phi(b_3, \phi(b, x)\phi(b_2, x) \\
&= \phi(b + b_2, \phi(b_3, x)) \\
&= \phi(b + b_2, z_3 z_1^{-1}) \\
\\
X_3 := \phi(b_4, z_3) &= \phi(b_4, z_1 \phi(b_3, x)) \\
&= \phi(b_4, \phi(a, x)\phi(b_3, x) \\
&= \phi(a + b_3, \phi(b_4, x)) \\
&= \phi(a + b_3, z_4 z_2^{-1}) \\
\\
&\vdots \\
\\
X_{n-2} := \phi(b_{n-1}, z_{n-2}) &= \phi(b_{n-1}, z_{n-3}\phi(b_{n-2}, x)) \\
&= \phi(b_{n-1}, \phi(b_{n-3}, x)\phi(b_{n-2}, x) \\
&= \phi(b_{n-3} + b_{n-2}, \phi(b_{n-1}, x)) \\
&= \phi(b + b_2, z_{n-1} z_{n-3}^{-1})
\end{aligned}$$

Considerando que n es par y utilizando la definición de los z_i para $i \in [2, n-1]$ vista anteriormente, podemos calcular z_n y z_1 de la siguiente forma:

$$\begin{aligned} z_n &= z_2\phi(b_2,x)^{-1} \\ &= z_4\phi(b_4,x)^{-1}\phi(b_2,x)^{-1} \\ &= \dots \\ &= z_{n-2}\phi(b_{n-2},x)^{-1}\phi(b_{n-4},x)^{-1}\dots\phi(b_4,x)^{-1}\phi(b_2,x)^{-1}, \end{aligned}$$

$$\begin{aligned} z_1 &= z_3\phi(b_3,x)^{-1} \\ &= z_5\phi(b_5,x)^{-1}\phi(b_3,x)^{-1} \\ &= \dots \\ &= z_{n-1}\phi(b_{n-1},x)^{-1}\phi(b_{n-3},x)^{-1}\dots\phi(b_5,x)^{-1}\phi(b_3,x)^{-1}. \end{aligned}$$

Ahora, sabiendo que $X_{n-1} = \log_x z_{n-1}(z_n z_{n-2}^{-1})$ y haciendo uso de lo anterior tendremos que:

$$\begin{aligned} X_{n-1} &= \phi(\log_x z_{n-1}, (z_n z_{n-2}^{-1})) \\ &= \phi(\log_x z_{n-1}, \phi(b_{n-2},x)^{-1}\phi(b_{n-4},x)^{-1}\dots\phi(b_4,x)^{-1}\phi(b_2,x)^{-1}) \\ &= \phi((-b_2 - b_4 - \dots - b_{n-2}), \phi(\log_x z_{n-1}, x)) \\ &= \phi((-b_2 - b_4 - \dots - b_{n-2}), z_{n-1}). \end{aligned}$$

De la misma manera pero ahora usando la definición de X_n:

$$\begin{aligned} X_n &= \phi(b, z_1 z_{n-1}^{-1}) \\ &= \phi(b, \phi(b_{n-1},x)^{-1}\phi(b_{n-3},x)^{-1}\dots\phi(b_5,x)^{-1}\phi(b_3,x)^{-1}) \\ &= \phi((-b_3 - b_5 - \dots - b_{n-1}), \phi(b,x)) \\ &= \phi((-b_3 - b_5 - \dots - b_{n-1}), z_n). \end{aligned}$$

Si ahora usamos el algoritmo $\mathcal{A}$ para obtener la clave K_1 del usuario U_1 entonces se tiene que

$$K_1(X_1^{n-1})^{-1}(X_2^{n-2})^{-1}\dots X_{n-1}^{-1} = n\phi(a, z_n) = \phi(ab, x)^n$$

de donde podemos obtener $\phi(ab, x)$ puesto que $\phi(ab, x)^n = \phi(ab, x^n)$.

■

Teorema 4.0.7. *El teorema anterior se puede extender para el caso en el que el número de usuarios n sea impar.*

Demostración. Basta con que uno de los n usuarios del sistema actúe como dos usuarios. ■

Teorema 4.0.8. *El protocolo garantiza secretismo si y sólo si el problema de decisional de Diffie-Hellman asociado a la acción* DDH-SAP *es intratable.*

Demostración. $\Rightarrow$) Si el problema DDH-SAP es tratable, entonces claramente podemos distinguir la clave obtenida tras la ejecución del protocolo de una cadena de bits arbitraria puesto que ésta resulta ser de la forma

$$K = \phi(s_1 s_2, x)\phi(s_2 s_3, x)\ldots\phi(s_{n-1}s_n, x)\phi(s_n s_1, x)$$

$\Leftarrow$) Demostraremos esta implicación por contradicción. Supongamos existe un algoritmo $\mathcal{A}$ capaz de distinguir la clave del usuario U_1 de una cadena de claves aleatoria en tiempo polinomial.

Sean pues un anillo A, un grupo G y una acción de A sobre G, $\phi : A \times G \to G$, un elemento de $x \in G$ y sean $z_1 = \phi(a, x)$, $z_n = \phi(b, x)$ y $\phi(c, x)$ con $a, b, c \in A$. Nuestro objetivo es resolver el problema DDH-SAP, esto es, decidir si $\phi(ab, x) = \phi(c, x)$. De este modo seleccionamos aleatoriamente b_i con $i \in [2, n-1]$, obteniendo $z_i := \phi(b_i, x)$ y $X_i := \phi(b_i, z_{i+1}z_{i-1}^{-1})$y calcula X_1 y X_n de la siguiente manera:

Teniendo en cuenta que

$$\phi(s_1, z_2 z_n^{-1}) = \phi(s_1, z_2)\phi(s_1, z_n)^{-1}$$

y que

$$\phi(s_n, z_1 z_{n-1}^{-1}) = \phi(s_n, z_1)\phi(s_n, z_{n-1})^{-1}$$

podemos definir entonces dos parejas de valores (X_1, X_n) usando los valores de $\phi(ab, x)$ y $\phi(c, x)$.

Sea la primera pareja:

$$\begin{array}{rcl} X_1 & := & \phi(b_2, \phi(a, x))\phi(c, x)^{-1} \\ & = & \phi(a, \phi(b_2, x))\phi(c, x)^{-1} \\ & = & \phi(a, z_2)\phi(c, x)^{-1} \\ X_n & := & \phi(c, x)\phi(b_{n-1}, \phi(b, x))^{-1} \\ & = & \phi(c, x)\phi(b, \phi(b_{n-1}, x))^{-1} \\ & = & \phi(c, x)\phi(b, z_{n-1})^{-1}. \end{array}$$

y análogamente X_1' y X_n':

$$\begin{array}{rcl} X_1 & = & \phi(a, z_2)\phi(ab, x)^{-1} \\ X_n & = & \phi(ab, x)\phi(b, z_{n-1})^{-1} \end{array}$$

De este modo, podemos usar un algoritmo que nos diferencie las claves K_1 y K_1' usando las parejas (X_1, X_n) y (X_1', X_n') respectivamente. Ahora bien, como sabemos que $K_1 = \phi(ab_1, x) \cdots \phi(ab_{n-1}, x)\phi(c, x)$ y $K_1' = \phi(ab_1, x) \cdots \phi(ab_{n-1}, x)\phi(ab, x)$, tenemos que podemos diferenciar $\phi(c, x)$ de $\phi(ab, x)$ y así quedaría resuelto el problema de decisión de *Diffie-Hellman.* ■

Capítulo 5

Seguridad contra ataques activos

5.1. Introducción

En nuestro estudio de la seguridad del protocolo presentamos en esta parte de la memoria un ataque activo contra el protocolo original introducido en [4]. Como hemos mencionado anteriormente, en un ataque activo, el atacante tratan de alterar, cambiar o modificar los datos. Se trata de un ataque directo a los usuarios del sistema. En este tipo, el atacante puede modificar la información o los datos durante la transmisión, o puede crear datos falsos y enviarlos al destinatario.

En nuestro caso, el atacante realiza un ataque del tipo de repetición, en el que los datos capturados son modificados antes de ser enviados al destinatario. De esta manera, el atacante obtiene acceso y la capacidad de hacer cualquier acción que un usuario autorizado pueda hacer en el sistema. El ataque que a continuación describiremos es una modificación del conocido como Man-in-the-Middle (visto en el capítulo 2). La diferencia principal es que, una vez obtenido acceso a la información difundida, el atacante puede permanecer escuchando dicha información sin necesidad de encriptarla convenientemente y enviarla al usuario cuya identidad suplanta y utiliza para tener acceso en el principio del ataque, es decir, el ataque pasa de ser activo en su inicio, a pasivo durante el resto de dicho ataque, a diferencia de lo que ocurre en el ataque clásico Man-in-the-Middle, anteriormente citado, tal y como se muestra en la Figura 5.1

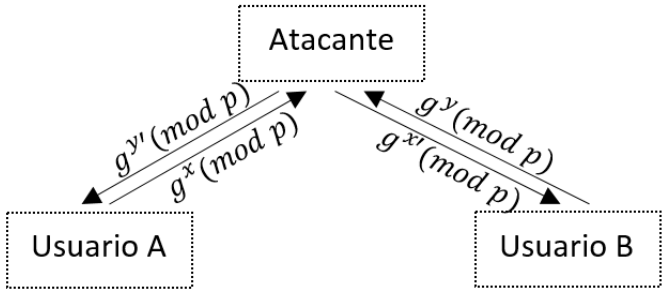

Figura 5.1: *Man-In-The-Middle* en *Diffie-Hellman*.

En el ataque que proponemos, el atacante obtendrá la clave K del grupo, la cual será creada en colaboración con los demás usuarios. El ataque consiste en detener el flujo de mensajes de un miembro del sistema, reemplazarlo durante el protocolo, y al final enviarle algunos valores necesarios para que tenga la misma clave del grupo como si dicho usuario atacado estuviese ejecutando el protocolo con toda normalidad. Estos resultados se recogen en [1].

5.2. Descripción de un ataque activo

En esta sección describimos las etapas que nos permiten atacar el protocolo original introducido en [4] de forma exitosa. Supondremos entonces que nos encontramos bajo las condiciones del protocolo vistas anteriormente, es decir, los usuarios del sistema eligen un grupo multiplicativo $\mathbb{Z}_p$, con p primo suficientemente grande y un elemento $g \in \mathbb{Z}_p$ de orden q. Los parámetros (p, g y q) son públicos y accesibles a todas las partes. Sean pues los usuarios U_i, $i = 1, \ldots, n$, que desean llevar a cabo dicho protocolo y obtener una clave K, consturída de modo colaborativo para obtener confidencialidad en sus comunicaciones y sea también un atacante activo $\mathcal{A}$ capaz de interceptar las comunicaciones, tanto entrantes, como salientes, de un usuario U_k del sistema. Los pasos a seguir para realizar el ataque son los siguientes:

1. Cada usuario U_i, $i = 1, \ldots, n$, escoge un elemento $r_i \in \mathbb{Z}_q$ y comparte su clave pública $z_i = g^{r_i}$ como en la *Ronda 1* del protocolo.

2. $\mathcal{A}$ detiene la clave pública del usuario U_k, se hace pasar por éste, enviando a los demás usuarios U_i, $i = 1, ..., n$, $i \neq k$, la clave $z'_k = g^a$, donde a es tal que $a - 1$ es invertible en $\mathbb{Z}_q$.

3. Al mismo tiempo, el atacante $\mathcal{A}$ detiene el mensaje z_{k+1} para el usuario U_k y lo reemplaza por $z'_{k+1} = z_{k-1}^a = (g^{r_{k-1}})^a$.

4. El usuario U_k empieza la *Ronda 2* y calcula $X_k = (z'_{k+1} z_{k-1}^{-1})^{r_k} = (z_{k-1}^{r_k})^{a-1}$, que será compartido con el resto de usuarios.

5. $\mathcal{A}$ detiene el mensaje X_k mientras el usuario U_k se encuentra en espera para recibir los X_i restantes, $i = 1, \ldots, n$, $i \neq k$.

6. Mientras U_k espera en la *Ronda 2*, $\mathcal{A}$ termina el protocolo de intercambio de clave con los demás participantes U_i, $i \ldots, n$, $i \neq k$, usando su clave privada a. Así obtiene la clave compartida tras el protocolo, que denotamos por K.

7. $\mathcal{A}$ halla b que será el inverso de $a - 1$ en $\mathbb{Z}_q$ y calcula $X_k^b = z_{k-1}^{r_k}$.

8. Finalmente, $\mathcal{A}$ genera una lista de elementos del grupo, $\{h_1, \ldots, h_{n-3}\}$ y le envía al usuario U_k la lista $\{X_1, \ldots, X_{k-1}, X_{k+1}, \ldots, X_n\}$ dada por

$$\begin{aligned} X_{k+1} &= z_{k-1}^{-r_k} h_1, \\ X_{k+j} &= h_{j-1}^{-1} h_j, \ \textit{para } j = 2, \ldots, n-3, \\ X_{k-2} &= K (X_k)^{-(n-1)} (z_{k-1})^{-2r_k} h_{n-3}^{-2} \textstyle\prod_{r=1}^{n-4} h_r^{-1}, \end{aligned}$$

   donde los índices son tomados en ciclo, es decir, módulo n.

Teorema 5.1. *Tras el ataque anterior, todos los usuarios U_i, $i = 1, \ldots, n$ y el atacante $\mathcal{A}$ comparten una clave común.*

Demostración. Es claro, que el atacante $\mathcal{A}$ comparte una clave común, K, con el resto de usuarios U_i, $i = 1, \ldots, n$, $i \neq k$. Ahora, el usuario U_k calcula

$$K_k = (z_{k-1})^{nr_k} \cdot X_k^{n-1} \cdot X_{k+1}^{n-2} \cdots X_{n+k-2} = K$$

□

Nota 5.2. *Observemos que el elemento X_{k-1} podría ser cualquier elemento pues éste no es usado para el cálculo de la clave. Sin embargo, en una ejecución del protocolo sin la mediación de un ataque activo como el que acabamos de mostrar, se tiene que $\prod_{i=1}^n X_i = 1$. El usuario U_k podría comprobar si esta condición se verifica. De este modo, y con objeto de evitar que el ataque sea detectado, una vez que se han calculado todos los X_i, $i \neq k-1$, podemos definir $X_{k-1} = \left(\prod_{i=1, i\neq k-1}^n X_i\right)^{-1}$*

Acabamos de probar la efectividad de nuestro ataque sobre el protocolo original introducido en [4]. Sin embargo, el mismo sería exitoso en el caso de llevarlo a cabo sobre un intercambio de clave en grupo determinado por una acción tal y como hemos definido en el capítulo anterior. Veamos un ejemplo concreto en el caso del grupo de puntos de una curva elíptica.

Ejemplo 5.3. *Supongamos que un grupo de cinco usuarios elige el cuerpo finito* $\mathbb{F}_p$ *con* $p = 83$*, la curva elíptica* $E : y^2 = x^3 + 36x + 82$ *sobre dicho cuerpo. Sea también un punto* $P = (10, 23)$ *de* $E(\mathbb{F}_{83})$*.* P *es de orden* 81*, por lo que la acción considerada por los cinco usuarios viene dada por:*

$$\begin{array}{rccl} \phi: & \mathbb{Z}_{81} \times E & \to & E \\ & (d_i, P) & \to & d_i P, \end{array}$$

donde d_i *es la clave privada de cada usuario. Cada usuario* U_i*,* $1 \leq i \leq 5$*, escoge su propio* d_i *y procede con la Ronda 1 del protocolo de la siguiente forma:*

- U_1 *elige* $d_1 = 13$ *y calcula* $z_1 = \phi(d_1, P) = d_1 P = 13 \cdot (10, 23) = (55, 14)$.
- U_2 *elige* $d_2 = 21$ *y calcula* $z_2 = \phi(d_2, P) = d_2 P = 21 \cdot (10, 23) = (76, 63)$.
- U_3 *elige* $d_3 = 47$ *y calcula* $z_3 = \phi(d_3, P) = d_3 P = 47 \cdot (10, 23) = (22, 20)$.
- U_4 *elige* $d_4 = 52$ *y calcula* $z_4 = \phi(d_4, P) = d_4 P = 52 \cdot (10, 23) = (25, 67)$.
- U_5 *elige* $d_5 = 26$ *y calcula* $z_5 = \phi(d_5, P) = d_5 P = 26 \cdot (10, 23) = (68, 63)$.

Los z_i *son públicos por lo que serán compartidos con todos los usuarios del sistema. Supongamos que un atacante activo* $\mathcal{A}$ *desea participar en el protocolo para obtener la misma clave que los miembros del sistema utilizando el ataque anterior. Para ello,* $\mathcal{A}$ *necesita tener un control sobre el flujo de mensajes recibidos por un usuario* U_k*, por ejemplo* U_3*. En primer lugar,* $\mathcal{A}$ *detiene la clave pública del usuario* U_3*, se hace pasar por este, enviando a los demás usuario* U_i*,* $i = 1, 2, 4, 5$*, la clave* $z_3' = \phi(a, P)$ *con* $a - 1$ *invertible en* $\mathbb{Z}_{81}$*. Sea* $a = 62$*,* $(a - 1)^{-1} \pmod{81} \equiv 4$*. Por lo que* $\mathcal{A}$ *comparte* $z_3' = \phi(62, P) = 62 \cdot (10, 23) = (6, 79)$*. Simultáneamente, el atacante* $\mathcal{A}$ *detiene el mensaje* z_4 *enviado al usuario que esta suplantando, y lo reemplaza por* $z_4' = \phi(a, z_2) = 62 \cdot (76, 63) = (51, 21)$.

El usuario U_3 empieza la Ronda 2 y calcula $X_3 = \phi(d_3, z_4' z_2^{-1}) = d_3 \cdot (z_4' - z_2) = 47 \cdot (56, 29) = (79, 66)$. El atacante $\mathcal{A}$ para el mensaje X_3 mientras el usuario U_3 se encuentra en espera para recibir los X_i restantes, es decir, para $i = 1, 2, 4, 5$. $\mathcal{A}$ termina el protocolo de intercambio de claves con los demás usuarios U_1, U_2, U_4 y U_5 (U_3 sigue en espera) usando su clave privada $a = 62$ de la siguiente manera:

- *Los miembros U_i, $i = 1, 2, 4, 5$, y el atacante $\mathcal{A}$, calculan los respectivos X_i:*
  - $X_1 = \phi(d_1, z_2 z_5^{-1}) = d_1(z_2 - z_5) = 13 \cdot (47, 41) = (4, 46)$.
  - $X_2 = \phi(d_2, z_3' z_1^{-1}) = d_2(z_3' - z_1) = 21 \cdot (33, 40) = (79, 17)$.
  - $X_3' = \phi(a, z_4 z_2^{-1}) = a(z_4 - z_2) = 62 \cdot (52, 36) = (37, 62)$.
  - $X_4 = \phi(d_4, z_5 z_3'^{-1}) = d_4(z_5 - z_3') = 52 \cdot (12, 1) = (35, 71)$.
  - $X_5 = \phi(d_5, z_1 z_4^{-1}) = d_5(z_1 - z_4) = 26 \cdot (41, 57) = (41, 26)$.
- *Los cuatro usuarios y el atacante calculan la clave del protocolo:*

$$\begin{aligned} K_1 &= \phi(5, \phi(d_1, z_5))\phi(4, X_1)\phi(3, X_2)\phi(2, X_3')X_4 \\ &= 5d_1 z_5 + 4X_1 + 3X_2 + 2X_3' + X_4 \\ &= 5 \cdot 13 \cdot (68, 63) + 4 \cdot (4, 46) + 3 \cdot (79, 17) + 2 \cdot (37, 62) + (35, 71) \\ &= (35, 12). \end{aligned}$$

$$\begin{aligned} K_2 &= \phi(5, \phi(d_2, z_1))\phi(4, X_2)\phi(3, X_3')\phi(2, X_4)X_5 \\ &= 5d_2 z_1 + 4X_2 + 3X_3' + 2X_4 + X_5 \\ &= 5 \cdot 21 \cdot (55, 14) + 4 \cdot (79, 17) + 3 \cdot (37, 62) + 2 \cdot (35, 71) + (41, 26) \\ &= (35, 12). \end{aligned}$$

$$\begin{aligned} K_3' &= \phi(5, \phi(a, z_2))\phi(4, X_3')\phi(3, X_4)\phi(2, X_5)X_1 \\ &= 5az_2 + 4X_3' + 3X_4 + 2X_5 + X_1 \\ &= 5 \cdot 62 \cdot (76, 63) + 4 \cdot (37, 62) + 3 \cdot (35, 71) + 2 \cdot (41, 26) + (4, 46) \\ &= (35, 12). \end{aligned}$$

$$\begin{aligned} K_4 &= \phi(5, \phi(d_4, z_3'))\phi(4, X_4)\phi(3, X_5)\phi(2, X_1)X_2 \\ &= 5d_4 z_3' + 4X_4 + 3X_5 + 2X_1 + X_2 \\ &= 5 \cdot 52 \cdot (6, 79) + 4 \cdot (35, 71) + 3 \cdot (41, 26) + 2 \cdot (4, 46) + (79, 17) \\ &= (35, 12). \end{aligned}$$

$$\begin{aligned} K_5 &= \phi(5,\phi(d_5,z_4))\phi(4,X_5)\phi(3,X_1)\phi(2,X_2)X_3' \\ &= 5d_5z_4 + 4X_5 + 3X_1 + 2X_2 + X_3' \\ &= 5\cdot 26\cdot(25,67) + 4\cdot(41,26) + 3\cdot(4,46) + 2\cdot(79,17) + (37,62) \\ &= (35,12). \end{aligned}$$

De esta manera, el atacante activo $\mathcal{A}$ obtiene la misma clave que los usuarios U_1, U_2, U_4 y U_5. En los dos últimos pasos del ataque, $\mathcal{A}$ se encargará de que el usuario U_3, al que ha suplantado durante el establecimiento de la clave del protocolo, obtenga la misma clave, procediendo de la siguiente forma:

- *$\mathcal{A}$ halla $b \equiv (a-1)^{-1}(\mod 81) \equiv 4$ y calcula $\phi(4,X_3) = 4X_3 = 4\cdot d_3(z_4' - z_2) = 4\cdot(79,66) = (56,54)$.*
- *Finalmente, $\mathcal{A}$ genera la lista de elementos:*

$$\{h_1, h_2\} = \{(30,5),(55,11)\}$$

y le envía al usuario U_3 la lista de los X_i, $i = 1,2,4,5$, que vienen dados por:

$$\begin{aligned} X_4 &= -d_3z_2 + h_1 \\ &= -bX_3 + h_1 \\ &= (56,-54) + (30,5) \\ &= (45,45), \\ X_5 &= h_2 - h_1 \\ &= (55,14) + (30,-5) \\ &= (76,63), \\ X_1 &= K - 4X_3 - 2d_3z_2 - 2h_2 - h_1 \\ &= (35,12) + (56,-54) + 2(56,-54) + 2(55,-14) + (30,-5) \\ &= (64,80). \end{aligned}$$

Por lo tanto, el usuario U_3 obtiene la misma clave que los demás usuarios haciendo el siguiente cálculo:

$$\begin{aligned} K_3 &= 5d_3z_2 + 4X_3 + 3X_4 + 2X_5 + X_1 \\ &= 5(56,54) + 4(79,66) + 3(45,45) + 2(76,63) + (64,80) \\ &= (35,12) = K. \end{aligned}$$

Ya hemos comentado anteriormente que el ataque tendría éxito en el caso de que el protocolo se lleve a cabo usando una acción tal y como hemos

definido en el caso general en el capítulo anterior. Sin embargo, en el caso de la acción definida en el tercer ejemplo de las aplicaciones del protocolo en la sección 3.2, tenemos un anillo $E_p^{(m)}$ actuando sobre un grupo abeliano, un anillo que, dependiendo de los valores de p y m, tal y como se expone en [6], puede que carezca casi de unidades, es decir, la posibilidad de llevar a cabo exitosamente el ataque es ínfima debido a la casi imposibilidad de encontrar un elemento a en $E_p^{(m)}$ tal que $a-1$ sea invertible. ¿Quiere decir esto que en tal caso el protocolo es seguro contra este ataque? La respuesta es no. Para ello puede llevarse a cabo una modificación del ataque tal y como a continuación describimos.

Recordemos que el atacante $\mathcal{A}$ persigue obtener el valor de la acción del elemento privado r_k del usuario U_k sobre el elemento público z_{k-1}, $\phi(r_k, z_{k-1})$, para lo que calcula $\phi((a-1)^{-1}, X_k)$ en el paso 7. Para ello, en el paso 3, en lugar de enviar el elemento $\phi(a, z_{k-1})$ al usuario U_k, le puede enviar el elemento

$$z_{k-1}\phi(a,g)^{-1}$$

siendo g el valor común público que se acuerda al comienzo del protocolo. Entonces el usuario U_k devuelve el valor

$$\phi(r_k, z_{k-1}\phi(a,g)^{-1})$$

Pero entonces, como el atacante conoce la clave pública de U_k, $\phi(r_k, g)$, entonces $\mathcal{A}$ puede calcular

$$\phi(a, \phi(r_k,g))\phi(r_k, z_{k-1}\phi(a,g)^{-1}) = \phi(ar_k, g)\phi(r_k, z_{k-1})\phi(r_k a, g)^{-1}$$

y así obtener el valor $\phi(r_k, z_{k-1})$ tal y como se pretendía.

Capítulo 6

Conclusiones

En esta memoria hemos sugerido una generalización del protocolo introducido por *Burmester* y *Desmedt* en [4], donde utilizan un grupo cíclico $\mathbb{Z}_p$, a cualquier acción de un semigrupo abeliano sobre un conjunto con estructura de grupo.

El análisis de la seguridad contra ataques pasivos concluye que nuestro protocolo sobre la acción definida garantiza privacidad y secretismo dependiendo de la dificultad para resolver los problemas relacionados con el protocolo de intercambio de claves de *Diffie-Hellman*.

En lo que respecta a la seguridad contra ataques activos del protocolo original de *Burmester* y *Desmedt*, hemos probado su vulnerabilidad, introduciendo una modificación de un ataque del tipo Man-in-the-Middle, pero en el que el atacante, tras una primera suplantación, puede permancer como un miembro más del grupo de comunicación sin ser detectado. Una parte fundamental del ataque es el uso de inversos en el grupo que lleva a cabo la acción. Sin embargo. también hemos visto que en los casos en los que carecemos de unidades, como es el del anillo $E_p^{(m)}$ actuando sobre un grupo abeliano, podemos ejecutar el ataque haciendo una pequeña modificación en éste.

Los resultados anteriores nos permiten afirmar la posibilidad de la extensión a otros escenarios más generales de este protocolo. Sin embargo, después de haber logrado llevar a cabo un ataque activo contra el mismo, aprovechándonos de que éste necesita de dos rondas, nos planteamos la siguiente

pregunta: ¿Es posible atacar cualquier protocolo que necesite la recepción y envío de más de un mensaje por los usuarios? Es decir, ¿podemos usar al usuario atacado como un oráculo que nos permita construir un nuevo tipo de ataques para este tipo de esquemas distribuidos?

Como posible solución a la cuestión anterior, podríamos pensar que podemos encontrar un esquema que permita autentificar que la información proviene de un miembro no malicioso del grupo, pero ¿será posible encontrar dicho esquema sin vulnerar la filosofía distributiva en cuanto a la obtención de la clave común, es decir, sin la existencia de autoridades centrales que la verifiquen?

Índice de figuras

Bibliografía

[1] M. Baouch, J.A. López-Ramos, R. Schnyder, B.Torrecillas. *An active attack on a distributed Group Key Exchange system.* sometido para su publicación. Accesible en arXiv: 1603.09090.

[2] R. Barua, R. Dutta and P. Sarkar. *Extending Joux's Protocol to Multi Party Key Agreement.* In *Progress in Cryptology-INDOCRYPT'03* volume 2904 of *Lecture Notes in Computer Science*, pages $205-217$. Springer, December 2003.

[3] K. Becker and U. Wille. *Communication Complexity of Group Key Distribution.* In *Proceedings of the 5th ACM CCS'98*, pages $1-6$. ACM Press, 1998.

[4] M. Burmester, I. Desmedt. *A secure and Scalable Group Key Exchange System*, *Information Proc. Letters* 94, 2005, $137-143$.

[5] J.-J. Climent, J. A. López-Ramos. *Public Key Protocols over the Ring* $E_p^{(m)}$. *Advances in Mathematics of Communications,* 2016.

[6] J.-J. Climent, P.R. Navarro and L. Tortosa. *An extension of the noncommutative Bergman's ring with a large number of noninvertible elements, Appl. Algebr. Eng. Comm,* 25(5) (2014), $347-367$.

[7] W.Diffie and M.E.Hellman. *New directions in cryptography. IEEE. Inform. Theory,* IT-22(6): $644-654$, 1976.

[8] T. ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithmms*, *IEEE. Inform. Theory,*, 31(4), $469-472$, 1985.

[9] I.Ingemarsson, D.T.Tang and C.K.Wong. *A conference key distribution system.. IEEE. Trans. Inform. Theory,* $28(5) : 714-720$, 1982.

[10] A. Joux. *A One Round Protocol for Tripartite Diffie-Hellman.* In *Algorithmic Number Theory, IV-th Symposium (ANTS IV)*, volume 1838 of *Lecture Notes in Computer Science*, pages $385-394$. Springer, July 2000.

[11] Y. Kim, A. Perrig, and G. Tsudik. *Communication-Efficient Group Key Agreement.* In *Proceedings of IFIP TC11 Sixteenth Annual Working Conference on Information Security (IFIP/Sec'01)*, volume 193 of *IFIP Conference Proceedings*, p. 229–244. Kluwer, 2001.

[12] Y. Kim, A. Perrig and G. Tsudik. *Simple and fault-tolerant key agreement for dynamic collaborative groups.* In *7th ACM Conference on Computer and Communications Security*, pages $235-244$. ACM Press, 2000.

[13] N.Koblitz. *A Course in Number Theory and Cryptography*, Graduate Texts in Mathematics 114, Springer, 1987.

[14] N. Koblitz, *Elliptic curve cyrptosystems, Maht. Comp.*, 48(177), 203-209, 1987.

[15] S. Lee, Y. Kim, K. Kim and D.-H. Ryu. *An Efficient Tree-Based Group Key Agreement Using Bilinear Map.* In *Proceedings of the First International Conference on Applied Cryptography and Network Security (ACNS'03)*, volume 2846 of *Lecture Notes in Computer Science*, pages $357-371$. Springer, 2003.

[16] J.A. López-Ramos, J. Rosenthal, D. Schipani, R. Schnyder. *Group key management based on semigroup actions,* por aparecer en Journal of Mathematics and its Applications. Accesible en arXiv:1509.01075.

[17] G. Maze, C. Monico, J. Rosenthal. *Public Key Cryptography based on Semigroup Actions, AMC,* vol.1(4), $489-507$, 2007.

[18] V.S. Miller, *Use of elliptic curves in cryptography*, *Advances in cryptography-CRYPTO'85 (Santa Barbara, Callif. 1985)*, $417-426$, Springer, Berlin, 1986.

[19] A.Perrig. *Efficient Collaborative Key Management Protocols for Secure Autonomous Group Communication.* In *Proceedings of the International Workshop on Cryptographic Techniques and Electronic Commerce* 1999, pages $192-202$. City University of Hong Kong Press, 1999.

[20] R. L. Rivest, A. Shamir y L. Adleman. *A method for obtaining difital signature and public-key cryptosystems. Communications of the ACM*, 21(2): 120 − 126 (1978).

[21] J. H. Silverman. *The Arithmetic of Elliptic Curves*, Springer, 1986.

[22] D. G. Steer, L. Strawczynski, Whitfield Diffie, and Michael J. Wiener. *A Secure Audio Teleconference System.* In *Advances in Cryptology – CRYPTO'88,* volume 403 of *Lecture Notes in Computer Science,* p. 520 − 528, Springer, 1990.

[23] M.Steiner, G.Tsudik and M. Waidner. *CLIQUES: A New Approach to Group Key Agreement.* In *Proceedings of the 18th International Conference on Distribuited Computing Systems (ICDCS'98)*, p. 380 − 387. *IEEE Computer Society Press,* 1998.

[24] M.Steiner, G.Tsudik and M. Waidner. *Diffie-Hellman Key Distribution Extended to Group Communication. ACM Conf. Comp. and Comm. Security*: 31 − 37, 1996.

[25] M.Steiner, G.Tsudik and M. Waidner. *Key Agreement in Dynamic Peer Groups. IEEE TPDS,* 11(8) : 769 − 780, 2000.