

Protocolos criptográficos

Llorenç Huguet Rotger

Josep Rifà Coma

Juan Gabriel Tena Ayuso

PID_00200954

Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació per a la Universitat Oberta de Catalunya), no hagáis un uso comercial y no hagáis una obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Índice

Introducción	5
Objetivos	7
1. Protocolos de gestión y distribución de claves	9
1.1. Protocolo de transporte de una clave privada	10
1.2. Protocolo de intercambio de dos claves de Needham-Schroeder	11
1.3. Protocolo de distribución de claves centralizado	11
1.4. Protocolo de acuerdo de claves de Diffie-Hellman	12
2. Protocolos de autenticación	15
2.1. Protocolo de tres pasos de Shamir.....	16
2.2. Protocolo de Omura	19
2.3. Protocolo de Needham-Schroeder.....	20
2.4. Protocolo de Kerberos.....	21
2.5. Protocolo STS	23
2.6. Otros protocolos: ISO, CCITT X.509, SSL	24
2.7. Protocolos de identificación de conocimiento nulo	26
3. Transacciones electrónicas seguras: dinero electrónico	30
3.1. Protocolo de Chaum	30
3.2. Transacciones sin rastro. Firmas digitales ciegas de Chaum ...	33
3.3. Sistemas de pago electrónicos	34
4. Protocolos de transferencia inconsciente	38
4.1. Protocolo de Rabin.....	38
4.2. Protocolos de compromiso de bits	40
4.3. Firma electrónica de contratos: Protocolo de Even	43
4.4. Protocolo de correo electrónico certificado.....	46
5. Esquemas umbral y reparto de secretos	49
5.1. Esquema de Shamir	49
6. Votaciones electrónicas	52
6.1. Garantizar la privacidad y la corrección de los resultados	53
6.2. Garantizar la auditoría de la votación	55

Ejercicios de autoevaluación	56
Solucionario	57
Bibliografía	59

Introducción

Las comunicaciones electrónicas ofrecen nuevas posibilidades en los intercambios de información, sobre todo en el campo de las transacciones comerciales y en la administración electrónica.

En el mundo no electrónico los intercambios de información, llevados a cabo con medios más convencionales, presentan problemas de seguridad y desconfianza que se resuelven a través de un arbitraje con terceras partes de confianza, como, por ejemplo, jueces, notarios y agentes postales, entre otros, que tradicionalmente son los que han dado seguridad a estos intercambios.

En las transacciones electrónicas, llevadas a cabo mediante protocolos de comunicación, la seguridad tiene un papel incluso más relevante, ya que necesita dar, además, una protección contra posibles amenazas, como son las manipulaciones desautorizadas de los datos o las falsificaciones.

Al conjunto de acciones bien definidas y coordinadas descritas por un cierto algoritmo, que permiten una interacción entre dos, o más, usuarios para llevar a cabo un intercambio de datos o de información, se le llama **protocolo**.

Los **protocolos criptográficos** son aquellos que, para llevar a cabo esta interacción, usan funciones criptográficas para poder asegurar los requisitos de seguridad en las comunicaciones entre los usuarios que intervienen en el intercambio: la confidencialidad, la integridad, la autenticidad y el no repudio.

Vamos a encontrar ejemplos de uso de los protocolos criptográficos en ámbitos tan diversos cómo: en comercio electrónico, donde tienen que utilizarse firmas compartidas; en correo electrónico seguro, donde se necesitará algo más que un acuse de recibo; en la implementación de sistemas electrónicos de pagos con la utilización de firmas ciegas o en sistemas de votación electrónica.

En muchos casos se necesitará, también, el arbitraje de una tercera parte de confianza para garantizar todos los requisitos de seguridad en estas transacciones electrónicas.

La diversidad de ámbitos hace que exista también una amplia variedad de protocolos criptográficos para dar respuesta a los diferentes objetivos, pero todos ellos se usan, generalmente, para eliminar desconfianzas.

Podemos dar la siguiente clasificación, según los objetivos que se quieran lograr:

- **Gestión y distribución de claves.** Permiten y garantizan la generación, almacenamiento, mantenimiento y distribución de claves de un sistema criptográfico de clave privada. En algunos casos, esta distribución se hará utilizando criptosistemas de clave pública.
- **Autenticación de usuario.** Permiten garantizar que el remitente de un mensaje, con quien se establece comunicación, es realmente quien pretende ser.
- **Transacciones electrónicas seguras.** Permiten realizar las operaciones bancarias habituales, en particular, implementar sistemas de pago electrónicos sustitutivos de las tarjetas de crédito y de débito en las transacciones económicas en el comercio electrónico, con una utilización muy especial en el caso de micropagos (menos de 10 €).
- **Transferencias inconscientes o transcordadas.** Permiten a un usuario enviar un mensaje o un secreto, entre dos posibles, a otro usuario. El usuario emisor no conoce cuál de los dos ha recibido el receptor. Estos protocolos permiten la firma electrónica de contratos.
- **Compromiso de bit.** Permiten a un usuario de una red comprometerse con la elección de un bit (o más, generalmente con una serie de bits) sin revelar tal elección hasta un momento posterior. El protocolo garantiza a la otra parte que el usuario no cambia su elección.
- **Compartición de secretos.** Permiten distribuir un cierto secreto entre un conjunto de participantes de manera que ciertos subconjuntos prefijados entre los participantes puedan, uniendo sus participaciones, recuperar el secreto.
- **Pruebas de conocimiento nulo.** Permiten a un usuario de una red convencer a otro que el primero posee una cierta información, sin revelar nada sobre el contenido de la misma.
- **Votaciones electrónicas.** Permiten realizar un proceso electoral electrónicamente, garantizando la privacidad de cada votante y la imposibilidad de fraude.

Objetivos

En los materiales didácticos de este módulo el estudiante encontrará los contenidos necesarios para alcanzar los objetivos siguientes:

- 1.** Conocer los protocolos más importantes de gestión y distribución de claves.
- 2.** Conocer los protocolos más importantes de autenticación.
- 3.** Conocer los protocolos más importantes de transacciones electrónicas seguras y sistemas de pago electrónico.
- 4.** Conocer los protocolos más importantes para implementar la firma de contratos y el correo electrónico certificado.
- 5.** Conocer los fundamentos de los esquemas de umbral.
- 6.** Conocer los fundamentos de las votaciones electrónicas.

1. Protocolos de gestión y distribución de claves

Uno de los problemas de la criptografía de clave privada está en la distribución de las claves entre los usuarios de una red de comunicaciones, en la que cada pareja de usuarios A, B necesita compartir una clave K_{AB} , para crear un canal privado virtual entre ambos.

Esta clave no puede ser enviada por la propia red de comunicaciones (que consideraremos insegura) y que, además, por motivos de seguridad, debe ser cambiada periódicamente. Muchas veces, K_{AB} es desechable y se la llama **clave de sesión**.

Por lo que hace a la gestión de claves, ésta comporta interrogantes, como asegurar quiénes asumen la responsabilidad de la creación de las claves (con diferentes alternativas: autoridad central o sistema totalmente descentralizado), construir diferentes tipos de claves (de comunicaciones, maestras, de sesión), definir los requisitos de seguridad en el almacenamiento de las mismas, etc.

En este apartado pondremos énfasis en los protocolos de distribución:

1) **Gestión de claves en los sistemas criptográficos de clave pública.** En un sistema criptográfico de clave pública, donde cada usuario U dispone de su pareja de claves: E_U para cifrar y D_U para descifrar, no es necesaria la distribución de las clave para cifrar; al contrario, han de estar accesibles para cualquiera que desee utilizarlas y poder comunicarse con el propietario de las mismas. Existe, sin embargo, el riesgo de la **impersonación**, es decir, un adversario C puede hacer creer que su clave pública E_C es la clave pública de otro usuario.

Consideramos cuatro esquemas posibles de gestión de claves públicas:

- **Anuncio público:** cada participante difunde su clave pública al resto de usuarios. El riesgo de impersonación es grande.
- **Directorio público:** mantenido por una cierta autoridad, una tercera parte de confianza, con acceso directo y libre (para lectura) por parte de cualquier usuario al directorio de claves TPD (*trusted public directory*). Cualquier usuario U registra en el TPD su identidad; Id_U ; y su clave pública, E_U , en persona o mediante comunicación autorizada, lo cual, en principio, elimina el riesgo de la impersonación.
- **Autoridad pública:** similar al anterior, pero los usuarios no tienen acceso directo al directorio de claves, sino que interaccionan con un centro de

Notación

A lo largo de este módulo, cuando nos referimos a un sistema criptográfico de clave pública, la notación que usaremos, asociada a un usuario U , será:

- Id_U : Identidad usuario.
- $E_U(m)$: Proceso de cifrar un mensaje m , con la clave pública de U .
- $D_U(c)$: Proceso de descifrar un criptograma c , con la clave privada de U .

distribución de claves *KDC* (*key directory centre*). Si el usuario *A* desea conocer la clave pública del usuario *B*, debe formular una petición expresa a la autoridad que mantiene el directorio.

- **Autoridad certificadora:** es una tercera parte de confianza, *T*, que expide a cada usuario *U* un certificado de su clave pública, ligada a su identidad Id_U , junto con otros datos. Este certificado va firmado con la clave privada de *T*. Así, un usuario *A* puede enviar a otro usuario *B* este certificado y éste puede comprobar, utilizando la clave pública de la autoridad E_T , la validez de la clave, la identidad del emisor, así como los otros datos incluídas, entre ellos el periodo de vigencia del certificado, etc.*

*Ver el subapartado 2.6 del módulo "Elementos de criptografía".

2) Distribución de claves en los sistemas criptográficos de clave privada.

En los sistemas criptográficos de clave privada, las claves se deben distribuir de forma centralizada puesto que en otro caso, el administrador de una red tendría que proporcionar claves a cada pareja de usuarios que quieran intercambiarse información secreta. Esto querría decir que el administrador necesitaría tener almacenadas $\binom{n}{2} = \frac{n(n-1)}{2} = O(n^2)$ claves, y cada usuario $(n-1)$ claves.

En un sistema criptográfico de clave privada, dos usuarios *A* y *B* pueden establecer una clave K_{AB} , compartida por ambos, por dos métodos:

- **Transporte de claves:** un usuario crea una clave y la transfiere, con seguridad, al otro usuario (alternativamente, puede ser una tercera parte quien la crea y transfiere a ambos).
- **Acuerdo de claves (*key agreement*):** la clave es calculada por los usuarios como una cierta función de la información suministrada por ambos usuarios. En principio ningún usuario, por sí mismo, puede predeterminar el valor de esta clave.

Un problema adicional que puede plantearse es el de la autenticación de las claves y de los usuarios que las acuerdan o envían. Tal autenticación será resuelta con los protocolos de distribución, en la mayoría de casos utilizando sistemas criptográficos de clave pública para distribuir una clave privada.

1.1. Protocolo de transporte de una clave privada

Este protocolo permite transferir una clave K_A , elegida por el usuario *A*, a otro usuario *B*, mediante un algoritmo de un solo paso.

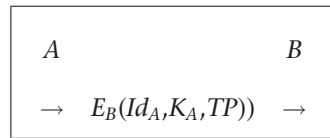
Supondremos que el usuario *A* tiene acceso a una copia autenticada de E_B y que usa un parámetro temporal, *TP*.

Protocolo

- $A \rightarrow B$. El usuario *A* envía a *B*: $E_B(Id_A, K_A, TP)$

Ahora, el usuario B puede descifrar lo que ha recibido, con su clave privada D_B y, de este modo, verificar la identidad del emisor, el parámetro temporal TP y asociar la clave K_A al usuario A .

Resumen de las transacciones del protocolo:



1.2. Protocolo de intercambio de dos claves de Needham-Schroeder

Este protocolo permite intercambiar dos claves secretas entre dos usuarios A y B , respectivamente, K_A y K_B , con solo 3 pasos, al mismo tiempo que los usuarios se autentican mutuamente.

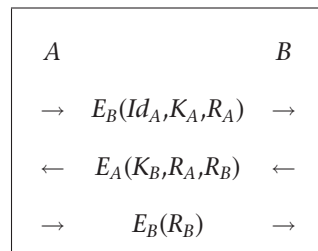
Protocolo

- $A \rightarrow B$. El usuario A elige un número aleatorio R_A y envía a B : $E_B(Id_A, K_A, R_A)$.
- $B \rightarrow A$. El usuario B descifra lo que ha recibido, con su clave privada D_B , y obtiene R_A , además de la clave pública y la identidad de A . Ahora B elige otro número aleatorio R_B y envía a A : $E_A(K_B, R_A, R_B)$.
- $A \rightarrow B$. El usuario A descifra lo que ha recibido, con su clave privada D_A , y obtiene R_B , además de la clave pública y la identidad de B , y envía a B : $E_B(R_B)$.

Autenticaciones

- **Autenticación de B por parte de A :** el usuario A comprueba que R_A es correcto, con lo cual autentica a B y confirma que ha recibido K_B .
- **Autenticación de A por parte de B :** el usuario B comprueba que R_B es correcto, con lo cual autentica a A y confirma que ha recibido K_A .

Resumen de las transacciones del protocolo:



1.3. Protocolo de distribución de claves centralizado

Este protocolo permite la distribución de una clave de sesión a dos usuarios A y B , para utilizar en un sistema criptográfico de clave privada.

Supongamos que A y B disponen de un sistema criptográfico de clave pública y que, en este caso, interviene una autoridad pública, T , que gestiona un centro de distribución de claves KDC .

Caso práctico

Se puede pensar en un protocolo donde se quiera usar una clave de sesión para utilizar con el sistema AES , mientras que la distribución se hace con un sistema ElGamal.

Para iniciar el protocolo, el usuario A comunica a la autoridad pública T , que se quiere comunicar con B y, para ello, solicita la identidad de B y una clave de sesión K_{AB} para compartir. Además, se utilizará un parámetro temporal TP , para controlar el plazo de vigencia.

Protocolo

- $T \rightarrow A$. La autoridad pública T envía a A : $E_A(Id_B, K_{AB}, TP, E_B(Id_A, K_{AB}, TP))$.

El usuario A descifra lo que ha recibido, con su clave privada D_A , y obtiene: Id_B , K_{AB} , el parámetro TP y $E_B(Id_A, K_{AB}, TP)$.

- $A \rightarrow B$. El usuario A envía a B : $E_B(Id_A, K_{AB}, TP)$.

El usuario B descifra lo que ha recibido, con su clave privada D_B , y obtiene: Id_A , K_{AB} y TP .

Finalmente, para iniciar la transmisión con total garantía, haría falta un acuse de recibo por parte de B a A , contrastando la validez del valor TP .

Resumen de las transacciones del protocolo:

$$\begin{array}{ccc} T & & A \\ \rightarrow & E_A(Id_B, K_{AB}, TP, E_B(Id_A, K_{AB}, TP)) & \rightarrow \end{array}$$

$$\begin{array}{ccc} A & & B \\ \rightarrow & E_B(Id_A, K_{AB}, TP) & \rightarrow \end{array}$$

1.4. Protocolo de acuerdo de claves de Diffie-Hellman

Este protocolo se basa en la función exponencial y el logaritmo discreto, en un cuerpo finito \mathbb{Z}_p , y no necesita ninguna otra intervención más que la de los propios usuarios. La seguridad del protocolo se basa, precisamente, en la intratabilidad del cálculo del logaritmo discreto.

Todos los usuarios conocen el valor del primo p y el de un elemento primitivo $\alpha \in \mathbb{Z}_p$.

Cada usuario U busca al azar $x_U \in \mathbb{Z}_p^*$, que guarda secreto, y hace público el valor de $y_U = \alpha^{x_U} \pmod{p}$. Así, cada usuario U coloca (Id_U, y_U) en el directorio público TPD .

Para compartir una clave de sesión, K_{AB} , entre dos usuarios A y B , éstos harán lo siguiente.

Protocolo

- El usuario A obtiene y_B del TDP , calcula $K_A = y_B^{x_A} \pmod{p} = \alpha^{x_B \cdot x_A} \pmod{p}$.
- El usuario B obtiene y_A del TDP , calcula $K_B = y_A^{x_B} \pmod{p} = \alpha^{x_A \cdot x_B} \pmod{p}$.

Ahora A y B ya pueden compartir una misma clave $K_{AB} = K_A = K_B$ (efectivamente: $\alpha^{x_A \cdot x_B} \pmod{p} = \alpha^{x_B \cdot x_A} \pmod{p}$) para intercambiarse mensajes, en un sistema criptográfico de clave privada, sin que A necesite del valor secreto x_B , ni B necesite del valor secreto x_A .

Resumen del protocolo:

A	B
$K_A = y_B^{x_A} \pmod{p}$	$K_B = y_A^{x_B} \pmod{p}$

Ejemplo 1.1.

A y B quieren intercambiar una clave de sesión K_{AB} , usando el cuerpo \mathbb{Z}_p , con $p = 1999$ y el elemento primitivo $\alpha = 33$. En realidad, este valor de p debería ser muy grande.

Supongamos que A escoge $x_A = 47$ y B escoge $x_B = 117$. Entonces, ambos usuarios harán los siguientes cálculos:

El usuario A calcula $y_A = \alpha^{x_A} \pmod{p} = 33^{47} \pmod{1999} = 1343$;
 El usuario B calcula $y_B = \alpha^{x_B} \pmod{p} = 33^{117} \pmod{1999} = 1991$;
 Así, $(Id_A, y_A = 1343)$, $(Id_B, y_B = 1991)$ figurarán en el TPD .

Protocolo

- El usuario A calcula $K_{AB} = y_B^{x_A} \pmod{p} = 1991^{47} \pmod{1999} = 1506$.
- El usuario B calcula $K_{BA} = y_A^{x_B} \pmod{p} = 1343^{117} \pmod{1999} = 1506$.

La clave secreta compartida por A y B será $K_{AB} = K_{BA} = 1506$.

Nota

El protocolo de acuerdo de claves de Diffie-Hellman es vulnerable, frente a ataques de impersonación.

Puede haber un espía activo-pasivo, C , que envíe mensajes a B , a partir de los que recibe de A . Los usuarios A y B creen que están interconectados cuando en realidad sus mensajes son filtrados y/o manipulados por C , puesto que es C quien realmente está conectado con A y con B .

**Simulador de cálculos
en \mathbb{Z}_p**

Para comprobar los cálculos de este ejemplo, podéis usar aplicativos de uso libre, como SAGE (<http://www.sagemath.org>).

Protocolo para impedir la impersonación

Este problema de autenticación de A por parte de B , o de B por parte de A , se puede resolver con la intervención de una autoridad pública T .

El usuario A envía a T la petición que quiere conectarse con B .

- $T \rightarrow A$. La autoridad pública T envía a A los siguientes certificados:

$$C_A = D_T(Id_A, E_A, TP) \text{ i } C_B = D_T(Id_B, E_B, TP)$$

donde TP es un parámetro temporal.

- $A \rightarrow B$. El usuario A calcula una clave de sesión K_{AB} y la envía a B , debidamente firmada y cifrada $E_B(C_A, C_B, X = D_A(K_{AB}))$.

El usuario B comprueba, aplicando su clave privada, D_B , a lo que acaba de recibir; el certificado de A , C_A , su certificado, C_B , y la clave pública de A , E_A , con la cual puede calcular la clave de sesión: $K_{AB} = E_A(X)$.

A partir de ahora, los usuarios A y B , ya pueden empezar la transmisión segura con la clave de sesión K_{AB} .

Resumen de las transacciones para evitar el ataque de impersonación:

T	A
$\rightarrow C_A = D_T(Id_A, E_A, TP); C_B = D_T(Id_B, E_B, TP) \rightarrow$	

A	B
$\rightarrow E_B(C_A, C_B, X = D_A(K_{AB})) \rightarrow$	

2. Protocolos de autenticación

Los protocolos de autenticación permiten verificar la identidad de un usuario A , ante un usuario B , de forma que B pueda confirmar que es el usuario A quien le ha enviado un cierto mensaje.

Son protocolos que demuestran la identidad de A a B (en este apartado, llamaremos a **verificador** al usuario B). A veces, la identificación consiste en demostrar la posesión de un cierto secreto (que puede ser una clave), sin necesidad de revelarlo.

En general, el usuario A deberá proporcionar una respuesta a un cierto *desafío* planteado por el verificador (normalmente, dentro de un plazo de tiempo determinado). Por esto, a veces, a estos protocolos se los denomina de **desafío-respuesta**.

Para la implementación, un protocolo de autenticación debería satisfacer los requisitos siguientes:

- Tendría que implementar conjuntamente con un protocolo de intercambio de claves. Es este último el que demuestra que no solo todos los participantes son quienes dicen ser, sino que, además, puede haber un intercambio de información.
- Tendrían que encadenarse los mensajes que se van cruzando dentro de una ejecución concreta de forma que no pudieran ser sacados de contexto.
- Tendría que evitarse la utilización de sellados de tiempo, lo que se conoce como *timestamps*, como herramienta de seguridad, aun cuando son útiles desde un punto de vista administrativo y documental.
- Tendría que evitarse que la revelación de las claves intercambiadas en una sesión comprometiera el secreto de las claves intercambiadas previamente.
- Tendría que poderse añadir a los datos a cifrar, en cada paso, algún otro dato seleccionado aleatoriamente por los participantes lo que se conoce como *add your own salt*, de manera que se impedirá que la otra parte disponga de cálculos que de otra manera no podría efectuar.

Definición 2.1 (Protocolo de autenticación seguro).

Diremos que un protocolo de autenticación es seguro si, en el momento en que un usuario acepta la identidad del otro, los registros que ambos han guardado de la sesión de autenticación coinciden. Y, además, no debe ser computacionalmente eficiente que un tercer usuario pueda recuperar el desafío aceptado.

Definición 2.2 (Protocolo de autenticación directo).

Diremos que el protocolo es de autenticación **directa** si acaba intercambiando mensajes entre los participantes usando las claves generadas previamente.

Ejemplo 2.1.

Veremos dos protocolos de desafío-respuesta y, en ambos casos, usaremos un sistema criptográfico de clave pública.

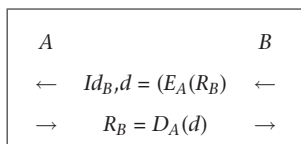
Protocolo 1:

Supongamos que A , para identificarse, quiere demostrar a B que posee la clave de descifrado D_A , de un sistema criptográfico de clave pública.

- $B \rightarrow A$. El verificador B , envía a A : $(Id_B, d = (E_A(R_B)))$, donde d es el desafío y R_B un número aleatorio escogido por B .
- $A \rightarrow B$. El usuario A , dentro del plazo establecido, con su clave privada D_A , recupera $R_B = D_A(d)$, y lo envía a B .

El verificador B , aceptará la identificación de A si el valor recibido coincide con R_B que él mismo había enviado.

Resumen de las transacciones del protocolo:

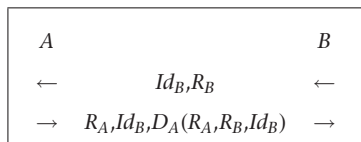


Protocolo 2:

- $B \rightarrow A$. El verificador B , envía a A : (Id_B, R_B) , donde R_B es un número escogido por B .
- $A \rightarrow B$. El usuario A , dentro del plazo establecido, usando su clave privada D_A envía: $(R_A, Id_B, D_A(R_A, R_B, Id_B))$, donde R_A es un número escogido por A , y lo envía a B .

El verificador B , aplicará, con la clave pública de A , E_A a la última componente del vector recibido y comprobará la coincidencia de Id_B y R_A con las dos primeras componentes del vector recibido y, además, que R_B coincide con el que él mismo le había enviado.

Resumen de las transacciones del protocolo:



2.1. Protocolo de tres pasos de Shamir

Este protocolo permite el envío de información secreta de A a B , sin intercambio previo de claves, a la vez que B tendrá la certeza de que el emisor es A .

Para llevar a cabo este protocolo necesitamos una función criptográfica que sea conmutativa para cada pareja de usuarios, $E_A \cdot E_B = E_B \cdot E_A$.

Protocolo

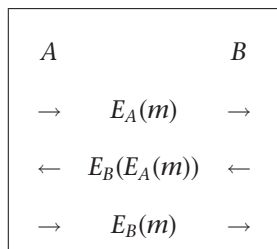
- $A \rightarrow B$. El usuario A cifra con su clave pública el mensaje m : $E_A(m)$, y lo envía a B .

El usuario B no puede calcular m , puesto que no conoce D_A .

- $B \rightarrow A$. El usuario B cifra con su clave pública lo que acaba de recibir: $E_B(E_A(m))$, y lo envía a A .
- $A \rightarrow B$. El usuario A , debido a la propiedad conmutativa presupuesta, puede descifrar con su clave privada lo que ha recibido y recuperar $E_B(m)$, que envía a B . Efectivamente: $D_A(E_B(E_A(m))) = D_A(E_A(E_B(m))) = E_B(m)$

Ahora B puede conocer m , descifrando con su clave privada lo que ha obtenido: $m = D_B(E_B(m))$ y, al mismo tiempo, estará seguro de que este mensaje lo ha enviado A .

Resumen de las transacciones del protocolo:



Nota

Este esquema no asegura la autenticación ni, en general, el secreto como se puede ver en el siguiente ejemplo.

Ejemplo 2.2.

Usaremos el sistema criptográfico de clave pública, de **Pohlig-Hellman** que, basando su seguridad en el problema del logaritmo discreto, consiste en:

- Escoger un grupo multiplicativo \mathbb{Z}_p^* , donde p es un número primo grande.
- Cada usuario elige un valor e_U , relativamente primo con $\phi(p) = p-1$ y calcula $d_U = e_U^{-1} \pmod{\phi(p)}$. La clave secreta de cada usuario será el valor d_U .
- Para un mensaje m y un criptograma c :

Cifrado: $E_A(m) = m^{e_A} \pmod{p}$ y
 Descifrado: $D_A(c) = c^{d_A} \pmod{p}$

Evidentemente, este sistema criptográfico de clave pública cumple la condición mencionada $E_A \cdot E_B = E_B \cdot E_A$.

Por lo tanto, es seguro respecto al secreto, pero no desde el punto de vista de la autenticación, como veremos en el ataque de impersonación de Massey-Omura.

Empecemos con un ejemplo numérico de este protocolo:

Tomando $p = 103$; si el usuario A escoge $e_A = 19$ ($d_A = e_A^{-1} \pmod{\phi(p)} = 19^{-1} \pmod{102} = 43$) y el usuario B escoge $e_B = 23$ ($d_B = e_B^{-1} \pmod{\phi(p)} = 23^{-1} \pmod{102} = 71$); suponiendo que el mensaje que A quiere enviar a B es $m = 10$, el protocolo de tres pasos de Shamir sería el siguiente:

- $A \rightarrow B$. El usuario A envía a B : $E_A(m) = 10^{19} \pmod{103} = 3$.

El usuario B no puede calcular $m = 10$, puesto que no conoce D_A .

- $B \rightarrow A$. El usuario B calcula $E_B(E_A(m)) = E_B(3) = 3^{23} \pmod{103} = 95$.
- $A \rightarrow B$. El usuario A puede calcular el valor de $E_B(m)$, mediante su clave privada D_A : $E_B(m) = D_A(E_B(E_A(m))) = D_A(95) = 95^{43} \pmod{103} = 27$. El usuario A envía el valor 27 a B .

Este valor coincide, efectivamente, con $E_B(10) = 10^{23} \pmod{103} = 27$.

Ataque de impersonación de Massey-Omura

Veamos ahora, como en el ejemplo anterior, que un espía C se puede interponer entre los dos usuarios A y B violando el secreto de la transmisión.

Este tipo de ataques, también son conocidos como *man-in-the-middle*.

Protocolo

Si el usuario A quiere enviar un mensaje m , cifrado, al usuario B , calcula $E_A(m) = m^{e_A} \pmod{p}$.

- $A \rightarrow B$. El usuario A envía $E_A(m)$ a B

Si un espía C intercepta este mensaje cifrado puede retornar a A :

$$E_C(E_A(m)) = (m^{e_A})^{e_C} \pmod{p}$$

- $A \leftarrow C$. el espía C envía $E_C(E_A(m))$ a A .

El usuario A no té ningún mecanismo para comprobar que lo que recibe sea de B .

Si A no desconfía, sigue con el tercer paso del protocolo de tres pasos de Shamir:

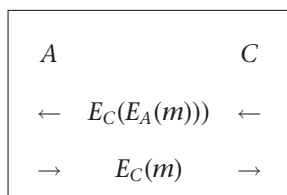
$$D_A((m^{e_A})^{e_C}) \pmod{p} = ((m^{e_A})^{e_C})^{d_A} \pmod{p} = m^{e_C} \pmod{p}.$$

- $A \rightarrow B$. El usuario A piensa que envía $E_B(m)$ y, en cambio, está enviando $E_C(m) = m^{e_C} \pmod p$.

El espía C al recibir este valor, y dado que él conoce D_C , puede calcular: $(m^{e_C})^{d_C} \pmod p = m$.

De este modo el espía C ha podido conocer el mensaje m que A quería enviar a B .

Resumen de las transacciones del ataque de impersonación



Ejemplo 2.3.

Usaremos el mismo sistema criptográfico de clave pública, de **Pohlig-Hellman** que en el ejemplo anterior.

Tomemos $p = 103$; ya conocemos las claves públicas y privadas de A y B , respectivamente: $\{e_A = 19, d_A = 43\}$ y $\{e_B = 23, d_B = 71\}$.

Suponiendo que los parámetros del espía C sean $\{e_C = 5, d_C = 41\}$ el ataque de autenticación seguiría los siguientes pasos. Supongamos que el mensaje que se quiere enviar es $m = 10$:

- $A \rightarrow B$. El usuario A calcula $E_A(m) = 10^{19} \pmod{103} = 3$ y lo envía a B .

El espía C intercepta este mensaje y calcula:

$$E_C(E_A(m)) = (m^{e_A})^{e_C} \pmod p = 3^5 \pmod{103} = 37.$$

- $A \leftarrow C$. El espía C envía a A : $E_C(E_A(m)) = 37$.

El usuario A no tiene ningún mecanismo para comprobar que lo que recibe es de B y, al no desconfiar, descifra el mensaje recibido, mediante su clave privada d_A : $D_A(E_C(E_A(m))) = D_A(37) = 37^{43} \pmod{103} = 90$

- $A \rightarrow B$. El usuario A piensa que envía a B el valor $90 = D_A(E_B(E_A(m))) = m^{e_B} \pmod p$, pero en realidad está enviando $D_A(E_C(E_A(m))) = E_C(m)$.

El espía C recibe este valor y dado que él posee d_C puede calcular: $(m^{e_C})^{d_C} \pmod p = m$. Para ello calcula: $D_C(90) = 90^{41} \pmod{103} = 10$; y, por tanto, C conoce el mensaje $m = 10$ que A enviaba a B .

2.2. Protocolo de Omura

Este es un protocolo semejante al de Diffie-Hellman de acuerdo de claves, que permite la autenticación de dos usuarios, sin que nadie más que ellos intervenga.

Sea $\alpha \in \mathbb{Z}_p$ un elemento primitivo y supongamos que cada usuario U tiene una clave secreta x_U y una clave pública $y_U = \alpha^{x_U} \pmod{p}$.

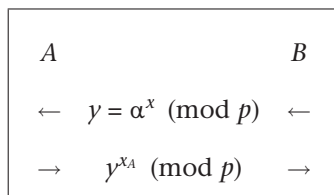
Protocolo

Una vez que A ha informado a B que él es A y que lo quiere probar, el protocolo consiste en los dos pasos siguientes:

- $B \rightarrow A$. El usuario B escoge al azar un valor x , calcula $y = \alpha^x \pmod{p}$ y envía y a A .
- $A \rightarrow B$. El usuario A calcula $y^{x_A} \pmod{p}$ y envía el resultado a B .

El usuario B verifica la igualdad entre lo que acaba de recibir ($y^{x_A} \pmod{p}$) y el cálculo a partir de la clave pública de A y el valor x por él escogido ($(y_A)^x \pmod{p}$). En caso de igualdad, acepta la autenticación de A y, de lo contrario, la rechaza.

Resumen de las transacciones del protocolo



Ejemplo 2.4.

Utilizamos los parámetros del ejemplo 1.1.

Sea el cuerpo \mathbb{Z}_p , con $p = 1999$ y el elemento primitivo $\alpha = 33$.

Supongamos las claves públicas y privadas de A y B , respectivamente:
 $\{x_A = 47, y_A = \alpha^{x_A} \pmod{p} = 1343, x_B = 117, y_B = \alpha^{x_B} \pmod{p} = 1991\}$

Los pasos del protocolo son (suponiendo que el usuario B ha sido avisado de que el usuario A quería autenticarse):

- $B \rightarrow A$. B escoge el valor $x = 13$, calcula $y = \alpha^{13} \pmod{1999} = 319$ y lo envía a A .
- $A \rightarrow B$. A calcula $y^{x_A} \pmod{p} = 319^{47} \pmod{1999} = 1465$ y envía el resultado a B .

El usuario B verifica si el valor recibido coincide con $(y_A)^x \pmod{p}$. Efectivamente:
 $1343^{13} \pmod{1999} = 1465$ y el usuario B da por autenticado al usuario A .

2.3. Protocolo de Needham-Schroeder

Este es un protocolo de autenticación e intercambio de claves, usando un sistema criptográfico de clave pública y mediante un centro de distribución de

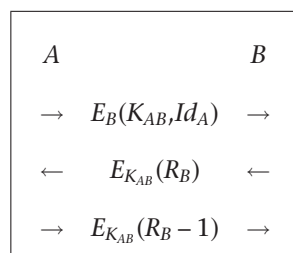
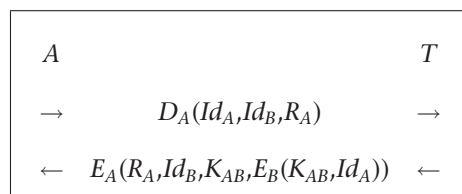
claves, *KDC*, gestionado por la autoridad pública *T*, que sigue los siguientes pasos.

Protocolo

- $A \rightarrow T$. El usuario *A* envía a *T*: $D_A(Id_A, Id_B, R_A)$, donde R_A es un número aleatorio escogido por *A*.
- $T \rightarrow A$. La autoridad *T* devuelve a *A*: $E_A(R_A, Id_B, K_{AB}, E_B(K_{AB}, Id_A))$, donde K_{AB} es la clave de sesión, generada por el *KDC*, que han de utilizar *A* y *B* en su intercambio.
- $A \rightarrow B$. El usuario *A* envía $E_B(K_{AB}, Id_A)$ a *B*.
- $B \rightarrow A$. El usuario *B* envía, cifrando con la clave de sesión: $E_{K_{AB}}(R_B)$ a *A*, donde R_B es un número aleatorio escogido por *B*.
- $A \rightarrow B$. El usuario *A* devuelve $E_{K_{AB}}(R_B - 1)$ a *B*.

Cuando el usuario *B*, descifrando con la clave de sesión, K_{AB} , recibe el valor de $R_B - 1$, da por autenticado al usuario *A*.

Resumen de las transacciones del protocolo:



2.4. Protocolo de Kerberos

Este es un protocolo de distribución de claves de sesión, mediante un centro de distribución de claves, *KDC*, que proporciona autenticación del usuario y establece la clave de sesión entre dos usuarios *A* y *B*, usando un sistema criptográfico de clave privada y terceras partes de confianza.

Hay otras versiones del protocolo para poder utilizar sistemas criptográficos de clave pública.

Consideramos T , el servidor de autenticación Kerberos, como tercera parte de confianza.

El protocolo de Kerberos se basa en el protocolo que acabamos de ver de Needham-Schroeder. Se usan vales y sellado de tiempo para asegurar la identidad de los usuarios.

La utilización de los vales y del periodo de validez de la clave K_{AB} evita que pueda haber ataques usando claves de sesión utilizadas previamente.

Protocolo

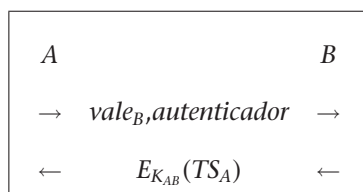
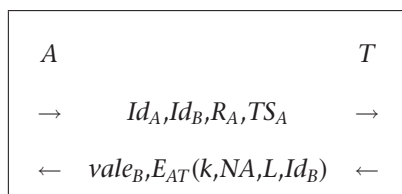
- $A \rightarrow T$. El usuario A envía T : (Id_A, Id_B, R_A, TS_A) , donde R_A es un valor aleatorio escogido por A y TS_A es un sellado de tiempo de A .
- $T \rightarrow A$. El servidor de autenticación envía a A : $(vale_B, E_{AT}(K_{AB}, R_A, L, Id_B))$, donde E_{AT} es la operación de cifrar con la clave privada que comparten A y T , L indica el periodo de validez de la clave K_{AB} y el $vale_B = E_{BT}(K_{AB}, Id_A, L)$.
- $A \rightarrow B$. El usuario A envía a B : $(vale_B, autenticador)$, donde $autenticador = E_{K_{AB}}(Id_A, TS_A)$, creado por A y que contiene el sellado de tiempo, TS_A cifrado con la clave de sesión K_{AB} .
- $B \rightarrow A$. El usuario B envía a A : $E_{K_{AB}}(Id_A, TS_A)$

Autenticación Kerberos

Kerberos permite la autenticación de A ante B . Incluye sellado de tiempo y una autoridad de certificación *on-line*, que comparte una clave privada con cada usuario para hacer las operaciones de cifrado E_{AT}, E_{BT} . El $vale_B$ va acompañado del autenticador.

A partir de aquí, los usuarios A y B comparten una clave de sesión K_{AB} y están mutuamente autenticados.

Resumen de las transacciones del protocolo:



2.5. Protocolo STS

Este protocolo responde al acrónimo (*station-to-station*) y fue creado por Diffie, Oorschot, Wiener como una propuesta de protocolo seguro para la autenticación y el intercambio de claves, basado en un sistema criptográfico de clave pública.

Supondremos que, al igual que en el protocolo de Diffie-Hellman, todos los usuarios conocen el valor del número primo p y del elemento primitivo $\alpha \in \mathbb{Z}_p$ y que cada usuario U sube (Id_U, γ_U) al directorio público TPD (recordar que cada usuario había escogido un valor $x_U \in \mathbb{Z}_p^*$, que guardaba secreto, y calculaba el valor público $\gamma_U = \alpha^{x_U} \pmod{p}$).

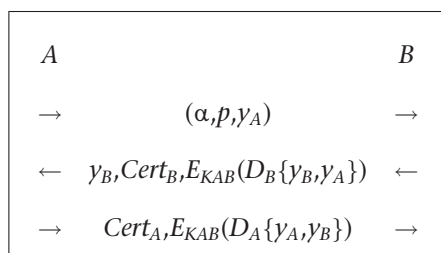
Protocolo

- $A \rightarrow B$. El usuario A escoge un valor x_A y calcula $\gamma_A = \alpha^{x_A} \pmod{p}$. Entonces, envía a B : (α, p, γ_A) .
- $B \rightarrow A$. El usuario B escoge un valor x_B y calcula $\gamma_B = \alpha^{x_B} \pmod{p}$ así como también calcula la clave de sesión $K_{AB} = \gamma_A^{x_B} \pmod{p}$. Entonces, envía a A : $(\gamma_B, Cert_B, E_{K_{AB}}(D_B\{\gamma_B, \gamma_A\}))$, donde $\{\gamma_B, \gamma_A\}$ significa el resultado de aplicar una función de *hash* a los parámetros γ_B, γ_A .
- $A \rightarrow B$. El usuario A también puede calcular la clave de sesión $K_{AB} = \gamma_B^{x_A} \pmod{p}$ y envía a A : $(Cert_A, E_{K_{AB}}(D_A\{\gamma_A, \gamma_B\}))$

En este protocolo, $Cert_A = (Id_A, E_A, \alpha, p, D_T(Id_A, E_A, \alpha, p))$, donde $D_T(Id_A, E_A, \alpha, p)$ es la firma de una autoridad de certificación T sobre estos parámetros.

En el último paso del protocolo, el usuario A queda autenticado ante el usuario B .

Resumen de las transacciones del protocolo:



Consideraciones de diseño

Los valores p y α los escoge cada usuario y mejor si son diferentes, puesto que los ataques más eficientes al logaritmo discreto se basan en la construcción de tablas a partir del conocimiento de p . En el segundo paso B envía a A su certificado $Cert_B = (Id_B, E_B, D_T(B, E_B))$. De este modo A podrá calcular E_B y comprobar que γ_B se corresponde con el *hash* firmado $D_B\{\gamma_B, \gamma_A\}$. Esto da autenticidad a B . El usuario B tiene que utilizar los parámetros pasados por A en el primer paso, en lugar de buscarlos en un directorio público. En el tercer paso el certificado de A asegura a B que estos parámetros son correctos.

2.6. Otros protocolos: ISO, CCITT X.509, SSL

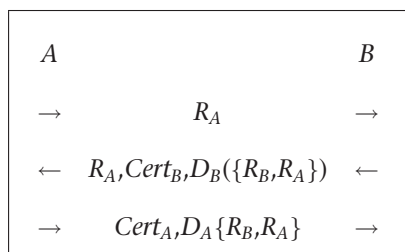
Para acabar este apartado, veamos otros protocolos que se pueden considerar estándares:

1) ISO (CD 9798-3. 1991). Es equivalente al protocolo STS en el cual sustituimos las exponenciales α^x, α^y por valores aleatorios R_A, R_B . Sin embargo, de este modo obtenemos un protocolo de autenticación pero sin intercambio de claves.

Protocolo

- $A \rightarrow B$. El usuario A envía R_A a B .
- $B \rightarrow A$. El usuario B envía R_A , junto con $(Cert_B, D_B(\{R_B, R_A\}))$, a A .
- $A \rightarrow B$. El usuario A envía B : $Cert_A, D_A(\{R_B, R_A\})$

Resumen de las transacciones del protocolo:



2) CCITT X.509. Es un algoritmo de intercambio de claves, con autenticación, de tres pasos y se basa en un sistema criptográfico de clave pública.

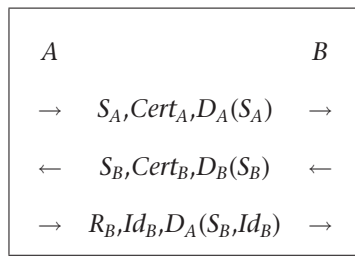
El usuario A que quiere intercambiar una clave K_{AB} , con B , tiene una copia autenticada de E_B .

Protocolo

- $A \rightarrow B$. El usuario A envía a B : $(S_A, Cert_A, D_A(S_A))$, donde $S_A = (R_A, Id_A, E_B(K_{AB}))$, y R_A es un número aleatorio escogido por A .
- $B \rightarrow A$. El usuario B envía a A : $(S_B, Cert_B, D_B(S_B))$, donde $S_B = (R_B, Id_B, E_A(K_{AB}))$, y R_B es un número aleatorio escogido por B .
- $A \rightarrow B$. El usuario A envía a B : $(R_B, Id_B, D_A(S_B, Id_B))$.

Evidentemente, B ha podido autenticar A y descifrando: $D_B(E_B(K_{AB})) = K_{AB}$, obtiene la clave de sesión K_{AB} que empleará para comunicarse con A .

Resumen de las transacciones del protocolo:



3) **SSL (Secure Socket Layer)**. Es un protocolo abierto creado por *Netscape* y ha llegado a ser un estándar para Internet. Se implementa entre la capa de aplicación (HTTP, Telnet, FTP,...) y la capa de transporte (TCP).

El SSL es transparente para la aplicación que lo utiliza y añade los siguientes servicios seguros a una conexión TCP/IP:

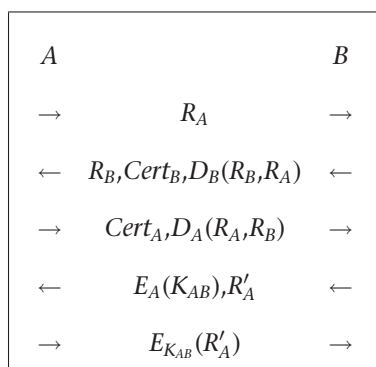
- Cifrado de datos usando los sistemas criptográficos de clave privada: DES, RC4.
- Autenticación, usando los sistemas criptográficos de clave pública: RSA, DSA. Implementa tanto la autenticación mutua emisor/receptor como la de servidor/cliente.
- Integridad de los datos mediante funciones *hash MAC (message authentication code)*: SHA, MD5.

El protocolo SSL sigue los siguientes pasos:

- 1) $A \rightarrow B$. El usuario *A* envía un valor aleatorio R_A a *B*.
- 2) $B \rightarrow A$. El usuario *B* envía a *A*: $(R_B, Cert_B, D_B(R_B, R_A))$.
- 3) $A \rightarrow B$. El usuario *A* envía, opcionalmente, a *B*: $(Cert_A, D_A(R_A, R_B))$.
- 4) $B \rightarrow A$. El usuario *B* escoge una clave de sesión K_{AB} y envía a *A*: $(E_A(K_{AB}), R'_A)$.
- 5) $A \rightarrow B$. El usuario *A* envía a *B*: $E_{K_{AB}}(R'_A)$.

En este último paso, el usuario *B* cifra con la clave K_{AB} , el valor aleatorio R'_A que le había enviado *A* en el paso anterior.

Resumen de las transacciones del protocolo:



Fases del SSL

- Fragmentación del mensaje en bloques de medida menor que 214 bytes. TCP añade fiabilidad.
- Posibilidad de compresión (opcional).
- Cálculo de un MAC.
- Cifrado de los datos y del MAC (*message authentication code*).
- Transmisión por TCP.

Observaciones

- El paso tercero es opcional.
- En el cuarto paso, *B* ha escogido la clave de sesión, K_{AB} .
- En el quinto paso, *A*, cifra con la clave K_{AB} , el valor aleatorio R'_A que le había enviado *B*, en el paso anterior.

2.7. Protocolos de identificación de conocimiento nulo

En los protocolos de desafío-respuesta, vistos anteriormente, aunque el usuario A no revele el secreto, el usuario B puede conseguir alguna información durante el proceso.

Definición 2.3 (Prueba de conocimiento nulo).

Una prueba de conocimiento nulo es un proceso interactivo donde el candidato convence al verificador, hasta un nivel aceptable, que conoce, o tiene algún secreto, sin que el verificador pueda extraer ninguna información de la prueba que no pudiera haber extraído por cualquier otro procedimiento, con o sin la participación del candidato o, incluso, si este miente en la prueba.

Condiciones de una prueba de conocimiento nulo

- 1) Si A posee el secreto, siempre podrá convencer a B que acepte su demostración.
- 2) Si A no posee el secreto, la probabilidad que engañe a B puede hacerse tan pequeña como se quiera, repitiendo el procedimiento el suficiente número de veces.

Las pruebas de conocimiento nulo adoptan la forma de una demostración interactiva, implicando un cierto número de etapas y, en cada una de ellas, se seguirá el siguiente protocolo:

Protocolo básico

- $A \rightarrow B$. El usuario A quiere probar algo al verificador B y le envía algún elemento para su identificación.
- $B \rightarrow A$. El verificador B presenta un desafío a A .
- $A \rightarrow B$. El usuario A tiene que efectuar unos cálculos privadamente y enviar al verificador B una respuesta al desafío planteado.

Si alguna de las respuestas es incorrecta, B deduce que A no dispone del secreto y rechaza su identidad. Por el contrario, si en todas las etapas la respuesta es correcta, entonces B acepta que A conoce (o tiene) el secreto.

Protocolo de Fiat-Shamir

Sea $n = p_1 \cdot p_2$, que todo el mundo puede conocer, generado por una tercera parte de confianza T y que mantiene secretos los valores de los números primos p_1 y p_2 .

Cada usuario U elige un elemento $x_U \in \mathbb{Z}_n^*$ y calcula $y_U = x_U^2 \pmod{n}$. Así, la clave secreta de A será x_A y la clave pública y_A , que se registra en el directorio público de T .

Protocolo

- $A \rightarrow B$. El usuario A genera al azar un valor $r \in \mathbb{Z}_n^*$, calcula $y_1 = r^2 \pmod n$ y lo envía a B , junto con un mensaje diciendo que quiere probar su identidad.
- $B \rightarrow A$. El verificador B envía a A un bit, al azar: $y_2 \in \{0,1\}$.
- $A \rightarrow B$ El usuario A calcula y_3 de la siguiente manera:

$$\text{si } y_2 = 0, \text{ entonces } y_3 = r \pmod n$$

$$\text{si } y_2 = 1, \text{ entonces } y_3 = r \cdot x_A \pmod n$$

y envía a B el valor y_3 .

Finalmente, la verificación la lleva a cabo el usuario B que comprueba que

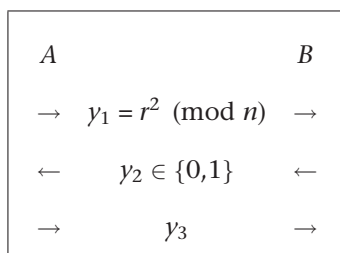
$$\text{si } y_2 = 0, \text{ entonces } y_3^2 = r^2 \pmod n = y_1$$

$$\text{si } y_2 = 1, \text{ entonces } y_3^2 = r^2 \cdot y_A \pmod n = y_1 \cdot y_A \pmod n$$

Si no se cumple la verificación, B rechaza la identidad de A .

Este protocolo tiene que llevarse a cabo unas cuantas veces. Si se repite k veces, la probabilidad que alguien impersona a A es 2^{-k} , y puede ser tan pequeña como queramos, dependiente del valor de k .

Resumen de las transacciones del protocolo:



Impersonación

Un usuario C impersonando a A puede enviar a B $y_1 = \frac{r^2}{y_A} \pmod n$. Entonces, podría responder correctamente $y_3 = r$, solo en el supuesto de que $y_2 = 1$. Si $y_2 = 0$ no podrá efectuar los cálculos. Por lo tanto, solo tiene una probabilidad de acierto de 0,5 en cada ronda.

Ejemplo 2.5. Supongamos los valores, $n = p_1 \cdot p_2 = 5 \cdot 11 = 55$; la clave secreta de A , $x_A = 13$, y la clave pública de A , $y_A = x_A^2 \pmod n = 13^2 \pmod 55 = 4$

Con estas premisas, el protocolo sigue los siguientes pasos:

- $A \rightarrow B$. El usuario A toma el valor $r = 30$ y calcula $y_1 = r^2 \pmod n = 30^2 \pmod 55 = 20$ y lo envía a B , junto con un mensaje diciendo que quiere probar su identidad.
- $B \rightarrow A$. El verificador B envía a A un bit, al azar: $y_2 \in \{0,1\}$. Supongamos $y_2 = 1$.
- $A \rightarrow B$. El usuario A calcula y_3 .
Como que $y_2 = 1$, entonces $y_3 = r \cdot x_A \pmod n = 30 \cdot 13 \pmod 55 = 5$ y lo envía a B .

Finalmente, la verificación la lleva a cabo el usuario B , que comprueba: puesto que $y_2 = 1$, si $y_3^2 = r^2 \cdot y_A \pmod{n} = y_1 \cdot y_A \pmod{n}$.

Efectivamente: $20 \cdot 4 \pmod{55} = 25 = 5^2$, y da por buena la identidad de A .

Protocolo de Schnorr

Este protocolo de identificación se basa en el problema del logaritmo discreto y hace intervenir a una tercera parte de confianza que desempeña las siguientes acciones:

Escoge dos primos p y q , de unos 1024 bits y 160 bits respectivamente, tales que $q \mid p-1$. También escoge un valor β , tal que $1 \leq \beta \leq p-1$, de orden q , y un parámetro de seguridad t , tal que: $t \geq 40$ y $2^t \leq q$.

La autoridad T envía a cada participante, firmada mediante su clave privada, $D_T(p, q, \beta)$.

Por otra parte, el usuario que se quiere identificar, A , dispone de su identificación Id_A , su clave privada, que consiste en un número a tal que $0 \leq a \leq q-1$ y la correspondiente clave pública es $v = \beta^{-a} \pmod{p}$; así como también del certificado emitido por T : $Cert_A = (Id_A, v, D_T(Id_A, v))$.

Protocolo

$A \rightarrow B$. El usuario A escoge aleatoriamente un valor r , tal que $1 \leq r \leq (p-1)$ y envía a B el $Cert_A$ y $x = \beta^r \pmod{p}$

$B \rightarrow A$. Después de que B ha verificado la clave pública de A , le envía el valor e tal que $1 \leq e \leq 2^t$.

$A \rightarrow B$. El usuario A envía al verificador B : $y = (a \cdot e + r) \pmod{q}$.

El verificador B acepta la identidad de A si $\beta^y \cdot v^e \pmod{p} = x$.

Efectivamente, como β es de orden q , entonces:

$$\beta^y \cdot v^e \pmod{p} = \beta^{a \cdot e + r} \cdot v^e \pmod{p} = \beta^{a \cdot e} \cdot \beta^r \cdot \beta^{-a \cdot e} \pmod{p} = \beta^r \pmod{p} = x$$

Resumen de las transacciones del protocolo:

A		B
→	$Cert_A, x = \beta^r \pmod{p}$	→
←	$e : 1 \leq e \leq 2^t$	←
→	$y = (a \cdot e + r) \pmod{q}$	→

Ejemplo 2.6.

Supongamos que una tercera parte de confianza T ha escogido:

- Los valores de los primos $p = 103$ y $q = 17$. Obsérvese que $\frac{102}{17} = 6$.
- Para encontrar el valor de β busca un elemento primitivo en \mathbb{Z}_p ; por ejemplo $\alpha = 6$ y toma $\beta = \alpha^6 \pmod{103} = 100$. Efectivamente, de la construcción se deriva que el orden de β es 17.
- Para simplificar el ejemplo, omitiremos sin perder rigor el parámetro de seguridad t , $t \geq 40$ y $2^t \leq q$.

La autoridad T envía a cada participante copia autenticada de $(p = 103, q = 17, \beta = 100)$.

Los parámetros de A , aparte del identificador Id_A , son su clave privada que suponemos $a = 11$ y su correspondiente clave pública $v = 100^{-11} \pmod{103} = 8$. Así, el certificado emitido por T es: $Cert_A = (Id_A, 8, D_T(Id_A, 8))$.

Protocolo

$A \rightarrow B$. El usuario A escoge aleatoriamente un valor $r = 5$ y envía a B el $Cert_A$ y $x = 100^5 \pmod{103} = 66$.

$B \rightarrow A$. Después de que B ha verificado la clave pública de A , le envía el valor que ha escogido: $e = 13$.

$A \rightarrow B$. El usuario A envía al verificador B : $y = 11 \cdot 13 + 5 \pmod{17} = 12$.

El verificador B acepta la identidad de A puesto que $\beta^y \cdot v^e \pmod{p} = 100^{12} \cdot 8^{13} \pmod{103} = 64 \cdot 30 \pmod{103} = 66$ que coincide con el valor de $x = 66$ previamente calculado.

3. Transacciones electrónicas seguras: dinero electrónico

Un problema usual en el mundo de las transacciones es que un usuario A quiera hacer una transacción con un banco, B , de forma que, de una parte, B debe poder autenticar a A y, por otra parte, A debe tener un certificado de B que le asegure que este no negará la transacción ya realizada.

El problema es fácil de resolver si utilizamos las firmas electrónicas como sustitutos del dinero, como propone David Chaum (a finales de los ochenta).

3.1. Protocolo de Chaum

Supongamos un comprador, A , que quiere interactuar con B , que ahora consideraremos que es el banco, para pedirle poner a disposición de una cierta cantidad de dinero que, posiblemente más tarde, querrá utilizar para poder pagar alguna compra.

Protocolo

- $A \rightarrow B$. El usuario A genera, al azar, un número m grande y crea un documento personalizado, doc_{AB} , en el cual explica al banco que quiere disponer de $X \text{ €}$. Firma el número escogido, calculando $D_A(m)$ y usando la clave pública de B le envía: $(Id_A, E_B(D_A(m)), doc_{AB})$
- $B \rightarrow A$. El banco recibe los mensajes, identifica A y lee doc_{AB} . Con la clave pública de A , y con su clave privada, puede recuperar el número m . Ahora ya puede descontar los $X \text{ €}$ de la cuenta de A . El banco firmará el número m , añadiendo un documento doc_{BA} firmado, $D_B(doc_{BA})$. Enviará a A estas dos firmas, cifradas con la clave pública de A : $(E_A(D_B(m)), E_A(D_B(doc_{BA})))$.

Al usuario A le interesa tener $(D_B(doc_{BA}))$, y no solo doc_{BA} , para poder demostrar ante terceros, en caso necesario, que el banco le ha descontado $X \text{ €}$ de su cuenta y que estos corresponden a su número m .

Si ahora hacemos intervenir a un vendedor V , cuando A va a comprar a V , el correspondiente protocolo seguirá los siguientes pasos:

- $A \rightarrow V$. El comprador A envía a V : $(Id_A, D_B(m), D_B(doc_{BA}))$. El vendedor podrá autenticar los mensajes firmados por B y conocer la identidad del comprador A y el hecho de que dispone de un valor de $X \text{ €}$.

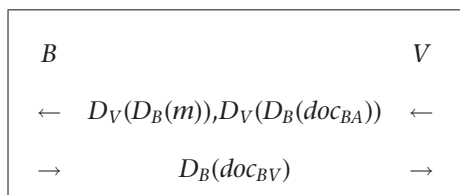
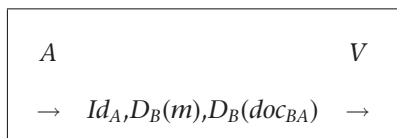
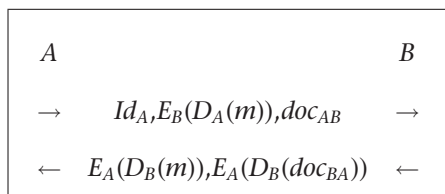
Nota

El comprador A podría enviar estos valores cifrados, con E_V , y firmados con D_A : $D_A(E_V(D_B(m)))$ y $D_A(E_V(D_B(doc_{BA})))$.

- $V \rightarrow B$. El vendedor V envía al banco: $D_V(D_B(m))$. El banco comprueba que el mensaje es correcto (correspondiente a un número firmado previamente por el propio banco); deposita en la cuenta de V los $X \text{ €}$ y escribe en una lista de números caducados el número m , para evitar que le vuelva a ser presentado otra vez.
- $B \rightarrow V$. El banco envía firmado un doc_{BV} de la transacción hecha al vendedor V : $D_B(doc_{BV})$.

A partir de este momento, V ya puede dar la mercancía a A , junto con un documento firmado, $D_V(doc_{VA})$ de haber cobrado del banco.

Resumen de las transacciones del protocolo:



Ejemplo 3.1.

Utilizemos como parámetros de los criptosistemas RSA del usuario A , del banco B y del vendedor V , respectivamente:

- $n_A = 7 \cdot 17 = 119$ y $e_A = 5$; entonces $\phi(n_A) = 6 \cdot 16 = 96$ y $d_A = 5^{-1} \pmod{96} = 77$.
Por lo tanto: clave pública $(e_A, n_A) = (5, 119)$ y clave secreta $(d_A, n_A) = (77, 119)$.
- $n_B = 5 \cdot 11 = 55$ y $e_B = 3$; entonces $\phi(n_B) = 4 \cdot 10 = 40$ y $d_B = 3^{-1} \pmod{40} = 27$.
Por lo tanto: clave pública $(e_B, n_B) = (3, 55)$ y clave secreta $(d_B, n_B) = (27, 55)$.
- $n_V = 3 \cdot 19 = 57$ y $e_V = 7$; entonces $\phi(n_V) = 2 \cdot 18 = 36$ y $d_V = 7^{-1} \pmod{36} = 31$.
Por lo tanto: clave pública $(e_V, n_V) = (7, 57)$ y clave secreta $(d_V, n_V) = (31, 57)$.

El protocolo de Chaum seguirá los siguientes pasos:

Protocolo

El usuario A hará:

Generar al azar $m = 8$.

Calcular $D_A(m) = m^{d_A} \pmod{n_A} = 8^{77} \pmod{119} = 43$.

Generar un documento personalizado, doc_{AB} en el que explica al banco que quiere disponer de 300 €.

- $A \rightarrow B$. El comprador A enviará al banco: $(Id_A, E_B(D_A(m)), doc_{AB})$.

Es decir: su identidad, $E_B(D_A(m)) = E_B(43) = 43^3 \pmod{55} = 32$ y el documento doc_{AB} .

El banco, B , una vez recibido $E_B(D_A(m)) = 32$, hará:

Aplicar $D_B(E_B(D_A(m))) = D_B(32) = 32^{27} \pmod{55} = 43 (= D_A(m))$.

Aplicar $E_A(D_A(m)) = E_A(43) = 43^5 \pmod{119} = 8 (= m)$.

Descontar 300 € de la cuenta de A y firmar el número m :

$D_B(m) = D_B(8) = 8^{27} \pmod{55} = 2$.

Además, el banco redacta un documento, doc_{BA} y lo envía firmado a A : $D_B(doc_{BA})$.

- $B \rightarrow A$. El banco envía a A :

$E_A(D_B(m)) = E_A(2) = 2^5 \pmod{119} = 32$.

$E_A(D_B(doc_{BA}))$.

Supongamos ahora la intervención del vendedor V , con cuentas en el mismo banco B .

Si A hace un pedido al vendedor por el valor de 300 €, le entrega $D_B(m) = 2$ y $D_B(doc_{BA})$.

- $V \rightarrow B$. El vendedor entrega al banco:

$D_V(D_B(m)) = D_V(2) = 2^{31} \pmod{57} = 41$ y $D_V(D_B(doc_{BA}))$.

El banco hará:

Comprobar que el mensaje es correcto $E_V(D_V(D_B(m))) = D_B(m)$ y que también lo es el documento doc_{VB} .

Depositar en la cuenta del vendedor 300 €.

Escribir en una lista de números caducados el número m para evitar que se pueda volver a usar.

- $B \rightarrow V$. El banco envía firmado un doc_{BV} de la transacción hecha al vendedor V : $D_B(doc_{BV})$

A partir de este momento, V ya puede dar la mercancía a A , junto con un documento firmado, $D_V(doc_{VA})$ de haber cobrado del banco.

3.2. Transacciones sin rastro. Firmas digitales ciegas de Chaum

En el protocolo que acabamos de describir hay *autenticidad* pero no *privacidad*. El banco sabe (por el número m del billete) que A ha comprado a V y, en general, podrá seguir el rastro de las operaciones comerciales de A .

Para obviar este problema, D. Chaum (1992) propone una variante del protocolo usando firmas digitales ciegas, que consiste en hacer firmar al banco un documento que contiene un número escondido.

De este modo, el usuario A sacará a la luz el número y lo hará servir junto con el documento firmado por el banco, cuando le convenga. Por otra parte, el banco no sabrá a quién corresponden los números de los billetes que se utilizan.

El método de Chaum es válido para cualquier sistema criptográfico de clave pública (como RSA) que cumpla: $E_U(xy) = E_U(x)E_U(y)$ y $D_U(xy) = D_U(x)D_U(y)$ para cualquier usuario U .

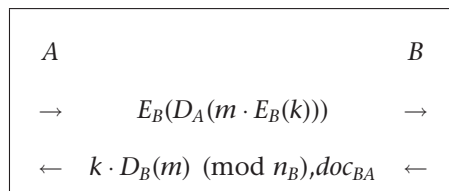
Vamos a desarrollar este protocolo suponiendo que estamos usando el criptosistema *RSA*.

Protocolo

- $A \rightarrow B$. El usuario A , antes de dar el número m al banco, selecciona otro número grande, k , escogido al azar y del cual puede calcular $k^{-1} \pmod{n_B}$. El usuario A calcula $E_B(k)$ y envía: $m \cdot E_B(k) \pmod{n_B}$, firmado y secreto a B . Es decir, envía a B : $E_B(D_A(m \cdot E_B(k)))$.
- $B \rightarrow A$. El banco descifra y autentica el mensaje recibido y obtiene $m \cdot E_B(k) \pmod{n_B}$. A continuación firma este mensaje calculando $D_B(m \cdot E_B(k)) = k \cdot D_B(m) \pmod{n_B}$. (Observar que el banco no conoce el número m que acaba de firmar, solo conoce $k \cdot D_B(m) \pmod{n_B}$).
Ahora el banco devuelve, como en el caso general, un documento a A , doc_{BA} , y también el número firmado $k \cdot D_B(m) \pmod{n}$.

El usuario A calcula $k \cdot D_B(m) \cdot k^{-1} \pmod{n_B} = D_B(m)$ y obtiene un número firmado por el banco que, cuando el vendedor lo presente al banco, será hecho efectivo y, además, el banco desconocerá al propietario del número del billete.

Resumen de las transacciones del protocolo:



Nivel de seguridad de las firmas ciegas

Las firmas ciegas garantizan la integridad, confidencialidad y autenticidad de los datos, al mismo tiempo que se garantiza el anonimato del comprador. El comprador está protegido ante posibles actuaciones fraudulentas de falta de entrega de mercancías o servicios por parte de vendedores, puesto que en cualquier momento el comprador puede revelar su identidad y, en este caso, seguir la traza del flujo del dinero.

Ejemplo 3.2. (Firma ciega de Chaum)

A desea que el Banco, B , le firme el mensaje $m = 65$.

Supongamos que las claves pública y privada de B son: $n_B = 851$, $e_B = 13$, $d_B = 61$.

(Los parámetros de B son: $p_B = 23$, $q_B = 37$, $\phi(n_B) = 792$, todos ellos guardados en un lugar seguro, junto con d_B).

Supongamos que A escoge $k = 51$, y calcula $51^{-1} \pmod{851} = 267$.

- $A \rightarrow B$. El usuario A calcula:

$$M = m \cdot k^{e_B} \pmod{n_B} = 65 \cdot 51^{13} \pmod{851} = 65 \cdot 458 \pmod{851} = 836 \text{ y envía } M \text{ al Banco.}$$

- $B \rightarrow A$. El Banco hace:

firma este mensaje recibido, M , con su clave privada:

$$M^{d_B} \pmod{n_B} = 836^{61} \pmod{851} = 220,$$

y lo envía a A , junto con un documento doc_{BA} .

El usuario A , como conoce $k^{-1} \pmod{n_B} = 267$, puede calcular: $276 \cdot 220 \pmod{851} = 21$, que es la firma del banco de m , sin que el banco conozca m .

3.3. Sistemas de pago electrónicos

Con la aparición y generalización del comercio electrónico se ha hecho necesaria la creación de sistemas electrónicos de pago adaptados a la situación no presencial de los usuarios involucrados.

Los sistemas de pago electrónico todavía utilizan, mayoritariamente, dinero asociado a una tarjeta emitida por una entidad financiera, pero hoy ya podemos hablar de dinero electrónico, en el sentido que acabamos de ver, que lo hace un sistema de pago más adecuado, y más seguro, en las transacciones electrónicas.

Los sistemas de pago electrónico en la actualidad son, básicamente:

- **Cheques electrónicos.** Sustitutos de los cheques de papel, consisten en un mensaje con una firma digital, representando un valor monetario, que se hace efectivo a través de una tercera parte de confianza y hace uso de las redes interbancarias existentes.
- **Moneda electrónica.** Sustituta de la moneda física que debe preservar el anonimato del comprador, permitiendo realizar pagos que no queden registrados y no vinculen a los usuarios con sus compras. Cuando la transferencia se hace *off-line*, se deberá garantizar la que sea posible la transferencia y la seguridad frente a la falsificación o del uso de la misma moneda en más de un pago.

Simplificación

Para más claridad en los pasos del protocolo, nos hemos ahorrado firmarlo y cifrarlo.

El banco no conoce el número que firma

Con esta operación, el banco ha firmado el mensaje original $m = 65$, sin conocer su valor. Observar que 21 es el mismo valor que obtendría el banco si hubiera firmado con su clave privada el mensaje $m = 65$: Es decir: $65^{61} \pmod{851} = 21$.

- **Tarjeta de crédito.** Aun cuando todavía es el sistema de pago más utilizado, los receptores se pueden poner en contacto con el banco para verificar la disponibilidad de fondo, aunque, normalmente, no se realiza la verificación de la identidad del usuario. A diferencia de la moneda electrónica, las tarjetas de crédito identifican a su propietario y, además, los pagos se pueden vincular entre ellos.
- **Micropagos.** Diseñados especialmente para reducir los costes de comunicación, almacenamiento y procesamiento relacionados con el pago cuando las cantidades a transferir son pequeñas. De este modo se permite un cierto relajamiento de las medidas de seguridad, puesto que los riesgos están más controlados, y en la mayoría de casos el anonimato del usuario que realiza el pago se sacrifica para reducir costes.

Los sistemas de pago vienen definidos por las reglas mediante las cuales los participantes en una operación de compra/venta (comprador, bancos emisor y adquirente, comercio, pasarela de pago y autoridad certificadora) intercambian dinero y productos o servicios.

Los procedimientos, o pasos, en los sistemas de pago con moneda electrónica son, básicamente:

1) Establecimiento de la cuenta. Es una operación que se realiza una sola vez, en la cual se vincula la identidad o el seudónimo del comprador a la nueva cuenta. El comprador dispondrá de un par de claves correspondientes a un sistema criptográfico de clave pública y de un certificado autenticando su clave pública.

2) Retirada de fondo: obtención de la moneda electrónica. La entidad financiera emisora carga en la cuenta del comprador el valor solicitado previamente para llevar a cabo una compra o el valor de la moneda electrónica solicitada. La moneda electrónica puede ser de débito si se extrae su valor de la cuenta del usuario antes del pago (este procedimiento requiere la autenticación del usuario), o de crédito si el pago se hace posteriormente y, en este caso, la identificación de la cuenta del pagador se realiza durante el depósito.

3) Pago. El comprador entrega la información necesaria y suficiente al comercio para hacer el pago de los bienes que quiere comprar. También puede dar esta información a la pasarela de pago.

4) Depósito. La entidad financiera adquirente recibe la información y cobra de la entidad emisora, a través de la red financiera una vez ha comprobado que no se ha depositado dos veces la misma moneda. Finalmente, ingresa el valor de la compra, o de las monedas electrónicas implicadas, en la cuenta del comercio. En sistemas que permiten la transferencia de las monedas electrónicas también puede optar por transferir la moneda a un tercero.

Una primera clasificación de los sistemas de pago puede venir marcada por el importe a transferir. Así hablaremos de **micropagos**, cuando el importe de la operación es inferior a los 10 €. Por encima de esta cantidad se hablará de **macropagos**. Aunque todo sistema de pago debe satisfacer una serie de requisitos de seguridad, es obvio que en el caso de micropagos estos serán más laxos para que el coste de la seguridad sea proporcionado al valor que se transfiere. No es lo mismo proteger un pago de 10 €, que proteger uno de 100.000 €.

Una segunda clasificación es la de sistemas *on-line* y sistemas *off-line*. Los protocolos *on-line* exigen el acceso a un servidor para cada transacción.

Para los sistemas de pago *on-line*, el concepto de dinero electrónico, propuesto por D. Chaum, permite conjugar las prestaciones que ofrecen las redes telemáticas con las propiedades intrínsecas de los sistemas de pago tradicional: anonimato, privacidad y dificultad de falsificación. La diferencia entre el dinero electrónico y el de curso legal actual es su soporte, que pasa de ser físico a una cadena de bits.

La no falsificación, o evitar el doble uso, del dinero electrónico exige una verificación, contra una cierta base de datos, para asegurarse de que no haya sido utilizado previamente. Si la verificación se hace *off-line* necesita mayor protección del anonimato, es decir, que no se pueda establecer ningún vínculo entre el dinero electrónico y la identidad del propietario, que si se hace *on-line*. Tal como hemos visto antes, estos sistemas se basan en firmas ciegas.

En general, el dinero electrónico está pensado para ser utilizado en Internet o, actualmente, en sistemas basados en terminales móviles. Se trata de reemplazar las monedas y billetes de curso legal por un sistema informacional, manteniendo las prestaciones de anonimato y no trazabilidad utilizando criptografía de clave pública. Los billetes y monedas electrónicas se almacenan localmente, pero se requiere que el usuario tenga una cuenta bancaria asociada, a partir de la que se descontará la cantidad que corresponda al billete o moneda electrónica.

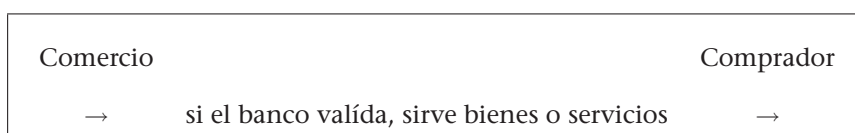
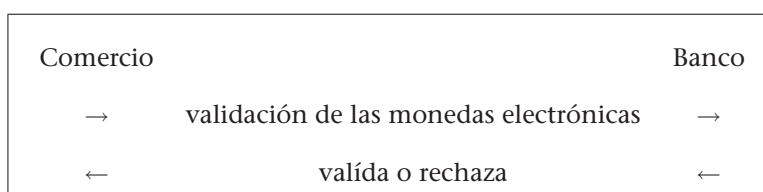
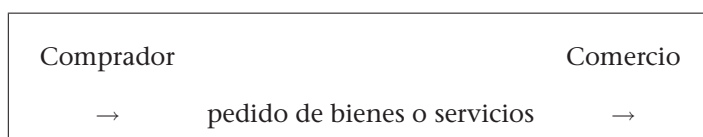
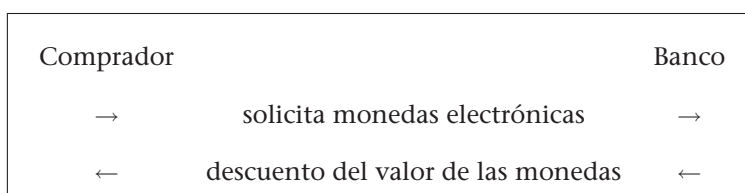
El funcionamiento general de este sistema es parecido al que se ha descrito en el protocolo de Chaum:

- 1) El comprador solicita monedas electrónicas. Estas se identifican con un número de serie. Este número permite garantizar la unicidad puesto que permite detectar duplicidades. El anonimato del propietario de las monedas electrónicas se protege mediante firmas ciegas.
- 2) Una vez el banco ha descontado el valor correspondiente de las monedas electrónicas entregadas al comprador, estas se almacenan en el terminal del comprador.

- 3) Cuando el comprador hace el pedido de bienes o servicios al vendedor, le entrega las monedas electrónicas correspondientes al valor de la compra.
- 4) El vendedor, antes de servir el pedido, entrega las monedas electrónicas al banco y le solicita su validación.
- 5) El servidor del banco accede a una base de datos donde consta el número de serie de todas las monedas que han sido utilizadas. Con esta operación se detecta si ha habido doble uso. Si no hay duplicidad, se responde al vendedor sobre la autenticidad de las monedas electrónicas y, si se realiza la operación de compra, automáticamente se ingresa el importe en su cuenta y se añade el número de serie correspondiente a la base de datos correspondiente.
- 6) El vendedor entrega los productos al comprador.

A nivel de seguridad, se garantiza la integridad, confidencialidad y autenticidad, al mismo tiempo que se facilita el anonimato del comprador. El comprador está protegido ante posibles actuaciones deshonestas de vendedores en cuanto a no entregas de productos se refiere, puesto que en un momento dado el comprador puede desvelar su identidad y en este caso se podría trazar el flujo del dinero.

Resumen de las transacciones del protocolo:



4. Protocolos de transferencia inconsciente

La transferencia inconsciente o transcordada (*oblivius transfer*) consiste en la transferencia de un secreto entre dos usuarios A y B de forma que la probabilidad de que B lo obtenga es de un 50%, sin que A pueda saber si B lo ha obtenido, o no.

Estos tipos de protocolos dan lugar a otros protocolos más complejos, como los de **compromiso de bits** o los de **prueba de conocimiento nulo** y, en general, a los protocolos que permiten la firma electrónica de contratos.

4.1. Protocolo de Rabin

El secreto consiste en la factorización del valor n_A de la clave pública del sistema RSA de A .

Protocolo

- $B \rightarrow A$. El usuario B escoge un valor x , $1 \leq x \leq n_A - 1$, y envía a A : $z = x^2 \pmod{n_A}$
- $A \rightarrow B$. El usuario A calcula las cuatro raíces cuadradas de:

$$z \pmod{n_A} = \{x, n_A - x, y, n_A - y\}$$

y envía una de ellas a B (esto solo lo puede hacer A , porque conoce los valores p_A y q_A , con los cuales ha calculado n_A).

Sea v la raíz enviada a B .

El usuario B comprueba si v coincide con y , o con $n_A - y$. Si coincide, se podrá factorizar n_A ; de lo contrario no se podrá.

Resumen de las transacciones del protocolo

A		B
←	$z = x^2 \pmod{n_A}$	←
→	$v = z^{-1/2} \pmod{n_A}$	→

Nota

Hay una equivalencia computacional entre factorizar $n_A = p_A \cdot q_A$ y calcular raíces cuadradas en \mathbb{Z}_{n_A} , como queda justificado en el cuadro y en el teorema siguientes.

Cuadrados y raíces cuadradas en \mathbb{Z}_p , donde p es un número primo

Si $\beta \in \mathbb{Z}_p$ puede pasar que exista, o no, $x \in \mathbb{Z}_p$ tal que $x^2 = \beta$.

En el primer caso se dice que β es un residuo cuadrático. En el segundo, se dice que β es un residuo no-cuadrático.

A excepción de $0 \in \mathbb{Z}_p$, hay exactamente $\frac{p-1}{2}$ residuos cuadráticos (QR) y $\frac{p-1}{2}$ residuos no cuadráticos (QNR) en \mathbb{Z}_p .

$$\text{Si } \beta \in \mathbb{Z}_p, \text{ es } \beta^{\frac{p-1}{2}} = \begin{cases} 1 & \text{si } \beta \text{ es QR} \\ -1 & \text{si } \beta \text{ es QNR} \end{cases}$$

- Es computacionalmente eficiente (hay un algoritmo determinista-polinomial) saber si $\beta \in \mathbb{Z}_p$ es QR o QNR.
- No es computacionalmente eficiente encontrar un elemento QNR. De todos modos, sí hay algoritmos no deterministas de complejidad polinómica que resuelven el problema de manera sencilla.
- Es computacionalmente eficiente calcular la raíz cuadrada de un elemento QR.

Teorema 4.1. Si $n = p_1 \cdot p_2$, donde p_1 y p_2 son números primos diferentes e impares, y si α y γ son raíces cuadradas, esencialmente diferentes, (es decir $\alpha \neq \pm\gamma$), de un cierto QR, entonces:

$$\text{mcd}(\alpha + \gamma, n) = p_1, \text{ o } \text{mcd}(\alpha + \gamma, n) = p_2$$

Ejemplo 4.1.

Supongamos $n_A = 5 \cdot 11 = 55$.

- $B \rightarrow A$. El usuario B escoge un valor $x = 13$ y envía a A : $z = 13^2 \pmod{55} = 4$.
- $A \rightarrow B$. El usuario A calcula las cuatro raíces cuadradas de $4 \pmod{55} = \{13, 55 - 13, 2, 55 - 2\} = \{13, 42, 2, 53\}$ y envía una de ellas a B . Sea $v = 2$ la raíz enviada.

El usuario B comprueba si $v = 2$ coincide con y , o con $55 - y$. Como coincide, se podrá factorizar $n_A = 55$ puesto que $\text{mcd}(13 + 2, 55) = \text{mcd}(15, 55) = 5$.

Igual hubiera pasado si $v = 53$, puesto que $\text{mcd}(13 + 53, 55) = \text{mcd}(66, 55) = 11$, que es un factor de $n_A = 55$.

Si se hubiera enviado $v = 42$ no se hubiera podido factorizar n_A , puesto que $\text{mcd}(13 + 42, 55) = \text{mcd}(55, 55) = 55$.

Igualmente para $v = 13$, puesto que $\text{mcd}(13 + 13, 55) = \text{mcd}(26, 55) = 1$.

Cuadrados y raíces cuadradas en \mathbb{Z}_n , donde $n = p_1 \cdot p_2$

Hay exactamente $\frac{n-1}{4}$ elementos QR en \mathbb{Z}_n y cada uno de ellos tiene, exactamente, cuatro raíces cuadradas.

Ejemplo: En \mathbb{Z}_{15} , el elemento $4 \in \mathbb{Z}_{15}$ es un QR y tiene cuatro raíces, que son: $\{2, 13, 7, 8\}$.

Los QR en \mathbb{Z}_n , son aquellos elementos en los que el módulo p_1 y, también, el módulo p_2 , son QR.

Probabilidad de que B conozca n_A es $1/2$

Del teorema también se deduce que la probabilidad de que B conozca $n_A = p_1 \cdot p_2$ es $1/2$ puesto que el espacio de muestras tiene 4 resultados posibles (las 4 raíces cuadradas de z), de las cuales solo 2 sirven para factorizar n_A (y y $n_A - y$). Así, si cada bit de un secreto se transmite mediante transferencia inconsciente, la probabilidad de que B pueda conocer el secreto es de 2^{-t} , si t es el número de bits del secreto.

4.2. Protocolos de compromiso de bits

En este protocolo, un usuario A se compromete frente a otro usuario B con un valor, de tal manera que A no lo pueda cambiar y B no lo pueda descubrir hasta que el propio A no haya abierto el compromiso.

Es un protocolo útil en otros protocolos, como el de lanzamiento de monedas o el de pruebas de conocimiento nulo, y debe cumplir las siguientes propiedades:

- Se debe poder comprometer cualquiera de los dos valores posibles para cada bit.
- De la apertura del compromiso, B solo obtendrá el valor del bit comprometido.
- No se debe poder modificar el valor comprometido, aunque se cambie la manera de abrir el compromiso.
- El usuario B no debe poder conocer nada sobre la manera de abrir los compromisos aunque haya visto abrir algunos.

1) Protocolo: Usando transferencia inconsciente

Para comprometer un bit b , el usuario A escoge n bits aleatorios b_i , tales que $b_1 \oplus b_2 \oplus \dots \oplus b_n = b$, y se siguen los tres pasos siguientes, donde \oplus es la operación XOR.

Protocolo

- 1) **Compromiso:** El usuario A envía a B cada bit b_i por orden, mediante transferencia inconsciente.
- 2) **Apertura:** El usuario A envía a B todos los bits b_i .
- 3) **Verificación:** El usuario B compara los bits b_i , recibidos dentro del plazo del compromiso, con los correspondientes de la apertura.

Nota

La probabilidad de fraude, por parte de A , es menor que $\frac{1}{2}$. Si se ejecuta el protocolo k veces, la probabilidad de fraude será menor que 2^{-k} .

2) Protocolo: Lanzamiento de moneda por teléfono

El protocolo de Rabin puede ser utilizado para el problema de tirar una moneda al aire, y resolver una apuesta entre dos usuarios A y B de manera telemática.

Otros protocolos de compromiso de bits

En la literatura hay otras implementaciones de protocolos de compromiso de bits, usando sistemas criptográficos de clave secreta, logaritmos discretos, funciones de *hash*, residuos cuadráticos o, incluso, grafos no isomorfos.

Apuesta

Cara \iff B puede factorizar n

Cruz \iff B no puede factorizar n

Protocolo

$A \rightarrow B$: El usuario A construye, al azar, dos números primos, diferentes y grandes: p_1, p_2 y calcula $n = p_1 \cdot p_2$. Envía n al usuario B .

$B \rightarrow A$: El usuario B escoge, al azar, un elemento primitivo, $\alpha \in \mathbb{Z}_n$, calcula $\beta = \alpha^2 \pmod{n}$ y envía β al usuario A .

$A \rightarrow B$: El usuario A calcula las cuatro raíces cuadradas de β y envía una al usuario B . Sea γ la raíz enviada.

$B \rightarrow A$: El usuario B le dice a A si ha salido CARA o CRUZ.

Tal como hemos visto antes, si $\gamma \neq \pm\alpha$, el usuario B podrá factorizar n . De lo contrario, si $\gamma = \pm\alpha$, no podrá factorizar n .

Resumen de las transacciones del protocolo:

A		B
→	n	→
←	$\beta = \alpha^2 \pmod{n}$	←
→	$\gamma = \beta^{-1/2} \pmod{n}$	→
←	CARA o CRUZ	←

Este protocolo fue propuesto por Mario Blum (1982) y se trata, tal como hemos visto antes, de resolver una apuesta entre dos usuarios A y B , distantes entre sí, mediante el lanzamiento de una moneda a cara o cruz. Si uno de los dos hace trampa, sin necesidad de un tercero el otro lo puede detectar.

Puede haber dos casos posibles de trampa:

trampa de A

- El usuario A lanza la moneda y anota el resultado.

- El usuario B hace su apuesta y se la dice a A .
- El usuario A le dice a B , que justamente ha salido lo contrario.

trampa de B

- El usuario A lanza la moneda y anota el resultado.
- El usuario B hace su apuesta y se la dice a A .
- El usuario A le dice a B , que justamente no ha salido lo que él había dicho.
- El usuario B niega su primera apuesta y replica a A diciéndole que precisamente lo que ha salido era lo que él había apostado.

¿Cómo resolver el problema de manera que si uno hace trampa el otro lo detecte? Las soluciones pasan por el protocolo de transferencia inconsciente de Rabin que hemos visto, o el protocolo propuesto por Blum, basado en una función unidireccional sobre un conjunto de números tal que la mitad de ellos son pares y la otra mitad impares.

El esquema general del protocolo, una vez el usuario A ha escogido un número de Blum: $n = p_1 \cdot p_2$ es:

Protocolo

- $A \rightarrow B$. El usuario A escoge un valor $x \in \mathbb{Z}_n$ y calcula $y = x^2 \pmod n$ y $z = y^2 \pmod n$. El usuario A envía z a B .
- $B \rightarrow A$. El usuario B , una vez recibido z , apuesta por la paridad de y (es decir, si es o no un número par).
- $A \rightarrow B$. El usuario A le informa si ha acertado o no. Además le muestra los valores de x y y , al mismo tiempo que le demuestra que n es un número de Blum.

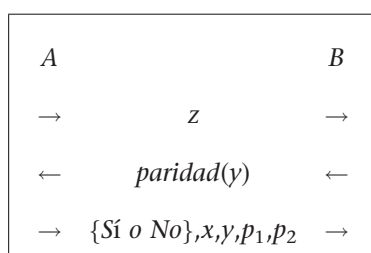
Números de Blum

Son los números $n = p_1 \cdot p_2$ tales que ambos primos p_1 y p_2 son congruentes con 3 del módulo 4. En este caso, aparte del cero y de p_1 y p_2 , la mitad de los residuos cuadráticos en \mathbb{Z}_n son pares y la otra mitad son impares. Además, el valor de z identifica y , ya que de las cuatro raíces cuadradas de z , únicamente hay una que, a su vez, sea un residuo cuadrático.

El usuario B comprueba que $y = x^2 \pmod n$ y $z = y^2 \pmod n$.

Ambos usuarios A y B han actuado con un 50% de probabilidad de engaño en el primero y el segundo paso del protocolo.

Resumen de las transacciones del protocolo:



Ejemplo 4.2.

El usuario A escoge un número de Blum, por ejemplo: $n = 7 \cdot 11 = 77$.

- $A \rightarrow B$. El usuario A escoge un valor $x = 13$ y calcula:
 $y = x^2 \pmod{n} = 13^2 \pmod{77} = 15$ y
 $z = y^2 \pmod{n} = 15^2 \pmod{77} = 71$.
 A envía z a B .
- $B \rightarrow A$. El usuario B , una vez recibido z , apuesta por la paridad de y .
 Suponemos que apuesta por y par.
- $A \rightarrow B$. El usuario A le informa de que no ha acertado en la apuesta.
 Para demostrarlo, le envía los valores de $x = 13$, $y = 15$ y $n = 7 \cdot 11 = 77$.
 El usuario A le demuestra que n es un número de Blum y B puede verificar el valor de z , a partir de los valores de x y y recibidos.

En este caso, B comprueba que se había equivocado en su apuesta.

4.3. Firma electrónica de contratos: Protocolo de Even

Se trata de permitir la firma electrónica de un documento entre dos usuarios A y B , sin intermediarios, de forma que se cumplan estas dos condiciones:

- Que ambos usuarios queden obligados a culminar la firma, solo a partir de un punto del protocolo.
- Que la firma no pueda falsificarse y, además, pueda ser comprobada por el otro usuario.

1) Protocolo básico

Suponemos que el usuario A dispone de dos claves secretas y las correspondientes claves públicas de un sistema criptográfico de clave pública: (E_{A1}, D_{A1}) ; (E_{A2}, D_{A2}) .

Suponemos que el usuario B elige una clave secreta K_B .

- $A \rightarrow B$. El usuario A envía a B sus dos claves públicas E_{A1} y E_{A2} .
- $B \rightarrow A$. El usuario B escoge una de las dos claves y cifra su clave K_B , enviando el resultado a A .
- $A \rightarrow B$. El usuario A escoge una de sus dos claves privadas y descifra lo que ha recibido de B .
- $A \rightarrow B$. El usuario A cifra el primer bloque del mensaje a firmar, usando la supuesta clave encontrada en el paso anterior, y lo envía a B .

Transferencia inconsciente

Los pasos de este protocolo se corresponden básicamente con una transferencia inconsciente.

El usuario B descifrará con la clave K_B el bloque de firma recibido.

El usuario A repetirá el tercero y cuarto paso para cada bloque de firma y B siempre descifrará con la clave K_B el bloque de firma recibido.

Cuando se hayan acabado todos los bloques de firma, el usuario A repetirá el cuarto paso, utilizando ahora la otra clave privada, y B descifrará con la clave K_B el bloque de firma recibido.

Si A y B han escogido al azar la misma clave (con una probabilidad del 50%), B descifrará un mensaje con sentido en la primera ronda. Si no, recibirá un mensaje sin sentido y tendrá que esperar hasta recibir el último bloque de la segunda ronda para obtener el texto en claro.

Sin embargo, A no tiene ninguna manera de saber cuándo B ha podido descifrar correctamente el criptograma, lo cual le fuerza a acabar el protocolo.

2) Protocolo de Even

Este protocolo se basa en sistemas criptográficos de clave privada y sigue los siguientes pasos:

Inicialmente, tanto el usuario A como el usuario B escogen entre un conjunto de $2n$ claves de un sistema criptográfico de clave privada:

El usuario A escoge el conjunto $\{K_1, K_2, \dots, K_n, K_{n+1}, \dots, K_{2n}\}$, tomadas en parejas: $\{(K_1, K_{n+1}), (K_2, K_{n+2}), \dots, (K_n, K_{2n})\}$.

El usuario B escoge el conjunto $\{K_1^*, K_2^*, \dots, K_n^*, K_{n+1}^*, \dots, K_{2n}^*\}$, tomadas en parejas $\{(K_1^*, K_{n+1}^*), (K_2^*, K_{n+2}^*), \dots, (K_n^*, K_{2n}^*)\}$.

Protocolo

- $A \rightarrow B$. El usuario A cifra un mensaje M_A , conocido por B , con las $2n$ claves:

$E_{K_1}(M_A), E_{K_2}(M_A), \dots, E_{K_{2n}}(M_A)$ y envía los $2n$ criptogramas ordenados a B .

El usuario A se comprometerá, más adelante, a firmar el contrato si B puede presentarle algún par de claves (K_i, K_{n+i}) .

- $B \rightarrow A$. El usuario B cifra un mensaje M_B , conocido por A , con las $2n$ claves: $E_{K_1}^*(M_B), E_{K_2}^*(M_B), \dots, E_{K_{2n}}^*(M_B)$ y envía los $2n$ criptogramas ordenados a A .

El usuario B se comprometerá, más adelante, a firmar el contrato si A puede presentarle algún par de claves (K^*i, K_{n+i}^*) .

- $A \rightarrow B$. El usuario A envía a B cada par (K_i, K_{n+i}) ordenados mediante una transferencia inconsciente; es decir, enviando K_i o K_{n+i} con igual probabilidad.
- $B \rightarrow A$. El usuario B hará lo mismo, enviando a A , ordenadamente, los pares (K^*i, K_{n+i}^*)

Nota

En este punto A y B tienen la mitad de las claves uno del otro.

Si la longitud de cada clave es de L bits, entonces los usuarios A y B realizan el siguiente bucle, $1 \leq i \leq 2n$, para las claves K_i y K_i^* que no se han usado en los pasos anteriores:

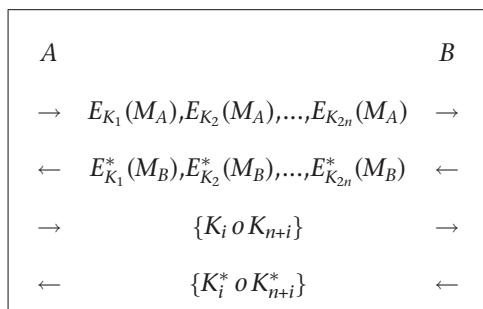
Algoritmo 4.2

```

for  $1 \leq j \leq L$ 
  begin
     $A$  envía a  $B$  el bit  $j$ -ésimo de todas estas claves  $K_i$ 
     $B$  envía a  $A$  el bit  $j$ -ésimo de todas estas claves  $K_i^*$ 
  end
    
```

Observar que el algoritmo es parecido al del protocolo de compromiso bit a bit. Al realizar este bucle completo, A y B tienen las $2n$ claves uno del otro y se supone que pueden firmar el contrato.

Resumen de las transacciones del protocolo:



4.4. Protocolo de correo electrónico certificado

Pese a que los sistemas actuales de correo electrónico permiten el acuse de recibo por parte del receptor (el emisor pide al receptor que le comunique que ha recibido el correo que le ha enviado, sin que esto implique la aceptación de su contenido); cuando enviamos un correo electrónico, ¿cómo podemos estar seguros de que el mensaje enviado ha llegado al destinatario autorizado y solo él conoce el contenido?

La propuesta de este protocolo es la implementación del correo electrónico certificado.

Así, si un usuario A desea enviar un mensaje electrónico como correo certificado a un usuario B , le descubrirá el mensaje, es decir, le enviará la clave, solo después de que el usuario B le haya enviado el acuse de recibo correspondiente.

Esto es tal como sucede con el correo regular certificado; antes de ver el mensaje hemos firmado que hemos recibido el sobre que lo contiene.

El protocolo que sigue es muy parecido al que hemos visto de *Even* para la firma de contratos.

Protocolo

El usuario A escoge aleatoriamente $n + 1$ claves $\{a_0, a_1, \dots, a_n\}$ de un sistema criptográfico de clave privada. Las claves $\{a_1, \dots, a_n\}$ serán la parte izquierda de la clave KE_{A_i} , y la parte derecha, KD_{A_i} será $\{a_{n+1}, a_{n+2}, \dots, a_{2n}\}$, donde $a_{n+i} = a_0 \oplus a_i$, para $1 \leq i \leq n$.

Los usuarios A y B se ponen de acuerdo en un mensaje de validación V .

El usuario A , con la clave a_0 cifra el documento M : $C_0 = E_{a_0}(M)$ y después cifra el mensaje de validación V con las $2n$ claves secretas; $\{KE_{A_i}, KD_{A_i}\}$ para $1 \leq i \leq n$:

Cifrado de validación de la parte izquierda: $VE_{A_i} = E_{KE_{A_i}}(V)$.

Cifrado de validación de la parte derecha: $VD_{A_i} = E_{KD_{A_i}}(V)$.

- $A \rightarrow B$. El usuario A envía a B , el valor C_0 y los pares ordenados (VE_{A_i}, VD_{A_i}) , para $1 \leq i \leq n$.

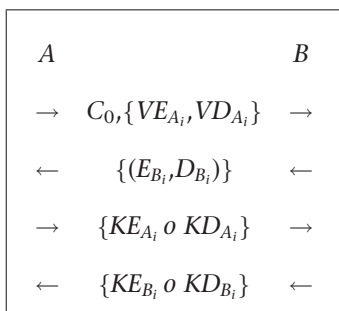
De forma similar, el usuario B genera n pares de claves KE_{B_i} y KD_{B_i} . También generará n pares de mensajes **acuse de recibo de la parte izquierda**: RE_i y **acuse de recibo de la parte derecha**: RD_i y cifra las parejas (RE_i, RD_i) en el

sistema criptográfico de clave privada, con las claves (KE_{B_i}, KD_{B_i}) obteniendo (E_{B_i}, D_{B_i}) .

- $B \rightarrow A$. El usuario B envía a A las parejas ordenadas (E_{B_i}, D_{B_i}) .
- $A \rightarrow B$. Mediante una transferencia inconsciente, el usuario A envía a B una de las dos claves secretas: KE_{A_i} o KD_{A_i} .
- $B \rightarrow A$. Mediante una transferencia inconsciente, el usuario B envía a A : KE_{B_i} o KD_{B_i} .

Este proceso se repite hasta haber enviado los n valores de las claves de cada usuario.

Resumen de las transacciones del protocolo:



Comprobación:

- El usuario B usa las claves enviadas por A , KE_{A1} o KD_{A1} , para comprobar que al descifrar $D_{KE_{A_i}}(V)$ o $D_{KD_{A_i}}(V)$ obtiene el mensaje de validación V .
- El usuario A usa las claves enviadas por B , KE_{B_i} o KD_{B_i} , para comprobar que al descifrar $D_{KE_{B_i}}(E_{B_i})$ o $D_{KD_{B_i}}(D_{B_i})$ obtiene siempre RE_i o RD_i .

Estos pasos son en realidad un protocolo de **compromiso** de bit entre los usuarios A y B .

Verificación:

- El usuario A ha obtenido todas las claves de B y comprueba todas las parejas de acuse de recibo; es decir, comprueba la parte izquierda y la parte derecha del acuse de recibo.
- El usuario B ha obtenido todas las claves de A y comprueba que todos los envíos de A contienen el mensaje de validación V .

Realmente, el usuario A habrá debido mostrar todas las claves a B , para que B pueda comprobar que A ha usado el cálculo $a_{n+i} = a_0 \oplus a_i$.

- Dado que el usuario B dispone de todas las claves de A , puede calcular:
 $a_0 = KE_{A_i} \oplus KD_{A_i}$.

Observar que para el cálculo anterior cualquiera de las parejas de accuse de recibo es válida.

- El usuario B descifra el criptograma C_0 , enviado en el segundo paso, recuperando, así el documento M : $D_{a_0}(C_0) = M$.

5. Esquemas umbral y reparto de secretos

Los protocolos para compartir secretos tratan de resolver el siguiente problema: dado un secreto, repartir unos fragmentos de información entre varias personas, de forma que ciertas agrupaciones de estas personas puedan recuperar el secreto, pero las restantes agrupaciones no sean capaces de obtenerlo.

Por ejemplo, un banco tiene una cámara acorazada que tiene que abrirse cada mañana con una cierta clave (el secreto), que comparten 5 cajeros encargados de tal apertura. El banco puede estar interesado en que, para abrirla, al menos hagan falta 3 de los 5 cajeros.

Así, formalmente, un secreto se divide en n participaciones, que se reparten entre n participantes. La finalidad es que la presencia de k de estos participantes ($k \leq n$) sea necesaria y suficiente para reconstruir el secreto.

Definición 5.1 (Esquema umbral).

Para un par de valores k y n , $k < n$, un (k, n) -esquema umbral consiste en n participaciones $D_1, D_2, \dots, D_n \in \mathbb{Z}_p$ y un valor secreto $D \in \mathbb{Z}_p$ tal que:

- 1) El conocimiento de k , o más, participaciones D_i es suficiente para calcular la clave secreta D .
- 2) El conocimiento de $k - 1$, o menos participaciones, deja la clave D completamente indeterminada.

5.1. Esquema de Shamir

Este esquema trabaja con polinomios en un cuerpo finito, donde el valor secreto será el término independiente de un cierto polinomio. La idea del esquema se basa en el hecho de que un polinomio de grado $k - 1$ se puede caracterizar por sus k coeficientes (la clave) o bien dando valores en k puntos diferentes (las participaciones).

Sea \mathbb{Z}_p un cuerpo finito. Supongamos que el valor que queremos guardar secreto es $D = a_0 \in \mathbb{Z}_p$ y que $a_1, a_2, \dots, a_{k-1} \in \mathbb{Z}_p$ son valores escogidos al azar, los cuales vamos a considerar que son los coeficientes del polinomio $A(x) = a_0 + a_1x + \dots + a_{k-1}x^{k-1} \in \mathbb{Z}_p[x]$.

Consideramos n elementos diferentes $x_1, x_2, \dots, x_n \in \mathbb{Z}_p$ y calculamos, para cada uno de ellos $A(x_i) = D_i$.

La participación que se dará al i -ésimo participante es D_i .

Cualquier grupo de k o más participantes permite reconstruir, gracias al teorema de interpolación de *Lagrange*, el polinomio $A(x)$ y, por lo tanto, calcular la clave $D = a_0$.

En efecto, sean $\{i_1, \dots, i_k\}$ los k participantes, entonces:

$$A(x) = \sum_{i_j=i_1}^{i_k} D_{i_j} \cdot q_{i_j}(x), \text{ donde:}$$

$$q_{i_j}(x) = \prod_{i_t \neq i_j}^{i_k} \frac{(x-x_{i_t})}{(x_{i_j}-x_{i_t})},$$

de donde podremos encontrar el secreto $D = a_0$.

Ejemplo 5.1.

Para más claridad en las operaciones, trabajaremos en \mathbb{R} .

Supongamos que el secreto es $D = 15$ en un (3,5)-esquema umbral. Escogeremos un polinomio $A(x)$, de grado 2, que tenga por término independiente $a_0 = D = 15$.

Sea, por ejemplo, $A(x) = 16x^2 + 2x + 15$ y los cinco valores $x_i = \{1,3,4,7,10\}$ con los cuales calcularemos $D_i = A(x_i)$.

Así tendremos los cinco puntos $\{(1,33),(3,165),(4,279),(7,813),(10,1635)\}$.

Supongamos que los usuarios 1,2,4 se agrupan para obtener el secreto.

$$q_1(x) = \frac{(x-x_2)}{(x_1-x_2)} \cdot \frac{(x-x_4)}{(x_1-x_4)} = \frac{(x-3)}{-2} \cdot \frac{(x-7)}{-6} = \frac{1}{12}(x^2 - 10x + 21)$$

$$q_2(x) = \frac{(x-x_1)}{(x_2-x_1)} \cdot \frac{(x-x_4)}{(x_2-x_4)} = \frac{(x-1)}{2} \cdot \frac{(x-7)}{-4} = \frac{-1}{8}(x^2 - 8x + 7)$$

$$q_4(x) = \frac{(x-x_1)}{(x_4-x_1)} \cdot \frac{(x-x_2)}{(x_4-x_2)} = \frac{(x-1)}{6} \cdot \frac{(x-3)}{4} = \frac{1}{24}(x^2 - 4x + 3), \text{ y}$$

$$A(x) = 33 \cdot q_1(x) + 165 \cdot q_2(x) + 813 \cdot q_4(x) = 16x^2 + 2x + 15,$$

con lo cual hemos recuperado el secreto $D = a_0 = 15$.

Nota

También podemos calcular el valor a_0 , del esquema umbral, resolviendo el sistema de ecuaciones, para $i = j_1, \dots, j_k$:

$$a_0 + a_1 x_i + \dots + a_{k-1} x_i^{k-1} = D_i$$

El determinante de este sistema es de tipo Vandermonde y todos los valores x_i son diferentes, por lo tanto, el sistema tiene solución única.

Interpolación de Lagrange

El teorema de interpolación de *Lagrange* nos permite calcular los coeficientes a_i de manera computacionalmente más sencilla que resolviendo el sistema de ecuaciones correspondiente.

Mentirosos en el esquema umbral de Shamir

Supongamos que un participante, por ejemplo el número 1, es un mentiroso y da el valor D_1^* ($D_1^* = D_1 + \epsilon$) como su participación.

En este caso la clave que el grupo de k participantes encontrará, es:

$$a_0^* = -\varphi(0) \frac{D_1^*}{\psi'(x_1)} - \varphi(0) \sum_{i \neq 1} \frac{D_i}{\psi'(x_i)} = a_0 - \epsilon \frac{\varphi(0)}{\psi'(x_1)}.$$

El mentiroso conoce D_1, D_1^* y, además, ϵ y a_0^* .

A partir de la ecuación anterior puede calcular el valor correcto a_0 .

Los otros participantes no pueden calcular el valor correcto a_0 .

Para evitar a los mentirosos en un esquema umbral, J. Rifà (*How to avoid the Cheaters succeed in the Key Sharing Scheme. Desings, Codes and Cryptography* (1993)) propone un esquema $(2k - 1, 2n)$ en el cual se dan dos participaciones a cada uno de los n participantes, de modo que para calcular la clave se necesita un mínimo de k participantes.

Llamemos $\varphi(x) = (x - x_1) \dots (x - x_k)$ y $\psi(x) = x\varphi(x)$, entonces:

$$a_0 = \frac{\text{Det.}(D_i, x_i, x_i^2, \dots, x_i^{k-1})}{\text{Det.}(1, x_i, x_i^2, \dots, x_i^{k-1})} = \frac{\sum_i (-1)^{i+1} D_i \prod_{j \neq i} x_j \prod_{s > t \neq i} (x_s - x_t)}{\prod_{s > t} (x_s - x_t)} =$$

$$\sum_i D_i \frac{\prod_{j \neq i} -x_j}{\varphi'(x_i)} = - \sum_i D_i \frac{\prod x_j}{x_i \varphi'(x_i)} = - \sum_i D_i \frac{\varphi(0)}{\psi'(x_i)} = -\varphi(0) \sum_i \frac{D_i}{\psi'(x_i)}.$$

El valor de $\psi'(x_i)$ no depende de D_i , pero el cálculo de la clave $a_0 = D$ necesita de estos valores D_i .

6. Votaciones electrónicas

Una votación realizada electrónicamente necesita garantizar una serie de condiciones formalmente análogas a las de una votación tradicional:

- **Democracia:** Solo las personas registradas en el censo pueden emitir su voto y solamente pueden hacerlo una vez.
- **Transparencia:** Ningún voto puede ser eliminado ni alterado.
- **Privacidad:** No se puede establecer ninguna relación entre un voto y un votante.
- **No coercibilidad:** Para evitar coacciones el votante no puede demostrar cuál ha sido el sentido de su voto.
- **Verificabilidad:** Cada votante, y eventualmente un auditor, puede comprobar que el voto ha sido correctamente contabilizado.

Agradecimiento

Los autores agradecen la colaboración en este apartado de Jordi Puiggalí, VP Research and Development de SCYTL (<http://www.scytl.com>).

En una votación presencial, estas condiciones quedan garantizadas por una urna transparente, la cabina de votación y el escrutinio público. En el caso electrónico, el proceso electoral se lleva a cabo en una dimensión lógica (por ejemplo, mediante un conjunto de programas que se ejecutan en un ordenador), y por lo tanto, no auditable por observadores o por el propio votante. Por esto, para asegurar que los procesos electorales se llevan a cabo de forma totalmente honesta y privada, existen diferentes propuestas basadas en la utilización de protocolos criptográficos.

Los protocolos criptográficos por voto electrónico se basan sobre todo en conseguir dos objetivos principalmente:

- **Garantizar la privacidad de los votantes y la corrección de los resultados:** asegurando que todos los votos que se han utilizado para obtener los resultados pertenecen a votantes válidos (por ejemplo, que forman parte de la lista del censo y no han sido suplantados), verificando que un votante no emita más de un voto, y haciendo que no pueda correlacionar en ningún momento la papeleta del voto y la identidad del votante.
- **Facilitar la auditoría de la elección:** permitiendo tanto a votantes como a observadores verificar que los votos emitidos contienen la opción del voto original seleccionado por el votante y, por lo tanto, que el resultado refleja totalmente la intención de voto de los votantes.

Para poder lograr estos objetivos los esquemas criptográficos propuestos combinan otros protocolos o algoritmos criptográficos más básicos: autenticación, firmas ciegas, pruebas de conocimiento nulo, reparto de secretos, etc.

6.1. Garantizar la privacidad y la corrección de los resultados

Desde el punto de vista de la privacidad de los votantes y la corrección de los resultados, los protocolos criptográficos parten de la base del cifrado y posterior firma digital de los votos. El cifrado se utiliza por evitar que la intención de voto sea visible para cualquier administrador o persona que disponga de privilegios sobre el sistema de voto. En este sentido, se utilizan como base algoritmos de clave pública (como por ejemplo RSA), dejando la clave privada en manos del equivalente a una mesa electoral para realizar el descifrado de los votos. De este modo, los votos se cifrarían con una clave pública de la mesa electoral en el mismo terminal desde el que el votante ha realizado la selección de voto, para evitar que su contenido se sepa antes de guardar el voto en la urna (que en este caso sería electrónica). Los votos solo se podrían descifrar cuando lo decidiera la mesa electoral, que es la que tiene acceso a la clave privada que los descifra.

Para evitar que los votos se puedan manipular una vez emitidos, los votos cifrados se firmarían digitalmente, preferentemente con la clave de identidad del elector (por ejemplo, DNI electrónico). De este modo, cualquier intento de manipulación del voto se detectaría, puesto que invalidaría la firma digital del mismo. Además, la firma digital permite verificar si el votante que ha emitido el voto pertenece realmente al censo electoral. De este modo se puede asegurar que los resultados se han obtenido de votos de ciudadanos que realmente podían participar.

Este mecanismo de cifrado y firma se conoce con el nombre de “*doble sobre*”, pues es equivalente al mecanismo de dos sobres utilizados en el voto postal. El sobre de dentro es el que protege la privacidad del voto y por lo tanto sería el que se obtiene al cifrar el voto. El sobre de fuera es el que contiene el sobre de dentro y la identidad del votante, para poder verificar que el voto proviene de un votante válido antes de introducirlo en la urna. En el caso del voto electrónico, sería la firma digital que se hace sobre el voto. Del mismo modo que en el voto postal el sobre externo de los votantes válidos se separa del sobre interno antes de hacer el recuento, en el voto electrónico se podría separar la firma digital del voto cifrado antes de descifrar el voto y hacer el recuento.

Aun cuando estas medidas podrían parecer suficientes para garantizar la integridad y privacidad, no se considera que lo sean en un entorno de voto electrónico. Por ejemplo, si se hace un descifrado directo de los votos, sería fácil correlacionar los votos en claro con los votantes: solo haría falta verificar el orden en el que han votado los votantes y el que se obtiene al descifrar los votos. Por esta razón, los protocolos criptográficos hacen propuestas de mecanismos de descifrado de voto que rompen cualquier correlación entre los votos descifrados y el orden en el que han sido emitidos. En este sentido se pueden destacar dos familias de propuestas que implementan un proceso de

descifrado que rompe esta correlación: la de los protocolos de recuento homomórfico y la de los protocolos de mezcla.

Los protocolos de recuento homomórfico se fundamentan en la obtención de los resultados sin descifrar los votos individuales. Para conseguir este objetivo, las operaciones de recuento se realizan directamente sobre los votos cifrados y solo se descifran los resultados de estas operaciones. Para poder operar con los votos cifrados sin tenerlos que descifrar, utilizan, para el cifrado de los votos, algoritmos criptográficos con propiedades homomórficas (por ejemplo, ElGamal). Estas propiedades permiten hacer operaciones directamente con la información cifrada sin necesidad de descifrarla. Un ejemplo sería multiplicar dos datos cifrados para obtener el cifrado del producto de los valores de estos datos: $E(x) \cdot E(y) = E(x \cdot y)$. De este modo solo haría falta descifrar el resultado de multiplicar los datos cifrados (por ejemplo, $E(x \cdot y)$) para obtener el producto de los valores (por ejemplo, $x \cdot y$). También existen algoritmos criptográficos en los que el resultado de multiplicar los datos cifrados es el cifrado de la suma de los contenidos: $E(x) \cdot E(y) = E(x + y)$. Estos son normalmente los utilizados por los protocolos homomórficos, ya que los votos se representan como un vector de valores binarios con tantas posiciones como opciones de voto disponibles. Si una opción ha sido seleccionada, el valor de su posición será 1, mientras que si no lo ha sido será 0. Por ejemplo, si hay cuatro opciones y la segunda y la tercera han sido seleccionadas el voto se representaría formalmente de forma simplificada como: (1,1,1,0). De este modo, al multiplicar los votos cifrados lo que estamos haciendo es sumar las veces que las opciones han sido escogidas y solo haría falta descifrar el resultado de multiplicar todos los votos cifrados para obtener los resultados. Por ejemplo:

$$E((1,0,0,1)) \cdot E((1,0,0,0)) \cdot E((0,1,0,0)) = E((2,1,0,1))$$

Por lo tanto, al descifrar, obtendríamos que la primera opción es la ganadora con dos votos, mientras que la segunda y tercera habrían obtenido un solo voto. Aun cuando estos protocolos son bastante eficientes, tienen algunas limitaciones, como el hecho de que solo funcionan en elecciones en las que los votos se pueden representar de forma numérica o que son poco escalables en elecciones de muchos candidatos. Estas limitaciones no existen en los protocolos basados en la mezcla.

Los protocolos de mezcla utilizan el mismo concepto que en las elecciones tradicionales: mezclar la urna antes de abrirla para garantizar que los votos no se encuentran en el mismo orden que se han introducido (por ejemplo, evitar que el voto de encima sea el que ha emitido el último votante). Esta mezcla sería equivalente a aplicar una permutación a los votos de la urna. El problema es que aplicar solamente una permutación no garantiza que el voto no se pueda rastrear, pues solo hace falta buscar dónde se encontraba un voto con el mismo cifrado antes de aplicarle la permutación. Por lo tanto, estos protocolos, además de permutar las posiciones de los votos, también

deben hacer un descifrado parcial o recifrado del voto. De este modo, no se puede correlacionar los votos una vez permutados y redescifrados con los votos cifrados originales, ya que serán todos diferentes.

En el caso del descifrado parcial, implica que el votante, además de cifrar el voto con la clave pública de la mesa electoral, también lo debe cifrar con la clave pública de la entidad que hace la mezcla. La entidad que ejecuta la mezcla dispone de una clave privada para hacer un primer descifrado parcial del voto una vez permutado. En algunas implementaciones, hay más de una entidad realizando el proceso de mezcla y descifrado parcial, con lo cual resulta que el votante debe hacer tantos cifrados como entidades participen.

En el caso de mezcla con recifrado, es necesario utilizar un algoritmo con propiedades homomórficas para poder utilizar la misma clave pública para recifrar el voto. De este modo, no hace falta descifrar el voto tantas veces como ha sido recifrado. Por lo tanto, este proceso es totalmente transparente al votante, que solo debe cifrar el voto una vez con la clave pública de la mesa electoral. El proceso de mezcla y recifrado se puede hacer tantas veces como entidades intervengan en el proceso. En cualquier caso, la última entidad es la que acaba constituyendo la mesa electoral y descifra los votos.

6.2. Garantizar la auditoría de la votación

Los procesos descritos anteriormente garantizan principalmente la privacidad del voto, pero ¿qué pasa si la entidad que hace el descifrado total presenta un resultado que no se corresponde con la intención de voto?

Para detectar esta situación, se utilizan los mecanismos de auditoría de los procesos de descifrado. Estos procesos utilizan mecanismos criptográficos como pruebas de conocimiento nulo para demostrar que el proceso de descifrado o recifrado no ha modificado el voto cuando estos se llevan a término. En este sentido destacamos dos tipos de pruebas de conocimiento nulo importantes: las de correcto descifrado y las de correcto recifrado. En el primer caso, se permite demostrar matemáticamente de forma irrefutable que el texto descifrado estaba contenido en el voto cifrado. En el caso de recifrado, la prueba demuestra que el valor del recifrado se ha obtenido al recifrar el texto previamente cifrado por el votante. En cualquier caso, no hace falta entregar la clave privada para demostrar que el valor descifrado es correcto, sino una prueba matemática irrefutable sobre el contenido descifrado que puede ser validada de forma universal (por ejemplo, por cualquier auditor o votante). De este modo se puede garantizar que los resultados reflejan de forma irrefutable los contenidos de los votos cifrados. De lo contrario, se pueden aislar los votos conflictivos para una auditoría posterior.

Ejercicios de autoevaluación

1. Considerar $n = 917641387$, producto de dos números primos $p = 12347$ i $q = 74323$. Nos dan $r = 10000$ que es un QR (residuo cuadrático) en \mathbb{Z}_n . Calcular las cuatro raíces cuadradas de r .

2. En el protocolo de lanzamiento de una moneda el usuario A escoge el número de Blum $n = 713 = 31 \cdot 23$. El número n es conocido por B , pero no su descomposición en factores!

- a) Haced el seguimiento de todos los pasos del protocolo, suponiendo que A escoge el valor $x = 220$ y B contesta diciendo que y es par. Al final del protocolo, tras la respuesta de A enviando $x, y, 31, 23$ a B , explicar el porqué B está convencido de que no se ha hecho trampa.
- b) Haced el seguimiento del protocolo, suponiendo ahora que A escoge $x = 15$ y B ha conocido, antes de contestar, el valor de los primos $31, 23$, factores de n . Qué debe contestar sobre la paridad de y para ganar?

3. En una red de comunicaciones cada usuario U tiene su propio algoritmo público de cifrar E_U y su algoritmo privado de descifrar D_U .

En esta red todos los mensajes que van de un usuario A a un usuario B deben ser enviados bajo el formato $(E_B(m), A)$. La identificación A se envía de forma que el receptor B sepa quién es el emisor.

El receptor B , tras leer el mensaje m que ha recibido, automáticamente tiene que devolver $(E_A(m), B)$ a la red para que el emisor A lo lea y sepa que su primer mensaje ha sido recibido correctamente.

- a) Demostrar que un tercer usuario C de la red puede calcular el mensaje m que A había enviado a B .
- b) Demostrar que las comunicaciones tampoco son seguras si cambiamos el protocolo inicial por el siguiente:
- A envía $E_B(E_B(M), A)$ a B
 - B , automáticamente devuelve $E_A(E_A(m), B)$ a A .

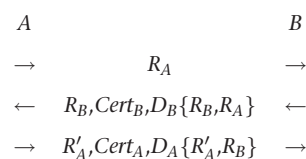
4. Un sistema público, basado en el logaritmo discreto para distribuir claves entre usuarios de una red, podría ser el siguiente:

- a) Es conocido de todos los usuarios un cuerpo finito \mathbb{F}_{p^s} y un elemento primitivo α de este cuerpo.
- b) El usuario U escoge aleatoriamente $1 < m_U < q^{s-1}$ y escribe en el fichero público su clave $c_U = \alpha^{m_U}$, (el valor m_U lo mantiene secreto).
- c) Cuando el usuario A quiere contactar con el usuario B , usa como clave $(c_B)^{m_A}$.

Mostrar que el sistema anterior es seguro y auténtico. Y, en el caso concreto $\mathbb{F}_{p^s} = \mathbb{F}_{2^{10}} = \mathbb{Z}_2[x] / (x^{10} + x^3 + 1)$, el usuario A , del que conocemos su clave privada $m_A = 2$, quiere contactar con el usuario B , del que sabemos su clave $c_B = \alpha + \alpha^5 + \alpha^7$, escrita en el fichero público.

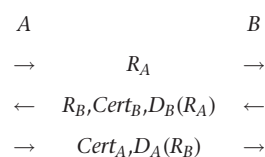
¿Cuál es la clave para sus comunicaciones comunes?

5. Demostrar que la siguiente variación del protocolo de autenticación de la ISO es inseguro.



$Cert_X = (X, E_X, D_S\{X, E_X\})$, S es la autoridad central de certificaciones, $\{x\}$ significa un *hash* sobre x y R'_A es un valor aleatorio diferente de R_A .

6. Demostrar que la siguiente variación del protocolo de autenticación de la ISO es inseguro.



$Cert_X = (X, E_X, D_S\{X, E_X\})$, S es la autoridad central de certificaciones y $\{x\}$ significa un *hash* sobre x .

Solucionario

1. Primero calcularemos las raíces cuadradas de $r = 10000$ a \mathbb{Z}_p i \mathbb{Z}_q , respectivamente, donde $p = 12347$ i $q = 74323$. Es fácil ver que $s_1 = 100 \pmod{p}$ y $s_2 = -100 = 12247 \pmod{p}$ son las raíces en \mathbb{Z}_p y $r_1 = 100 \pmod{q}$ i $r_2 = -100 = 74223 \pmod{q}$ son las raíces en \mathbb{Z}_q .

Como que p y q son primos entre sí, su máximo común divisor valdrá 1 y con el algoritmo extendido de Euclides hallaremos dos números A, B tales que $Ap + Bq = 1$.

Notar que los cuatro valores $m_{ij} = r_i Ap + s_j Bq$, donde $i, j \in \{1, 2\}$, cumplen que $m_{ij}^2 \equiv r \pmod{n}$, o sea que se trata de las raíces buscadas.

En nuestro caso: $1 = 30531 \cdot p - 5072 \cdot q$, o sea $A = 30531$ y $B = -5072$.

Por lo tanto, las cuatro raíces cuadradas de r son:

$$\begin{aligned} A \cdot p \cdot 100 + B \cdot q \cdot 100 &= 100 \\ A \cdot p \cdot 100 + B \cdot q \cdot 12247 &= 144632658 \\ A \cdot p \cdot 74223 + B \cdot q \cdot 100 &= 773033423 \\ A \cdot p \cdot 74223 + B \cdot q \cdot 12247 &= 917665981 \end{aligned}$$

2.

a) En el primer paso A envía $z = y^2 = (x^2)^2 = 639 \pmod{n}$. En el segundo paso B contesta diciendo que apuesta a que y es par. El usuario B pierde la apuesta, puesto que $y = x^2 = 629 \pmod{n}$ es impar. Al enviarle A el valor de x y la descomposición de n en factores, puede comprobar que n es un entero de Blum y que y es impar.

b) En el primer paso A envía $z = y^2 = (15^2)^2 = 2 \pmod{n}$. En este caso, al conocer B la descomposición de n en factores puede calcular las cuatro raíces cuadradas de z . Estas son 225, 271, 442, 488 y la única que es QR es 255. Con esto sabe que $y = 255$ es impar y, por lo tanto, debe contestar *impar* si quiere ganar la apuesta.

3. Efectivamente, el usuario C puede tomar $E_B(m)$, que está en la red bajo el formato $(E_B(m), A)$, y reenviar a la red $(E_B(m), C)$. La respuesta de B también sería automática y respondería con $(E_C(m), B)$. Por lo tanto, el usuario C solo tiene que descifrar $E_C(m)$ con su clave privada y encontrará m .

En el segundo supuesto, la red tampoco es segura. En este caso, el usuario C puede recuperar $E_B(E_B(m), A)$ y enviar a B lo siguiente (que cumple con los requisitos para ser aceptado para la transmisión): $E_B(E_B(E_B(m), A), C))$. El usuario B contesta automáticamente con: $E_C(E_C(E_B(m), A), C))$. A partir de aquí, el usuario C puede calcular $E_B(m)$ y enviar a la red $E_B(E_B(m), C)$. Ahora, la respuesta automática de B es $E_C(E_C(m), B)$ y el usuario C puede calcular m .

4. La clave que usan conjuntamente A y B es: $\alpha^{m_A \cdot m_B}$, que solo puede ser calculada por A y B . La seguridad y autenticidad se basa en que, para los otros usuarios, el problema es el de calcular el logaritmo discreto en el cuerpo \mathbb{F}_{p^s} .

En el caso específico de $\mathbb{F}_{2^{10}}$ tenemos que la clave común es $\alpha^{m_A \cdot m_B} = C_B^{m_A} = (\alpha + \alpha^5 + \alpha^7)^2 = \alpha^{428}$.

El cálculo anterior lo hemos hecho utilizando SAGE:

Primero definimos el anillo de polinomios en la indeterminada X :

```
sage: P.<X> = GF(2)[ ]%
```

Después definimos el cuerpo finito $\mathbb{F}_{2^{10}}$ a través del polinomio $X^{10} + X^3 + 1$ y llamamos $\alpha = [X]$.

```
sage: F.<alpha>=GF(2^10,'alpha',modulus=X^10+X^3+1)
```

Finalmente, calculamos $(\alpha + \alpha^5 + \alpha^7)^2$.

```
sage: (alpha +alpha^5+alpha^7)^2.
```

```
alpha^7 + alpha^4 + alpha^3 + alpha^2 + 1.
```

```
sage: log((alpha +alpha^5+alpha^7)^2,alpha)
```

```
428
```

5. Es un caso claro de utilización del *interleaving attack* o ataque del jugador de ajedrez.

El segundo paso del protocolo autentica B ante A . Aquí no hay ningún problema. El problema está en el tercer paso del protocolo. Un usuario C podría aprovechar el segundo paso del intercambio entre A y B para enviar R_B a A y esperar su contestación, que entonces C enviaría a B haciéndose pasar por A .

6. También es un caso claro del *interleaving attack* o ataque del jugador de ajedrez. El segundo paso del protocolo tendría que autenticar B ante A , pero no lo hace puesto que un impostor C puede tomar R_A del primer paso y empezar otro protocolo con B , con el que conseguirá lo necesario para impersonar a B .

El tercer paso del protocolo tampoco autentica a A frente a B , puesto que el mismo impostor con el valor R_B del segundo paso del protocolo anterior puede empezar otro protocolo con A y conseguir la firma $D_A(R_B)$.

Bibliografía

Menezes, A. J.; Dorschot, P. C.; Vanstone, S. A. (2001). *Handbook of applied cryptography* (5a. ed.). Boca Ratón: CRC Press.

Rifà, J. (1995). *Seguretat computacional*. Cerdanyola del Vallès: Servei de publicacions de la UAB.

Schneier, B. (1996). *Applied cryptography: protocols, algorithms and source code in C* (2a. ed.). Nueva York: John Wiley & Sons.

