# Protocols criptogràfics

Llorenç Huguet Rotger
Josep Rifà Coma
Juan Gabriel Tena Ayuso

PID_00185090

UOC
Universitat Oberta de Catalunya

# Índex

# Introducció

Les comunicacions electròniques ofereixen noves possibilitats en els intercanvis d'informació, sobretot en el camp de les transaccions comercials i en l'administració electrònica.

En el món no electrònic els intercanvis d'informació, duts a terme amb mitjans més convencionals, presenten problemes de seguretat i desconfiança que es resolen per mitjà d'un arbitratge amb una tercera part de confiança com, per exemple, jutges, notaris i agents postals, entre d'altres, que tradicionalment han donat seguretat a aquests intercanvis.

En les transaccions electròniques, dutes a terme mitjançant protocols de comunicació, la seguretat té un paper fins i tot més rellevant perquè necessita donar, a més, una protecció contra possibles amenaces, com són les manipulacions desautoritzades de les dades o les falsificacions.

El conjunt d'accions ben definides i coordinades, descrites per un cert algorisme, que permeten una interacció entre dos o més usuaris per a dur a terme un intercanvi de dades o d'informació, s'anomena **protocol**.

Els **protocols criptogràfics** són aquells que, per a dur a terme aquesta interacció, usen funcions criptogràfiques per tal d'assegurar els requisits de seguretat en les comunicacions entre els usuaris que intervenen en l'intercanvi: la confidencialitat, la integritat, l'autenticitat i el no-rebuig.

Trobarem usos dels protocols criptogràfics en àmbits tan diversos com: en comerç electrònic, en què s'han d'utilitzar signatures compartides; en correu electrònic segur, en què es necessitarà quelcom més que un acusament de rebut; en la implementació de sistemes electrònics de pagaments, amb la utilització de signatures cegues, o en sistemes de votació electrònica.

En molts casos es necessitarà, també, l'arbitratge d'una tercera part de confiança per garantir tots els requisits de seguretat en aquestes transaccions electròniques.

La diversitat d'àmbits fa que hi hagi una àmplia varietat de protocols criptogràfics per a donar resposta als diferents objectius, però tots s'usen, generalment, per a eliminar desconfiances.

Podem fer la classificació següent, segons els objectius que es volen assolir:

- **Gestió i distribució de claus**: permeten i garanteixen la generació, emmagatzemament, manteniment i distribució de claus d'un sistema criptogràfic

de clau privada. En alguns casos, aquesta distribució es farà utilitzant criptosistemes de clau pública.

- **Autenticació d'usuari**: permeten garantir que el remitent d'un missatge, amb el qual s'estableix comunicació, és realment qui pretén ser.

- **Transaccions electròniques segures**: permeten fer les operacions bancàries habituals, en particular implementar sistemes de pagament electrònics substitutius de les targetes de crèdit i de dèbit en les transaccions econòmiques en el comerç electrònic, amb una utilització molt especial en el cas de micropagaments (menys de 10 €).

- **Transferències inconscients o transcordades**: permeten a un usuari enviar un missatge o un secret, entre dos possibles, a un altre usuari. L'usuari emissor no coneix quin dels dos ha rebut el receptor. Aquests protocols permeten la signatura electrònica de contractes.

- **Compromís de bit**: permeten a un usuari d'una xarxa comprometre's amb una elecció d'un bit (o més generalment amb una sèrie de bits) sense revelar tal elecció fins a un moment posterior. El protocol garanteix a l'altra part que l'usuari no canvia la seva elecció.

- **Repartiment de secrets**: permeten distribuir un cert secret, entre un conjunt de participants, de forma que certs subconjunts prefixats entre aquests puguin, unint les seves participacions, recuperar el secret.

- **Proves de coneixement nul**: permeten a un usuari d'una xarxa convèncer a un altre que posseeix una certa informació, sense revelar-li res sobre el contingut d'aquesta.

- **Votacions electròniques**: permeten fer un procés electoral electrònicament, garantint la privadesa de cada votant i la impossibilitat de frau.

# Objectius

En els materials didàctics d'aquest mòdul l'estudiant trobarà els continguts necessaris per a assolir els objectius següents:

**1.** Conèixer els protocols més importants de gestió i distribució de claus.

**2.** Conèixer els protocols més importants d'autenticació.

**3.** Conèixer els protocols més importants de transaccions electròniques segures i sistemes de pagament electrònic.

**4.** Conèixer els protocols més importants per implementar la signatura de contractes i el correu electrònic certificat.

**5.** Conèixer els fonaments dels esquemes de llindar.

**6.** Conèixer els fonaments de les votacions electròniques.

# 1. Protocols de gestió i distribució de claus

Un dels problemes de la criptografia de clau privada és la distribució de les claus entre els usuaris d'una xarxa de comunicacions, en la qual cada parella d'usuaris $A$, $B$ necessita compartir una clau $K_{AB}$, per a crear un canal privat virtual entre tots dos.

Aquesta clau no pot ser enviada per la xarxa de comunicacions mateixa (que considerarem insegura) i, a més, per motius de seguretat, ha de ser canviada periòdicament. Moltes vegades, $K_{AB}$ és d'un sol ús i es denomina **clau de sessió**.

Pel que fa la gestió de claus, aquesta comporta qüestions com qui assumeix la responsabilitat de la creació de les claus (amb diferents alternatives: autoritat central o sistema totalment descentralitzat), diferents tipus de claus (de comunicacions, mestres, de sessió), els requisits de seguretat en l'emmagatzematge, etc.

En aquest apartat farem èmfasi en els protocols de distribució:

**1) Gestió de claus en els sistemes criptogràfics de clau pública**. En un sistema criptogràfic de clau pública, en què cada usuari $U$ disposa de la seva parella de claus: $E_U$ per a xifrar i $D_U$ per a desxifrar, no és necessària la distribució de la clau de xifrar; ans al contrari, han d'estar accessibles per a qualsevol que les vulgui utilitzar per a comunicar-se amb el propietari. Hi ha, no obstant això, el risc de la **impersonació** és a dir, un adversari $C$ pot fer creure que la seva clau pública $E_C$ és la clau pública d'un altre usuari.

**Notació**

Al llarg d'aquest mòdul, quan ens referim a un sistema criptogràfic de clau pública, la notació que usarem, associada a un usuari $U$, serà:

- $Id_U$: identitat usuari.
- $E_U(m)$: procés de xifrar un missatge $m$, amb la clau pública de $U$.
- $D_U(c)$: procés de desxifrar un criptograma $c$, amb la clau privada de $U$.

Considerem quatre esquemes possibles de gestió de claus públiques:

- **Anunci públic**: cada participant difon la seva clau pública a la resta d'usuaris. El risc d'impersonació és gran.

- **Directori públic**: mantingut per una certa autoritat, una tercera part de confiança, amb accés directe i lliure (per a lectura) per part de qualsevol usuari al directori de claus TPD (*trusted public directory*). Tot usuari $U$ registra en el TPD la seva identitat, $Id_U$, i la seva clau pública, $E_U$, en persona o mitjançant comunicació autoritzada, la qual cosa, en principi, elimina el risc de la impersonació.

- **Autoritat pública**: similar a l'anterior, però els usuaris no tenen accés directe al directori de claus, sinó que interaccionen amb un centre de distri-

bució de claus KDC (*key directory centre*). Si l'usuari $A$ vol conèixer la clau pública de l'usuari $B$, ha de formular una petició expressa a l'autoritat que manté el directori.

- **Autoritat certificadora**: és una tercera part de confiança, $T$, que expedeix a cada usuari $U$ un certificat de la seva clau pública, lligada a la seva identitat $Id_U$, juntament amb altres dades. Aquest certificat està signat amb la clau privada de $T$. Així, un usuari $A$ pot enviar a un altre usuari $B$ aquest certificat i aquest pot comprovar (utilitzant la clau pública de l'autoritat $E_T$) la validesa de la clau, la identitat de l'emissor, i també les altres dades incloses, entre aquestes el període de vigència del certificat, etc*.

*** Vegeu el subapartat 2.6 del mòdul "Elements de criptografia".**

**2) Distribució de claus en els sistemes criptogràfics de clau privada**. En els sistemes criptogràfics de clau privada les claus s'han de distribuir de manera centralitzada ja que, si no, l'administrador d'una xarxa hauria de proporcionar claus a cada parella d'usuaris que es volen intercanviar informació secreta. Això vol dir que l'administrador necessitaria tenir emmagatzemades $\binom{n}{2} = \frac{n(n-1)}{2} = O(n^2)$ claus, i cada usuari $(n-1)$ claus.

En un sistema criptogràfic de clau privada, dos usuaris $A$ i $B$ poden establir una clau $K_{AB}$, compartida per tots dos, per dos mètodes:

- **Transport de claus**: un usuari crea una clau i la transfereix, amb seguretat, a l'altre usuari (alternativament, pot ser una tercera part qui la crea i transfereix a tots dos).

- **Acord de claus** (*key agreement*): la clau és derivada pels usuaris com una certa funció d'informació subministrada per tots dos usuaris. En principi cap usuari, per si mateix, no pot predeterminar el valor d'aquesta clau.

Un problema addicional que es pot plantejar és el de l'autenticació de les claus i dels usuaris que les acorden o envien. Tal autenticació serà resolta en els protocols de distribució, en la majoria de casos utilitzant sistemes criptogràfics de clau pública per a distribuir una clau privada.

### 1.1. Protocol de transport d'una clau privada

Aquest protocol permet transferir una clau $K_A$, triada per l'usuari $A$, a un altre usuari $B$, mitjançant un algorisme d'un sol pas.

Suposarem que l'usuari $A$ té accés a una còpia autenticada de $E_B$ i que usa un paràmetre temporal, $TP$.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ envia a $B$: $E_B(Id_A, K_A, TP)$

Ara, l'usuari $B$ pot desxifrar el que ha rebut, amb la seva clau privada $D_B$, i d'aquesta manera verificar la identitat de l'emissor, el paràmetre temporal $TP$ i associar la clau $K_A$ a l'usuari $A$.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $E_B(Id_A,K_A,TP))$ | $\rightarrow$ |

### 1.2. Protocol d'intercanvi de dues claus de Needham-Schroeder

Aquest protocol permet intercanviar dues claus secretes entre dos usuaris $A$ i $B$, respectivament $K_A$ i $K_B$, amb només 3 passos, al mateix temps que els usuaris s'autentiquen mútuament.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ tria un nombre aleatori $R_A$ i envia a $B$: $E_B(Id_A,K_A,R_A)$.
- $B \rightarrow A$. L'usuari $B$ desxifra el que ha rebut, amb la seva clau privada $D_B$, i obté $R_A$, a més de la clau pública i la identitat de $A$. Ara $B$ tria un altre nombre aleatori $R_B$ i envia a $A$: $E_A(K_B,R_A,R_B)$
- $A \rightarrow B$. L'usuari $A$ desxifra el que ha rebut, amb la seva clau privada $D_A$, i obté $R_B$, a més de la clau pública i la identitat de $B$, i envia a $B$: $E_B(R_B)$.

**Autenticacions**

- **Autenticació de *B* per part de *A***: l'usuari $A$ comprova que $R_A$ és correcte, amb la qual cosa autentica $B$ i confirma que ha rebut $K_B$.
- **Autenticació d'*A* per part de *B***: l'usuari $B$ comprova que $R_B$ és correcte, amb la qual cosa autentica $A$ i confirma que ha rebut $K_A$.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $E_B(Id_A,K_A,R_A)$ | $\rightarrow$ |
| $\leftarrow$ | $E_A(K_B,R_A,R_B)$ | $\leftarrow$ |
| $\rightarrow$ | $E_B(R_B)$ | $\rightarrow$ |

### 1.3. Protocol de distribució de claus centralitzat

Aquest protocol permet la distribució d'una clau de sessió per a dos usuaris $A$ i $B$, per utilitzar en un sistema criptogràfic de clau privada.

Suposarem que $A$ i $B$ disposen d'un sistema criptogràfic de clau pública i, en aquest cas, intervindrà una autoritat pública, $T$, que gestiona un centre de distribució de claus KDC.

**Cas pràctic**

Es pot pensar en un protocol en què es vol usar una clau de sessió per a utilitzar amb el sistema AES, mentre que la distribució es fa amb un sistema ElGamal.

Per a iniciar el protocol, primer l'usuari $A$ diu a l'autoritat pública $T$ que es vol comunicar amb $B$, tot sol·licitant la identitat de $B$ i una clau de sessió $K_{AB}$ per a compartir. A més, s'utilitzarà un paràmetre temporal $TP$, per al control del termini de vigència.

**Protocol**

- $T \rightarrow A$. L'autoritat pública $T$ envia a $A$: $E_A(Id_B,K_{AB},TP,E_B(Id_A,K_{AB},TP))$.

L'usuari $A$ desxifra el que ha rebut, amb la seva clau privada $D_A$, i obté: $Id_B$, $K_{AB}$, el paràmetre $TP$ i $E_B(Id_A,K_{AB},TP)$.

- $A \rightarrow B$. L'usuari $A$ envia a $B$: $E_B(Id_A,K_{AB},TP)$.

L'usuari $B$ desxifra el que ha rebut, amb la seva clau privada $D_B$, i obté: $Id_A$, $K_{AB}$ i $TP$.

Finalment, per a iniciar la transmissió, amb total garantia, caldria un acusament de rebut de la recepció per part de $B$ a $A$, contrastant la validesa del valor $TP$.

*Resum de les transaccions del protocol*:

| $T$ | | $A$ |
|---|---|---|
| $\rightarrow$ | $E_A(Id_B,K_{AB},TP,E_B(Id_A,K_{AB},TP))$ | $\rightarrow$ |

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $E_B(Id_A,K_{AB},TP)$ | $\rightarrow$ |

## 1.4. Protocol d'acord de claus de Diffie-Hellman

Aquest protocol es basa en la funció exponencial i el logaritme discret, en un cos finit $\mathbb{Z}_p$, i no necessita cap més intervenció que la dels usuaris mateixos. La seguretat del protocol es basa, precisament, en la intractabilitat del càlcul del logaritme discret.

Tots els usuaris coneixen el valor del primer $p$ i el d'un element primitiu $\alpha \in \mathbb{Z}_p$.

Cada usuari $U$ cerca a l'atzar $x_U \in \mathbb{Z}_p^*$, que guarda secret, i fa públic el valor de $y_U = \alpha^{x_U} \pmod{p}$. Així, cada usuari $U$ posa $(Id_U,y_U)$ en el directori públic $TPD$.

Per a compartir una clau de sessió, $K_{AB}$, entre dos usuaris $A$ i $B$, aquests faran el següent.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ obté $y_B$ del TDP, calcula $K_A = y_B^{x_A} \pmod p = \alpha^{x_B \cdot x_A} \pmod p$.

- $B \rightarrow A$. L'usuari $B$ obté $y_A$ del TDP, calcula $K_B = y_A^{x_B} \pmod p = \alpha^{x_A \cdot x_B} \pmod p$.

Ara $A$ i $B$ ja poden compartir una mateixa clau $K_{AB} = K_A = K_B$ (efectivament: $\alpha^{x_A \cdot x_B} \pmod p = \alpha^{x_B \cdot x_A} \pmod p$) per a intercanviar-se missatges, en un sistema criptogràfic de clau privada, sense que $A$ necessiti el valor secret $x_B$, ni $B$ necessiti el valor secret $x_A$.

*Resum de les transaccions del protocol*:

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

**Exemple 1.1**

$A$ i $B$ volen intercanviar una clau de sessió $K_{AB}$, usant el cos $\mathbb{Z}_p$, amb $p = 1999$ i l'element primitiu $\alpha = 33$. En realitat, aquest valor de $p$ hauria de ser molt gran.

Suposem que $A$ escull $x_A = 47$ i $B$ escull $x_B = 117$. Aleshores, tots dos usuaris faran els càlculs següents:

$A$ calcula $y_A = \alpha^{x_A} \pmod p = 33^{47} \pmod{1999} = 1343$;

$B$ calcula $y_B = \alpha^{x_B} \pmod p = 33^{117} \pmod{1999} = 1991$;

Així, $(Id_A, y_A = 1343)$, $(Id_B, y_B = 1991)$ figuraran en el *TPD*.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ calcula $K_{AB} = y_B^{x_A} \pmod p = 1.991^{47} \pmod{1999} = 1506$.

- $B \rightarrow A$. L'usuari $B$ calcula $K_{BA} = y_A^{x_B} \pmod p = 1.343^{117} \pmod{1999} = 1506$.

La clau secreta compartida per $A$ i $B$ serà $K_{AB} = K_{BA} = 1506$.

**Nota**

El protocol d'acord de claus de Diffie-Hellman és vulnerable enfront d'atacs d'impersonació.

Hi pot haver un espia actiu-passiu, $C$, que enviï missatges a $B$, a partir dels que rep de $A$. Els usuaris $A$ i $B$ creuen que estan interconnectats quan en realitat els seus missatges són filtrats o manipulats per $C$, ja que és $C$ qui realment està connectat amb $A$ i amb $B$.

**Simulador de càlculs en $\mathbb{Z}_p$**

Per a comprovar els càlculs d'aquest exemple podeu usar programes d'ús lliure, com per exemple el SAGE (http://www.sagemath.org).

**Protocol per a impedir la impersonació**

Aquest problema d'autenticació de $A$ per part de $B$, o de $B$ per part de $A$, es pot resoldre amb la intervenció d'una autoritat pública $T$.

L'usuari $A$ envia al $T$ la petició que es vol connectar amb $B$.

- $T \rightarrow A$. L'autoritat pública $T$ envia a $A$ els certificats següents:

  $C_A = D_T(Id_A,E_A,TP)$ i $C_B = D_T(Id_B,E_B,TP)$

  en què $TP$ és un paràmetre temporal.

- $A \rightarrow B$. L'usuari $A$ calcula una clau de sessió $K_{AB}$ i l'envia a $B$, degudament signada i xifrada: $E_B(C_A,C_B,X = D_A(K_{AB}))$.

L'usuari $B$ comprova, aplicant la seva clau privada, $D_B$, al que acaba de rebre; el certificat de $A$, $C_A$, el seu certificat, $C_B$, i la clau pública de $A$, $E_A$, amb la qual pot calcular la clau de sessió: $K_{AB} = E_A(X)$.

A partir d'ara, els usuaris $A$ i $B$ ja poden començar la transmissió segura amb la clau de sessió $K_{AB}$.

*Resum de les transaccions per a evitar l'atac d'impersonació*:

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

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

## 2. Protocols d'autenticació

Els protocols d'autenticació permeten verificar la identitat d'un usuari *A*, davant un usuari *B*, de manera que *B* pugui tenir la certesa que és l'usuari *A* qui li ha enviat un cert missatge.

Són protocols que demostren la identitat de *A* a *B* (en aquest apartat, *B* l'anomenarem el **verificador**). A vegades, la identificació consisteix a demostrar la possessió d'un cert secret (que pot ser una clau), sense necessitat de revelar-lo.

En general, l'usuari *A* haurà de proporcionar una resposta a un cert *desafiament* plantejat pel verificador (normalment, dintre d'un termini de temps determinat). Per això, a vegades, a aquests protocols es denominen de **desafiament-resposta**.

Respecte a la implementació, un protocol d'autenticació hauria de satisfer les característiques següents:

- S'hauria d'implementar juntament amb un protocol d'intercanvi de claus. És aquest últim el que demostra que no solament tothom és qui diu ser, sinó que hi pot haver un intercanvi d'informació.
- S'haurien d'encadenar els missatges que es van creuant dins d'una execució concreta de manera que no poguessin ser trets del context.
- S'hauria d'evitar la utilització de segells de temps, cosa que es coneix com a *timestamps*, com a eina de seguretat, tot i que són útils des d'un punt de vista administratiu i documental.
- S'hauria d'evitar que la revelació de les claus intercanviades en una sessió comprometés el secret de les claus intercanviades prèviament.
- S'hauria de poder afegir a les dades per xifrar, en cada pas, alguna dada seleccionada aleatòriament per un mateix, cosa que es coneix com a *add your own salt*, per tal d'impedir que l'altra part pugui disposar de càlculs que d'altra manera no podria efectuar.

**Definició 2.1 (Protocol d'autenticació segur)**.

Direm que un protocol d'autenticació és segur si, en el moment que un usuari accepta la identitat de l'altre, els registres que tots dos han desat en la sessió d'autenticació coincideixen. I, a més, no ha de ser computacionalment eficient que un tercer usuari pugui recuperar el desafiament acceptat.

**Definició 2.2 (Protocol d'autenticació directa)**.

Direm que el protocol és d'autenticació **directa** si s'acaba intercanviant missatges entre els participants usant les claus intercanviades prèviament.

**Exemple 2.1**

Veurem dos protocols de desafiament-resposta i en tots dos casos usarem un sistema criptogràfic de clau pública.

**Protocol 1**:

Suposem que $A$ per a identificar-se vol demostrar a $B$ que posseeix la clau de desxifratge $D_A$ d'un sistema criptogràfic de clau pública.

- $B \rightarrow A$. El verificador $B$, envia a $A$: $(Id_B,d = (E_A(R_B))$, en què $d$ és el desafiament i $R_B$ un nombre aleatori triat per $B$.

- $A \rightarrow B$. L'usuari $A$, dintre del termini establert, amb la seva clau privada $D_A$, recupera $R_B = D_A(d)$, i l'envia a $B$.

El verificador $B$ acceptarà la identificació de $A$ si el nombre rebut coincideix amb el valor $R_B$ que ell mateix havia enviat.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\leftarrow$ | $Id_B,d = (E_A(R_B)$ | $\leftarrow$ |
| $\rightarrow$ | $R_B = D_A(d)$ | $\rightarrow$ |

**Protocol 2**:

- $B \rightarrow A$. El verificador $B$ envia a $A$: $(Id_B,R_B)$, en què $R_B$ és un nombre triat per $B$.

- $A \rightarrow B$. L'usuari $A$, dintre del termini establert, usant la seva clau privada $D_A$ envia: $(R_A,Id_B,D_A(R_A,R_B,Id_B))$, en què $R_A$ és un nombre triat per $A$, i l'envia a $B$.

El verificador $B$ aplicarà, amb la clau pública de $A$, $E_A$ al darrer component del vector rebut i comprovarà la coincidència de $Id_B$ i $R_A$ amb els dos primers components del vector rebut i, a més, que $R_B$ coincideix amb el que ell mateix li havia enviat.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\leftarrow$ | $Id_B,R_B$ | $\leftarrow$ |
| $\rightarrow$ | $R_A,Id_B,D_A(R_A,R_B,Id_B)$ | $\rightarrow$ |

## 2.1. Protocol de tres passos de Shamir

Aquest protocol permet l'enviament d'informació secreta de $A$ a $B$, sense intercanvi previ de claus, alhora que $B$ podrà tenir la certesa que $A$ és l'emissor.

Per a dur a terme aquest protocol necessitem una funció criptogràfica que sigui commutativa per a cada parella d'usuaris, $E_A \cdot E_B = E_B \cdot E_A$.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ xifra amb la seva clau pública el missatge $m$: $E_A(m)$, i l'envia a $B$.

  L'usuari $B$ no pot calcular $m$, ja que no coneix $D_A$.

- $B \rightarrow A$. L'usuari $B$ xifra amb la seva clau pública el que acaba de rebre: $E_B(E_A(m))$, i ho envia a $A$.

- $A \rightarrow B$. L'usuari $A$, a causa de la propietat commutativa pressuposada, pot desxifrar amb la seva clau privada el que ha rebut i recuperar $E_B(m)$, que envia a $B$. Efectivament: $D_A(E_B(E_A(m)) = D_A(E_A(E_B(m)) = E_B(m)$

Ara $B$ pot conèixer $m$, i desxifra amb la seva clau privada el que ha obtingut, $m = D_B(E_B(m)))$ i, al mateix temps, estarà segur que aquest missatge l'ha enviat $A$.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $E_A(m)$ | $\rightarrow$ |
| $\leftarrow$ | $E_B(E_A(m))$ | $\leftarrow$ |
| $\rightarrow$ | $E_B(m)$ | $\rightarrow$ |

**Nota**

Aquest esquema no assegura l'autenticació ni, en general, secret, tal com es pot veure en l'exemple següent.

**Exemple 2.2**

Usarem el sistema criptogràfic de clau pública, de **Pohlig-Hellman** que, basant la seva seguretat en el problema del logaritme discret, consisteix en el següent:

- Escollir un grup multiplicatiu $\mathbb{Z}_p^*$, en què $p$ és un nombre primer gran.

- Cada usuari tria un valor $e_U$, relativament primer amb $\phi(p) = p - 1$ i calcula $d_U = e_U^{-1} \pmod{\phi(p)}$. La clau secreta de cada usuari serà el valor $d_U$.

- Per un missatge $m$ i un criptograma $c$:

$$\text{Xifratge: } E_A(m) = m^{e_A} \pmod{p} \text{ i}$$
$$\text{Desxifratge: } D_A(c) = c^{d_A} \pmod{p}$$

Evidentment, aquest sistema criptogràfic de clau pública compleix la condició esmentada $E_A \cdot E_B = E_B \cdot E_A$. Per tant, és segur respecte del secret, però no des del punt de visa de l'autenticació, tal com veurem en l'atac d'impersonació de Massey-Omura.

Comencem amb un exemple numèric d'aquest protocol:

Agafant $p = 103$, si l'usuari $A$ agafa $e_A = 19$ ($d_A = e_A^{-1} \pmod{\phi(p)} = 19^{-1} \pmod{102} = 43$) i l'usuari $B$ agafa $e_B = 23$ ($d_B = e_B^{-1} \pmod{\phi(p)} = 23^{-1} \pmod{102} = 71$);

suposant que el missatge que volen enviar és $m = 10$, el protocol de tres passos de Shamir seria el següent:

- $A \rightarrow B$. L'usuari $A$ envia a $B$: $E_A(m) = 10^{19} \pmod{103} = 3$.

  L'usuari $B$ no pot calcular $m = 10$, ja que no coneix $D_A$.

- $B \rightarrow A$. L'usuari $B$ calcula $E_B(E_A(m) = E_B(3) = 3^{23} \pmod{103} = 95$.

- $A \rightarrow B$. L'usuari $A$ pot calcular el valor de $E_B(m)$, mitjançant la seva clau privada $D_A$: $E_B(m) = D_A(E_B(E_A(m) = D_A(95) = 95^{43} \pmod{103} = 27$. L'usuari $A$ envia el valor 27 a $B$.

Aquest valor coincideix, efectivament, amb $E_B(10) = 10^{23} \pmod{103} = 27$.

### Atac d'impersonació de Massey-Omura

Vegem ara que en l'exemple anterior un espia $C$ es pot interposar entre els dos usuaris $A$ i $B$ i pot violar el secret de la transmissió.

Aquest tipus d'atacs, també són coneguts com a *man-in-the-middle*.

### Protocol

Si l'usuari $A$ vol enviar un missatge $m$, xifrat, a l'usuari $B$, calcula $E_B(m) = m^{e_B} \pmod{p}$.

- $A \rightarrow B$. l'usuari $A$ envia $E_B(m)$ a $B$.

Si un espia $C$ intercepta aquest missatge xifrat i es vol fer passar per $A$ pot xifrar aquest criptograma i enviar a $B$:

$$E_C(E_B(m)) = (m^{e_B})^{e_C} \pmod{p}.$$

- $C \rightarrow B$. L'espia $C$ envia $E_C(E_B(m))$ a $B$.

$B$ no té cap mecanisme per a comprovar que el que rep sigui de $A$.

Si $B$ no desconfia, desxifra el missatge rebut:

$$D_B((m^{e_B})^{e_C}) \pmod{p} = (((m^{e_B})^{e_C})^{d_B} \pmod{p} = m^{e_C} \pmod{p}.$$

- $B \rightarrow A$. L'usuari $B$ pensa que envia $E_C = m^{e_C} \pmod{p}$ a $A$.

Però en realitat és l'espia $C$ qui rep aquest valor, i com que ell posseeix $D_C$ pot calcular: $(m^{e_C})^{d_C} \pmod{p} = m$.

D'aquesta manera l'espia $C$ ha pogut conèixer el missatge $m$ que $A$ volia enviar a $B$, sense que ni $A$ ni $B$ hagin detectat l'intrús.

*Resum de les transaccions de l'atac d'impersonació*

| $C$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $E_C(E_B(m))$ | $\rightarrow$ |
| $\leftarrow$ | $E_C(m))$ | $\leftarrow$ |

**Exemple 2.3**

Usarem el mateix sistema criptogràfic de clau pública, de Pohlig-Hellman, que en l'exemple anterior.

Agafant $p = 103$, ja coneixem les claus públiques i privades de $A$ i $B$, respectivament: $\{e_A = 19, d_A = 43\}$ y $\{e_B = 23, d_B = 71\}$.

Suposant que l'espia $C$ té de paràmetres $\{e_C = 5, d_C = 41\}$ l'algorisme de Massey-Omura seguiria els passos següents, tot suposant que el missatge que volen enviar és $m = 10$:

- $A \rightarrow B$. L'usuari $A$ calcula $E_B(m) = 10^{23} \pmod{103} = 27$ i ho envia a $B$.

L'espia $C$ intercepta aquest missatge i calcula:

$E_C(E_B(m)) = (m^{e_B})^{e_C} \pmod{p} = 27^5 \pmod{103} = 80$.

- $C \rightarrow B$. L'espia $C$ envia a $B$, en substitució del que li enviava $A$: $E_C(E_B(m)) = 80$.

L'usuari $B$ no té cap mecanisme per a comprovar que el que rep sigui de $A$ i, en no desconfiar, desxifra el missatge rebut, mitjançant la seva clau privada $d_B$:

$D_B(E_C(E_B(m)) = D_B(80) = 80^{71} \pmod{103} = 90$

- $B \rightarrow A$. L'usuari $B$ pensa que envia a $A$ el valor $90 = D_B(E_C(E_B(m)) = m^{e_C} \pmod{p}$.

Però, en realitat, és l'espia $C$ qui rep aquest valor i, com que ell posseeix $d_C$, pot calcular: $(m^{e_C})^{d_C} \pmod{p} = m$. Per a fer-ho, calcula: $D_C(D_B(E_C(E_B(m)))) = D_C(90) = 90^{41} \pmod{103} = 10$ i per tant, $C$ coneix el missatge $m = 10$ que $A$ enviava a $B$, sense que cap dels dos, criptogràficament, hagi pogut detectat res.

## 2.2. Protocol d'Omura

Aquest és un protocol, semblant al de Diffie-Hellman, d'acord de claus, que permet l'autenticació de dos usuaris, sense cap més intervinent.

Sigui $\alpha \in \mathbb{Z}_p$ un element primitiu i suposem que cada usuari $U$ té una clau secreta $x_U$ i una clau pública $y_U = \alpha^{x_U} \pmod{p}$.

### Protocol

Un cop $A$ ha informat a $B$ que és $A$ i li ho vol provar, el protocol consisteix en els dos passos següents:

- $B \rightarrow A$. L'usuari $B$ tria a l'atzar un valor $x$, calcula $y = \alpha^x \pmod p$ i envia $y$ a $A$.

- $A \rightarrow B$. L'usuari $A$ calcula $y^{x_A} \pmod p$ i envia el resultat a $B$.

L'usuari $B$ verifica la igualtat entre el que acaba de rebre ($y^{x_A} \pmod p$) i el càlcul a partir de la clau pública de $A$ i el valor $X$ escollit per ell ($(y_A)^x \pmod p$). En cas d'igualtat, accepta l'autenticació de $A$ i, en cas contrari, la rebutja.

*Resum de les transaccions del protocol*

| $A$ | | $B$ |
|---|---|---|
| $\leftarrow$ | $y = \alpha^x \pmod p$ | $\leftarrow$ |
| $\rightarrow$ | $y^{x_A} \pmod p$ | $\rightarrow$ |

**Exemple 2.4**

Reprenguem els paràmetres de l'exemple 1.1.

Sigui el cos $\mathbb{Z}_p$, amb $p = 1999$ i l'element primitiu $\alpha = 33$.

Suposem les claus públiques i privades de $A$ i $B$, respectivament:

$\{x_A = 47, y_A = \alpha^{x_A} \pmod p = 1343, x_B = 117, y_B = \alpha^{x_B} \pmod p = 1991\}$

El passos del protocol són ara (suposant que l'usuari $B$ ha estat avisat que l'usuari $A$ es vol autenticar):

- $B \rightarrow A$. $B$ tria el valor $x = 13$, calcula $y = \alpha^{13} \pmod{1999} = 319$ i l'envia a $A$.

- $A \rightarrow B$. $A$ calcula $y^{x_A} \pmod p = 319^{47} \pmod{1999} = 1465$ i envia el resultat a $B$.

L'usuari $B$ verifica si el valor rebut coincideix amb $= (y_A)^x \pmod p$. Efectivament: $1343^{13} \pmod{1999} = 1465$ i l'usuari $B$ dóna per autenticat l'usuari $A$.

## 2.3. Protocol de Needham-Schroeder

Aquest és un protocol d'autenticació i intercanvi de claus, usant un sistema criptogràfic de clau pública i mitjançant un centre de distribució de claus, KDC, gestionat per l'autoritat pública $T$, que segueix els passos següents.

**Protocol**

- $A \rightarrow T$. L'usuari $A$ envia a $T$: $D_A(Id_A,Id_B,R_A)$, en què $R_A$ és un nombre aleatori escollit per $A$.

- $T \rightarrow A$. L'autoritat $T$ retorna a $A$: $E_A(R_A,Id_B,K_{AB},E_B(K_{AB},Id_A))$, en què $K_{AB}$ és la clau de sessió, generada pel *KDC*, que han d'utilitzar $A$ i $B$ en el seu intercanvi.

- $A \rightarrow B$. L'usuari $A$ envia $E_B(K_{AB},Id_A)$ a $B$.

- $B \rightarrow A$. L'usuari $B$ envia, xifrant amb la clau de sessió, $E_{K_{AB}}(R_B)$ a $A$, en què $R_B$ és un nombre aleatori escollit per $B$.

- $A \rightarrow B$. L'usuari $A$ retorna $E_{K_{AB}}(R_B - 1)$ a $B$.

Quan l'usuari $B$, desxifrant amb la clau de sessió, $K_{AB}$, rep el valor de $R_B - 1$, dóna per autenticat l'usuari $A$.

*Resum de les transaccions del protocol*:

| $A$ | | $T$ |
|---|---|---|
| $\rightarrow$ | $D_A(Id_A,Id_B,R_A)$ | $\rightarrow$ |
| $\leftarrow$ | $E_A(R_A,Id_B,K_{AB},E_B(K_{AB},Id_A))$ | $\leftarrow$ |

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $E_B(K_{AB},Id_A)$ | $\rightarrow$ |
| $\leftarrow$ | $E_{K_{AB}}(R_B)$ | $\leftarrow$ |
| $\rightarrow$ | $E_{K_{AB}}(R_B - 1)$ | $\rightarrow$ |

## 2.4. Protocol de Kerberos

Aquest és un protocol de distribució de claus de sessió, mitjançant un centre de distribució de claus, KDC, que proporciona autenticació de l'usuari i estableix la clau de sessió entre dos usuaris $A$ i $B$, usant un sistema criptogràfic de clau privada i terceres parts de confiança.

Hi ha altres versions del protocol per a poder utilitzar sistemes criptogràfics de clau pública.

Considerem $T$, el servidor d'autenticació Kerberos, com a tercera part de confiança.

El protocol de Kerberos es basa en el protocol que acabem de veure de Needham-Schroeder. Es fan servir tiquets i segells de temps, per a assegurar la identitat dels usuaris.

La utilització dels tiquets i del període de validesa de la clau $K_{AB}$ evita que hi pugui haver atacs utilitzant claus de sessió utilitzades prèviament.

**Protocol**

**Autenticació Kerberos**

Kerberos permet l'autenticació de $A$ davant $B$. Inclou segells de temps i una autoritat de certificació en línia, que comparteix una clau privada amb cada usuari per a fer les operacions de xifratge $E_{AT}$,$E_{BT}$. El $ticket_B$, acompanyat de l'autenticador.

- $A \rightarrow T$. L'usuari $A$ envia a $T$: $(Id_A,Id_B,R_A,TS_A)$, en què $R_A$ és un valor aleatori triat per $A$ i $TS_A$ és un segell de temps de $A$.

- $T \rightarrow A$. El servidor d'autenticació envia a $A$: $(ticket_B,E_{AT}(K_{AB},R_A,L,Id_B)$,

  en què $E_{AT}$ és l'operació de xifrar amb la clau privada que comparteixen $A$ i $T$, $L$ indica el període de validesa de la clau $K_{AB}$ i el $ticket_B = E_{BT}(K_{AB},Id_A,L)$.

- $A \rightarrow B$. L'usuari $A$ envia a $B$: $(ticket_B,autenticador)$, en què $autenticador = E_{K_{AB}}(Id_A,TS_A)$, creat per $A$ i conté el segell de temps, $TS_A$, xifrat amb la clau de sessió $K_{AB}$.

- $B \rightarrow A$. L'usuari $B$ envia a $A$: $E_{K_{AB}}(Id_A,TS_A)$

A partir d'això, els usuaris $A$ i $B$ comparteixen una clau de sessió $K_{AB}$ i estan mútuament autenticats.

*Resum de les transaccions del protocol*:

| $A$ | | $T$ |
|---|---|---|
| $\rightarrow$ | $Id_A,Id_B,R_A,TS_A$ | $\rightarrow$ |
| $\leftarrow$ | $ticket_B,E_{AT}(k,N_A,L,Id_B)$ | $\leftarrow$ |

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $ticket_B,autenticador$ | $\rightarrow$ |
| $\leftarrow$ | $E_{K_{AB}}(TS_A)$ | $\leftarrow$ |

## 2.5. Protocol STS

Aquest protocol respon a l'acrònim *station-to-station* i fou creat per Diffie, Oorschot, Wiener com una proposta de protocol segur per a l'autenticació i l'intercanvi de claus, basat en un sistema criptogràfic de clau pública.

Suposarem que, igual que en el protocol de Diffie-Hellman, tots els usuaris coneixen el valor del primer $p$ i de l'element primitiu $\alpha \in \mathbb{Z}_p$ i que cada usuari $U$ puja $(Id_U,y_U)$ al directori públic TPD (recordeu que cada usuari havia escollit un valor $x_U \in \mathbb{Z}_p^*$, que guardava secret, i calculava el valor públic $y_U = \alpha^{x_U} (\text{mod})$.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ tria un valor $x_A$ i calcula $y_A = \alpha^{x_A} \pmod p$. Aleshores, envia a $B$: $(\alpha,p,y_A)$.

- $B \rightarrow A$. L'usuari $B$ tria un valor $x_B$ i calcula $y_B = \alpha^{x_B} \pmod p$ i també calcula la clau de sessió $K_{AB} = y_A^{x_B} \pmod p$.

  Aleshores, envia a $A$: $(y_B,Cert_B,E_{K_{AB}}(D_B\{y_B,y_A\})$,

  en què $\{y_B,y_A\}$ significa el resultat d'aplicar una funció resum als paràmetres $y_B,y_A$.

- $A \rightarrow B$. L'usuari $A$ també pot calcular la clau de sessió $K_{AB} = y_B^{x_A} \pmod p$ i envia a $A$: $(Cert_A,E_{K_{AB}}(D_A\{y_A,y_B\})$

**Consideracions de disseny**

Els valors $p$ i $\alpha$ els tria cada usuari i millor si són diferents, ja que els atacs més eficients al logaritme discret es basen en la construcció de taules a partir del coneixement de $p$.
En el segon pas $B$ envia a $A$ el seu certificat $Cert_B = (Id_B,E_B,D_T(B,E_B))$. D'aquesta manera $A$ podrà treure $E_B$ i comprovar que $y_B$ es correspon amb el resum signat $D_B\{y_B,y_A\}$. Això dóna autenticitat a $B$.
L'usuari $B$ ha d'utilitzar els paràmetres passats per $A$ en el primer pas, en lloc de cercar-los en un directori públic. En el tercer pas el certificat de $A$ assegura a $B$ que aquests paràmetres són correctes.

En aquest protocol, $Cert_A = (Id_A,E_A,\alpha,p,D_T(Id_A,E_A,\alpha,p))$, en què $D_T(Id_A,E_A,\alpha,p)$ és la signatura d'una autoritat de certificació $T$ sobre aquests paràmetres.

En el darrer pas del protocol, l'usuari $A$ queda autenticat davant l'usuari $B$.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $(\alpha,p,y_A)$ | $\rightarrow$ |
| $\leftarrow$ | $y_B,Cert_B,E_{KAB}(D_B\{y_B,y_A\})$ | $\leftarrow$ |
| $\rightarrow$ | $Cert_A,E_{KAB}(D_A\{y_A,y_B\})$ | $\rightarrow$ |

## 2.6. Altres protocols: ISO, CCITT X.509, SSL

Per acabar aquest apartat, vegem altres protocols que es poden considerar estàndard:

**1) ISO (CD 9798-3. 1991)**. És equivalent al protocol STS, en el qual substituïm les exponencials $\alpha^x,\alpha^y$ per valors aleatoris $R_A,R_B$. No obstant això, d'aquesta manera obtenim un protocol d'autenticació però sense intercanvi de claus.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ envia $R_A$ a $B$.

- $B \rightarrow A$. L'usuari $B$ envia $R_A$, juntament amb $(Cert_B, D_B(\{R_B, R_A\}))$, a $A$.

- $A \rightarrow B$. L'usuari $A$ envia a $B$: $Cert_A, D_A(\{R_B, R_A\})$.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $R_A$ | $\rightarrow$ |
| $\leftarrow$ | $R_A, Cert_B, D_B(\{R_B, R_A\})$ | $\leftarrow$ |
| $\rightarrow$ | $Cert_A, D_A\{R_B, R_A\}$ | $\rightarrow$ |

**2) CCITT X.509**. És un algorisme d'intercanvi de claus, amb autenticació, de tres passos i és basat en un sistema criptogràfic de clau pública.

L'usuari $A$, que vol intercanviar una clau $K_{AB}$, amb $B$, té una còpia autenticada de $E_B$.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ envia a $B$ $(S_A, Cert_A, D_A(S_A))$, en què

  $S_A = (R_A, Id_A, E_B(K_{AB}))$, i $R_A$ és un nombre aleatori escollit per $A$.

- $B \rightarrow A$. L'usuari $B$ envia a $A$ $(S_B, Cert_B, D_B(S_B))$, en què

  $S_B = (R_B, Id_B, E_A(K_{AB}))$, i $R_B$ és un nombre aleatori escollit per $B$.

- $A \rightarrow B$. L'usuari $A$ envia a $B$: $(R_B, Id_B, D_A(S_B, Id_B))$.

Evidentment, $B$ ha pogut autenticar $A$ i desxifrant $D_B(E_B(K_{AB}) = K_{AB}$, obté la clau de sessió $K_{AB}$, que emprarà per a comunicar-se amb $A$.

*Resum de les transaccions del protocol:*

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $S_A, Cert_A, D_A(S_A)$ | $\rightarrow$ |
| $\leftarrow$ | $S_B, Cert_B, D_B(S_B)$ | $\leftarrow$ |
| $\rightarrow$ | $R_B, Id_B, D_A(S_B, Id_B)$ | $\rightarrow$ |

**3) SSL (secure socket layer)**. És un protocol obert creat per Netscape i ha esdevingut un estàndard per a Internet. S'implementa entre la capa d'aplicació (HTTP, Telnet, FTP...) i la capa de transport (TCP).

L'SSL és transparent per a l'aplicació que l'utilitza i afegeix els serveis segurs següents a una connexió TCP/IP:

- Xifratge de dades fent servir els sistemes criptogràfics de clau privada: DES, RC4.
- Autenticació, fent servir els sistemes criptogràfics de clau pública: RSA, DSA. Implementa tant l'autenticació mútua emissor/receptor com la de servidor/client.
- Integritat de les dades mitjançant funcions resum MAC (*message authentication code*): SHA, MD5.

El protocol SSL segueix els passos següents:

**1)** $A \rightarrow B$. L'usuari $A$ envia un valor aleatori $R_A$ a $B$.

**2)** $B \rightarrow A$. L'usuari $B$ envia a $A$: $(R_B, Cert_B, D_B(R_B, R_A))$.

**3)** $A \rightarrow B$. L'usuari $A$ envia, opcionalment, a $B$: $(Cert_A, D_A(R_A, R_B))$.

**4)** $B \rightarrow A$. L'usuari $B$ escull una clau de sessió $K_{AB}$ i envia a $A$: $(E_A(K_{AB}), R'_A)$.

**5)** $A \rightarrow B$. L'usuari $A$ envia a $B$: $E_{K_{AB}}(R'_A)$.

En aquest darrer pas, l'usuari $B$ xifra amb la clau $K_{AB}$ el valor aleatori $R'_A$ que li havia enviat $A$ en el pas anterior.

**Fases de l'SSL**

- Fragmentació del missatge en blocs de mida més petita que 214 bytes. TCP afegeix fiabilitat.
- Possibilitat de compressió (opcional).
- Càlcul d'un MAC.
- Xifratge de les dades i del MAC (*message authentication code*).
- Transmissió per TCP.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $R_A$ | $\rightarrow$ |
| $\leftarrow$ | $R_B, Cert_B, D_B(R_B, R_A)$ | $\leftarrow$ |
| $\rightarrow$ | $Cert_A, D_A(R_A, R_B)$ | $\rightarrow$ |
| $\leftarrow$ | $E_A(K_{AB}), R'_A$ | $\leftarrow$ |
| $\rightarrow$ | $E_{K_{AB}}(R'_A)$ | $\rightarrow$ |

**Observacions**

- El pas tercer és opcional.
- En el quart pas, $B$ ha escollit la clau de sessió, $K_{AB}$.
- En el cinquè pas, $A$ xifra amb la clau $K_{AB}$ el valor aleatori $R_A$ que li havia enviat $B$ en el pas anterior.

## 2.7. Protocols d'identificació de coneixement nul: Fiat-Shamir i Schnorr

En els protocols de desafiament-resposta, vistos abans, encara que l'usuari $A$ no reveli el secret, l'usuari $B$ pot aconseguir alguna informació durant el procés.

**Definició 2.3 (Prova de coneixement nul)**.

Una prova de coneixement nul és un procés interactiu en què el candidat convenç el verificador, fins a un nivell acceptable, que coneix, o té, algun secret, sense que el verificador pugui extreure cap informació de la prova que no pogués haver extret per qualsevol altre procediment, amb la participació del candidat o sense o, fins i tot, si aquest menteix en la prova.

**Condicions d'una prova de coneixement nul**

**1)** Si $A$ posseeix el secret, sempre podrà convèncer a $B$ que accepti la seva demostració.

**2)** Si $A$ no posseeix el secret, la probabilitat que enganyi $B$ es pot fer tan petita com es vulgui, repetint prou vegades les etapes.

Les proves de coneixement nul adopten la forma d'una demostració interactiva, impliquen un cert nombre d'etapes i, en cadascuna, se seguirà el protocol següent:

**Protocol bàsic**

- $A \rightarrow B$. L'usuari $A$ vol provar quelcom al verificador $B$ i li envia algun element per a la identificació.
- $B \rightarrow A$. El verificador $B$ presenta un desafiament a $A$.
- $A \rightarrow B$. L'usuari $A$ ha d'efectuar uns càlculs privadament i enviar al verificador $B$ una resposta al desafiament plantejat.

Si alguna de les respostes és incorrecta, $B$ dedueix que $A$ no disposa del secret i rebutja la seva identitat. Per contra, si en totes les etapes la resposta és correcta, aleshores $B$ accepta que $A$ té el secret.

## Protocol de Fiat-Shamir

Sigui $n = p_1 \cdot p_2$, que tothom pot conèixer, generat per una tercera part de confiança $T$ i que manté secrets els valors dels primers $p_1$ i $p_2$.

Cada usuari $U$ tria un element $x_U \in \mathbb{Z}_n^*$ i calcula $y_U = x_U^2 \pmod{n}$. Així, la clau secreta de $A$ serà $x_A$ i la clau pública $y_A$, que registra al directori públic de $T$.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ genera, a l'atzar, un valor $r \in \mathbb{Z}_n^*$, calcula $y_1 = r^2 \pmod{n}$ i ho envia a $B$, juntament amb un missatge que diu que vol provar la seva identitat.
- $B \rightarrow A$. El verificador $B$ envia a $A$ un bit, a l'atzar: $y_2 \in \{0,1\}$.
- $A \rightarrow B$ L'usuari $A$ calcula $y_3$ de la manera següent:

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

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

i envia a $B$ el valor $y_3$.

Finalment la verificació la porta a terme l'usuari $B$, que comprova que

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

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

Si no es compleix la verificació, $B$ rebutja la identitat de $A$.

Aquest protocol s'ha de portar a terme unes quantes vegades. Si es repeteix $k$ vegades, la probabilitat que algú impersoni $A$ és $2^{-k}$, i pot ser tan petita com vulguem, depenent de valor de $k$.

**Impersonació**

Un usuari $C$ impersonant $A$ pot enviat a $B$ $x = \frac{r^2}{y_A} \pmod{n}$. Aleshores, podria respondre correctament $y_3 = r$, només en el cas que $y_2 = 1$. Si $y_2 = 0$ no podrà efectuar els càlculs. Per tant, només té una probabilitat d'encert $\frac{1}{2}$ en cada ronda.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $y_1 = r^2 \pmod{n}$ | $\rightarrow$ |
| $\leftarrow$ | $y_2 \in \{0,1\}$ | $\leftarrow$ |
| $\rightarrow$ | $y_3$ | $\rightarrow$ |

**Exemple 2.5**

Suposem els valors $n = p_1 \cdot p_2 = 5 \cdot 11 = 55$, la clau secreta de $A$, $x_A = 13$, i la clau pública de $A$, $y_A = x_A^2 \pmod{n} = 13^2 \pmod{55} = 20$

Amb aquestes premisses, el protocol segueix els passos següents:

- $A \rightarrow B$. L'usuari $A$ agafa el valor $r = 30$ i calcula $y_1 = r^2 \pmod{n} = 30^2 \pmod{55} = 4$ i ho envia a $B$, juntament amb un missatge en què diu que vol provar la seva identitat.
- $B \rightarrow A$. El verificador $B$ envia a $A$ un bit, a l'atzar: $y_2 \in \{0,1\}$. Suposem $y_2 = 1$.
- $A \rightarrow B$. L'usuari $A$ calcula $y_3$.

Com que $y_2 = 1$, llavors $y_3 = r \cdot x_A \pmod{n} = 30 \cdot 13 \pmod{55} = 5$ i l'envia a $B$.

Finalment, la verificació la porta a terme l'usuari $B$, que comprova, ja que $y_2 = 1$, si $y_3^2 = r^2 \cdot y_A \pmod{n} = y_1 \cdot y_A \pmod{n}$.

Efectivament, $20 \cdot 4 \pmod{55} = 25 = 5^2$, i dóna per bona la identitat de $A$.

### Protocol de Schnorr

Aquest protocol d'identificació és basat en el problema del logaritme discret i fa intervenir una tercera part de confiança, la qual fa les accions següents:

Escull dos primers $p$ i $q$, d'uns 1024 bits i 160 bits, respectivament, tals que $q \mid p-1$. També escull un valor $\beta$, tal que $1 \leq \beta \leq p-1$, d'ordre $q$, i un paràmetre de seguretat $t$, tal que: $t \geq 40$ i $2^t \leq q$.

L'autoritat $T$ envia a cada participant, signada mitjançant la seva clau privada, $D_T(p,q,\beta)$.

Per altra banda, l'usuari que es vol identificar, $A$, disposa de la seva identificació $Id_A$, la seva clau privada, que consisteix en un nombre $a$ tal que $0 \leq a \leq q-1$, i la clau pública corresponent és $v = \beta^{-a} \pmod p$; també es disposa del certificat emès per $T$: $Cert_A = (Id_A, v, D_T(Id_A, v))$.

#### Protocol

$A \rightarrow B$. L'usuari $A$ escull aleatòriament un valor $r$, tal que $1 \leq r \leq (p-1)$, i envia a $B$ el $Cert_A$ i $x = \beta^r \pmod p$

$B \rightarrow A$. Després que $B$ ha verificat la clau pública de $A$, li envia el valor $e$ tal que $1 \leq e \leq 2^t$.

$A \rightarrow B$. L'usuari $A$ envia al verificador $B$: $y = (a \cdot e + r) \pmod q$.

El verificador $B$ accepta la identitat de $A$ si $\beta^y \cdot v^e \pmod p = x$.

Efectivament, com que $\beta$ és d'ordre $q$, aleshores:

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

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $Cert_A, x = \beta^r \pmod p$ | $\rightarrow$ |
| $\leftarrow$ | $e : 1 \leq e \leq 2^t$ | $\leftarrow$ |
| $\rightarrow$ | $y = (a \cdot e + r) \pmod q$ | $\rightarrow$ |

**Exemple 2.6**

Suposem que una tercera part de confiança $T$ ha escollit:

- Els valors dels primers $p = 103$ i $q = 17$. Observeu que $\frac{102}{17} = 6$.

- Per a trobar el valor de $\beta$ cerca un element primitiu de $\mathbb{Z}_p$, com per exemple $\alpha = 6$, i agafa $\beta = \alpha^6 \pmod{103} = 100$. Efectivament, de la construcció es deriva que l'ordre de $\beta$ és 17.

- Per simplificar l'exemple ometrem, sense perdre rigor, el paràmetre de seguretat $t$, $t \geq 40$ i $2^t \leq q$.

L'autoritat $T$ envia a cada participant còpia autenticada de ($p = 103$,$q = 17$,$\beta = 100$).

Els paràmetres de $A$, a part de l'identificador $Id_A$, són la seva clau privada, que suposem $a = 11$, i la seva clau pública corresponent $v = 100^{-11} \pmod{103} = 8$. Així el certificat emès per $T$ és: $Cert_A = (Id_A,8,D_T(Id_A,8))$.

**Protocol**

$A \rightarrow B$. L'usuari $A$ escull aleatòriament un valor $r = 5$ i envia a $B$ el $Cert_A$ i $x = 100^5 \pmod{103} = 66$.

$B \rightarrow A$. Després que $B$ ha verificat la clau pública de $A$, li envia el valor que ha escollit: $e = 13$.

$A \rightarrow B$. L'usuari $A$ envia al verificador $B$: $y = 11 \cdot 13 + 5 \pmod{17} = 12$.

El verificador $B$ accepta la identitat de $A$, ja que $\beta^y \cdot v^e \pmod{p} = 100^{12} \cdot 8^{13} \pmod{103} = 64 \cdot 30 \pmod{103} = 66$, que coincideix amb el valor de $x = 66$ prèviament calculat.

# 3. Transaccions electròniques segures: diners electrònics

Un problema usual en el món de les transaccions és que un usuari $A$ vulgui fer una transacció amb un banc, $B$, de manera que, per una banda, $B$ ha de poder autenticar $A$ i, per altra banda, $A$ ha de tenir un certificat de $B$ que li asseguri que aquest no es desdirà de la transacció ja feta.

Aquest problema és fàcil de resoldre utilitzant les signatures electròniques com a substituts del diner, tal com va proposar David Chaum (al final dels vuitanta).

## 3.1. Protocol de Chaum

Suposem un comprador, $A$, que vol interaccionar amb $B$, que ara considerarem que és el banc, per demanar-li de disposar d'una certa quantitat de diners que, possiblement més tard, voldrà utilitzar per a poder pagar alguna compra.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ genera, a l'atzar, un nombre $m$ gran i crea un document personalitzat, $doc_{AB}$, en el qual explica al banc que vol disposar de $X$ €. Signa el nombre escollit, tot calculant $D_A(m)$ i usant la clau pública de $B$ li envia: $(Id_A, E_B(D_A(m)), doc_{AB})$

- $B \rightarrow A$. El banc rep els missatges, identifica $A$ i llegeix $doc_{AB}$. Amb la clau pública de $A$, i amb la seva clau privada, pot recuperar el nombre $m$. Ara ja pot descomptar els $X$ € del compte d'$A$. El banc signarà el nombre $m$, afegint-hi un document $doc_{BA}$ signat, $D_B(doc_{BA})$. Enviarà a $A$ aquestes dues signatures, xifrades amb la clau pública de $A$: $(E_A(D_B(m)), E_A(D_B(doc_{BA})))$.

A l'usuari $A$ li interessa tenir $(D_B(doc_{BA})$, i no solament $doc_{BA}$, per a poder demostrar davant tercers, en cas necessari, que el banc li ha descomptat $X$ € del seu compte i que aquests corresponen al seu nombre $m$.

Si ara fem intervenir un venedor $V$, quan $A$ va a comprar a $V$, el protocol corresponent seguirà els passos següents:

- $A \rightarrow V$. El comprador $A$ envia a $V$: $(Id_A, D_B(m), D_B(doc_{BA}))$. El venedor podrà autenticar els missatges signats per $B$ i conèixer la identitat del comprador $A$ i el fet que disposa d'un valor de $X$ €.

**Nota**

El comprador $A$ podria enviar aquests valors xifrats, amb $E_V$, i signats amb $D_A$: $D_A(E_V(D_B(m)))$ i $D_A(E_V(D_B(doc_{BA})))$.

- $V \rightarrow B$. El venedor $V$ envia al banc: $D_V(D_B(m))$. El banc comprova que el missatge és correcte (corresponent a un nombre signat prèviament pel banc mateix); diposita en el compte de $V$ els $X \in$ i escriu en una llista de nombres caducats el nombres $m$, per evitar que li torni a ser presentat un altre cop.

- $B \rightarrow V$. El banc envia signat un $doc_{BV}$ de la transacció feta al venedor $V$: $D_B(doc_{BV})$.

A partir d'aquest moment, $V$ ja pot donar la mercaderia a $A$, juntament amb un document signat, $D_V(doc_{VA})$ d'haver cobrat del banc.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $Id_A, E_B(D_A(m)), doc_{AB}$ | $\rightarrow$ |
| $\leftarrow$ | $E_A(D_B(m)), E_A(D_B(doc_{BA}))$ | $\leftarrow$ |

| $A$ | | $V$ |
|---|---|---|
| $\rightarrow$ | $Id_A, D_B(m), D_B(doc_{BA})$ | $\rightarrow$ |

| $B$ | | $V$ |
|---|---|---|
| $\leftarrow$ | $D_V(D_B(m)), D_V(D_B(doc_{BA}))$ | $\leftarrow$ |
| $\rightarrow$ | $D_B(doc_{BV})$ | $\rightarrow$ |

**Exemple 3.1**

Suposem els paràmetres dels criptosistemes RSA de l'usuari $A$, del banc $B$ i del venedor $V$, respectivament:

- $n_A = 7 \cdot 17 = 119$ i $e_A = 5$; aleshores $\phi(n_A) = 6 \cdot 16 = 96$ i $d_A = 5^{-1} \pmod{96} = 77$. Per tant: clau pública $(e_A, n_A) = (5,119)$ i clau secreta $(d_A, n_A) = (77,119)$.

- $n_B = 5 \cdot 11 = 55$ i $e_B = 3$; aleshores $\phi(n_B) = 4 \cdot 10 = 40$ i $d_B = 3^{-1} \pmod{40} = 27$. Per tant: clau pública $(e_B, n_B) = (3,55)$ i clau secreta $(d_B, n_B) = (27,55)$.

- $n_V = 3 \cdot 19 = 57$ i $e_V = 7$; aleshores $\phi(n_V) = 2 \cdot 18 = 36$ i $d_V = 7^{-1} \pmod{36} = 31$. Per tant: clau pública $(e_V, n_V) = (7,57)$ i clau secreta $(d_V, n_V) = (31,57)$.

El protocol de Chaum seguirà els passos següents:

**Protocol**

L'usuari $A$ farà:

Generar a l'atzar $m = 8$.

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

Generar un document personalitzat, $doc_{AB}$, en el qual explica al banc que vol disposar de 300 €.

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

  És a dir: la seva identitat, $E_B(D_A(m)) = E_B(43) = 43^3 \pmod{55} = 32$ i el document $doc_{AB}$.

  El banc, $B$, un cop rebut $E_B(D_A(m)) = 32$, farà:

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

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

  Descomptar 300 € del compte de $A$ i signar el nombre $m$:

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

  A més, el banc redacta un document, $doc_{BA}$, i l'envia signat a $A$: $D_B(doc_{BA})$.

- $B \rightarrow A$. El banc envia a $A$:

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

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

  Suposem ara la intervenció del venedor $V$, lligat al mateix banc $B$.

  Si $A$ fa una comanda al venedor pel valor de 300 €, li lliura $D_B(m) = 2$ i $D_B(doc_{BA})$

- $V \rightarrow B$. El venedor lliura al banc:

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

  El banc farà:

  Comprovar que el missatge és correcte $E_V(D_V(D_B(m))) = D_B(m)$ i també ho és el document $doc_{VB}$.

  Dipositar en el compte del venedor 300 €.

  Escriure en una llista de nombres caducats el nombre $m$ per evitar que es pugui tornar a usar.

- $B \rightarrow V$. El banc envia signat un $doc_{BV}$ de la transacció feta al venedor $V$: $D_B(doc_{BV})$.

  A partir d'aquest moment, $V$ ja pot donar la mercaderia a $A$, juntament amb un document signat, $D_V(doc_{VA})$, d'haver cobrat del banc.

## 3.2. Transaccions sense rastre. Signatures digitals cegues de Chaum

En el protocol que acabem de descriure hi ha *autenticitat* però no *privadesa*. El banc sap (pel nombre $m$ del bitllet) que $A$ ha comprat a $V$ i, en general, podrà saber el rastre de les operacions comercials de $A$.

Per obviar aquest problema, D. Chaum (1992) proposa una variant del protocol usant signatures digitals cegues, que consisteix a fer signar al banc un document que conté un nombre amagat.

D'aquesta manera, l'usuari $A$ treurà a la llum el nombre i el farà servir juntament amb el document signat pel banc, quan li convingui. Per altra banda, el banc no sabrà a qui corresponen els nombres dels bitllets que s'utilitzen.

El mètode de Chaum és vàlid per a qualsevol sistema criptogràfic de clau pública (com RSA) que compleixi $E_U(xy) = E_U(x)E_U(y)$ i $D_U(xy) = D_U(x)D_U(y)$ per a qualsevol usuari $U$.

**Nivell de seguretat de les signatures cegues**

Les signatures cegues garanteixen la integritat, confidencialitat i autenticitat de les dades, al mateix temps que es garanteix l'anonimat del comprador.
El comprador està protegit davant possibles actuacions fraudulentes de manca d'entrega de bens o serveis per part de venedors, ja que en qualsevol moment el comprador pot revelar la seva identitat i, en aquest cas, traçar el flux dels diners.

Aquest protocol el desenvoluparem suposant que estem usant el criptosistema RSA.

**Protocol**

- $A \to B$. L'usuari $A$, abans de donar el nombre $m$ al banc, selecciona un altre nombre gran, $k$, triat a l'atzar i del qual pot calcular $k^{-1} \pmod{n_B}$.

  L'usuari $A$ calcula $E_B(k)$ i envia $m \cdot E_B(k) \pmod{n_B}$, signat i secret a $B$. És a dir, envia a $B$: $E_B(D_A(m \cdot E_B(k))$.

- $B \to A$. El banc desxifra i autentica el missatge rebut i obté $m \cdot E_B(k) \pmod{n_B}$. A continuació signa aquest missatge tot calculant $D_B(m \cdot E_B(k)) = k \cdot D_B(m) \pmod{n_B}$. (Observeu que el banc no coneix el nombre $m$ que acaba de signar, només coneix $k \cdot D_B(m) \pmod{n_B}$.)

  Ara el banc retorna, com en el cas general, un document a $A$, $doc_{BA}$, i també el nombre signat $k \cdot D_B(m) \pmod{n_B}$.

L'usuari $A$ calcula $k \cdot D_B(m) \cdot k^{-1} \pmod{n_B} = D_B(m)$ i obté un nombre signat pel banc que, quan el venedor el presenti al banc, serà fet efectiu i, a més a més, el banc desconeixerà el propietari del nombre del bitllet.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\to$ | $E_B(D_A(m \cdot E_B(k)))$ | $\to$ |
| $\leftarrow$ | $k \cdot D_B(m) \pmod{n_B}, doc_{BA}$ | $\leftarrow$ |

**Exemple 3.2. (Signatura cega de Chaum)**

$A$ vol que el Banc, $B$, li firmi el missatge $m = 65$.

Suposem les claus pública i privada de $B$: $n_B = 851$, $e_B = 13$, $d_B = 61$.

(Els paràmetres de $B$ són $p_B = 23$, $q_B = 37$, $\phi(n_B)= 792$, tots desats en un lloc segur, juntament amb $d_B$.)

Suposem que $A$ escull $k = 51$, i calcula $51^{-1} \pmod{851} = 267$.

- $A \rightarrow B$. L'usuari $A$ calcula:

  $M = m \cdot k^{e_B} \pmod{n_B} = 65 \cdot 51^{13} \pmod{851} = 65 \cdot 458 \pmod{851} = 836$ i envia $M$ al Banc.

- $B \rightarrow A$. El banc fa:

  firma aquest missatge rebut, $M$, amb la seva clau privada:

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

  i l'envia a $A$, juntament amb un document $doc_{BA}$.

L'usuari $A$, com que coneix $k^{-1} \pmod{n_B} = 267$, pot calcular: $276 \cdot 220 \pmod{851} = 21$.

que és la signatura del banc de $m$, sense que el banc conegui $m$.

**Simplificació**

Per més claredat en els passos del protocol, ens hem estalviat signar-lo i xifrar-lo.

**El banc no coneix el nombre que signa**

Amb aquesta operació, el banc ha signat el missatge original $m = 65$, sense conèixer-ne el valor. Observeu que 21 és el mateix valor que obtindria el Banc si hagués signat amb la seva clau privada el missatge $m = 65$: És a dir: $65^{61} \pmod{851} = 21$.

## 3.3. Sistemes de pagament electrònics

Amb l'aparició i generalització del comerç electrònic s'ha fet necessària la creació de sistemes electrònics de pagament adaptats a la situació no presencial dels usuaris involucrats.

Els sistemes de pagament electrònic encara utilitzen, majoritàriament, diners associats a una targeta emesa per una entitat financera, però avui ja podem parlar de diner electrònic, en el sentit que acabem de veure, que és un sistema de pagament més adequat, i més segur, en les transaccions electròniques.

Els sistemes de pagament electrònic actualment són, bàsicament:

- **Xecs electrònics**. Substituts dels xecs de paper, consistent en un missatge amb una signatura digital, que representa un valor monetari, que es fa efectiu per mitjà d'una tercera part de confiança i fa ús de les xarxes interbancàries existents.

- **Moneda electrònica**. Substituta de la moneda física que ha de preservar l'anonimat del comprador i permet fer pagaments que no quedin registrats i no vinculin els usuaris amb les seves compres. Quan la transferència es faci fora de línia, s'haurà de garantir la transferibilitat i la seguretat enfront de la falsificació o de l'ús de la moneda en més d'un pagament.

- **Targeta de crèdit**. Tot i que encara és el sistema de pagament més utilitzat, els receptors es poden posar en contacte amb el banc per a verificar la disponibilitat de fons, però normalment no es fa la verificació de la identitat de l'usuari. A diferència de la moneda electrònica, les targetes de crèdit identifiquen el seu propietari i, a més, els pagaments es poden vincular entre aquests.

- **Micropagaments**. Dissenyats especialment per a reduir els costos de comunicació, emmagatzemament i processament relacionats amb el pagament quan les quantitats per transferir són petites. D'aquesta manera es permet relaxar les mesures de seguretat, ja que els riscos estan més controlats, i en la majoria de casos l'anonimat de l'usuari que fa el pagament se sacrifica per reduir costos.

Els sistemes de pagament estan definits per les regles mitjançant les quals els participants en una operació de compravenda (comprador, bancs emissor i adquirent, comerç, passarel·la de pagament i autoritat certificadora) intercanvien diners i productes o serveis.

Els procediments, o passos, en els sistemes de pagament amb moneda electrònica són, bàsicament:

**1) Establiment del compte**. És una operació que només es fa una sola vegada, en la qual es vincula la identitat o el pseudònim del comprador al nou compte. El comprador disposarà d'un parell de claus corresponents a un sistema criptogràfic de clau pública i d'un certificat que autentica la seva clau pública.

**2) Retirada de fons: obtenció de la moneda electrònica**. L'entitat financera emissora carrega, al compte del comprador, el valor sol·licitat prèviament per a dur a terme una compra o el valor de la moneda electrònica sol·licitada. La moneda electrònica pot ser de dèbit, si s'extreu el valor del compte de l'usuari abans del pagament (aquest procediment requereix l'autenticació de l'usuari), o de crèdit, si el pagament es fa posteriorment i, en aquest cas, la identificació del compte del pagador es fa durant el dipòsit.

**3) Pagament**. El comprador lliura la informació necessària i suficient al comerç per a fer el pagament dels béns que vol comprar. També pot donar aquesta informació a la passarel·la de pagament.

**4) Dipòsit**. L'entitat financera adquirent rep la informació i cobra de l'entitat emissora, per mitjà de la xarxa financera, un cop ha comprovat que no s'ha dipositat dues vegades la mateixa moneda. Finalment, ingressa el valor de la compra, o de les monedes electròniques implicades, en el compte del comerç. En sistemes que permeten la transferibilitat de les monedes electròniques també pot optar per transferir la moneda a un tercer.

Una primera classificació dels sistemes de pagament pot estar marcada per l'import per transferir. Així, parlarem de **micropagaments** quan l'import de l'operació és inferior als 10 €. Més enllà dels 10 € seran **macropagaments**. Encara que tot sistema de pagament ha de satisfer una sèrie de requisits de seguretat, és obvi que en el cas dels micropagaments seran més laxos, per tal que el cost de la seguretat sigui proporcional al valor que es transfereix. No és el mateix protegir un pagament de 10 € que protegir-ne un de 100.000 €.

Una segona classificació és la de sistemes en línia i sistemes fora de línia. Els protocols en línia exigeixen l'accés a un servidor per a cada transacció.

Per als sistemes de pagament en línia, el concepte de diner electrònic, proposat per D. Chaum, permet conjugar les prestacions que ofereixen les xarxes telemàtiques amb les propietats intrínseques dels sistemes de pagament tradicional: anonimat, privadesa i dificultat de falsificació. La diferència entre el diner electrònic i el de curs legal actual és el seu suport, que passa de ser físic a una cadena de bits.

La no-falsificació, o evitar el doble ús, del diner electrònic, exigeix una verificació, contra una certa base de dades, per assegurar-se que no hagi estat utilitzat prèviament. Si la verificació es fa fora de línia necessita més protecció de l'anonimat, és a dir, que no es pugui establir cap vincle entre el diner electrònic i la identitat del propietari, que si es fa en línia. Tal com hem vist abans, aquests sistemes es basen en signatures cegues.

En general, el diner electrònic està pensat per a ser utilitzat a Internet o, actualment, en sistemes basats en terminals mòbils, i es tracta de reemplaçar les monedes i bitllets de curs legal per un sistema informacional, mantenint les prestacions d'anonimat i no traçabilitat utilitzant criptografia de clau pública. Els bitllets i monedes electròniques s'emmagatzemen localment, però requereixen que l'usuari tingui un compte bancari associat, a partir del qual es descomptarà la quantitat que correspongui al bitllet o moneda electrònica.

El funcionament general d'aquest sistema és semblant al que s'ha descrit en el protocol de Chaum:

**1)** El comprador sol·licita monedes electròniques. Aquestes s'identifiquen amb un número de sèrie. Aquest número permet garantir la unicitat, ja que permet detectar duplicitats. L'anonimat del propietari de les monedes electròniques es protegeix mitjançant signatures cegues.

**2)** Un cop el banc ha descomptat el valor corresponent de les monedes electròniques lliurades al comprador; aquestes s'emmagatzemen en el terminal del comprador.

**3)** Quan el comprador fa la comanda de béns o serveis al venedor, li lliura les monedes electròniques corresponents al valor de la compra.

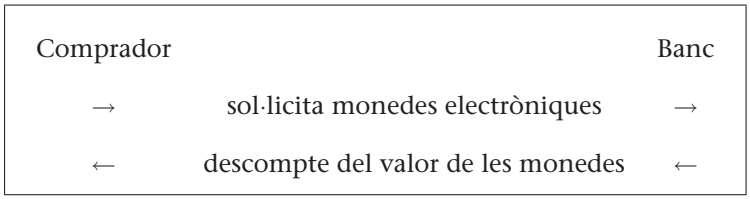
**4)** El venedor, abans de servir la comanda, lliura les monedes electròniques al banc i li sol·licita la validació.

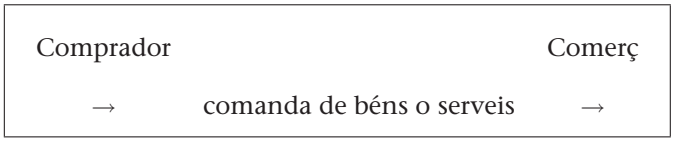
**5)** El servidor del banc accedeix a una base de dades en què consta el número de sèrie de totes les monedes que han estat utilitzades. Amb aquesta operació es detecta si hi ha hagut doble ús. Si no hi ha duplicitat, es respon al venedor sobre l'autenticitat de les monedes electròniques i, si es fa l'operació de compra, automàticament s'ingressa l'import en el seu compte i s'afegeix el número de sèrie corresponent a la base de dades corresponent.

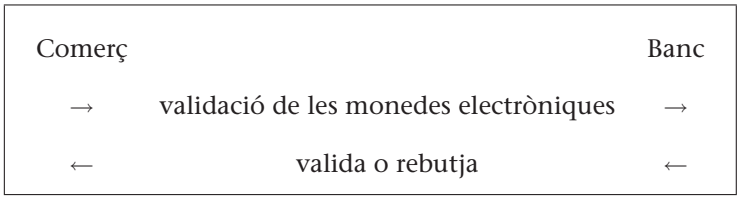
**6)** El venedor lliura els productes al comprador.

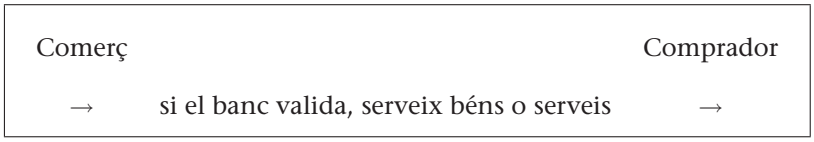
A escala de seguretat, es garanteix la integritat, confidencialitat i autenticitat, al mateix temps que es facilita l'anonimat del comprador. El comprador està protegit davant possibles actuacions deshonestes de venedors quant a no-lliuraments de productes es refereix, ja que en un moment donat el comprador pot desvetllar la seva identitat i en aquest cas es podria traçar el flux dels diners.

*Resum de les transaccions del protocol*:



# 4. Protocols de transferència inconscient

La transferència inconscient o transcordada (*oblivious transfer*) consisteix en la transferència d'un secret entre dos usuaris $A$ i $B$ de manera que la probabilitat que $B$ l'obtingui sigui d'un 50%, sense que $A$ pugui saber si $B$ l'ha obtingut o no.

Aquests tipus de protocols donen lloc a altres protocols més complexos, com els de **compromís de bits** o els de **prova de coneixement nul** i, en general, als protocols que permeten la signatura electrònica de contractes.

## 4.1. Protocol de Rabin

El secret consisteix en la factorització del valor $n_A$ de la clau pública del sistema RSA de $A$.

**Protocol**

- $B \rightarrow A$. L'usuari $B$ tria un valor $x$, $1 \leq x \leq n_A - 1$, i envia a $A$: $z = x^2 \pmod{n_A}$

- $A \rightarrow B$. L'usuari $A$ calcula les quatre arrels quadrades de:

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

  i n'envia una a $B$ (això només ho pot fer $A$, perquè coneix els valors $p_A$ i $q_A$, amb els quals ha calculat $n_A$).

  Sigui $v$ l'arrel enviada a $B$.

L'usuari $B$ comprova si $v$ coincideix amb $y$, o amb $n_A - y$. Si coincideix, es podrà factoritzar $n_A$; en cas contrari no es podrà.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\leftarrow$ | $z = x^2 \pmod{n_A}$ | $\leftarrow$ |
| $\rightarrow$ | $v = z^{-1/2} \pmod{n_A}$ | $\rightarrow$ |

**Nota**

Hi ha una equivalència computacional entre factoritzar $n_A = p_A \cdot q_A$ i calcular arrels quadrades a $\mathbb{Z}_{n_A}$, com queda justificat en el quadre i en el teorema següents.

**Quadrats i arrels quadrades a $\mathbb{Z}_p$**

Si $\beta \in \mathbb{Z}_p$ pot passar que existeixi, o no, $x \in \mathbb{Z}_p$ tal que $x^2 = \beta$.

En el primer cas es diu que $\beta$ és un residu quadràtic. En el segon, es diu que $\beta$ és un residu no-quadràtic.

Llevat de $0 \in \mathbb{Z}_p$, hi ha exactament $\frac{p-1}{2}$ residus quadràtics (QR) i $\frac{p-1}{2}$ residus no quadràtics (QNR) en $\mathbb{Z}_p$.

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

- És computacionalment eficient (hi ha un algorisme determinista-polinòmic) saber si $\beta \in \mathbb{Z}_p$ és QR o QNR.
- No és computacionalment eficient trobar un element QNR. De totes maneres, sí que hi ha algorismes no deterministes, de complexitat polinòmica, que resolen el problema de manera senzilla.
- És computacionalment eficient calcular l'arrel quadrada d'un element QR.

**Teorema 4.1**. Si $n = p_1 \cdot p_2$, en què $p_1$ i $p_2$ són nombres primers diferents i senars, i si $\alpha$ i $\gamma$ són arrels quadrades, essencialment diferents (és a dir, $\alpha \neq \pm\gamma$) d'un cert QR, llavors:

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

**Quadrats i arrels quadrades en $\mathbb{Z}_n$, en què $n = p_1 \cdot p_2$**

Hi ha exactament $\frac{n-1}{4}$ elements QR en $\mathbb{Z}_n$ i cadascun té, exactament, quatre arrels quadrades.
**Exemple**: En $\mathbb{Z}_{15}$, l'element $4 \in \mathbb{Z}_{15}$ és un QR i té quatre arrels, que són: $\{2,13,7,8\}$. Els QR en $\mathbb{Z}_n$, són aquells elements en què mòdul $p_1$ i, també, mòdul $p_2$, són QR

**Exemple 4.1**

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

- $B \rightarrow A$. L'usuari $B$ tria un valor $x = 13$ i envia a $A$: $z = 13^2 \pmod{55} = 4$.
- $A \rightarrow B$. L'usuari $A$ calcula les quatre arrels quadrades de 4 (mod 55) = $\{13,55-13,2,55-2\} = \{13,42,2,53\}$ i n'envia una a $B$.
  Sigui $v = 2$ l'arrel enviada.

L'usuari $B$ comprova si $v = 2$ coincideix amb $y$, o amb $55 - y$. Com que coincideix, es podrà factoritzar $n_A = 55$, ja que $mcd(13 + 2,55) = mcd(15,55) = 5$.

Igual hauria passat si $v = 53$, ja que $mcd(13 + 53,55) = mcd(66,55) = 11$, que és un factor de $n_A = 55$.

Si s'hagués enviat $v = 42$ no s'hauria pogut factoritzar $n_A$, ja que

$mcd(13 + 42,55) = mcd(55,55) = 55$.

Igual per a $v = 13$, ja que $mcd(13 + 13,55) = mcd(26,55) = 1$.

**La probabilitat que $B$ conegui $n_A$ és 1/2**

Del teorema també es dedueïx que la probabilitat que $B$ conegui $n_A = p_1 \cdot p_2$ és 1/2, ja que l'espai de mostres té 4 resultats possibles (les 4 arrels quadrades de $z$), dels quals només 2 serveixen per a factoritzar $n_A$ ($y$ i $n_A - y$). Així, si cada bit d'un secret es transmet mitjançant transferència inconscient, la probabilitat que $B$ pugui conèixer el secret és de $2^{-t}$, si $t$ és el nombre de bits del secret.

## 4.2. Protocols de compromís de bits

En aquest protocol, un usuari $A$ es compromet enfront d'un altre usuari $B$, amb un valor, de tal manera que $A$ no el pugui canviar i $B$ no el pugui descobrir fins que $A$ mateix no hagi obert el compromís.

És un protocol útil en altres protocols com el de llançament de monedes o el de proves de coneixement nul, i ha de complir les propietats següents:

- S'ha de poder comprometre qualsevol dels dos valors possibles per a cada bit.
- De l'obertura del compromís, $B$ només obtindrà el valor del bit compromès.
- No s'ha de poder modificar el valor compromès, encara que es canvïi la manera d'obrir el compromís.
- L'usuari $B$ no ha de poder aprendre res sobre la manera d'obrir els compromisos encara que n'hagi vist obrir alguns.

**Altres protocols de compromís de bits**

En la literatura hi ha altres implementacions de protocols de compromís de bits, usant sistemes criptogràfics de clau secreta, logaritmes discrets, funcions resum, residus quadràtics o, fins i tot, grafs no isomorfs.

### 1) Protocol: amb transferència inconscient

Per a comprometre un bit $b$, l'usuari $A$ escull $n$ bits aleatoris $b_i$, tals que $b_1 \oplus b_2 \oplus \cdots \oplus b_n = b$, i se segueixen els tres passos següents, en què $\oplus$ és l'operació XOR.

**Protocol**

**1) Compromís**: l'usuari $A$ envia a $B$ cada bit $b_i$ per ordre, mitjançant transferència inconscient.

**2) Obertura**: l'usuari $A$ envia a $B$ tots els bits $b_i$.

**3) Verificació**: l'usuari $B$ compara els bits $b_i$ rebuts, dins el termini de compromís, amb els corresponents de l'obertura.

**Nota**

La probabilitat de frau, per part de $A$, és més petita que $\frac{1}{2}$. Si s'executa el protocol $k$ vegades, la probabilitat de frau serà menor que $2^{-k}$.

### 2) Protocol: llançament de moneda per telèfon

El protocol de Rabin pot ser utilitzat pel problema de llençar una moneda a l'aire, per a resoldre una travessa entre dos usuaris $A$ i $B$ de manera telemàtica.

**Travessa**:

Cara $\iff$ B pot factoritzar $n$

Creu $\iff$ B no pot factoritzar $n$

**Protocol**

$A \to B$. L'usuari $A$ construeix, a l'atzar, dos nombres primers, diferents i grans: $p_1$,$p_2$ i calcula $n = p_1 \cdot p_2$. Envia $n$ a l'usuari $B$.

$B \to A$. L'usuari $B$ tria, a l'atzar, un element primitiu, $\alpha \in \mathbb{Z}_n$, calcula $\beta = \alpha^2 \pmod{n}$ i envia $\beta$ a l'usuari $A$.

$A \to B$. L'usuari $A$ calcula les quatre arrels quadrades de $\beta$ i n'envia una a l'usuari $B$. Sigui $\gamma$ l'arrel enviada.

$B \to A$. L'usuari $B$ diu a $A$ si ha sortit cara o creu.

Tal com hem vist abans, si $\gamma \neq \pm\alpha$, l'usuari $B$ podrà factoritzar $n$. Altrament, si $\gamma = \pm\alpha$, no podrà factoritzar $n$.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\to$ | $n$ | $\to$ |
| $\leftarrow$ | $\beta = \alpha^2 \pmod{n}$ | $\leftarrow$ |
| $\to$ | $\gamma = \beta^{-1/2} \pmod{n_A}$ | $\to$ |
| $\leftarrow$ | *CARA o CREU* | $\leftarrow$ |

**3) Protocol: llançament de moneda**

Aquest protocol fou proposat per Mario Blum (1982) i es tracta, tal com hem vist abans, de resoldre una travessa entre dues persones $A$ i $B$, distants entre si, mitjançant el llançament d'una moneda, a cara o creu. Es tracta que si un dels dos fa trampa, sense necessitat d'un tercer, l'altre ho pot detectar.

Hi pot haver dos casos possibles de trampa:

**Trampa de *A***

- L'usuari $A$ llança la moneda i anota el resultat.
- L'usuari $B$ fa la seva travessa i ho diu a $A$.
- L'usuari $A$ diu a $B$ que justament ha sortit el contrari.

**Trampa de *B***

- L'usuari *A* llança la moneda i anota el resultat.
- L'usuari *B* fa la seva travessa i ho diu a *A*.
- L'usuari *A* diu a *B* que justament no ha sortit el que ell havia dit.
- L'usuari *B* es desdiu i replica a *A* dient-li que precisament el que ha sortit era el que ell havia apostat.

Com es pot resoldre el problema per tal que si un fa trampa l'altre ho detecti?

Les solucions impliquen el protocol de transferència inconscient de Rabin, que hem vist, o el protocol proposat per Blum, basat en una funció unidireccional sobre un conjunt de nombres tal que la meitat són parells i l'altra meitat senars.

L'esquema general del protocol, un cop l'usuari *A* ha escollit un nombre de Blum $n = p_1 \cdot p_2$ és:

**Nombres de Blum**

Són els nombres $n = p_1 \cdot p_2$ tals que tots dos primers $p_1$ i $p_2$ són congruents amb 3 mòdul 4.
D'aquesta manera es mantindrà la paritat dels residus quadràtics: $y = x^2 \pmod n$ i $z = y^2 \pmod n$.

**Protocol**

- $A \rightarrow B$. L'usuari *A* escull un valor $x \in \mathbb{Z}_n$ i calcula $y = x^2 \pmod n$ i $z = y^2 \pmod n$. L'usuari *A* envia *z* a *B*.
- $B \rightarrow A$. L'usuari *B*, un cop rebut *z*, aposta per la paritat de *y* (és a dir, si és o no residu quadràtic).
- $A \rightarrow B$. L'usuari *A* l'informa si ha encertat, o no. A més, li mostra els valors de *x* i *y*, al mateix temps que li demostra que *n* és un nombre de Blum.

L'usuari *B* comprova que $y = x^2 \pmod n$ i $z = y^2 \pmod n$.

Tots dos usuaris *A* i *B* han actuat amb un 50% de probabilitat d'engany en el primer i segon pas del protocol.

*Resum de les transaccions del protocol*:

| *A* | | *B* |
|---|---|---|
| $\rightarrow$ | $z$ | $\rightarrow$ |
| $\leftarrow$ | $paritat(y)$ | $\leftarrow$ |
| $\rightarrow$ | $\{Si\,o\,No\},x,y$ | $\rightarrow$ |

**Exemple 4.2**

L'usuari $A$ escull un nombre de Blum, com per exemple: $n = 7 \cdot 11 = 77$.

- $A \rightarrow B$. L'usuari $A$ escull un valor $x = 13$ i calcula:

  $y = x^2 \pmod{n} = 13^2 \pmod{77} = 15$ i

  $z = y^2 \pmod{n} = 15^2 \pmod{77} = 71$.

  $A$ envia $z$ a $B$.

- $B \rightarrow A$. L'usuari $B$, un cop rebut $z$, aposta per la paritat de $y$.

  Suposem que aposta per $y$ parell.

- $A \rightarrow B$. L'usuari $A$ l'informa que no ha encertat en l'aposta.

  Per demostrar-ho, li envia els valors de $x = 13$, $y = 15$ i $n = 77$.

  L'usuari $A$ li demostra que $n$ és un nombre de Blumn i $B$ pot verificar el valor de $z$ a partir dels valors de $x$ i $y$ rebuts.

En aquest cas, $B$ comprova que s'havia equivocat en la travessa.

## 4.3. Signatura electrònica de contractes: protocol d'Even

Es tracta de permetre la signatura electrònica d'un document entre dos usuaris $A$ i $B$, sense intermediaris, de manera que es compleixin aquestes dues condicions:

- Que tots dos usuaris quedin obligats a culminar la signatura, només a partir d'un punt del protocol.

- Que la signatura no es pugui falsificar i, a més, pugui ser comprovada per l'altre usuari.

**1) Protocol bàsic**

Suposem que l'usuari $A$ disposa de dues claus secretes i les claus públiques corresponents d'un sistema criptogràfic de clau pública: $(E_{A1},D_{A1})$; $(E_{A2},D_{A2})$.

Suposem que l'usuari $B$ tria una clau secreta $K_B$.

**Transferència inconscient**

Els passos d'aquest protocol es corresponen bàsicament a una transferència inconscient.

- $A \rightarrow B$. L'usuari $A$ envia a $B$ les seves dues claus públiques $E_{A1}$ i $E_{A2}$.

- $B \rightarrow A$. L'usuari $B$ escull una de les dues claus i xifra la seva clau $K_B$, i envia el resultat a $A$.

- $A \rightarrow B$. L'usuari $A$ escull una de les dues seves claus privades i desxifra el que ha rebut de $B$.

- $A \rightarrow B$. L'usuari $A$ xifra el primer bloc del missatge per signar, usant el valor trobat abans, i l'envia a $B$.

L'usuari $B$ desxifrarà amb la clau $K_B$ el bloc de signatura rebut.

L'usuari $A$ repetirà el tercer i quart pas per a cada bloc de signatura i $B$ sempre desxifrarà amb la clau $K_B$ el bloc de signatura rebut.

Quan s'hagin acabat tots els blocs de signatura, l'usuari $A$ repetirà el quart pas, utilitzant ara l'altra clau privada, i $B$ desxifrarà amb la clau $K_B$ el bloc de signatura rebut.

Si $A$ i $B$ han escollit a l'atzar la mateixa clau (amb una probabilitat del 50%), $B$ desxifrarà un missatge amb sentit en la primera volta. Si no, rebrà un missatge sense sentit i haurà d'esperar fins a rebre l'últim bloc de la segona volta per a obtenir el text en clar.

No obstant això, $A$ no té cap manera de saber quan $B$ ha pogut desxifrar correctament el criptograma, la qual cosa força a acabar el protocol.

**2) Protocol d'Even**

Aquest protocol es basa en sistemes criptogràfics de clau privada i segueix els passos següents:

Inicialment, tant l'usuari $A$ com l'usuari $B$ escullen un conjunt de $2n$ claus en un sistema criptogràfic de clau privada:

L'usuari $A$ tria el conjunt $\{K_1,K_2,\cdots,K_n,K_{n+1},\cdots,K_{2n}\}$, agafades en parelles: $\{(K_1,K_{n+1}),(K_2,K_{n+2}),\cdots,(K_n,K_{2n})\}$.

L'usuari $B$ tria el conjunt $\{K_1^*,K_2^*,\cdots,K_n^*,K_{n+1}^*,\cdots,K_{2n}^*\}$, agafades en parelles: $\{(K_1^*,K_{n+1}^*),(K_2^*,K_{n+2}^*),\cdots,(K_n^*,K_{2n}^*)\}$.

**Protocol**

- $A \rightarrow B$. L'usuari $A$ xifra un missatge $M_A$, conegut per $B$, amb les $2n$ claus:

  $E_{K_1}(M_A),E_{K_2}(M_A),\ldots,E_{K_{2n}}(M_A)$ i envia els $2n$ criptogrames ordenats a $B$.

  L'usuari $A$ es comprometrà, més endavant, a signar el contracte si $B$ li pot presentar algun parell de claus $(K_i,K_{n+i})$.

- $B \rightarrow A$. L'usuari $B$ xifra un missatge $M_B$, conegut per $A$, amb les $2n$ claus $E_{K_1}^*(M_B),E_{K_2}^*(M_B),\ldots,E_{K_{2n}}^*(M_B)$ i envia els $2n$ criptogrames ordenats a $A$.

L'usuari $B$ es comprometrà, més endavant, a signar el contracte si $A$ li pot presentar algun parell de claus $(K_i^*, K_{n+i}^*)$.

- $A \rightarrow B$. L'usuari $A$ envia a $B$ cada parell $(K_i, K_{n+i})$ ordenats mitjançant una transferència inconscient; és a dir, enviant $K_i$ o $K_{n+i}$ amb igual probabilitat.

- $B \rightarrow A$. El mateix farà $B$, i enviarà a $A$, ordenadament, els parells $(K_i^*, K_{n+i}^*)$

**Nota**

En aquest punt $A$ i $B$ tenen la meitat de les claus un de l'altre.

Si la llargada de cada clau és de $L$ bits, aleshores els usuaris $A$ i $B$ fan el bucle següent, $1 \leq i \leq 2n$, per les claus $K_i$ i $K_i^*$ que no s'han usat en els passos anteriors:

**Algorisme 4.2**

for $1 \leq i \leq L$

begin

$A$ envia a $B$ el bit $j$-èsim de totes aquestes claus $K_i$

$B$ envia a $A$ el bit $j$-èsim de totes aquestes claus $K_i^*$

end

Observeu que l'algorisme es semblant al del protocol de compromís bit a bit. En fer aquest bucle complet, $A$ i $B$ tenen les $2n$ claus un de l'altre i se suposa que poden signar el contracte.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $E_{K_1}(M_A), E_{K_2}(M_A), \ldots, E_{K_{2n}}(M_A)$ | $\rightarrow$ |
| $\leftarrow$ | $E^*_{K_1}(M_B), E^*_{K_2}(M_B), \ldots, E^*_{K_{2n}}(M_B)$ | $\leftarrow$ |
| $\rightarrow$ | $\{K_i \, o \, K_{n+i}\}$ | $\rightarrow$ |
| $\leftarrow$ | $\{K_i^* \, o \, K_{n+i}^*\}$ | $\leftarrow$ |

## 4.4. Protocol de correu electrònic certificat

Malgrat que els sistemes actuals de correu electrònic permeten l'acusament de rebut per part del receptor (l'emissor demana al receptor que li comuniqui

que ha rebut el correu que li ha enviat, sense que això impliqui l'acceptació del contingut), quan enviem un correu electrònic, com podem estar segurs que el missatge enviat ha arribat al destinatari autoritzat i només ell en coneix el contingut?

La proposta d'aquest protocol és la implementació del correu electrònic certificat.

Així, si un usuari $A$ vol enviar un missatge electrònic com a correu certificat a un usuari $B$, li descobrirà el missatge, és a dir, li enviarà la clau, només després que l'usuari $B$ li hagi enviat l'acusament de rebut corresponent.

Això és talment el que passa amb el correu certificat; abans de veure el missatge hem signat que hem rebut el sobre que el conté.

El protocol que segueix és molt similar al que hem vist d'Even per a la signatura de contractes.

**Protocol**

L'usuari $A$ escull aleatòriament $n+1$ claus $\{a_0,a_1,\cdots,a_n\}$ d'un sistema criptogràfic de clau privada. Les claus $\{a_1,\cdots,a_n\}$ seran la part esquerra de la clau $KE_{A_i}$, i la part dreta, $KD_{A_i}$ serà $\{a_{n+1},a_{n+2},\cdots,a_{2n}\}$, en què $a_{n+i} = a_0 \oplus a_i$, per $1 \leq i \leq n$.

Els usuaris $A$ i $B$ es posen d'acord en un missatge de validació $V$.

L'usuari $A$, amb la clau $a_0$ xifra el document $M$, $C_0 = E_{a_0}(M)$ i, després, xifra el missatge de validació $V$ amb les $2n$ claus secretes; $\{KE_{A_i}, KD_{A_i}\}$ per a $1 \leq i \leq n$:

Xifratge de validació de la part esquerra: $VE_{A_i} = E_{KE_{A_i}}(V)$.

Xifratge de validació de la part dreta: $VD_{A_i} = E_{KD_{A_i}}(V)$.

- $A \rightarrow B$. L'usuari $A$ envia a $B$, el valor $C_0$ i els parells ordenats $(VE_{A_i},VD_{A_i})$, per a $1 \leq i \leq n$.

De manera similar, l'usuari $B$ genera $n$ parelles de claus $KE_{B_i}$ i $KD_{B_i}$. També generarà $n$ parelles de missatges **acusament de rebut de la part esquerra:** $RE_i$ i **acusament de rebut de la part dreta:** $RD_i$ i xifra les parelles de $(RE_i,RD_i)$ en el sistema criptogràfic de clau privada, amb les claus de $(KE_{B_i},KD_{B_i})$ obtenint $(E_{B_i},D_{B_i})$.

- $B \rightarrow A$. L'usuari $B$ envia a $A$ les parelles ordenades $(E_{B_i},D_{B_i})$.
- $A \rightarrow B$. Mitjançant una transferència inconscient, l'usuari $A$ envia a $B$ una de les dues claus secretes: $KE_{A_i}$ o $KD_{A_i}$.
- $B \rightarrow A$. Mitjançant una transferència inconscient, l'usuari $B$ envia a $A$: $KE_{B_i}$ o $KD_{B_i}$.

Aquest procés es repeteix fins haver enviat els $n$ valors de les claus de cada usuari.

*Resum de les transaccions del protocol*:

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $C_0, \{VE_{A_i}, VD_{A_i}\}$ | $\rightarrow$ |
| $\leftarrow$ | $\{(E_{B_i}, D_{B_i})\}$ | $\leftarrow$ |
| $\rightarrow$ | $\{KE_{A_i}\, o\, KD_{A_i}\}$ | $\rightarrow$ |
| $\leftarrow$ | $\{KE_{B_i}\, o\, KD_{B_i}\}$ | $\leftarrow$ |

**Comprovació**:

- L'usuari $B$ usa les claus enviades per $A$, $KE_{A_1}$ o $KD_{A_1}$, per a comprovar que en desxifrar $D_{KE_{A_i}}(V)$ o $D_{KD_{A_i}}(V)$ obté el missatge de validació $V$.

- L'usuari $A$ usa les claus enviades per $B$, $KE_{B_i}$ o $KD_{B_i}$, per a comprovar que en desxifrar $D_{KE_{B_i}}(E_{B_i})$ o $D_{KD_{B_i}}(D_{B_i})$ obté sempre $RE_i$ o $RD_i$.

Aquests passos són en realitat un protocol de **compromís** de bit entre els usuaris $A$ i $B$.

**Verificació**:

- L'usuari $A$ ha obtingut totes les claus de $B$ i comprova totes les parelles d'acusaments de rebut; és a dir, comprova la part esquerra i la part dreta dels acusaments de rebut.

- L'usuari $B$ ha obtingut totes les claus de $A$ i comprova que tots els enviaments de $A$ contenen el missatge de validació $V$.

  Realment, l'usuari $A$ haurà hagut de mostrar totes les claus a $B$, per tal que $B$ pugui comprovar que $A$ ha usat el càlcul $a_{n+i} = a_0 \oplus a_i$.

- Com que l'usuari $B$ disposa de totes les claus de $A$, pot calcular: $a_0 = KE_{A_i} \oplus KD_{A_i}$.

  Observeu que per a fer el càlcul anterior qualsevol de les parelles d'acusament de rebut és vàlida.

- L'usuari $B$ desxifra el criptograma $C_0$, enviat en el segon pas, i recupera així el document $M$: $D_{a_0}(C_0) = M$.

# 5. Esquemes llindar i repartiment de secrets

Els protocols per a compartir secrets tracten de resoldre el problema següent: donat un secret, repartir uns fragments d'informació entre diverses persones, de manera que certes agrupacions d'aquestes persones puguin recuperar el secret, però les restants agrupacions no siguin capaces d'obtenir-lo.

Per exemple, un banc té una càmera cuirassada que s'ha d'obrir cada matí amb una certa clau (el secret), que comparteixen 5 caixers encarregats de tal obertura. El banc pot estar interessat que, per a obrir-la, almenys calguin 3 dels 5 caixers.

Així, formalment, un secret es divideix en $k$ participacions, que es reparteixen entre $n$ participants. La finalitat és que la presència de $k$ d'aquests participants ($k \leq n$) sigui necessària i suficient per a reconstruir el secret.

**Definició 5.1 (Esquema llindar)**.

Per a un parell de valors $k$ i $n$, $k < n$, un $(k,n)$-esquema llindar consisteix en $n$ participacions $D_1,D_2,\ldots,D_n \in \mathbb{Z}_p$ i un valor secret $D \in \mathbb{Z}_p$ tal que:

**1)** El coneixement de $k$, o més participacions $D_i$, és suficient per a calcular la clau secreta $D$.

**2)** El coneixement de $k-1$, o menys participacions, deixa la clau $D$ completament indeterminada.

## 5.1. Esquema de Shamir

Aquest esquema treballa amb polinomis en un cos finit $i$, en què el valor secret serà el terme independent d'un cert polinomi. La idea de l'esquema es basa en el fet de que un polinomi de grau $k$–1 es pot caracteritzar pels seus $k$ coeficients (la clau) o bé donant valors en $k$ punts diferents (les participacions).

Sigui $\mathbb{Z}_p$ un cos finit. Suposem que el valor que volem desar secret és $D = a_0 \in \mathbb{Z}_p$ i que $a_1,a_2,\ldots,a_{k-1} \in \mathbb{Z}_p$ són valors agafats a l'atzar, que considerem com a coeficients del polinomi $A(x) = a_0 + a_1x + \ldots + a_{k-1}x^{k-1} \in \mathbb{Z}_p[x]$.

Considerem $n$ elements diferents $x_1,x_2,\ldots,x_n \in \mathbb{Z}_p$ i calculem, per a cadascun, $A(x_i) = D_i$.

**La participació que es donarà en l'$i$-èsim participant és $D_i$.**

Qualsevol grup de $k$, o més participants, permet reconstruir, gràcies al teorema d'interpolació de Lagrange, el polinomi $A(x)$ i, per tant, calcular la clau $D = a_0$.

En efecte, siguin $\{i_1, \cdots, i_k\}$ els $k$ participants; aleshores:

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

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

d'on podrem retrobar el secret $D = a_0$.

**Interpolació de Lagrange**

El teorema d'interpolació de Lagrange ens permet calcular els coeficients $a_i$ de manera computacionalment més senzilla que no pas resolent el sistema d'equacions corresponent.

**Exemple 5.1**

Per a més claredat de les operacions, treballarem en $\mathbb{R}$.

Suposem que el secret és $D = 15$ en un (3,5)-esquema llindar. Escollirem un polinomi $A(x)$, de grau 2, que tingui per terme independent $a_0 = D = 15$.

Sigui, per exemple, $A(x) = 16x^2 + 2x + 15$ i els cinc valors $x_i = \{1,3,4,7,10\}$, amb els quals calcularem $D_i = A(x_i)$.

Així tindrem els cinc punts $\{(1,33),(3,165),(4,279),(7,813),(10,1635)\}$.

Suposem que els usuaris 1,2,4 s'agrupen per a calcular el secret. Aleshores:

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

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

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

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

amb la qual cosa hem recuperat el secret $D = a_0 = 15$.

**Nota**

També podem calcular el valor $a_0$, de l'esquema llindar resolent el sistema d'equacions, per a $i = j_1, \ldots, j_k$:

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

El determinant d'aquest sistema és de tipus Vandermonde i tots els valors $x_i$ són diferents, i per tant el sistema té solució única.

Diguem $\varphi(x) = (x - x_1) \ldots (x - x_k)$ i $\psi(x) = x\varphi(x)$, llavors:

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

**Mentiders en l'esquema llindar de Shamir**

Suposem que un participant, per exemple el número 1, és un mentider i dóna el valor $D_1^*$ ($D_1^* = D_1 + \varepsilon$) com la seva participació.
En aquest cas la clau que el grup de $k$ participants trobarà, és:
$a_0^* = -\varphi(0)\frac{D_1^*}{\psi'(x_1)} - \varphi(0)\sum_{i \neq 1} \frac{D_i}{\psi'(x_i)} = a_0 - \varepsilon\frac{\varphi(0)}{\psi'(x_1)}$.
El mentider coneix $D_1, D_1^*$ i, a més a més, $\varepsilon$ i $a_0^*$.
A partir de l'equació anterior pot calcular el valor correcte $a_0$.
**Els altres participants no poden calcular la clau correcta** $a_0$.
Per evitar els mentiders en un esquema de llindar, J. Rifà (*How to avoid the Cheaters succeed in the Key Sharing Scheme. Desings, Codes and Cryptography*, 1993) proposa un esquema $(2k - 1, 2n)$ en el qual es donen dues participacions a cadascun dels $n$ participants, de manera que per a calcular la clau es necessiten un mínim de $k$ participants.

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

El valor de $\psi'(x_i)$ no depèn de $D_i$, però el càlcul de la clau $a_0 = D$ necessita d'aquests valors $D_i$.

# 6. Votacions electròniques

Una votació feta electrònicament necessita garantir una sèrie de condicions formalment anàlogues a les d'una votació tradicional:

- **Democràcia**: només les persones registrades en el cens poden emetre el seu vot i solament ho poden fer una vegada.
- **Transparència**: cap vot pot no ser eliminat ni alterat.
- **Privadesa**: no es pot establir cap relació entre un vot i un votant.
- **No-coercibilitat**: per a evitar coaccions el votant no pot demostrar com ha estat el sentit del seu vot.
- **Verificabilitat**: cada votant, i eventualment un auditor, pot comprovar que el vot ha estat correctament comptabilitzat.

**Agraïment**

Els autors agraeixen la col·laboració en aquest apartat de Jordi Puiggalí, VP Research and Development de SCYTL (http://www.scytl.com).

En una votació presencial, aquestes condicions queden garantides per una urna transparent, la cabina de votació i l'escrutini públic. En el cas electrònic, el procés electoral es porta a terme en una dimensió lògica (és a dir, mitjançant un conjunt de programes que s'executen en un ordinador), i per tant, no auditable per observadors o per al votant. Per això, per assegurar que els processos electorals es duen a terme de manera totalment honesta i privada, hi ha diferents propostes basades en la utilització de protocols criptogràfics.

Els protocols criptogràfics per a vot electrònic es basen sobretot a aconseguir dos objectius principalment:

- **Garantir la privadesa dels votants i la correcció dels resultats**: assegurant que tots els vots que s'han utilitzat per a obtenir els resultats pertanyen a votants vàlids (és a dir, que formen part de la llista del cens i no han estat suplantats), verificant que un votant no emeti més d'un vot, i fent que no es pugui correlacionar en cap moment la papereta del vot i la identitat del votant.

- **Facilitar l'auditoria de l'elecció**: permetent tant a votants com observadors verificar que els vots emesos contenen l'opció del vot original seleccionada pel votant i, per tant, que el resultat reflecteix totalment la intenció de vot dels votants.

Per a poder assolir aquests objectius els esquemes criptogràfics proposats combinen altres protocols o algorismes criptogràfics més bàsics: autenticació, firmes cegues, proves de coneixement nul, repartiment de secrets, etc.

### 6.1. Garantir la privadesa i la correcció dels resultats

Des del punt de vista de la privadesa dels votants i la correcció dels resultats, els protocols criptogràfics parteixen de la base del xifratge i la signatura digital posterior dels vots. El xifratge s'utilitza per a evitar que la intenció de vot sigui visible per a qualsevol administrador o persona que disposi de privilegis sobre el sistema de vot. En aquest sentit, s'utilitzen com a base algorismes de clau pública (com per exemple RSA), i es deixa la clau privada en mans de l'equivalent a una mesa electoral per a fer el desxifratge dels vots. D'aquesta manera, els vots es xifrarien amb una clau pública de la mesa electoral en el mateix terminal des del qual el votant ha fet les seleccions de vot, per evitar que el contingut se sàpiga abans de desar el vot en l'urna (que en aquest cas seria electrònica). El vots només es podrien desxifrar quan ho decidís la mesa electoral, que és la que té accés a la clau privada que els desxifra.

Per a evitar que els vots es puguin manipular un cop emesos, els vots xifrats se signarien digitalment preferentment amb la clau d'identitat de l'elector (per exemple, DNI electrònic). D'aquesta manera, qualsevol intent de manipulació del vot es detectaria, ja que invalidaria la signatura digital del vot. A més, la signatura digital permet verificar si el votant que ha emès el vot pertany realment al cens electoral. D'aquesta manera es pot assegurar que els resultats s'han obtingut de vots de votants que realment podien participar.

Aquest mecanisme de xifratge i signatura es coneix com el del *doble sobre*, ja és equivalent al mecanismes de dos sobres utilitzats en el vot postal. El sobre de dins és el que protegeix la privadesa del vot i, per tant seria el que s'obté en xifrar el vot. El sobre de fora és el que conté el sobre de dins i la identitat del votant, per a poder verificar que el vot prové d'un votant vàlid abans d'introduir-lo a l'urna. En el cas del vot electrònic seria la signatura digital que es fa sobre el vot. De la mateixa manera que en el vot postal el sobre extern dels votants vàlids se separa del sobre intern abans de fer el recompte, en el vot electrònic es podria separar la signatura digital del vot xifrat abans de desxifrar el vot i fer el recompte.

Tot i que aquestes mesures podrien semblar suficients per a garantir la integritat i privadesa, no es considera que ho siguin en un entorn de vot electrònic. Per exemple, si es fa un desxifratge directe dels vots seria fàcil correlacionar els vots en clar amb els votants: només caldria verificar l'ordre en el qual han votat els votants i el que s'obté en desxifrar els vots. Per aquesta raó, els protocols criptogràfics fan propostes de mecanismes de desxifratge de vot que trenquen qualsevol correlació entre els vots desxifrats i l'ordre en el qual han estat emesos. En aquest sentit es poden destacar dues famílies de propostes que implementen un procés de desxifratge que trenca aquesta correlació: la dels protocols de recompte homomòrfic i la dels protocols de mescla.

Els protocols de recompte homomòrfic es fonamenten a obtenir els resultats sense haver de desxifrar els vots individuals. Per a aconseguir aquest objec-

tiu, les operacions de recompte les fan directament sobre els vots xifrats i només desxifren els resultats d'aquestes operacions. Per a poder fer operacions amb els vots xifrats sense haver-los de desxifrar, utilitzen, per al xifratge dels vots, algorismes criptogràfics amb propietats homomòrfiques (per exemple, ElGamal). Aquestes propietats permeten fer operacions directament amb la informació xifrada sense necessitat de desxifrar-la. Un exemple seria multiplicar dues dades xifrades per a obtenir el xifratge del producte dels valors d'aquestes dades: $E(x)\cdot E(y) = E(x\cdot y)$. D'aquesta manera només caldria desxifrar el resultat de multiplicar les dades xifrades (és a dir, $E(x\cdot y)$) per a obtenir el producte dels valors (és a dir, $x\cdot y$). També hi ha algorismes criptogràfics en els quals el resultat de multiplicar les dades xifrades és el xifratge de la suma dels continguts: $E(x)\cdot E(y) = E(x+y)$. Aquests són normalment els utilitzats pels protocols homomòrfics, ja que els vots es representen com un vector de valors binaris amb tantes posicions com opcions de vot disponibles. Si una opció ha estat seleccionada el valor de la seva posició serà 1, mentre que si no ho ha estat serà 0. Per exemple, si hi ha quatre opcions i la primera, la segona i la tercera han estat seleccionades, el vot es representaria formalment de manera simplificada com: (1,1,1,0). D'aquesta manera, en multiplicar els vots xifrats el que estem fent és sumar les vegades que les opcions han estat triades i només caldria desxifrar els resultats de multiplicar tots els vots xifrats per a obtenir els resultats. Per exemple:

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

Per tant, en desxifrar, obtindríem que la primera opció és la guanyadora, amb dos vots, mentre que la segona i tercera haurien obtingut un sol vot. Tot i que aquests protocols són bastant eficients, tenen algunes limitacions, com el fet que només funcionen en eleccions en les quals els vots es poden representar de manera numèrica o que són poc escalables en eleccions de molts candidats. Aquestes limitacions no existeixen en els protocols basats en la mescla.

El protocols de mescla utilitzen el mateix concepte que en les eleccions tradicionals: sacsejar l'urna abans d'obrir-la per garantir que els vots no es troben en el mateix ordre que s'han introduït (és a dir, evitar que el vot del damunt sigui el que ha emès l'últim votant). Aquest sacsejament seria equivalent a aplicar una permutació als vots de l'urna. El problema és que aplicar només una permutació no garanteix que el vot no es pugui rastrejar, ja que només cal buscar on es trobava un vot amb el mateix xifratge abans d'aplicar-li la permutació. Per tant, en aquests protocols, a més de permutar les posicions dels vots, també es fa un desxifratge parcial o rexifratge del vot. D'aquesta manera, no és possible correlacionar els vots un cop permutats i redesxifrats amb els vots xifrats originals, ja que seran tots diferents.

En el cas del desxifratge parcial implica que el votant, a més de xifrar el vot amb la clau pública de la mesa electoral, també l'ha de xifrar amb la clau pública de l'entitat que fa la mescla. L'entitat que executa la mescla disposa

d'una clau privada per a fer un primer desxifratge parcial del vot un cop permutat. En algunes implementacions, hi ha més d'una entitat que fa el procés de mescla i desxifratge parcial, amb la qual cosa resulta que el votant ha de fer tants xifratges com entitats participin.

En el cas de mescla amb rexifratge, és necessari utilitzar un algorisme amb propietats homomòrfiques per a poder utilitzar la mateixa clau pública per a rexifrar el vot. D'aquesta manera, no cal desxifrar el vot tantes vegades com ha estat rexifrat. Per tant, aquest procés és totalment transparent al votant, que només ha de xifrar el vot una vegada amb la clau pública de la mesa electoral. El procés de mescla i rexifratge es pot fer tantes vegades com entitats intervinguin en el procés. En qualsevol cas, l'última entitat és la que acaba constituint la mesa electoral i desxifra els vots.

### 6.2. Garantir l'auditoria de la votació

Els processos descrits anteriorment, garanteixen principalment la privadesa del vot, però que passa si l'entitat que fa el desxifratge total presenta un resultat que no es correspon amb la intenció de vot?

Per a detectar aquesta situació, s'utilitzen els mecanismes d'auditoria dels processos de desxifratge. Aquests processos utilitzen mecanismes criptogràfics, com proves de coneixement nul, per a demostrar que el procés de desxifratge o rexifratge no ha modificat el vot quan aquests es duen a terme. En aquest sentit destaquem dos tipus de proves de coneixement nul importants: les de desxifratge correcte i les de re-xifratge correcte. En el primer cas, es permet demostrar matemàticament de manera irrefutable que el text desxifrat estava contingut en el vot xifrat. En el cas de re-xifratge, la prova demostra que el valor del rexifratge s'ha obtingut en rexifrar el text prèviament xifrat pel votant. En qualsevol dels casos, no cal lliurar la clau privada per a demostrar que el valor desxifrat és correcte, sinó una prova matemàtica irrefutable sobre el contingut desxifrat que pot ser validada de manera universal (és a dir, per qualsevol auditor o votant). D'aquesta manera es pot garantir que els resultats reflecteixen de manera irrefutable els continguts dels vots xifrats. En cas contrari es poden aïllar els vots conflictius amb una auditoria posterior.

# Exercicis d'autoavaluació

**1.** En una xarxa de comunicacions cada usuari $U$ té el seu propi algorisme públic de xifrar $E_U$ i el seu algorisme privat de desxifrar $D_U$.
En aquesta xarxa tots els missatges que van d'un usuari $A$ a un usuari $B$ han de ser enviats sota el format $(E_B(m),A)$. L'adreça $A$ s'envia de manera que el receptor $B$ sàpiga qui és l'emissor.
El receptor $B$, després de llegir el missatge $m$ que ha rebut, automàticament ha d'enviar $(E_A(m),B)$ a la xarxa perquè l'emissor $A$ ho llegeixi i sàpiga que el seu primer missatge havia estat rebut correctament.

**a**) Demostreu que un tercer usuari $C$ de la xarxa pot retrobar el missatge $m$ que $A$ havia enviat a $B$.

**b**) Demostreu que la xarxa tampoc no és segura si canviem el protocol inicial pel següent:

$A$ envia $E_B(E_B(M),A)$ a $B$

$B$, automàticament retorna $E_A(E_A(m),B)$ a $A$.

**2.** Un sistema públic, basat en el logaritme discret, per a distribuir claus entre usuaris d'una xarxa podria ser el següent:

**a**) És conegut de tots els usuaris un cos finit $\mathbb{F}_{p^s}$ i un element primitiu $\alpha$ d'aquest cos.

**b**) L'usuari $U$ calcula aleatòriament $1 < m_U < q^{s-1}$ i escriu en el fitxer públic la seva clau $c_U = \alpha^{m_U}$ (el valor $m_U$ el manté secret).

**c**) Quan l'usuari $A$ vol contactar amb l'usuari $B$, fa servir com a clau $(c_B)^{m_A}$.

Demostreu que el sistema anterior és segur i autèntic. I, en el cas concret $\mathbb{F}_{p^s} = \mathbb{F}_{2^{10}} = Z_2[x] \,/\, x^{10} + x^3 + 1$, l'usuari $A$, del qual coneixem la seva clau privada $m_A = 2$, vol contactar amb l'usuari $B$, del qual coneixem la seva clau $c_B = \alpha + \alpha^5 + \alpha^7$, escrita en el fitxer públic. Quina és la clau per a les seves comunicacions comunes?

**3.** Demostreu que la variació següent del protocol d'autenticació de l'ISO és insegura.

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $R_A$ | $\rightarrow$ |
| $\leftarrow$ | $R_B,Cert_B,D_B\{R_B,R_A\}$ | $\leftarrow$ |
| $\rightarrow$ | $R'_A,Cert_A,D_A\{R'_A,R_B\}$ | $\rightarrow$ |

$Cert_X = (X,E_X,D_S\{X,E_X\})$, $S$ és l'autoritat central de certificacions, $\{x\}$ vol dir una funció resum sobre $x$ i $R'_A$ és un valor aleatori diferent de $R_A$.

**4.** Demostreu que la variació següent del protocol d'autenticació de l'ISO és insegura.

| $A$ | | $B$ |
|---|---|---|
| $\rightarrow$ | $R_A$ | $\rightarrow$ |
| $\leftarrow$ | $R_B,Cert_B,D_B(R_A)$ | $\leftarrow$ |
| $\rightarrow$ | $Cert_A,D_A(R_B)$ | $\rightarrow$ |

$Cert_X = (X,E_X,D_S\{X,E_X\})$, $S$ és l'autoritat central de certificacions i $\{x\}$ vol dir una funció resum sobre $x$.

# Solucionari

**1)** Efectivament, l'usuari $C$ pot agafar $E_B(m)$, que està a la xarxa sota el format $(E_B(m),A)$, i reenviar a la xarxa $(E_B(m),C)$. La resposta de $B$ també seria automàtica i respondria amb $(E_C(m),B)$. Per tant, l'usuari $C$ només ha de desxifrar $E_C(m)$ amb la seva clau privada i retrobar $m$.

En el segon supòsit, la xarxa tampoc no és segura. En aquest cas, l'usuari $C$ pot recuperar $E_B(E_B(m),A))$ i enviar a $B$ el següent (que compleix els requisits per a ser acceptat per a ser transmès): $E_B\big(E_B(E_B(m),A),C)\big)$. L'usuari $B$ contesta automàticament amb: $E_C\big(E_C(E_B(m),A),C)\big)$. A partir d'això, l'usuari $C$ pot calcular $E_B(m)$ i enviar a la xarxa $E_B(E_B(m),C)$. Ara, la resposta automàtica de $B$ és $E_C(E_C(m),B)$ i l'usuari $C$ pot calcular $m$.

**2)** La clau que fan servir conjuntament $A$ i $B$ és $\alpha^{m_A \cdot m_B}$, que només pot ser calculada per $A$ i $B$. La seguretat i autenticitat es basa en el fet que per als altres usuaris el problema és el de calcular el logaritme discret al cos $\mathbb{F}_{p^s}$.

En el cas específic de $\mathbb{F}_{2^{10}}$ tenim que la clau comuna és $\alpha^{m_A \cdot m_B} = C_B^{m_A} = (\alpha + \alpha^5 + \alpha^7)^2 = \alpha^{428}$.

El càlcul anterior l'hem fet utilitzant el SAGE:

Primer definim l'anell de polinomis en la indeterminada $X$:

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

Després definim el cos finit $\mathbb{F}_{2^{10}}$ per mitjà del polinomi $X^{10} + X^3 + 1$ i anomenem *alpha* = $[X]$.

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

Finalment, calculem $(\alpha + \alpha^5 + \alpha^7)^2$.

```
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
```

**3)** És un cas clar d'utilització de l'*interleaving attack* o atac del jugador d'escacs.

El segon pas del protocol autentica $B$ davant de $A$. Aquí no hi ha cap problema. El problema és en el tercer pas del protocol. Un usuari $C$ podria aprofitar el segon pas de l'intercanvi entre $A$ i $B$ per a enviar $R_B$ a $A$ i esperar la seva contesta, que llavors $C$ enviaria a $B$ fent-se passar per $A$.

**4)** També és un cas clar de l'*interleaving attack* o atac del jugador d'escacs. El segon pas del protocol hauria d'autenticar $B$ davant de $A$, però no ho fa, ja que un impostor $C$ pot agafar $R_A$ del primer pas i endegar un protocol amb $B$, amb la qual cosa aconseguirà el necessari per a impersonar $B$.

El tercer pas del protocol tampoc no autentica $A$ davant de $B$, ja que el mateix impostor amb el valor $R_B$ del segon pas del protocol anterior pot endegar un altre protocol amb $A$ i aconseguir la firma $D_A(R_B)$.

# Bibliografia

**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.). Nova York: John Wiley & Sons.