Protocols criptogràfics

PID_00255019

Jordi Herrera Joancomartí
Cristina Pérez Solà

Temps mínim de dedicació recomanat: 3 hores

Universitat Oberta de Catalunya

Jordi Herrera Joancomartí

Llicenciat en Matemàtiques per la Universitat Autònoma de Barcelona i doctor per la Universitat Politècnica de Catalunya. Els seus àmbits de recerca són la criptografia, les criptomonedes i la tecnologia *blockchain*. Ha publicat nombrosos textos docents i més de cent articles de recerca en revistes i congressos nacionals i internacionals. Ha dirigit nou tesis doctorals i ha estat investigador principal de diversos projectes de recerca nacionals. Ha participat com a avaluador per a agències de recerca de diversos països europeus i també per a la Comissió Europea. Actualment és professor agregat del departament d'Enginyeria de la Informació i les Comunicacions a la Universitat Autònoma de Barcelona.

Cristina Pérez Solà

Doctora en Informàtica per la Universitat Autònoma de Barcelona i la Universitat Catòlica de Lovaina. Actualment és professora dels Estudis d'Informàtica, Multimèdia i Telecomunicacions de la Universitat Oberta de Catalunya. Els seus àmbits de recerca són les criptomonedes basades en *blockchain* i, en especial, els aspectes relacionats amb la seguretat i la privadesa d'aquestes. També està interessada en els problemes de privacitat que sorgeixen arran de l'ús de les xarxes socials i en l'adaptació de tècniques de mineria de dades a la naturalesa específica d'aquest tipus de xarxes.

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats per la professora: Helena Rifà Pous

Segona edició: febrer 2021

Av. Tibidabo, 39-43, 08035 Barcelona
Autoria: Jordi Herrera Joancomartí, Cristina Pérez Solà
Producció: FUOC

Índex

Introducció

Més enllà dels mecanismes per a xifrar i desxifrar missatges, el cert és que la criptografia permet construccions més elaborades que continuen tenint el mateix objectiu que els criptosistemes: protegir la informació. Així, ens podem trobar diferents situacions on ens calguin protocols que ens garanteixin un seguit de propietats de seguretat que els criptosistemes per si sols no poden proporcionar. És en aquest punt on intervenen els protocols criptogràfics, protocols entre dos o més usuaris que utilitzen mecanismes criptogràfics per protegir la informació.

En aquest mòdul didàctic estudiarem diversos protocols criptogràfics cada un d'ells amb un propòsit diferent. Llevat de l'esquema de compartició de secrets, els protocols descrits en aquest mòdul didàctic són protocols en els que hi intervenen dos usuaris i no es contempla l'existència de cap tercera part de confiança. Així, les operacions es realitzen, sovint de forma conjunta, entre els dos usuaris per aconseguir l'objectiu del protocol. La suposició que es fa en tot moment és que els usuaris poden actuar de forma deshonesta de manera que és important que els propis protocols incorporin els mecanismes de seguretat necessaris per tal que, en cas que una part actuï de forma maliciosa, l'altra part no se'n vegi afectada o, com a mínim, pugui detectar l'engany.

Objectius

Els materials didàctics d'aquest mòdul han de permetre assolir els objectius següents:

1. Entendre que les aplicacions de la criptografia no es restringeixen a proporcionar secret i autenticitat.
2. Conèixer els esquemes de compartició de secrets.
3. Entendre el concepte de prova de coneixement nul i la seva aplicabilitat.
4. Veure com cal desenvolupar protocols en contextos en què no hi ha confiança entre les parts que participen en el protocol.

1. El protocol de tres passos de Shamir

El protocol de tres passos de Shamir, va ser proposat per Adi Shamir, tot i que no el va publicar mai. El protocol permet establir una comunicació secreta entre dues parts sense cap intercanvi previ de claus. La base del protocol és una funció de xifratge commutativa respecte a les claus. És a dir, serà el mateix xifrar un missatge m amb una clau k_1 i el resultat tornar-lo a xifrar amb una clau k_2 que xifrar-lo primer amb la clau k_2 i el resultat xifrar-lo amb la k_1:

$$E_{k_1}(E_{k_2}(m)) = E_{k_2}(E_{k_1}(m))$$

Passem a descriure el **protocol de tres passos de Shamir** en el qual l'Alice vol fer arribar el missatge m a en Bob. Per a fer-ho, l'Alice disposarà d'una clau per a xifrar, k_A^e i una clau per a desxifrar k_A^d i el Bob també tindrà una clau per xifrar k_B^e i una per desxifrar k_B^d. Denotarem per $E_{k_A^e}(m)$ l'acció de xifrar el missatge m amb la clau de xifratge k_A^e de l'Alice. Igualment, denotarem per $D_{k_A^d}(c)$ el desxifratge del missatge c amb la clau de desxifratge k_A^d de l'Alice.

A l'esquema de la taula 1 es poden veure els diferents passos del protocol i la informació que s'intercanvien els usuaris que hi participen.

Taula 1. Esquema de tres passos de Shamir

Pas	Alice		Bob
1.	Calcula $c_1 = E_{k_A^e}(m)$	$\xrightarrow{c_1}$	
2.		$\xleftarrow{c_2}$	Calcula $c_2 = E_{k_B^e}(c_1) = E_{k_B^e}(E_{k_A^e}(m))$
3.	Calcula $c_3 = D_{k_A^d}(E_{k_B^e}(E_{k_A^e}(m))) =$ $= D_{k_A^d}(E_{k_A^e}(E_{k_B^e}(m))) = E_{k_B^e}(m)$	$\xrightarrow{c_3}$	
4.			Calcula $m = D_{k_B^d}(c_3) = D_{k_B^d}(E_{k_B^e}(m))$

Com podem veure, al final del protocol l'Alice ha fet arribar a en Bob el missatge m de manera segura, ja que en cap dels missatges que s'han intercanviat en cada un dels tres passos el missatge m no ha viatjat en clar. Així, un atacant que estigui analitzant les comunicacions entre A i B no podrà extreure cap informació d'm. Noteu, a més, que en cap moment s'ha produït un intercanvi de claus. L'Alice només coneix k_A^e i k_A^d i en Bob només k_B^e i k_B^d.

Als subapartats 1.1. i 1.2. veurem alguns criptosistemes que tenen la propietat de commutativitat de claus i quins resultats presenten quan s'utilitzen com a esquema de xifratge en el protocol de tres passos de Shamir.

1.1. El xifratge de Vernam i el protocol de tres passos de Shamir

Un dels criptosistemes que hem vist al llarg del curs és el criptosistema de Vernam, que seria el criptosistema més segur, ja que, utilitzant una bona clau, ens aporta seguretat incondicional. Recordem que el mecanisme tant de xifratge com de desxifratge d'aquest criptosistema és molt simple. Donat un missatge m expressat en bits i una clau k de la mateixa mida que el missatge també expressada en bits, la funció de xifratge consisteix a fer una XOR entre el missatge i la clau, és a dir, $E_k(m) = m \oplus k = c$. D'altra banda, per a desxifrar el missatge c simplement haurem de fer de nou una XOR amb la mateixa clau k amb la qual hem xifrat $D_k(c) = c \oplus k = m$.

Si ens hi fixem, aquest criptosistema presenta commutativitat de claus, ja que si tenim dues claus k_1 i k_2 es compleix que:

$$E_{k_1}(E_{k_2}(m)) = (m \oplus k_2) \oplus k_1 = (m \oplus k_1) \oplus k_2 = E_{k_2}(E_{k_1}(m))$$

Això passa perquè l'operació XOR és commutativa.

Així, si utilitzem el criptosistema de Vernam per al protocol de tres passos de Shamir entre A i B, tenim que les claus de xifrar i desxifrar per a cada usuari són la mateixa, és a dir, $k_A^e = k_A^d = k_A$ i $k_B^e = k_B^d = k_B$. Així mateix, en els tres intercanvis d'informació del protocol es generaran els missatges mostrats a l'esquema de la taula 2.

Taula 2. Esquema de tres passos de Shamir amb el xifratge de Vernam

Pas	Alice		Bob
1.	Calcula $c_1 = m \oplus k_A$	$\xrightarrow{c_1}$	
2.		$\xleftarrow{c_2}$	Calcula $c_2 = c_1 \oplus k_B$
3.	Calcula $c_3 = c_2 \oplus k_A = m \oplus k_B$	$\xrightarrow{c_3}$	
4.			Calcula $m = c_3 \oplus k_B$

Tot i que aparentment hem aconseguit desenvolupar el protocol correctament utilitzant un dels criptosistemes més segurs que hi ha, el problema el trobem en el fet que un atacant que està veient la comunicació en té prou a prendre nota dels tres missatges xifrats que s'intercanvien l'Alice i en Bob, ja que un cop intercepta c_1,c_2 i c_3 per a obtenir el missatge xifrat m només cal que faci una suma XOR dels tres:

$$c_1 \oplus c_2 \oplus c_3 = (m \oplus k_A) \oplus (m \oplus k_A \oplus k_B) \oplus (m \oplus k_B) = m$$

Per tant, podem concloure que a l'hora d'utilitzar un criptosistema per al protocol de tres passos de Shamir no en tindrem prou a assegurar-nos que compleixi la commutativitat de les claus, sinó que caldrà anar en compte amb la relació que tenen els missatges una vegada han estat xifrats.

Aquest fet ens fa veure que, més enllà d'aquest exemple concret, en la creació de protocols criptogràfics no només és important que cada una de les eines criptogràfiques que s'utilitza sigui segura, sinó que, a més, la seva combinació ho continuï essent, fet que, com hem vist, no sempre succeeix.

1.2. El criptosistema d'exponenciació

Un altre esquema amb commutativitat de claus el va proposar el mateix Adi Shamir. Aquest sistema es basa en l'exponenciació i la seva seguretat recau en la dificultat del càlcul del logaritme discret. És un criptosistema semblant al de l'RSA però no s'han de confondre, ja que en aquest cas les dues claus que s'utilitzen, una per a xifrar i l'altra per a dexifrar, són dues claus secretes que únicament estan en possessió d'un sol usuari.

En primer lloc, es tria un paràmetre per a l'intercanvi, un primer p gran. Totes les operacions es realitzaran al cos $\mathbb{Z}_p$. L'Alice genera les seves claus de la manera següent. Tria com a clau de xifratge k_A^e un valor aleatori i com a clau de desxifratge calcula el valor k_A^d tal que $k_A^e \cdot k_A^d = 1 \pmod{p-1}$. La funció de xifratge per a un missatge m serà $E_{k_A^e}(m) = m^{k_A^e} \pmod{p}$. La funció de desxifratge d'un missatge c serà $D_{k_A^d}(c) = c^{k_A^d} \pmod{p}$. De la mateixa manera, en Bob generarà les seves claus k_B^e i k_B^d i utilitzarà les mateixes funcions de xifratge i desxifratge. Amb aquestes condicions el protocol queda descrit a l'esquema de la taula 3.

Taula 3. Esquema de tres passos de Shamir amb el xifratge d'exponenciació

Pas	Alice		Bob
1.	Calcula $c_1 = m^{k_A^e} \pmod{p}$	$\xrightarrow{c_1}$	
2.		$\xleftarrow{c_2}$	Calcula $c_2 = (c_1)^{k_B^e} \bmod p$
3.	Calcula $c_3 = (c_2)^{k_A^d} \bmod p = m^{k_B^e} \bmod p$	$\xrightarrow{c_3}$	
4.			Calcula $m = (c_3)^{k_B^d} \bmod p$

Fixeu-vos que, en aquest cas, un atacant que intercepti els tres missatges de la comunicació, c_1,c_2 i c_3 no podrà obtenir cap informació sobre el missatge transmès, ja que les claus per a xifrar només les coneixen A i B.

Exemple de protocol de tres passos de Shamir amb el xifratge d'exponenciació

En aquest exemple suposarem que els dos usuaris treballen amb el paràmetre $p = 131$. A més, l'usuari A disposarà de la clau de xifratge $k_A^e = 21$ i de la clau de desxifratge $k_A^d = (k_A^e)^{-1} \pmod{p-1} = 31$. D'altra banda, l'usuari B també tindrà el seu parell de claus. La de xifratge serà $k_B^e = 27$ i la de desxifratge $k_B^d = (k_B^e)^{-1} \pmod{p-1} = 53$.

Amb aquests paràmetres, l'usuari A vol enviar en secret el missatge $m = 15$ a B, i per a fer-ho els passos del protocol seran els que es mostren a la taula 4.

Taula 4. Esquema d'enviament del missatge

Pas	Alice		Bob
1.	$c_1 = 15^{21} \pmod{131} = 125$	$\xrightarrow{125}$	
2.		$\xleftarrow{27}$	$c_2 = (125)^{27} \bmod 131 = 27$
3.	$c_3 = (27)^{31} \bmod 131 = 129$	$\xrightarrow{129}$	
4.			$m = (129)^{53} \bmod 131 = 15$

2. Esquemes de compartició de secrets

Quan volem emmagatzemar un secret cal tenir en compte que hi ha situacions en què el secret no pot ser guardat de manera centralitzada perquè hi ha el perill que aquesta centralització esdevingui un punt feble en la seguretat. En aquestes situacions el concepte de centralització pot tenir diferents vessants. Per exemple, imaginem-nos que tenim el codi d'obertura d'una caixa forta però no volem que estigui custodiat per una sola persona perquè té el perill que aquesta persona pugui marxar amb tots els diners. Voldríem poder distribuir aquest codi de manera que més d'una persona fos necessària per a l'obertura de la caixa forta.

Una altra situació, potser més quotidiana, és l'emmagatzemament de contrasenyes. Si emmagatzemem la contrasenya en un únic lloc, si aquest lloc sofrís algun incident perdríem la clau. Podríem solucionar aquest problema guardant la mateixa clau en diferents llocs, però això implicaria una reducció de la seguretat, ja que les probabilitats que algú la trobi són més grans. Tal com hem fet amb la caixa forta, podríem repartir el valor de la clau en diferents fragments. Fixeu-vos que en aquest cas, la possibilitat de poder recuperar la clau només amb alguns fragments (i no necessàriament amb tots) és important, ja que si els necessitem tots per a recuperar-la, tornem a estar en el punt de partida: si un dels llocs en què hi ha un dels fragments de la clau sofrís algun incident no podríem recuperar la clau i també l'hauríem perduda.

Per a resoldre aquests tipus de situacions tenim els esquemes de compartició de secrets. Aquests esquemes van ser proposats de manera independent l'any 1979 per Adi Shamir i George Blakley.

Un **esquema de compartició de secrets llindar** (m,n) (en anglès (m,n)-*threshold secret sharing scheme*) és un esquema que permet distribuir un secret en n fragments diferents de manera que si s'ajunten m o més fragments es pot recuperar el secret, però no és possible obtenir cap informació del secret si es disposa de menys d'm fragments.

Si assumim l'escenari en el qual volem repartir un secret S entre diferents usuaris, un esquema de compartició de secrets llindar (m,n) està format per n usuaris, $u_1, \cdots, u_n$. Cada usuari té el seu fragment s_i del secret S. A més, cal que es compleixin les propietats següents:

1) Per a tot $i = 1, \cdots, n$, l'usuari u_i només coneix el seu fragment s_i.

2) El secret S es pot obtenir a partir d'm valors s_i diferents per a qualsevol $i \in \{1, \cdots, n\}$.

3) Donats $m-1$ valors diferents, s_i, no es pot obtenir cap informació d'S.

2.1. Esquema de compartició de secrets polinòmic

Un esquema per a compartir secrets llindar (m,n) força utilitzat és el proposat per Adi Shamir basat en la interpolació polinòmica.

Suposem que volem compartir el secret S utilitzant un esquema de llindar (m,n). Això vol dir que hem de crear n fragments i que en tenim prou tenint-ne m per a reconstruir-lo, però que menys d'aquesta quantitat no ens serà suficient. En aquest tipus d'esquema hi haurà un superusuari, el gestor, que serà l'encarregat de, partint del secret S, generar els n fragments. Com a paràmetres públics tindrem un nombre primer p tal que $p > n$ i $p > S$.

Per a construir els fragments, el gestor construeix un polinomi $a(x)$ de grau $m-1$ amb coeficients a_i a $\mathbb{Z}_p$, és a dir, $a(x) \in \mathbb{Z}_p[x]$. Aquest polinomi tindrà com a coeficients valors aleatoris, llevat del terme independent, que serà exactament el valor secret S, és a dir, podem expressar el polinomi de la manera següent:

$$a(x) = S + a_1 x + a_2 x^2 + \cdots + a_{m-1} x^{m-1} \pmod{p}$$

La generació dels fragments es realitzarà de la manera següent: el gestor tria n valors aleatoris de $\mathbb{Z}_p$, $\{x_1, \cdots, x_n\}$, i per a cada valor calcula la seva avaluació pel polinomi, és a dir, $a(x_i) = S + a_1 x_i + a_2 x_i^2 + \cdots + a_{m-1} x_i^{m-1} \pmod{p}$; el polinomi $a(x)$ es manté en secret i només el coneix el gestor, però es pot eliminar una vegada s'han generat els fragments.

Cada participant rep com a fragment del secret el parell $\{x_i, a(x_i)\}$, és a dir, un valor x_i i la seva avaluació en el polinomi $a(x_i)$. Podrem recuperar el secret si tenim m fragments plantejant el sistema d'equacions següent.

$$\begin{aligned}
a(x_1) &= S + a_1 x_1 + a_2 x_1^2 + \cdots + a_{m-1} x_1^{m-1} \pmod{p} \\
a(x_2) &= S + a_1 x_2 + a_2 x_2^2 + \cdots + a_{m-1} x_2^{m-1} \pmod{p} \\
&\vdots \quad \vdots \\
a(x_m) &= S + a_1 x_m + a_2 x_m^2 + \cdots + a_{m-1} x_m^{m-1} \pmod{p}
\end{aligned}$$

Si ens hi fixem, en aquest sistema tenim m incògnites, corresponents als m coeficients dels polinomis $S,a_1,a_2,\cdots,a_{m-1}$ i també hi ha m equacions, per la qual cosa en resoldre'l obtindrem el valor de les incògnites i en particular la que ens interessa, que és el valor secret S. A més, aquest sistema sempre tindrà solució, un solució única, perquè hi intervé el determinant de Vandermonde.

Exemple de protocol de compartició de secrets llindar (3,5)

Suposem que tenim cinc usuaris u_1,u_2,u_3,u_4,u_5 que volen repartir-se el valor secret $S = 673$. Per a fer-ho utilitzaran l'esquema de compartició de secrets polinòmic de Shamir i treballaran amb el primer $p = 1931$. Descriurem els dos processos d'un esquema de compartició de secrets: la generació dels fragments i la recuperació del secret.

Generació dels fragments

Donat que amb tres usuaris n'hi haurà prou per recuperar el secret, el gestor construirà un polinomi de grau 2 amb coeficients a $\mathbb{Z}_{1931}$ on el terme independent sigui el secret $S = 673$. Així, el gestor triarà dos valors aleatoris per a crear el polinomi, per exemple, 436 i 806 i construirà el polinomi $a(x) = 673 + 806x + 436x^2$. Amb aquest polinomi, procedirà a construir els fragments de cada usuari avaluant el polinomi en una component x per a cada participant. Si suposem que u_1 té la component $x = 1$ i u_2 la component $x_2 = 2$, i així per a cada usuari, tindrem les avaluacions següents:

$$\begin{aligned}
a(1) &= 673 + 806 \cdot 1 + 436 \cdot 1^2 = 1915 \pmod{1931}\\
a(2) &= 673 + 806 \cdot 2 + 436 \cdot 2^2 = 167 \pmod{1931}\\
a(3) &= 673 + 806 \cdot 3 + 436 \cdot 3^2 = 1222 \pmod{1931}\\
a(4) &= 673 + 806 \cdot 4 + 436 \cdot 4^2 = 1218 \pmod{1931}\\
a(5) &= 673 + 806 \cdot 5 + 436 \cdot 5^2 = 155 \pmod{1931}
\end{aligned}$$

Per tant l'usuari u_1 rebrà el fragment [1,1915], l'usuari u_2 el fragment [2,167], l'usuari u_3 el fragment [3,1222], l'usuari u_4 el fragment [4,1218] i l'usuari u_5 el fragment [5,155].

Recuperació del secret

Suposem ara que tres dels cinc usuaris es reuneixen per a recuperar el secret. Suposem que són els usuaris u_1, u_4 i u_5 (però hauríem pogut triar qualssevol dels altres tres). Els fragments d'aquests usuaris són [1,1915], [4,1218] i [5,155], respectivament. Com que aquests valors són punts del polinomi utilitzat per a generar els fragments, podem plantejar el sistema d'equacions següent:

$$\begin{aligned}
S + a_1 \cdot 1 + a_2 \cdot 1^2 &= 1915 \pmod{1931}\\
S + a_1 \cdot 4 + a_2 \cdot 4^2 &= 1218 \pmod{1931}\\
S + a_1 \cdot 5 + a_2 \cdot 5^2 &= 155 \pmod{1931}
\end{aligned}$$

Com que només ens interessa resoldre el sistema per la variable S, que és el secret, podem aplicar el mètode de Kramer. Així doncs, obtenim el següent:

$$\frac{\begin{vmatrix} 1915 & 1 & 1 \\ 1218 & 4 & 16 \\ 155 & 5 & 25 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \end{vmatrix}} = \frac{352}{12} = 352 \cdot 161 = 673 \pmod{1931}$$

2.2. Problemàtiques dels esquemes de compartició de secrets

A la pràctica, els esquemes de compartició de secrets tenen un seguit de restriccions que fan que el seu ús requereixi construccions molt més complexes que les que hem presentat aquí.

El primer punt que cal tenir en compte en un esquema de compartició de secrets és la confiança que es diposita en el gestor del sistema. Fixeu-vos que el gestor és el que s'encarrega de generar el polinomi que permetrà crear els fragments de cada participant i, per a fer-ho, necessita el valor del secret. Per tant, cal que el gestor sigui una tercera part de confiança o bé que aquest procés es realitzi amb les garanties de seguretat necessàries.

D'altra banda, també ens podríem preguntar què passaria si un dels participants donés un valor aleatori en comptes del seu fragment. El cert és que el secret no es recuperaria i, encara més, no sabríem qui ha estat el culpable. I encara pitjor, l'atacant podria utilitzar el secret recuperat erròniament, el seu fragment fals i el seu fragment correcte per a recuperar el secret real sense l'ajut de la resta de participants, mentre que la resta de participants continuarien sense poder recuperar el secret.

Per tal de resoldre aquests problemes hi ha esquemes de compartició de secrets més elaborats que resisteixen aquest tipus d'atacs, però la seva discussió s'escapa de l'abast d'aquesta assignatura.

Fixeu-vos, però, que totes aquestes problemàtiques no s'apliquen quan només volem utilitzar l'esquema de compartició de secrets per a emmagatzemar una contrasenya de manera distribuïda i segura, ja que en aquest cas, tant el gestor com els usuaris que proporcionaran els fragments són tots el mateix.

3. Signatures cegues

Un altre dels protocols interessants en criptografia són les signatures cegues, un protocol que s'utilitza per a signar digitalment missatges de manera especial.

En un **protocol de signatura cega** (en anglès, *blind signature*) l'usuari A aconsegueix la signatura d'un missatge m per part de l'usuari B sense que B sàpiga quin missatge ha signat.

El concepte de signatura cega el va proposar David Chaum l'any 1982 per a fer-lo servir en esquemes de pagament anònim.

Per a imaginar-nos com funciona un protocol de signatura cega és interessant utilitzar una analogia en termes de papers i signatures manuscrites. La idea és que l'usuari A té el document que ha de signar B i en comptes de proporcionar-li directament (fet que faria que B en pogués veure el contingut), A el posa dins d'un sobre. La peculiaritat d'aquest sobre és que està fet de paper carbó, és a dir, si escrivim alguna cosa fora del sobre es calcarà a l'interior. En particular, si B fa una signatura manuscrita fora del sobre, quan posteriorment traiem el document de dins del sobre tindrem el document signat per les propietats de calca del sobre de paper carbó. A més, B no haurà pogut veure el contingut del document que ha signat.

Com amb altres protocols criptogràfics, no és òbvia quina utilitat pot tenir que un usuari pugui signar un document sense saber el que signa. Tot i això, al llarg d'aquest apartat veurem en quines situacions tenen aplicabilitat les signatures cegues.

3.1. Signatura cega amb RSA

Donat que un protocol de signatura cega pretén la signatura digital d'un missatge, aquest protocol sempre inclourà un esquema de signatura digital en concret. A continuació, veurem un protocol de signatura cega basat en RSA. Aquest mateix protocol el va idear David Chaum quan va proposar el concepte de signatura cega.

Denotarem per m el missatge que A vol tenir signat per B. B signarà digitalment els seus missatges amb un esquema RSA. Per fer-ho utilitzarà la seva clau privada d. La clau pública corresponent a aquesta clau privada la denotarem per (e,n). El protocol es desenvoluparà en els passos següents:

1) A tria un valor aleatori r a $\mathbb{Z}_n$, que sigui invertible, i el xifra amb la clau pública de B, és a dir, calcula $t = r^e \pmod{n}$. El valor t és el valor que utilitzarà per a tapar el missatge m que B ha de signar. Per a fer-ho, A calcularà $m' = m \cdot t \pmod{n}$ i enviarà el valor m' a B.

2) En rebre m', B simplement realitzarà la signatura sobre aquest valor de manera estàndard, utilitzant la seva clau privada d. Així obtindrà $s' = (m')^d \pmod{n}$ i enviarà el valor s' a A.

3) A destaparà la signatura feta per B simplement dividint la signatura que ha rebut de B, s', pel valor aleatori r generat en el primer pas, $s = \frac{s'}{r}$.

Vegem l'esquema del protocol gràficament a la taula 5.

Taula 5. Esquema gràfic del protocol

Pas	Alice		Bob
1.	Tria $r \in_R \mathbb{Z}_n$ Calcula $t = r^e \bmod n$ Tapat : calcula $m' = m \cdot t \bmod n$	$\xrightarrow{m'}$	
2.		$\xleftarrow{s'}$	Signa el valor m' calculant: $s' = (m')^d \bmod n$
3.	Obté la signatura de m calculant $s = \frac{s'}{r}$ (destapat)		

Fixeu-vos que el valor s destapat per A en el pas 3 efectivament correspon a la signatura del missatge original m. Això és així perquè:

$$s = \frac{s'}{r} = \frac{(m')^d}{r} = \frac{(m \cdot t)^d}{r} = \frac{m^d \cdot t^d}{r} = \frac{m^d \cdot (r^e)^d}{r} = \frac{m^d \cdot r}{r} = m^d \pmod{n}$$

Exemple de protocol de signatura cega amb RSA

Suposem que l'usuari A vol que l'usuari B li signi el missatge $m = 15$. L'usuari B utilitza per a realitzar signatures digitals el criptosistema RSA. La clau pública de B és $(e,n) = (19,551)$ i la corresponent clau privada $d = 451$. Amb aquests paràmetres, el protocol de signatura cega entre A i B serà el que es mostra a la taula 6.

Taula 6. Esquema gràfic del protocol

Pas	Alice		Bob
1.	Tria $25 \in_R \mathbb{Z}_{551}$ Calcula $t = 25^{19} = 310 \bmod 551$ Tapat : calcula $m' = 15 \cdot 310 = 242 \bmod 551$	$\xrightarrow{m'=242}$	
2.		$\xleftarrow{s'=14}$	Signa el valor $m' = 242$ calculant: $s' = 242^{451} = 14 \bmod 551$
3.	Obté la signatura de m calculant $s = \frac{14}{25} = 14 \cdot 529 = 243 \pmod{551}$		

Fixeu-vos que el valor $s = 243$ és efectivament la signatura del missatge original $m = 15$, ja que $s = 15^{451} = 243 \bmod 551$.

3.2. Aplicacions de les signatures cegues

Hi ha múltiples escenaris en què les signatures cegues són interessants d'utilitzar i la majoria d'ells tenen a veure amb la protecció de l'anonimat. Vegem com es poden fer servir a l'escenari següent per tenir identificadors anònims.

Suposeu un sistema amb una autoritat central que té identificats els seus usuaris. Per tal de permetre utilitzar els recursos del sistema de manera anònima, els usuaris poden obtenir uns pseudònims per part de l'autoritat central. Aquests pseudònims estan signats digitalment per l'autoritat central una vegada ha comprovat que l'usuari té suficients privilegis per a utilitzar els corresponents recursos. La signatura de l'autoritat central sobre el pseudònim ha de permetre la seva validació per una tercera part quan l'usuari vol utilitzar el pseudònim davant d'algun dels recursos del sistema on es vol autenticar.

Amb aquest escenari, si l'autoritat central signa els pseudònims dels usuaris de manera estàndard, els usuaris obtindran anonimat davant dels tercers amb qui s'autentifiquin utilitzant el pseudònim. Ara bé, no aconseguiran anonimat davant de l'autoritat central, ja que l'autoritat central, quan signa el pseudònim, sap la identitat real de l'usuari i, per tant, la correspondència entre la identitat real i el pseudònim, amb la qual cosa es trenca l'anonimat.

Una opció per a resoldre aquest problema és que l'autoritat central signi el pseudònim però utilitzant un protocol de signatura cega. D'aquesta manera, l'autoritat central donaria validesa al pseudònim però no sabria a qui correspon el pseudònim.

3.3. Protecció contra abusos en les signatures cegues

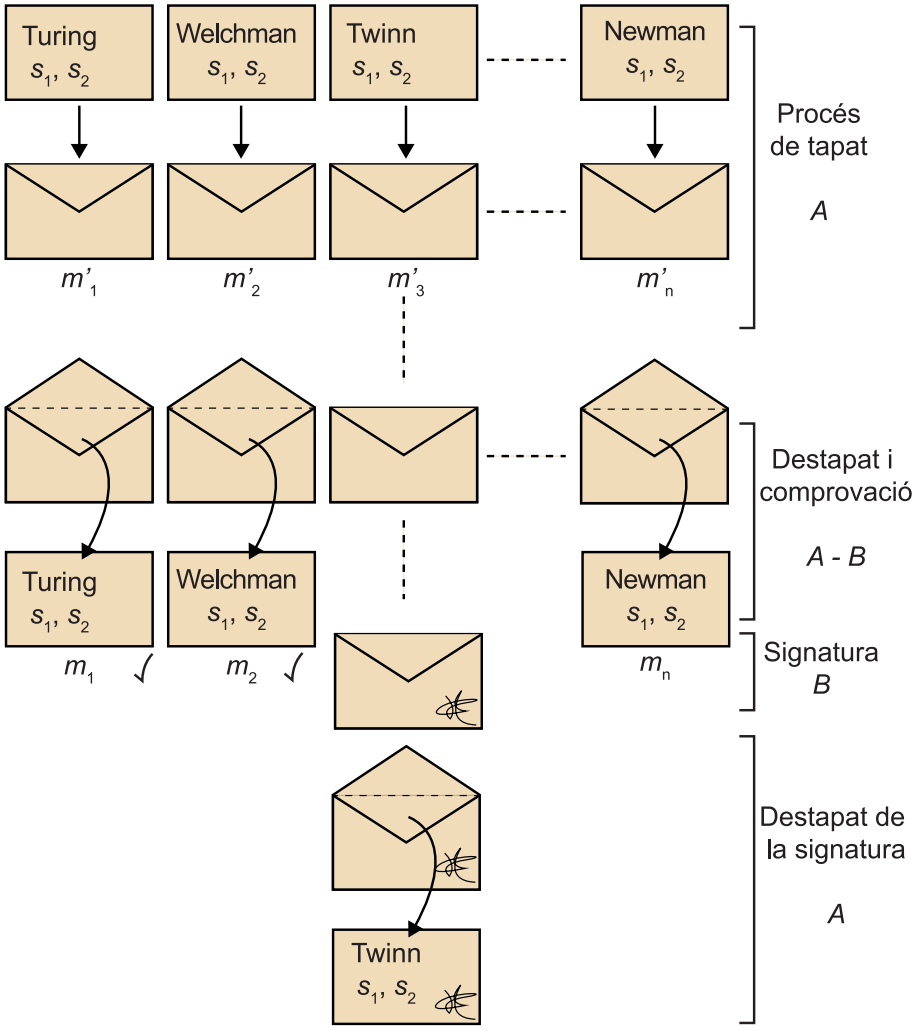
Malgrat que les signatures cegues són interessants d'utilitzar en alguns escenaris, el cert és que la possibilitat que un usuari signi un valor sense saber exactament el que signa pot comportar també alguns problemes de seguretat. Per exemple, com ja hem estudiat anteriorment, la realització d'una signatura digital és equivalent al desxifratge d'un missatge. Per tant, un usuari A que hagués interceptat un missatge xifrat c dirigit a B podria utilitzar un protocol de signatura cega per a tapar c, fer-lo signar per B i, d'aquesta manera, obtenir el missatge desxifrat.

Així mateix, en escenaris més complexos, el contingut del que signa B pot ser rellevant i A pot voler-lo modificar per treure'n profit. Per exemple, imaginemnos el cas descrit al subapartat 3.2. en el qual l'usuari A vol obtenir un pseudònim per a autenticar-se. B només li proporcionarà el pseudònim en funció dels privilegis que tingui A en el sistema. A més, el pseudònim ha d'incloure aquesta informació per tal que A el pugui fer servir. Un possible atac d'A seria presentar un pseudònim amb unes atribucions diferents de les que el sistema li permet. Si B ha de realitzar una signatura cega, no podrà verificar aquestes condicions i podria arribar a signar condicions no desitjades.

Per a evitar aquest tipus d'accions hi ha diferents estratègiques. La primera és utilitzar una clau específica per a les signatures cegues. És a dir, una clau pública que incorporés la mateixa semàntica de l'autorització. Per exemple, qualsevol pseudònim signat amb la clau pública que tingués el valor concret $PK_{CA}^{S_1,S_2}$ només serviria per a autenticar-se davant dels recursos S_1 i S_2. Per a autenticar-se davant del recurs S_3, per exemple, caldria tenir el pseudònim signat amb la clau pública $PK_{CA}^{S_3}$. A més, aquestes claus públiques de signatures cegues només es farien servir en aquest context i mai s'utilitzarien per a xifrar missatges, de manera que l'atac per al desxifratge no seria possible.

Tot i que aquesta protecció que associa una semàntica a una clau és factible, a la pràctica pot comportar la gestió d'un volum de claus molt gran. Per a evitar-ho una altra opció és utilitzar el procediment de "remenar i triar" per a assegurar que B no signa res fraudulent. El procés funciona tal com es mostra a la figura 1.

Figura 1. Esquema del mecanisme de remenar i triar



L'usuari A, en comptes d'enviar un únic valor tapat m' a B, calcula múltiples valors tapats $m'_1, m'_2, \cdots, m'_n$. És important que cada valor s'hagi tapat amb un element diferent, és a dir, per a cada m'_i tindrem un valor t_i diferent, seguint

la nomenclatura que hem utilitzat en l'esquema de signatura cega. Cada un d'aquests valors tapats $m'_1, m'_2, \cdots, m'_n$ conté certa informació que B ha de poder validar abans de signar i una altra informació diferent per a cada un dels valors m'_i. Per exemple, en el cas dels pseudònims per a l'autenticació, la part que ha de poder validar B és la part que indica a quins recursos permet accedir el pseudònim. Aquesta part ha de ser la mateixa per a tots els valors. La part que és diferent per a cada valor m'_i és la que indicarà el pseudònim que A farà servir.

Una vegada A ha enviat els n valors tapats $m'_1, m'_2, \cdots, m'_n$ a B, B demanarà a A que destapi $n-1$ valors, és a dir, A proporcionarà els corresponents t_i per a $n-1$ valors que B haurà triat aleatòriament. Una vegada destapats, B podrà comprovar que la part que ha de validar coincideix en tots i cada un dels $n-1$ valors que ha destapat. Si és així, assumirà que el valor que li resta per a destapar (el qual no pot destapar perquè A manté el corresponent t_i per a fer-ho) també compleix les condicions estipulades. Per tant, pot procedir a signar cegament aquest valor.

Fixeu-vos que cada un dels valors que ha destapat A pot contenir un pseudònim diferent, de manera que B no sap quin pseudònim hi haurà en el valor que ha signat. D'altra banda, la probabilitat que A pugui enganyar B per a aconseguir que signi algun contingut que no vulgui es pot fer tant petita com es vulgui, ja que el seu valor és d'$\frac{1}{n}$.

4. Proves de coneixement nul

Un dels usos de la criptografia és la gestió de la informació secreta. En ocasions la gestió d'aquesta informació pot comportar que ens interessi convèncer algú que coneixem certa informació secreta però sense revelar aquesta informació. Dit d'una altra manera, ens interessa un mecanisme per a poder demostrar que sabem un secret sense revelar-lo. Aquest concepte, batejat amb el nom de proves de coneixement nul, el van introduir S. Goldwasser, S. Micali i C. Rackoff l'any 1985.

Una **prova de coneixement nul** (en anglès, *zero-knowledge proof*) és un protocol entre dos usuaris pel qual l'usuari que actua de provador, P, permet demostrar que coneix un cert valor secret s davant d'un usuari verificador, V, sense proporcionar el valor s. Al final del protocol, V estarà convençut que P coneix el valor s i alhora V no haurà obtingut cap informació sobre aquest valor.

Per tant, una prova de coneixement nul ha de complir les propietats següents:

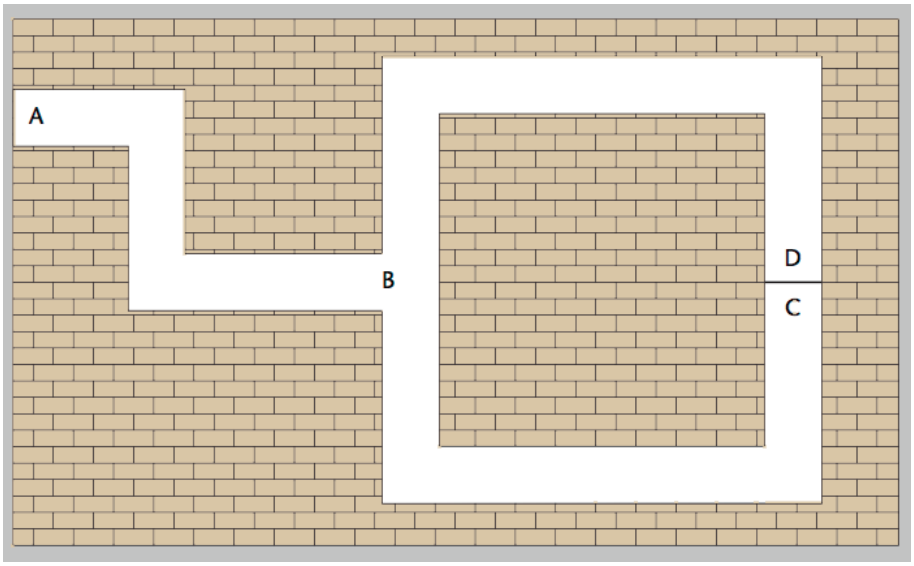
1) Correcció: si el provador coneix el valor s ha de poder convèncer al verificador que efectivament el coneix.

2) Robustesa: la probabilitat que el provador enganyi el verificador ha de ser molt petita. És a dir, si el provador no coneix el valor secret s, la probabilitat que la prova de coneixement nul s'executi correctament és molt petita.

3) Coneixement nul: un cop realitzada la prova de coneixement nul, el verificador no té cap informació sobre el valor secret s que el provador coneix. En particular, el verificador no pot provar davant una tercera persona, ni per mitjà d'una prova de coneixement nul, que coneix el secret.

Un exemple gràfic per a entendre la mecànica de la majoria de les proves de coneixement nul és el que van proposar J. J. Quisquater i L. Guillou. Com veiem a la figura 2, es van basar en una cova amb una entrada amb un únic camí. En un punt de la cova, el camí es bifurca i fa una volta fins a tornar-se a unir amb l'altra part del camí. Ara bé, el camí està tancat per una porta que s'obre per mitjà d'una paraula secreta.

Figura 2. Gràfic de la cova de l'exemple



En Pep (*P*) coneix aquesta paraula secreta i vol convèncer en Vicenç (*V*) que la coneix, però no vol donar-li aquesta clau. Per a fer-ho executen la prova següent de coneixement nul:

1) En Vicenç és queda a l'entrada de la cova (punt A del gràfic) mentre que en Pep entra dins i tria un dels dos camins fins a arribar a la porta. Per tant, pot estar al punt C o bé al punt D depenent de la tria que hagi fet.

2) Un cop en Pep hagi arribat davant de la porta, en Vicenç avançarà fins a la bifurcació (punt B). Des d'allí tria un dels dos camins, el de la dreta o el de l'esquerra i li fa un crit a en Pep perquè surti pel camí que ha triat.

3) Com que en Pep coneix la clau que obra la porta no tindrà cap problema per a sortir pel costat que en Vicenç li ha demanat.

Comprovem ara si amb aquest exemple es compleixen les tres propietats que hem indicat anteriorment.

Exemple rebuscat

Com veurem més endavant, aquest exemple il·lustra com funciona una prova de coneixement nul, però, òbviament, pel nostre propòsit n'hi hauria prou a fer entrar en Pep per la dreta i fer-lo sortir per l'esquerra.

1) Si en Pep coneix la clau sempre podrà sortir pel costat que en Vicenç li demana i, per tant, podrà demostrar que té el coneixement que vol provar.

Si tornéssim a fer l'experiment i el repetíssim tantes vegades com volguéssim, en Pep sortiria sempre pel costat que en Vicenç li demanés, ja que coneix la clau que obre la porta i, per tant, no tindria cap problema.

2) Si en Pep no coneix la clau de la porta no hauria de poder convèncer en Vicenç que sí que la coneix. Fixeu-vos que si no coneix la clau, en fer la prova en Pep tindria una probabilitat d'1/2 d'encertar el camí que li demanarà més tard en Vicenç, ja que si en endinsar-se en la cova l'encerta, després podrà sortir pel mateix costat i no li caldrà utilitzar la clau de la porta que de fet no sap. Ara bé, si el procés el repetim un altre cop, en Pep només té 1/4 de probabilitat d'enganyar-lo. Ja es veu que si repetim la prova n vegades la probabilitat que

en Pep enganyi en Vicenç és d'$1/2^n$. Així doncs, si en Vicenç vol estar segur amb probabilitat 0,999023 que en Pep sap la paraula secreta que obre la porta només cal que realitzin la prova deu vegades.

3) Un cop en Vicenç hagi pogut validar que en Pep coneix la clau, en Vicenç no haurà obtingut cap informació de la clau i tampoc pot utilitzar informació de la prova que ha fet amb en Pep, tot i que l'hagi repetida deu vegades, per a poder demostrar davant d'un tercer que coneix la clau.

En general, les proves de coneixement nul funcionen d'aquesta manera, és a dir, són iteratives de manera que en cada iteració hi ha una probabilitat del 50% d'encertar. A més, aquests tipus de protocols utilitzen la tècnica anomenada *challenge and response*, en què el verificador dona al provador una informació que ell ha generat aleatòriament per tal que el provador la completi utilitzant el secret que coneix. Aquesta tècnica també s'anomena sovint *cut and choose*, ja que fa referència al típic protocol de repartir un pastís entre dues persones, en el que una fa les parts (talla) i l'altre escull.

4.1. Prova del coneixement del logaritme discret

A continuació veurem un exemple concret d'una prova de coneixement nul aplicada al coneixement del logaritme discret d'un valor, prova que va ser proposada per D. Chaum, J. Evertse i J. Van de Graaf el 1987. Ja hem comentat anteriorment que el càlcul del logaritme discret té una complexitat elevada, és a dir, donats uns valors y,g i p és difícil trobar per a quin valor x es compleix que $y = g^x \pmod p$. Per tant, aquesta prova de coneixement nul permet al provador demostrar que coneix el valor x que compleix l'equació $y = g^x \pmod p$ sense necessitat de revelar aquest valor.

El protocol funciona de la manera següent. En primer lloc, el protocol estableix tres paràmetres públics (p,g,y), en què p és un nombre primer gran, y és un nombre enter tal que $y < p$ i g és un generador del grup multiplicatiu Z_p. El provador ha de demostrar al verificador que coneix el valor x que satisfà l'equació $y = g^x \pmod p$. Per a fer-ho el protocol fa els passos de la taula 7.

Taula 7. Esquema gràfic del protocol

Pas	Provador (P)		Verificador (V)
1.	Tria $r \in_R \mathbb{Z}_p \setminus \{0,1\}$ Calcula $c = g^r \pmod p$	$\xrightarrow{c}$	
2.		$\xleftarrow{b}$	Tria un bit aleatori $b \in_R \{0,1\}$
3.	Calcula $h = r + b \cdot x \pmod{p-1}$	$\xrightarrow{h}$	
4.			Verifica que: $c \cdot y^b = g^h \pmod p$

El protocol consisteix a repetir n vegades els quatre passos descrits anteriorment. Comprovem que es compleixen les propietats d'una prova de coneixement nul.

1) Correcció: en el cas que P conegui el valor x sempre podrà calcular el valor h en el tercer pas del protocol de manera que la validació que farà V en el pas quatre serà correcta.

2) Robustesa: per a verificar la propietat de robustesa, analitzarem com s'ho faria P per a intentar fer creure a V que coneix x sense realment saber-ho. Per a fer-ho, P ha de poder calcular el valor h del pas 3 sense conèixer r. Fixeu-vos que en cas que V li enviï a P el valor $b = 0$ en el pas 2, P calcularà $h = r + b \cdot x \pmod{p–1} = r \pmod{p–1}$ sense necessitat de saber x i aquest valor serà correcte i, per tant, superarà la validació del pas 4. Ara bé, si V tria $b = 1$ en el pas 2, aleshores P no pot calcular el valor h correcte (li falta el coneixement de x) de manera que no podrà concloure el protocol correctament. Fixeu-vos que la probabilitat que això passi és de 1/2, ja que és la probabilitat que té V en el pas 2 de triar un 0 o un 1. Per tant, si repetim el protocol n vegades, la probabilitat que P enganyi V és d'$\frac{1}{2^n}$.

Arribats a aquest punt, podríem pensar que no sembla que tingui sentit que V en el pas 2 enviï un 0, ja que en aquest cas, P no necessita conèixer x. Per tant, podríem concloure, erròniament, que en el pas 2, V podria enviar sempre un 1, forçant P a conèixer x. Ara bé, aquesta estratègia no és correcta. Fixem-nos que si V sempre tria $b = 1$, P pot generar un valor r en el pas 1, però en comptes d'enviar $c = g^r \pmod{p}$ a V pot enviar $c' = \frac{g^r}{y} \pmod{p}$. Aleshores, en el pas 3, P envia r en comptes de $r + x$, però la verificació del pas 4 serà correcta perquè $c' \cdot y = \frac{g^r}{y} \cdot y = g^r = g^h$.

Per tant, fixeu-vos que si P no sap si li arribarà un 0 o un 1 en el pas 2 (i P no coneix el secret) no sap quina estratègia d'engany ha de seguir en el pas 1, és a dir, si ha d'enviar $c = g^r \pmod{p}$ o bé $c' = \frac{g^r}{y} \pmod{p}$. Per tant, d'una manera o d'una altra té una probabilitat d'1/2 d'enganyar.

3) Coneixement nul: el protocol també té la propietat de coneixement nul, ja que després d'executar-lo, V només coneix el valor g^r rebut en el pas 1, valor que no té cap relació amb el secret x. A més, el valor h rebut en el pas 3 tant pot correspondre al valor r com al valor $r+x$ i ambdós es presenten com a valors aleatoris per a V i, per tant, no poden proporcionar cap informació d'r.

Exemple de protocol de prova de coneixement nul del logaritme discret

Suposem que els paràmetres del protocol són $p = 89$ i $g = 3$. El provador P coneix el logaritme discret de $y = 14 \pmod{89}$, que és $x = 9$. Suposarem que V tria el valor $b = 1$ en el pas 2. D'aquesta manera el protocol tindria els valors de la taula 8.

Taula 8. Esquema gràfic del protocol

Pas	Provador (P)		Verificador (V)
1.	Tria $r = 20 \in_R \mathbb{Z}_{89} \setminus \{0,1\}$ Calcula $c = 3^{20} = 73 \pmod{89}$	$\xrightarrow{c=73}$	
2.		$\xleftarrow{b=1}$	Tria un bit aleatori $b = 1$
3.	Calcula $h = 20 + 1 \cdot 9 = 29 \pmod{88}$	$\xrightarrow{h=29}$	
4.			Verifica que: $c \cdot y^b = 73 \cdot 14^1 = 43 \pmod{89}$ $g^h \pmod{p} = 3^{29} = 43 \pmod{89}$

4.2. Aplicacions de les proves de coneixement nul

Les proves de coneixement nul tenen diferents camps d'aplicació. El primer camp es troba en els sistemes d'autenticació. El tradicional mètode de contrasenya comença a ser insuficient per a certes aplicacions, ja que tant si aquesta, de manera incorrecta, es guarda en clar com si es guarda com a imatge d'una funció hash en algun moment l'usuari ha d'introduir-la en clar i és aleshores que pot ser interceptada. A més, la utilització de la mateixa informació per a diferents processos d'autenticació pot donar lloc a atacs de repetició en què un atacant utilitza informació d'una autenticació anterior per a autenticar-se posteriorment. Utilitzant proves de coneixement nul, donat que el verificador no pot obtenir cap informació sobre el valor secret que té el provador, la possibilitat d'atacs de repetició desapareix.

Un altre camp en què les proves de coneixement nul són importants és en la verificació de paràmetres en protocols criptogràfics més complexos. Per exemple, en protocols de votació electrònica, els votants han de proporcionar certs paràmetres per a poder realitzar la votació. Alguns d'aquests paràmetres han de ser secrets, per a preservar l'anonimat del vot, però a la vegada han de tenir certes característiques per tal que el protocol funcioni correctament. Les proves de coneixement nul s'utilitzen per a provar que un usuari coneix un paràmetre del protocol amb certes característiques sense haver de revelar cap informació del paràmetre en qüestió.

5. Protocol de transferència inconscient

Els protocols de transferència inconscient permeten que un usuari emissor "transmeti" informació a un altre usuari receptor de manera que al final de la transmissió, l'usuari receptor només obté una part de la informació "transmesa". A més, la particularitat d'aquests esquemes és que, d'una banda, l'emissor no sap quina informació finalment ha rebut el receptor i, d'altra banda, el receptor no obté cap informació de la informació que no li ha arribat.

El concepte de protocol de transferència inconscient va ser presentat per M. O. Rabin l'any 1981. La proposta de Rabin era un protocol en el qual l'emissor té un secret i, amb probabilitat 1/2 l'envia al receptor. Al final del protocol, el receptor pot tenir el secret o no tenir-lo (amb probabilitat 1/2) però l'emissor no pot saber si l'ha rebut o no. Aquest seria el que es coneix com a protocol de transferència inconscient 0-1. En aquest apartat, però, ens centrarem en els protocols de transferència inconscient 1-2.

0-1 OT

Aquest protocol es basa en la dificultat de calcular arrels quadrades modulars i amb la relació d'aquesta operació i la factorització d'enters.

En un **protocol de transferència inconscient 1-2** (en anglès, 1-2 *oblivious transfer*) l'usuari A té dos secrets s_0 i s_1. Al final de l'execució del protocol entre A i B, B obté un dels dos secrets amb igual probabilitat. A més, A no pot saber quin secret ha rebut B i B no obtindrà cap informació sobre el secret que no ha rebut.

A continuació veurem un exemple concret d'aquest tipus de protocol.

5.1. Protocol d'Even, Goldreich i Lempel

Aquest protocol va ser proposat el 1985 pels criptogràfs Shimon Even, Oded Goldreich i Abraham Lempel. La proposta utilitza el criptosistema RSA per tal de xifrar els valors secrets que hi intervenen. El protocol permet l'intercanvi inconscient 1-2 dels secrets s_0 i s_1 entre l'usuari A, que és qui coneix els dos valors, i l'usuari B, que és qui en rebrà un dels dos. El funcionament del protocol, així com les accions i els missatges que s'intercanvien en el protocol es mostra gràficament a l'esquema de la taula 9.

Taula 9. Esquema gràfic del protocol

Pas	Alice		Bob
1.	Secrets s_0 i s_1.		
	Generació de la clau:		
	$n = p \cdot q$ amb p,q primers		
	$e \cdot d = 1 \pmod{\phi(n)}$		
	Genera $x_0,x_1 \in_R \mathbb{Z}_n$	$\xrightarrow{(e,n,x_0,x_1)}$	
2.			Tria un bit aleatori b
			Genera $k \in_R \mathbb{Z}_n$
		$\xleftarrow{v}$	Calcula $v = x_b + k^e \pmod{n}$
3.	Calcula $k_0 = (v - x_0)^d \pmod{n}$		
	Calcula $k_1 = (v - x_1)^d \pmod{n}$		
	Calcula $s'_0 = s_0 + k_0 \pmod{n}$		
	Calcula $s'_1 = s_1 + k_1 \pmod{n}$	$\xrightarrow{(s'_0,s'_1)}$	
4.			Coneixent el valor b,
			calcula $s_b = s'_b - k \pmod{n}$

Al pas 1, A genera el parell de claus pública-privada i dos valors aleatoris. Envia la clau pública i els valors aleatoris a B. Al pas 2, B triarà un dels dos valors aleatoris i l'amagarà utilitzant una clau. Fixeu-vos que la clau que utilitza per a amagar el valor aleatori triat és k^e. Com que k ha estat triat aleatòriament, k^e també és un valor aleatori i el resultat v també aparenta un valor aleatori per a A, ja que no coneix ni k ni k^e. Al pas 3, A calcula dues claus k_0 i k_1 que utilitzarà per a amagar els secrets s_0 i s_1 de la transferència inconscient obtenint els valors s'_0 i s'_1. El punt important està en com es calculen aquestes claus k_0 i k_1. Si ens fixem, per exemple, en k_0, en el cas que B hagi triat el valor x_0 al pas 2 tenim que:

$$k_0 = (v - x_0)^d = (x_0 + k^e - x_0)^d = (k^e)^d = k$$

És a dir que A haurà amagat el valor s_0 amb la clau k que B ha triat al pas 2. Per tant, B podrà descobrir el valor s_0 en el pas 3 simplement restant-ne el valor k. En el cas que B hagi triat x_1 en comptes de x_0 en el pas 2, podrà recuperar el secret s_1, ja que k_1 serà igual a k. Fixeu-vos que en aquest segon cas (B ha triat x_1) B no pot fer res amb el valor s'_0 per a intentar esbrinar s_0, perquè no té cap informació de k_0, ja que:

$$k_0 = (v - x_0)^d = (x_1 + k^e - x_0)^d \neq k$$

Exemple de protocol de transferència inconscient 1-2

Suposem que l'Alice vol fer una transferència inconscient 1-2 a en Bob dels secrets $s_0 = 22$ i $s_1 = 34$. El protocol tindria l'estructura de la taula 10.

Taula 10. Esquema gràfic del protocol

Pas	Alice		Bob
1.	Secrets $s_0 = 22$ i $s_1 = 34$. Generació de la clau: $n = 19 \cdot 29 = 551$ $e = 19$ i $d = 451$ Genera $x_0 = 130, x_1 = 525$ aleatoris.	$\xrightarrow{(e=19, n=551, x_0=130, x_1=525)}$	
2.		$\xleftarrow{v}$	Tria un bit aleatori $b = 0$ Genera $k = 174$ Calcula $v = 130 + 174^{19} = 304 \pmod{551}$
3.	Calcula $k_0 = (304 - 130)^{451} = 174 \pmod{551}$ Calcula $k_1 = (304 - 525)^{451} = 26 \pmod{551}$ Calcula $s'_0 = 22 + 174 = 196 \pmod{551}$ Calcula $s'_1 = 34 + 26 = 60 \pmod{551}$	$\xrightarrow{(s'_0=196, s'_1=60)}$	
4.			Coneixent el valor $b = 0$, calcula $s_0 = 196 - 174 = 22 \pmod{551}$

5.2. Aplicacions de la transferència inconscient

Com ja hem dit abans, aquest protocol per si sol pot no tenir gaire interès, però és la base d'altres esquemes com la signatura de contractes. Suposem l'escenari en el qual dos usuaris A i B volen signar digitalment un contracte però cap d'ells vol enviar primer la signatura a l'altre per no estar en desavantatge. Vegem com es pot aplicar la transferència inconscient 1-2 per a solucionar aquesta situació.

L'usuari A descompon la seva signatura en $2n$ trossos de m bits cada un, que denotarem per $\{a_i, 1 \leq i \leq 2n\}$. L'usuari B fa el mateix amb la seva signatura i obté els trossos $\{b_i, 1 \leq i \leq 2n\}$. Així doncs, vegeu el procés:

1) A divideix els seus $2n$ trossos de la seva signatura en n parells, per exemple, (a_{2j-1}, a_{2j}) per $j = 1, \cdots, n$ i envia a B un element de cada parell utilitzant una transferència inconscient 1-2, per la qual cosa B rep a_{2j-1} o bé a_{2j}, per $j = 1, \cdots, n$, però A no sap quin dels elements ha rebut B (recordem que cada element del parell té un 50% de probabilitat de ser enviat).

2) Simultàniament al pas **1**, B fa exactament el mateix amb els seus $2n$ trossos de la seva signatura: els divideix en parells i envia un element de cada parell a A utilitzant una transferència inconscient 1-2.

3) A i B s'envien l'un a l'altre el primer bit de tots els seus trossos a_i i b_i per $i = 1, \cdots, 2n$, després el segon bit, i així fins al final. Si A vol enganyar B, només té la probabilitat d'$1/2^n$ d'aconseguir-ho, ja que B ja té n dels $2n$ nombres secrets del pas **1** i A no sap quins són. Simètricament, es pot aplicar el mateix si B vol enganyar A.

Fixeu-vos que d'aquesta manera, A i B poden intercanviar la signatura del contracte i cap d'ells no està mai en avantatge de més d'un bit.

6. Protocol multipart segur

En algunes aplicacions ens pot interessar que un conjunt d'usuaris realitzi un cert càlcul, de manera que, tot i que cada usuari aporta una entrada per a la realització del càlcul, al final del procés cada usuari només podrà obtenir el resultat del càlcul, però no podrà obtenir els valors d'entrada d'altres usuaris. Aquests tipus de protocols es coneixen com a protocols multipart segurs.

En un **protocol multipart segur** (en anglès, *multiparty computation*) un conjunt d'n participants cooperen per a avaluar el valor d'una funció f sobre un conjunt de valors $(v_1, \cdots, v_n)$ aportats pels participants. Com a sortida del protocol, cada usuari u_i obté l'avaluació de la funció $f(v_1, \cdots, v_n)$ però no obté cap informació sobre el contingut dels valors v_j per a $j \in [1,n]$ i $j \neq i$.

Com en la majoria de protocols criptogràfics, una solució simple en aquest escenari és la utilització d'una tercera part de confiança en la qual tothom confia. Aquesta tercera part és la que pot realitzar l'avaluació de la funció f i, com que tothom hi confia, tothom està segur que un cop cada usuari li ha lliurat el seu tros d'informació, no el mostrarà a cap altra part.

Justament, el que proporcionen els protocols multipart segurs és un mecanisme per a poder prescindir de la tercera part de confiança. Als subapartats 6.1. i 6.2. veurem dos exemples concrets de protocol multipart segur.

6.1. El problema del milionari

Un exemple d'un protocol de càlcul segur a múltiples bandes, en aquest cas a dues bandes, és el protocol proposat per C. Yao l'any 1982, conegut com el problema del milionari. En aquest escenari, dos milionaris, A i B, volen saber qui és el més ric però no volen revelar el valor de la seva fortuna. És a dir, la funció que volem avaluar de manera segura és una comparació de la mida de dos valors en què cada un dels dos participants aporta un valor.

Per tal de simplificar una mica el problema (de fet, seria equivalent a fitar la fortuna que tenen els participants) transformarem aquest problema en un problema equivalent que consistirà en el següent: l'Alice i en Bob volen saber qui és més gran sense dir quina edat tenen. Suposarem que tots dos són honestos i que utilitzen les seves edats reals.

Suposarem que l'Alice té x anys i en Bob y, i cap dels dos no en té més de 100, és a dir, $1 \leq x,y \leq 100$. Per a realitzar aquest protocol utilitzarem un criptosistema de clau pública. Així, tant A com B tindran cada un d'ells un parell de claus pública i privada que seran (E_A,D_A) i (E_B,D_B), respectivament. D'altra banda, també assumirem que tots dos usuaris coneixen la clau pública de l'altre participant. A més, A i B també es posen d'acord en la mida màxima que tindran dos dels valors utilitzats en el protocol, t_a i t_b. Així poden assegurar que els valors p_a i p_b triats al pas 6 són més petits que aquests dos valors.

El protocol funciona tal com es descriu a l'esquema de la taula 11.

Taula 11. Protocol del milionari

Pas	Alice		Bob
1.	Tria $t_a \in_R \mathbb{Z}$		Tria $t_b \in_R \mathbb{Z}$
2.	Calcula: $k_a = E_B(t_a)$ $K_a = k_a - x$		Calcula: $k_b = E_A(t_b)$ $K_b = k_b - y$
3.		$\xrightarrow{K_a}$	
4.		$\xleftarrow{K_b}$	
5.	Calcula: $f_i = D_A(K_b + i)$ per $1 \leq i \leq 100$		Calcula: $f'_i = D_B(K_a + i)$ per $1 \leq i \leq 100$
6.	Tria $p_a < t_b$ Calcula: $g_i = f_i \pmod{p_a}$ per $1 \leq i \leq 100$ assegurant que $\lvert g_i - g_j \rvert \geq 2$ per a $i \neq j, 1 \leq i,j \leq 100$ Crea la seqüència: $G = \{g_1, \cdots, g_x, g_{x+1} + 1, g_{x+2} + 1, \cdots, g_{100} + 1, p_a\}$		Tria $p_b < t_a$ Calcula: $g'_i = f'_i \pmod{p_b}$ per $1 \leq i \leq 100$ assegurant que $\lvert g'_i - g'_j \rvert \geq 2$ per a $i \neq j, 1 \leq i,j \leq 100$ Crea la seqüència: $G' = \{g'_1, \cdots, g'_y, g'_{y+1} + 1, g'_{y+2} + 1, \cdots, g'_{100} + 1, p_b\}$
7.		$\xrightarrow{G}$	
8.		$\xleftarrow{G'}$	
9.	Comprova: Si $G'_x = t_a \pmod{p_b}$, aleshores $y \geq x$, sinó $y < x$		Comprova: Si $G_y = t_b \pmod{p_a}$, aleshores $x \geq y$, sinó $x < y$

Com es pot veure en el protocol, la idea és que tant A com B creen una seqüència de valors, en aquest cas 100, que és el màxim de l'edat dels participants. La particularitat d'aquestes seqüències és que, per exemple, prenent la seqüència G que genera l'usuari A, per a índex inferiors o iguals a l'índex que determina l'edat de l'usuari A, el valors són congruents amb el valor aleatori que ha triat B, si l'edat de A és més gran que la d'B.

Les conclusions sobre qui té més edat que cada usuari obté en el pas 9 són correctes. Per exemple, el raonament respecte a la comprovació de l'usuari B seria la següent: si A té més edat que B, és a dir, $x \geq y$, aleshores el valor de la posició y de la seqüència que A envia a B en el pas 7 és $G_y = g_y$. Per tant, $G_y = f_y \pmod{p_a}$. Com que $f_y = D_A(K_b + y) = D_A(k_b - y + y) = D_A(k_b)$ i $k_b = E_A(t_b)$ ens queda que $f_y = D_A(E_A(t_b)) = t_b$ i, per tant, $G_y = t_b \pmod{p_a}$.

D'altra banda, si $x < y$, aleshores el valor G_y verifica que:

$$G_y = g_y + 1 \neq g_y = f_y = t_b \pmod{p_a}$$

6.2. El problema del milionari socialista

En aquest segon protocol, A i B tenen cada un la seva fortuna, representada pels valors x i y, respectivament, però en comptes de saber qui és més ric, el que volen saber és si la seva fortuna és igual o no. L'execució d'aquest protocol és més elaborada que la de l'exercici anterior, ja que el nombre de missatges que s'intercanvien és més elevat, a causa que el protocol utilitza com a subprotocol, en diverses etapes, el protocol d'intercanvi de claus de Diffie i Hellman.

El protocol defineix dos paràmetres generals: un nombre primer p i un valor $h \in \mathbb{Z}_p$ tal que $h \neq 1$. El valor de p ha de ser més gran que la fortuna tant de l'Alice com d'en Bob, és a dir, $x < p$ i $y < p$. El funcionament del protocol es mostra a l'esquema de la taula 12.

Taula 12. Protocol del milionari socialista

Pas	Alice		Bob
1.	Tria $a_1, a_2 \in_R \mathbb{Z}_p$		Tria $b_1, b_2 \in_R \mathbb{Z}_p$
2.	Calcula: $h^{a_1} \pmod p$ $h^{a_2} \pmod p$		Calcula: $h^{b_1} \pmod p$ $h^{b_2} \pmod p$
3		$\xrightarrow{(h^{a_1}, h^{a_2})}$	
4.		$\xleftarrow{(h^{b_1}, h^{b_2})}$	
5.	Verifica que: $h^{b_1} \neq 1 \pmod p$ $h^{b_2} \neq 1 \pmod p$		Verifica que: $h^{a_1} \neq 1 \pmod p$ $h^{a_2} \neq 1 \pmod p$
6.	Calcula: $g = (h^{b_1})^{a_1} \pmod p$ $f = (h^{b_2})^{a_2} \pmod p$		Calcula: $g = (h^{a_1})^{b_1} \pmod p$ $f = (h^{a_1})^{b_2} \pmod p$
7.	Tria $r \in_R \mathbb{Z}_p$		Tria $s \in_R \mathbb{Z}_p$
8.	Calcula: $P_a = f^r \pmod p$ $Q_a = h^r g^x \pmod p$		Calcula: $P_b = f^s \pmod p$ $Q_b = h^s g^y \pmod p$
9		$\xrightarrow{(P_a, Q_a)}$	
10.		$\xleftarrow{(P_b, Q_b)}$	
11.	Comprova que: $P_a \neq P_b \pmod p$ $Q_a \neq Q_b \pmod p$		Comprova que: $P_a \neq P_b \pmod p$ $Q_a \neq Q_b \pmod p$
12.	Calcula: $(Q_a Q_b^{-1})^{a_2}$		Calcula: $(Q_a Q_b^{-1})^{b_2}$
13.		$\xrightarrow{(Q_a Q_b^{-1})^{a_2}}$	
14.		$\xleftarrow{(Q_a Q_b^{-1})^{b_2}}$	
15.	Calcula: $c = ((Q_a Q_b^{-1})^{b_2})^{a_2} \pmod p$		Calcula: $c = ((Q_a Q_b^{-1})^{b_2})^{a_2} \pmod p$
16.	Comprova que: $c = P_a P_b^{-1} \pmod p$		Comprova que: $c = P_a P_b^{-1} \pmod p$

Com es pot veure en el protocol, els primers sis passos corresponen a un intercanvi de claus de Diffie-Hellman que permeten que A i B comparteixin dos valors f i g. La verificació final podria ser errònia en cas que els valors a_1, a_2, b_1, b_2 fossin 0, per aquest motiu es realitza la validació del pas 5.

Al final del protocol, en cas que la darrera comprovació del valor sigui correcta, tant A com B poden estar convençuts que els dos tenen la mateixa fortuna, ja que:

$$P_a P_b^{-1} = f^r (f^s)^{-1} = f^{r-s} = h^{a_2 b_2 (r-s)} \pmod{p}$$

D'altra banda, però:

$$\begin{aligned} c &= ((Q_a Q_b^{-1})^{b_2})^{a_2} \\ &= ((h^x g^x)(h^s g^s)^{-1})^{a_2 b_2} \\ &= (h^{(r-s)} g^{(x-y)})^{a_2 b_2} \\ &= (h^{(r-s)} (h^{a_1 b_1})^{(x-y)})^{a_2 b_2} \\ &= h^{a_2 b_2 (r-s)} h^{a_1 b_1 a_2 b_2 (x-y)} \\ &= P_a P_b^{-1} (h^{a_1 b_1 a_2 b_2 (x-y)}) \pmod{p} \end{aligned}$$

Així mateix, com que els valors a_1, a_2, b_1, b_2 han estat triats aleatòriament per A i B, l'única possibilitat que $c = P_a P_b^{-1} \pmod{p}$ és en el cas que $x = y$, és a dir, que A i B tinguin la mateixa fortuna.

En cas que la comprovació no sigui correcta voldrà dir que les fortunes no són iguals, però cap dels dos sabrà qui té una fortuna més gran.

Resum

En aquest mòdul didàctic hem estudiat diferents protocols criptogràfics que permeten assolir diferents objectius, tots ells relacionats amb la seguretat de la informació. En primer lloc, hem vist que dos usuaris es poden intercanviar un missatge de manera secreta sense haver intercanviat prèviament cap clau, utilitzant el protocol de tres passos de Shamir. També hem vist com funcionen els esquemes de compartició de secrets que permeten que un secret es descompongui en diferents fragments, de manera que amb la unió d'un nombre fixat de fragments es pot recuperar el secret però amb menys sigui impossible.

En segon lloc, hem estudiat també altres protocols en què la seva aplicació directa pot no ser del tot òbvia. Un exemple en són les signatures cegues, en què el signatari no coneix el missatge que està signant i aquest fet es pot aprofitar per a protocols d'autenticació anònima. Un altre exemple estudiat són les proves de coneixement nul, en què un usuari pot demostrar davant d'un altre que coneix un secret sense revelar-ne informació. També hem vist com funciona un protocol de transferència inconscient en què la comunicació entre dos usuaris es fa de manera probabilística; així l'emissor envia dos missatges i el receptor només en rep un. Ara bé, ni l'emissor sap quin missatge ha rebut el receptor ni el receptor pot triar quin dels dos rebre, ja que té una probabilitat del 50% de rebre'n un dels dos.

Finalment, hem analitzat dos exemples de protocols multipart segurs. En els protocols multipart segurs, n usuaris volen obtenir l'avaluació d'una funció $f(x_1,x_2,\cdots,x_n)$ proporcionant cada un d'ells una entrada de la funció x_i. El punt clau del protocol és que tots els usuaris han d'obtenir el resultat de l'avaluació de la funció, però no poden obtenir cap informació sobre les entrades que han proporcionat la resta d'usuaris. Els exemples estudiats han mostrat protocols en què intervenien dos usuaris, un d'ells permet avaluar la funció "menor o igual" i l'altre permet avaluar la funció d'igualtat.

Exercicis d'autoavaluació

1. Reproduïu el protocol de tres passos de Shamir per tal que A enviï el missatge $m = 20$ a B utilitzant el criptosistema d'exponenciació en què la clau de xifratge d'A val $k_A^e = 19$, la clau de desxifratge d'A val $k_A^d = 79$ i les corresponents claus de xifratge i desxifratge de B valen $k_B^e = 13$ i $k_B^d = 77$, respectivament. Suposarem, també, que utilitzen $p = 101$.

2. Utilitzeu un esquema de compartició de secrets de Shamir per a generar els fragments d'un sistema amb llindar (3,5) per a compartir el nombre secret 11. Preneu com a primer $p = 13$.

3. En un esquema de compartició de secrets polinòmic de Shamir amb llindar (3,6) els participants reben els fragments (58,137),(11,48),(50,99),(80,50),(104,33),(39,114). Tenint en compte que treballen a $\mathbb{Z}_{149}$, recupereu el secret.

4. En un esquema de compartició de secrets polinòmic de Shamir amb llindar $m = 3$, construït sobre $\mathbb{Z}_{13}$, l'usuari A té l'avaluació del polinomi per a $x = 1$, l'usuari B, $x = 2$, i l'usuari C, $x = 3$. Els tres usuaris es reuneixen per a poder trobar la clau del sistema. Tots tres usuaris fan trampa; els usuaris A i C sumen 2 a l'avaluació del polinomi en el seu punt, però la clau que recuperen és la correcta. Quina és la trampa que ha fet l'usuari B?

5. En un sistema d'autentificació anònima, l'usuari A té accés a un recurs S. Per a poder-hi accedir, l'autoritat de certificació CA li generarà una credencial que consistirà en la signatura d'un missatge m que contindrà una clau pública generada per l'usuari A i l'identificador del recurs S. Per tal que la credencial sigui anònima, la CA realitzarà una signatura cega de manera que no tindrà manera de saber quina és la clau pública que certifica i per tant quan A accedeixi al recurs la CA no podrà saber-ho. Ara bé, per a assegurar-se que A no accedeix a un recurs diferent, la signatura cega la realitzaran amb un protocol de triar i remenar. Així, A prepararà cinc missatges diferents m_i tals que $m_i = (Pk_i||S)$, en què PK_i serà una clau pública de la qual A coneix la corresponent clau privada i el símbol $||$ denota la concatenació. Expliciteu tots els missatges que s'intercanviaran A i la CA en aquest protocol. Suposeu que treballen a $\mathbb{Z}_{899}$ i que el criptosistema de clau pública que fem servir és l'RSA. Suposeu que el valor $S = 5$ i que el parell de claus (pública i privada) de la CA són $PK_{CA} = 19$,$SK_{CA} = 619$. Per a simplificar, no cal indicar les corresponents claus privades de les cinc claus públiques triades.

6. Voleu realitzar una prova de coneixement nul per a demostrar que coneixeu el logaritme discret amb base 7 de $y = 94$ a $\mathbb{Z}_{97}$, és a dir, el valor x tal que $y = 7^x \pmod{97}$. El problema és que realment no coneixeu el valor x però voleu enganyar un usuari fent una prova de coneixement nul i que es pugui convèncer que sí que el coneixeu. Afortunadament per a vosaltres, el generador pseudoaleatori que fa servir el provador té una vulnerabilitat i vosaltres podeu saber el valor dels bits que genera en el pas 2 del protocol. L'usuari en qüestió vol fer una prova de coneixement nul que li asseguri que coneixeu el valor amb probabilitat superior a 0,75. Desenvolupeu tot el protocol de prova de coneixement nul assumint que el generador aleatori de V produeix els bits següents: 010011100... Doneu el detall de les operacions i valors que s'intercanvien els usuaris en cada pas del protocol.

7. L'Alice i en Bob han estat de sort i els ha tocat la loteria, que reparteix com a màxim 5 milions. L'Alice ha tingut més sort que en Bob i li han tocat 4 milions, mentre que a en Bob n'hi han tocat 2. Com que cap d'ells vol dir quina quantitat li ha tocat, decideixen saber qui és més ric utilitzant el protocol del milionari. Desenvolupeu el protocol per tal que els dos puguin saber qui ha guanyat més diners sense saber quants diners li han tocat a l'altre. Suposarem que utilitzem com a sistema de clau pública l'RSA i el parell de claus pública-privada d'A val $[(e_A = 2573,n_A 5911),(d_A = 197,n_A = 5911)]$, mentre que el parell de B val $[(e = 3109,n_B = 5191),(d_B = 1795,n_B = 5191)]$.

Solucionari

1. Amb aquests paràmetres, l'usuari A enviarà en secret el missatge $m = 20$ a B amb el protocol de la taula 13.

Taula 13. Protocol de l'exercici 1

Pas	Alice		Bob
1.	$c_1 = 20^{19} \pmod{101} = 30$	$\xrightarrow{30}$	
2.		$\xleftarrow{77}$	$c_2 = (30)^{13} \bmod 101 = 77$
3.	$c_3 = (77)^{79} \bmod 101 = 9$	$\xrightarrow{9}$	
4.			$m = (9)^{77} \bmod 101 = 20$

2. El polinomi per a generar els fragments estarà compost pel terme independent 11, tindrà com a grau $m - 1 = 3 - 1 = 2$ i com a coeficients podem triar aleatòriament, per exemple, els nombres $x_1 = 8$ i $x_2 = 7$. D'aquesta manera el polinomi ens queda determinat per $a(x) = 7x^2 + 8x + 11 \bmod 13$.

Per a generar els fragments prenem cinc valors qualssevol més petits que p i calculem les seves imatges pel polinomi $a(x)$. Prenent com a valors $\{1,2,3,4,5\}$ tindrem:

$$a(1) = 7 + 8 + 11 = 0 \pmod{13}$$

$$a(2) = 28 + 16 + 11 = 3 \pmod{13}$$

$$a(3) = 63 + 24 + 11 = 7 \pmod{13}$$

$$a(4) = 112 + 32 + 11 = 12 \pmod{13}$$

$$a(5) = 175 + 40 + 11 = 5 \pmod{13}$$

Per tant, els fragments dels participants són: (1,0),(2,3),(3,7),(4,12),(5,5).

3. Donat que tenim un sistema de compartició llindar amb $m = 3$ podem triar, d'entre els diferents fragments, (58,137),(11,48),(50,99),(80,50),(104,33),(39,114), qualsevol conjunt de 3 punts per a recuperar el secret. Per exemple, si triem (50,99),(80,50),(39,114) podem plantejar el sistema d'equacions següent:

$$S + a_1 \cdot 50 + a_2 \cdot 50^2 = 99 \pmod{149}$$

$$S + a_1 \cdot 80 + a_2 \cdot 80^2 = 50 \pmod{149}$$

$$S + a_1 \cdot 39 + a_2 \cdot 39^2 = 114 \pmod{149}$$

Com que només ens interessa resoldre el sistema per la variable S, que és el secret, podem aplicar el mètode de Kramer i obtenim:

$$\frac{\begin{vmatrix} 99 & 50 & 116 \\ 50 & 80 & 142 \\ 114 & 39 & 31 \end{vmatrix}}{\begin{vmatrix} 1 & 50 & 116 \\ 1 & 80 & 142 \\ 1 & 39 & 31 \end{vmatrix}} = \frac{36}{120} = 36 \cdot 113 = 45 \pmod{149}$$

4. El gestor ha utilitzat el polinomi $a(x) = S + a_1x + a_2x^2$, en què S és la clau del sistema.

Quan els tres usuaris es reuneixen poden escriure el sistema següent:

$$\begin{aligned} f_1 + 2 &= S + a_1 + a_2 \\ f_2 + x &= S + 2a_1 + 4a_2 \\ f_3 + 2 &= S + 3a_1 + 9a_2 \end{aligned}$$

En aquest sistema f_1,f_2,f_3 són els fragments respectius de A, B i C, i x és la trampa que ha fet l'usuari B.

La solució per a la incògnita S en aquest sistema és la mateixa que pel sistema en el qual cap participant fa trampa, ja que l'enunciat indica que han recuperat el mateix secret, per tant:

$$\begin{aligned} f_1 &= S + a_1 + a_2 \\ f_2 &= S + 2a_1 + 4a_2 \\ f_3 &= S + 3a_1 + 9a_2 \end{aligned}$$

Així doncs, podem plantejar la igualtat següent:

$$\frac{\begin{vmatrix} f_1+2 & 1 & 1 \\ f_2+x & 2 & 4 \\ f_3+2 & 3 & 9 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{vmatrix}} = \frac{\begin{vmatrix} j & 1 & 1 \\ m & 2 & 4 \\ s & 3 & 9 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{vmatrix}}$$

A més, si realitzem les operacions dels determinants, ens queda que $6x = 3$ i, finalment, $x = 3 \cdot 6^{-1} = 7$, sempre treballant a $\mathbb{Z}_{13}$. Per tant, la trampa que ha fet l'usuari B ha estat sumar 7 al seu fragment.

5. La solució de l'exercici es mostra a la taula 14.

6. Com que A vol fer creure a B que coneix el logaritme discret amb un probabilitat de 0,75 això vol dir que caldrà executar de manera satisfactòria tres vegades del protocol. Com que A coneix el generador pseudoaleatori, sap que en la primera execució del protocol, al pas 2, V triarà $b = 0$, en la segona execució del protocol triarà $b = 1$ i en la tercera execució triarà $b = 0$. Per tant, per tal d'enredar V:

- En el primer protocol, en el pas 1 triarem qualsevol valor aleatori, per exemple, $r = 45$, que serà el mateix valor que retornarem al pas 3, $h = 45$. La validació del pas 4 feta per V serà correcta.

- A la segona execució del protocol, al pas 1 triarem, per exemple, $r = 5$. Però enviarem a V el valor $c = \frac{g^r}{y} \bmod p = \frac{7^5}{94} \bmod 97 = 56$. Aleshores en el pas 3 enviarem $h = r = 5$ i la validació que farà V en el pas 4 també serà correcta, ja que $c \cdot y^b = 56 \cdot 94^1 = 26 \bmod 97$ i $g^h = 7^5 = 26 \bmod 97$.

- A la tercera execució, aplicarà la mateixa estratègia que a la primera.

7. La solució d'aquest exercici es mostra a la taula 15.

Taula 14. Solució de l'exercici 5

Pas	Alice		Bob
1.	Genera 5 claus públiques: (3,8,10,11,14) Prepara els cinc missatges per a signar: $m_1 = (3\|\|5) = (35), m_2 = (85)$ $m_3 = (105), m_4 = (115), m_5 = (145)$ Genera els cinc valors per a tapar-los: (5,8,15,23,4) $t_1 = 5^{19} = 718 \pmod{899}$ $t_2 = 8^{19} = 872 \pmod{899}$ $t_3 = 15^{19} = 773 \pmod{899}$ $t_4 = 23^{19} = 895 \pmod{899}$ $t_5 = 4^{19} = 473 \pmod{899}$ Tapa els cinc missatges: $m_1 \xrightarrow{t_1} m'_1 = 35 \cdot 718 = 857 \pmod{899}$ $m_2 \xrightarrow{t_1} m'_2 = 85 \cdot 872 = 402 \pmod{899}$ $m_3 \xrightarrow{t_1} m'_3 = 105 \cdot 773 = 255 \pmod{899}$ $m_4 \xrightarrow{t_1} m'_4 = 115 \cdot 895 = 439 \pmod{899}$ $m_5 \xrightarrow{t_1} m'_5 = 145 \cdot 473 = 261 \pmod{899}$	$\xrightarrow{(m'_1,m'_2,m'_3,m'_4,m'_5)}$	
2.		$\xleftarrow{i=2}$	Tria $i = 2 \in_R \mathbb{Z}_5$
3.	Envia els valors t_j menys el t_2 seleccionat.	$\xrightarrow{(t_1,t_3,t_4,t_5)}$	
4.			Destapa els valors i comprova que el servei sol·licitat sigui $S = 5$ (últim dígit) $m'_1 \xrightarrow{t_1} m_1 = 35$ $m'_3 \xrightarrow{t_3} m_3 = 105$ $m'_4 \xrightarrow{t_4} m_4 = 115$ $m'_5 \xrightarrow{t_5} m_5 = 145$ Signa el valor no destapat:
		$\xleftarrow{s'_2=371}$	$s'_2 = 402^{619} = 371 \pmod{899}$
5.	Destapa el valor per a obtenir la signatura de m_2: $s'_2 \xrightarrow{t_2} s_2 = \frac{371}{8} = 833$ Com es pot veure coincideix $85^{619} = 833 \pmod{899}$		

Taula 15. Solució de l'exercici 7

Pas	Alice ($x = 4$)		Bob ($y = 2$)
1.	Tria $t_a = 1349 \in_R \mathbb{Z}$		Tria $t_b = 1547 \in_R \mathbb{Z}$
2.	Calcula:		Calcula:
	$k_a = E_B(t_a) = 1465$		$k_b = E_A(t_b) = 2212$
	$K_a = k_a - x = 1461$		$K_b = k_b - y = 2210$
3.		$\xrightarrow{K_a=1461}$	
4.		$\xleftarrow{K_b=2210}$	
5.	Calcula:		Calcula:
	$f_1 = D_A(K_b + 1) = 4217$		$f'_1 = D_B(K_a + 1) = 1177$
	$f_2 = D_A(K_b + 2 = 1547)$		$f'_2 = D_B(K_a + 2) = 573$
	$f_3 = D_A(K_b + 3) = 3556$		$f'_3 = D_B(K_a + 3) = 4426$
	$f_4 = D_A(K_b + 4) = 3569$		$f'_4 = D_B(K_a + 4) = 69$
	$f_5 = D_A(K_b + 5) = 884$		$f'_5 = D_B(K_a + 5) = 674$
6.	Tria $p_a = 239 < t_b$		Tria $p_b = 739 < t_a$
	Calcula:		Calcula:
	$g_1 = f_1 \pmod{p_a} = 154$		$g'_1 = f'_1 \pmod{p_b} = 438$
	$g_2 = f_2 \pmod{p_a} = 113$		$g'_2 = f'_2 \pmod{p_b} = 573$
	$g_3 = f_3 \pmod{p_a} = 210$		$g'_3 = f'_3 \pmod{p_b} = 731$
	$g_4 = f_4 \pmod{p_a} = 223$		$g'_4 = f'_4 \pmod{p_b} = 69$
	$g_5 = f_5 \pmod{p_a} = 167$		$g'_5 = f'_5 \pmod{p_b} = 674$
	Crea la seqüència:		Crea la seqüència:
	$G = \{154,113,210,223,168,239\}$		$G' = \{438,573,732,70,675,739\}$
7.		$\xrightarrow{G}$	
8.		$\xleftarrow{G'}$	
9.	Com que:		Com que:
	$G'_4 = 70 \neq 1349 = t_a \pmod{739}$, aleshores		$G_2 = 113 = 1547 = t_b \pmod{239}$, aleshores
	$y < x$ i, per tant, A té més diners que B.		$x \geq y$ i, per tant, A té tants o més diners que B.

Glossari

compromís de bit *m* Eina criptogràfica que permet a un usuari A comprometre's a un valor b davant d'algú B. B no podrà saber el valor b a què A s'ha compromès, però, posteriorment, A podrà obrir el compromís per a mostrar b a B.

esquema de compartició de secrets *m* Esquema pel qual es pot dividir un secret en diferents fragments de manera que amb un subconjunt de fragments es pot recuperar el secret.

esquema de compartició de secrets d'interpolació polinomial *m* Esquema de compartició de secrets en el qual els fragments del secret són punts del pla i el secret s'obté fent una interpolació polinòmica d'un cert nombre de punts.

esquema de compartició de secrets de llindar (m,n) *m* Esquema de compartició de secrets en el qual el secret es divideix en n fragments i se'n necessiten m per tal de recuperar-lo. A més, conèixer $m-1$ fragments no dona cap informació del secret.

protocol de tres passos de Shamir *m* Protocol que permet a dos usuaris intercanviar informació xifrada sense necessitat de compartir cap clau.

protocol multipart segur *m* Protocol que permet a un conjunt d'n participants cooperar per a avaluar el valor d'una funció f sobre un conjunt de valors $(v_1,\cdots,v_n)$ aportats pels participants. Com a sortida del protocol, cada usuari u_i obté l'avaluació de la funció $f(v_1,\cdots,v_n)$ però no obté cap informació sobre el contingut dels valors v_j per a $j \in [1,n]$ i $j \neq i$.

prova de coneixement nul *f* Protocol criptogràfic pel qual un participant P demostra a un altre V el coneixement d'alguna informació sense revelar-ne cap detall.

provador *m* Part encarregada de demostrar que coneix una informació en una prova de coneixement nul.

signatura cega *f* Protocol criptogràfic en el qual un usuari signa un missatge sense que pugui conèixer el contingut del missatge signat.

transferència inconscient *f* Protocol criptogràfic pel qual A emet missatges en direcció a B però desconeix quins en rebrà.

transferència inconscient 0-1 *f* Protocol criptogràfic pel qual A envia un missatge a B i B el rep amb probabilitat 1/2 però A desconeix si l'ha rebut o no.

transferència inconscient 1-2 *f* Protocol criptogràfic pel qual A envia dos missatges a B i B només en rep un dels dos amb probabilitat 1/2 però A desconeix quin dels dos ha rebut.

verificador *m* Part encarregada de verificar una prova de coneixement nul.

xifratge de Vernam *m* Esquema de xifratge que aconsegueix seguretat perfecta.

Bibliografia

Brassard, G.; Crepeau, C.; Chaum, D. (1988). "Minimum Disclosure Proofs of knowledge". *Journal of Computer and System Sciences* (vol. 37, núm. 2).

Chaum, D. (1983)."Blind signatures for untraceable payments". *Advances in Cryptology Proceedings of Crypto* (vol. 82, núm. 3, pàg. 199-203). Boston: Springer-Verlag.

Chaum, D.; Evertse, J. H.; Graaf, J. van de (1988). "An improved protocol for demonstrating possession of discrete logarithms and some generalizations". *Proceedings of Eurocrypt'87*. Berlín: Springer.

Even, S.; Goldreich, O.; Lempel, A. (1985). "A Randomized Protocol for Signing Contracts". *Communications of the ACM* (vol. 28, núm. 6, pàg. 637-647). Nova York: Association for Computing Machinery.

Jakobsson, M.; Yung, M. (1996). "Proving without knowing: On oblivious, agnostic and blindfolded provers". *Advances in Cryptology - CRYPTO '96* (vol. 1109, pàg. 186-200). Berlín: Springer.

Shamir, A. (1979). "How to Share a Secret". *Communications of the ACM* (vol. 24, núm. 11, pàg. 612-613). Nova York: Association for Computing Machinery.