Criptografia

Estudis d'Informàtica i Multimèdia (coordinador)

XP03/05024/02258

www.uoc.edu

Josep Domingo Ferrer

Llicenciat i doctor en Informàtica per la Universitat Autònoma de Barcelona. Llicenciat en Matemàtiques per la UNED. El seu àmbit de recerca és la seguretat de la informació. Autor de més de setanta publicacions, tant nacionals com internacionals. Actualment és catedràtic del Departament d'Enginyeria Informàtica i Matemàtiques de la Universitat Rovira i Virgili, on encapçala el grup CRISES.

Jordi Herrera Joancomartí

Llicenciat en Matemàtiques per la Universitat Autònoma de Barcelona i doctor per la Universitat Politècnica de Catalunya. El seu àmbit de recerca és la seguretat de la informació i, més concretament, la criptografia i la protecció del *copyright*. Actualment és professor propi als Estudis d'Informàtica i Multimèdia de la Universitat Oberta de Catalunya.

Helena Rifà Pous

Enginyera de Telecomunicacions per la Universitat Politècnica de Catalunya (2001). Ha treballat en consultoria de TIC a Grupo Penteo i actualment és la responsable de projectes I+i a Safelayer Secure Communications. Ha participat en diversos projectes relacionats amb les tecnologies de PKI d'àmbit estatal (CICYT, Profit, etc.) i europeu (Ten-Telecom, FP5, FP6).

Segona edició: febrer 2004

Av. Tibidabo 39-43. 08035 Barcelona
Disseny: Manel Andreu
Producció editorial: Eureca media, SL
Fotografies: Prisma, Index, A.G.E. Fotostock, Oronoz, Vision photo productions
Il·lustracions: Andrada
ISBN: 84-9788-055-2
Dipòsit legal: B-1.007-2004

Criptografia

Introducció

En aquesta assignatura presentem els fonaments i les aplicacions principals de la criptografia. Aquesta ciència, coneguda antigament com l'art de l'escriptura secreta, ha esdevingut avui dia un company imprescindible del desenvolupament de la societat de la informació. Els objectius cabdals als quals serveix la criptografia són la confidencialitat, la integritat i l'autenticitat en el tractament de la informació en format electrònic. Una de les aplicacions més notables d'aquesta disciplina és actualment el comerç electrònic.

A fi de facilitar la consecució dels objectius d'aquesta assignatura, combinem els continguts teòrics amb els pràctics i orientem aquests darrers amb vista a introduir l'estudiant en les aplicacions.

Objectius

Aquesta assignatura conté els materials didàctics necessaris perquè l'estudiant assoleixi els objectius següents:

1. Assimilar la història, la terminologia i els supòsits de la criptografia.

2. Conèixer els fonaments teòrics de la criptografia moderna.

3. Adquirir els coneixements necessaris per a implementar xifres o criptosistemes. Específicament, familiaritzar-se amb el programari criptogràfic disponible a Internet.

4. Comprendre el funcionament de les infraestructures de clau pública que possibiliten la implementació de criptografia de clau pública.

5. Entendre el funcionament dels protocols criptogràfics actualment en ús.

Contingut

Mòdul didàctic 6

Signatures digitals

Josep Domingo Ferrer

1. Signatura digital
2. Esquemes de signatura digital
3. Funcions *hash*

Mòdul didàctic 7

Infraestructura de clau pública

Helena Rifà Pous

1. Conceptes bàsics
2. Components d'una infraestructura de clau pública
3. Models de confiança
4. Cicle de vida de claus i certificats digitals
5. Estructures de dades bàsiques de PKIX
6. Format i procediments per a generar missatges: PKCS
7. Protocols que utilitzen infraestructura de clau pública

Mòdul didàctic 8

Aplicacions de la criptografia

Josep Domingo Ferrer, Jordi Herrera Joancomartí

1. Autentificació i identificació
2. Esquemes de compartició de secrets
3. Situacions de desconfiança mútua
4. Diners electrònics
5. Concessió de drets intransferibles
6. Eleccions electròniques

Bibliografia

Fuster, A.; De la Guía, D.; Hernández, L.; Montoya, F.; Muñoz, J. (1997). *Técnicas criptográficas de protección de datos*. Madrid: Ra-ma.

Menezes, A.J.; Dorschot, P.C.; Vanstone, S.A. (1997). *Handbook of applied cryptography. Boca Ratón* (etc.): CRC Press.

Pieprzyk, J.; Hardjono, T.; Seberry, J. (2003). *Fundamentals of computer security*. Berlín: Springer.

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

Simmons, G.J. (1992). *Contemporary Cryptology: the Science of Information Integrity*. Nova York: IEEE Press.

Introducció a la criptografia

Josep Domingo Ferrer

P03/05024/02259

UOC
www.uoc.edu

Índex

Introducció a la criptografia

Introducció

Criptografia és un terme d'origen grec que prové dels mots *krypto* ('amagar') i *grapho* ('escriure'). Podem dir que la **criptografia** és la ciència i l'estudi de l'escriptura secreta.

Inicialment, la criptografia va aparèixer per a resoldre la necessitat de comunicar-se en presència d'un adversari (normalment en un context militar o diplomàtic). Actualment, engloba molts altres problemes; per citar-ne només uns quants, podem parlar de xifratge, autentificació, distribució de claus, etc.

La criptografia moderna proporciona els fonaments teòrics necessaris per a poder:

- Entendre exactament els problemes que acabem d'enumerar.
- Avaluar els protocols que en teoria poden resoldre aquests problemes.
- Construir protocols en la seguretat dels quals puguem confiar.

Els protocols que resolen els problemes bàsics esmentats es poden emprar com a base per a resoldre altres problemes més complexos, com ara els sistemes de pagament electrònic segur, usats en el comerç electrònic, que fan servir protocols d'autentificació i de xifratge.

Objectius

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

1. Conèixer la terminologia bàsica emprada en criptografia.

2. Tenir una visió històrica de la criptografia.

3. Prendre consciència de l'omnipresència de la criptografia en el món actual.

1. Terminologia

Tu as tes procédés d'information que je ne pénètre point.
Guy de Maupassant

Una **xifra** o **criptosistema** és un mètode secret d'escriptura, mitjançant el qual un text en clar es transforma en un text xifrat*. El procés de transformar text en clar en text xifrat s'anomena *xifratge*; el procés invers, transformar text xifrat en text en clar, s'anomena *desxifratge*. Tant el xifratge com el desxifratge són controlats per una o més claus criptogràfiques.

*** A vegades s'anomena *criptograma*.**

La **criptografia** i una disciplina complementària anomenada **criptoanàlisi** es coneixen conjuntament amb el nom de **criptologia**. La criptografia s'ocupa del disseny de xifres. La criptoanàlisi s'ocupa de trencar xifres. La motivació del criptoanalista pot ser l'interès intrínsec de descobrir el text en clar xifrat i/o la clau emprada, o bé ser de caire cientificotècnic (verificació de la seguretat de la xifra). El vessant cientificotècnic de la criptoanàlisi és essencial per a la depuració de les xifres i és molt útil per al progrés de la criptografia.

La necessitat de la criptoanàlisi...

... és menys reconeguda socialment que la de la criptografia. "Els cavallers no llegeixen el correu dels altres", va respondre el 1929 el secretari d'estat nord-americà H.L. Stimson en saber que el seu departament trencava sistemàticament els telegrames diplomàtics xifrats de diversos països.

1.1. Xifres elementals

Hi ha dues menes bàsiques de xifres: les transposicions i les substitucions. A continuació descrivim i il·lustrem breument cadascuna d'aquestes xifres:

1) Una **xifra de transposició** reordena els bits o els caràcters del text en clar; la clau de la xifra és el criteri de reordenació emprat.

Exemple de xifra de transposició

Considereu la xifra que divideix el text en clar en grups de k lletres i inverteix l'ordre de les lletres dins de cada grup per a obtenir el text xifrat. La clau en aquest cas és k. Prenent el text en clar següent:

DALT DEL COTXE HI HA DUES NINES,

si fem servir $k = 5$ i negligim els espais en blanc, obtenim el text xifrat següent:

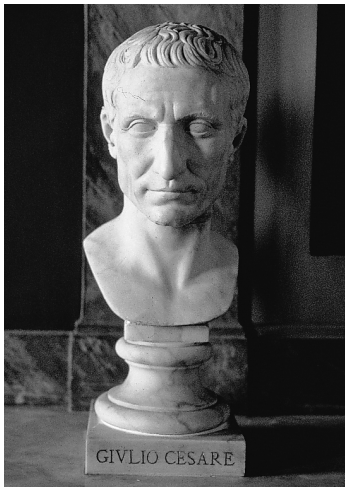
DTLADTOCLEHIHEXSEUDASENIN.

2) Una **xifra de substitució** canvia bits, caràcters o blocs de caràcters per substituts; la clau és el criteri de substitució emprat.

Exemple de xifra de substitució

Considereu la xifra que desplaça cada lletra de l'alfabet k posicions endavant (la lletra Z es desplaça cíclicament a l'inici de l'alfabet). La clau és k. Prenent el text en clar següent:

DALT DEL COTXE HI HA DUES NINES,



Els secrets de Cèsar

La xifra de substitució de l'exemple se sol anomenar **xifra de Cèsar**, perquè Juli Cèsar la feia servir amb $k = 3$ per comunicar-se amb Ciceró i altres amics seus.

si fem servir $k = 4$ i negligim els espais en blanc, obtenim el text xifrat que presentem a continuació:

HEPXHIPGSXBILMLEHYIWRMRIW.

1.2. Resistència de les xifres

Segons la resistència que tinguin als atacs dels criptoanalistes, les xifres es poden classificar de la manera següent:

a) Xifres trencables o febles: xifres per a les quals el criptoanalista té prou recursos de càlcul per a determinar el text en clar o la clau a partir del text xifrat, o per a determinar la clau a partir de parells de text en clar/text xifrat.

b) Xifres computacionalment segures o fortes: xifres que no poden ser trencades a partir d'una anàlisi sistemàtica amb els recursos de què disposa el criptoanalista.

c) Xifres incondicionalment segures: una xifra ho és si, independentment de la quantitat de text xifrat interceptada pel criptoanalista, no hi ha prou informació al text xifrat per a determinar el text en clar de manera única.

De fet, tan sols hi ha una xifra incondicionalment segura i veurem que en moltes situacions no és pràctica. La resta de xifres conegudes es poden trencar si els recursos de càlcul de l'enemic són il·limitats. Per tant, és més interessant parlar de xifres computacionalment segures.

Vegeu la xifra incondicionalment segura al subapartat 3.1 del mòdul didàctic "Fonaments de criptografia" d'aquesta assignatura.

1.3. Atacs criptoanalítics

Hi ha quatre mètodes bàsics d'atac criptoanalític:

1) En un **atac amb només text xifrat**, el criptoanalista ha de trobar la clau basant-se només en el text xifrat que ha pogut interceptar. El mètode de xifratge, la llengua en què és escrit el text en clar i algunes paraules probables es poden suposar coneguts.

Per a trobar la clau...

... d'un text que se sap que en clar és una ordre financera, el criptoanalista explotarà el fet que probablement inclourà paraules com "comprar", "vendre", "euro", etc.

2) En un **atac amb text en clar conegut**, el criptoanalista sap uns quants parells de text en clar/text xifrat i n'intenta deduir la clau o algun text en clar que no coneix.

Exemples d'atac amb text en clar conegut

Suposem que fem servir xifratge en les nostres sessions *telnet* contra un sistema UNIX; un espia que intercepti els nostres missatges sap que en una determinada posició apareixerà la forma xifrada de la paraula `login` i en una altra posició apareixerà la forma xifrada de `password`. A més, sap que és probable que més endavant aparegui la forma xifrada d'algunes comandes com `ls`, `pwd`, `whoami`, etc.

Els programes (codi font) xifrats són un altre exemple vulnerable a atacs amb text en clar conegut; en efecte, el criptoanalista sap que hi ha una bona part del text xifrat que correspon a paraules reservades del llenguatge.

3) En un **atac amb text en clar escollit**, el criptoanalista que intenta deduir la clau és capaç d'adquirir el text xifrat corresponent a un text en clar escollit per ell mateix. Aquest atac representa la situació més favorable per al criptoanalista, i és, per tant, el més perillós. Les bases de dades que guarden la informació en forma xifrada es presten a aquesta mena d'atacs si l'enemic pot inserir registres en clar i observar els canvis en el text xifrat emmagatzemat.

4) Un **atac amb text xifrat escollit** només té sentit en criptosistemes de clau pública, en els quals una de les dues transformacions de xifratge/desxifratge és pública. En aquesta modalitat d'atac, el criptoanalista és capaç d'adquirir el text en clar corresponent a un text xifrat escollit per ell mateix. Tot i que és poc probable que el text en clar que ha obtingut sigui intel·ligible, pot ajudar a deduir-ne la clau.

Vegeu els criptosistemes de clau pública a l'apartat 4 del mòdul didàctic "Xifres de clau pública" d'aquesta assignatura.

Actualment es considera que una xifra ofereix una seguretat acceptable només si pot resistir un atac amb text en clar conegut en què el criptoanalista té un nombre arbitrari de parells text en clar/text xifrat.

1.4. Atacs als sistemes informàtics i de comunicació

Els atacs criptoanalítics proven de trencar l'algorisme de xifratge suposant el coneixement d'una determinada informació. Ara bé, normalment cal un atac de tipus informàtic per a obtenir la informació necessària a l'hora de muntar un atac criptoanalític. De fet, si l'atac informàtic és prou hàbil, pot ser que no calgui recórrer a la criptoanàlisi: imaginem que un pirata informàtic aconsegueix entrar al nostre ordinador i llegir el fitxer on guardem la clau de la nostra xifra.

Per a muntar un atac criptoanalític, normalment és necessari muntar conjuntament un atac informàtic.

La criptografia protegeix dades enviades per un mitjà de comunicació o guardades en un sistema informàtic. La protecció té dos vessants:

- El **secret** o la **privacitat**, que permet preservar la confidencialitat de les dades, és a dir, impedir-ne la revelació no autoritzada.

- La **integritat** o **autenticitat**, que impedeix la modificació no autoritzada de les dades.

Els **atacs als sistemes de comunicació** per on circulen dades xifrades consisteixen en escoltes i se'n poden distingir dos tipus:

1) Els **atacs contra el secret**, que consisteixen en l'anomenada **escolta passiva***. L'enemic es limita a interceptar el text xifrat, normalment sense ser

*** En anglès, *eavesdropping*.**

detectat, amb la finalitat de deduir-ne la clau o el text en clar. L'ús de bons mètodes de xifratge pot fer estèrils aquesta mena d'atacs.

2) Els **atacs contra l'autenticitat**, que consisteixen en l'anomenada **escolta activa***. L'enemic es dedica a modificar missatges interceptats o a inserir-ne de completament inventats, amb la finalitat que el receptor accepti els missatges modificats o inventats com a bons. La criptografia no pot impedir que l'enemic faci aquesta mena d'atacs (per exemple, que torni a inserir un text xifrat anterior), però sí que permet al receptor detectar-los.

*** En anglès, *tampering*.**

Els **atacs a un sistema informàtic** en què es guarden dades xifrades també atempten contra el secret i l'autenticitat:

1) Els **atacs contra el secret** poden ser de tres tipus diferents:

a) L'**escombratge de memòria**, que fa referència a la cerca d'informació confidencial en l'emmagatzematge primari (memòria) o secundari (disc).

b) La **filtració**, que és la transmissió de dades confidencials a usuaris no autoritzats per part de processos amb accés legítim a les dades en clar.

c) L'**atac d'inferència**, que intenta deduir informació confidencial sobre un individu a partir de la correlació d'estadístiques publicades sobre grups d'individus.

Un atac d'inferència...

... pot servir per a deduir el sou d'un analista de sistemes concret a partir del sou mitjà dels analistes de sistemes de l'empresa.

2) Els **atacs contra l'autenticitat** inclouen els dos tipus següents:

a) La **falsificació**, que consisteix a modificar, inserir o esborrar dades.

b) La **destrucció accidental**, que fa referència a l'esborrament o la sobreescriptura no intencionada de dades.

3) Els **atacs mixtos**, que bàsicament es redueixen a l'anomenat **emmascarament**. Si un usuari aconsegueix d'entrar al sistema amb el compte d'un altre usuari, llavors pot accedir a informació confidencial de l'altre usuari (atac contra el secret) i fer-se passar per l'altre usuari davant de tercers (atac contra l'autenticitat). L'emmagatzemament de les contrasenyes en forma xifrada contribueix a dificultar l'emmascarament, però cal prendre precaucions suplementàries.

Mentre que la falsificació pot ser detectada (no impedida) per tècniques criptogràfiques, la destrucció accidental ens pot passar inadvertida fins i tot si fem servir la criptografia.

La criptografia pot fer estèril l'escombratge de memòria, però no pot combatre per si sola la filtració i la inferència.

Les tècniques criptogràfiques són suficients davant els atacs contra els sistemes de comunicació. En canvi, necessiten ser complementades per controls d'accés per contrarestar els atacs contra els sistemes informàtics.

2. Evolució històrica

Des de l'antiguitat fins a l'aparició dels ordinadors, la criptografia va ser més un art que una ciència. L'aparició de l'ordinador forçà la revolució científica de les tècniques criptogràfiques.

2.1. La criptografia com a art

El període que va des de l'antiguitat fins a l'any 1949 es pot anomenar *era de la criptografia precientífica*. Es tractava més aviat d'un art que d'una ciència, la qual cosa no vol dir que estigués mancada d'interès. Ja hem comentat que Juli Cèsar feia servir una xifra de substitució. No hi ha proves que demostrin que Brutus hagués trencat aquesta xifra, però és obvi que ara qualsevol noiet que sabés una mica de llatí no tindria cap problema per a sortir-se'n amb un atac amb només text xifrat sobre unes quantes frases xifrades. De fet, durant gairebé dos mil anys després de Cèsar, els criptoanalistes se'n sortien millor que els criptògrafs.

Atac a la xifra de Cèsar

Per a trencar la xifra de Cèsar, n'hi ha prou de comparar les freqüències de les lletres en el text xifrat amb les freqüències de les lletres en llatí clàssic; llavors se sap a quina lletra correspon la A, la B, etc.

L'any 1926, un enginyer de la companyia nord-americana AT&T anomenat G.S. Vernam va publicar una xifra remarcable per a ser usada amb el codi binari de Baudot. Com la de Cèsar, la **xifra de Vernam** consisteix a sumar una clau K aleatòria al text en clar M per a obtenir el text xifrat C. La diferència és que M, C i K prenen valors a $\{0, 1\}$ i que la suma és mòdul 2 (és a dir, una *o exclusiva*):

$$C = M \oplus K.$$

La innovació fonamental introduïda per Vernam fou fer servir la clau només una vegada, és a dir, xifrar cada bit de text en clar amb un nou bit de clau escollit a l'atzar. Això requereix la transferència segura (amb missatgers armats, per exemple) d'emissor a receptor de tants bits de clau com text en clar vulguem xifrar més tard. Malgrat aquest inconvenient, veurem que aquesta és l'única xifra incondicionalment segura*.

*** Aquesta propietat fou intuïda però no demostrada per Vernam.**

Durant la Segona Guerra Mundial, quan l'ús de la criptografia va ser generalitzat, es va començar a reconèixer que les matemàtiques podien ser útils en criptografia i en criptoanàlisi.

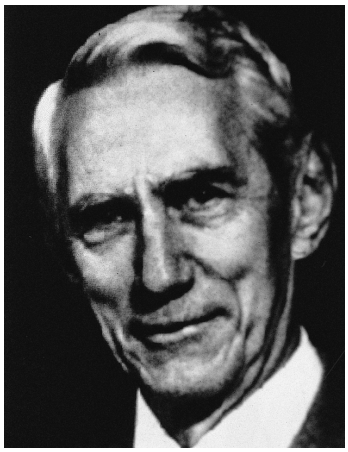
Turing i l'Enigma

Durant la Segona Guerra Mundial, un equip de matemàtics encapçalat per l'anglès A.M. Turing (inventor de la màquina que porta el seu nom) fou l'encarregat de trencar la xifra alemanya basada en les màquines de rotors *Enigma*.

2.2. La criptografia com a ciència moderna

La publicació l'any 1949 per part de C.E. Shannon de l'article "Communication Theory of Secrecy Systems" va inaugurar l'**era de la criptologia científ-**

fica de clau compartida. Shannon, que era enginyer i matemàtic, va elaborar una teoria dels sistemes secrets gairebé tan completa com la teoria de les comunicacions que havia publicat l'any anterior. Entre altres coses, l'article del 1949 demostrava la seguretat incondicional de la xifra de Vernam. Però a diferència de l'article sobre comunicacions del 1948, que va fer néixer la teoria de la informació com a disciplina, l'article del 1949 no va suposar un impuls comparable per a la recerca criptogràfica.



Claude Elwood Shannon, matemàtic nord-americà (nascut el 1916).

L'eclosió real de la criptografia s'esdevingué amb la publicació l'any 1976 per part de W. Diffie i M.E. Hellman de l'article "New Directions in Cryptography". Diffie i Hellman van mostrar per primer cop que era possible la comunicació secreta sense cap transferència de clau secreta entre l'emissor i el receptor, i van encetar així l'**època de la criptografia de clau pública** en la qual ens trobem actualment.

3. Aplicacions de la criptografia

Actualment, la criptografia és omnipresent en la vida quotidiana, per bé que d'una manera silenciosa. Desenvolupaments d'una actualitat tan candent com la telefonia mòbil, la televisió de pagament o el comerç electrònic no serien viables sense les tècniques criptogràfiques. En particular, el desenvolupament de sistemes de comerç electrònic segur és una activitat que a hores d'ara absorbeix una bona quantitat de mà d'obra informàtica.

En aquest apartat pretenem donar una visió ràpida i estimulant d'algunes de les aplicacions més vistents de la criptografia. Confiem que això estimularà l'alumne a continuar endavant amb l'assignatura.

3.1. Seguretat de les comunicacions

L'objectiu i l'ús primari de la criptografia és proporcionar seguretat en les comunicacions i, en la mesura que pugui, seguretat en els sistemes informàtics. A continuació veiem alguns camps concrets on cal aplicar-la:

1) En una **xarxa de paquets commutats** com IP o X.25, la seguretat de la informació transmesa es pot aconseguir amb un xifratge d'enllaç (al nivell d'enllaç de la jerarquia OSI) o bé extrem a extrem (als nivells alts de la jerarquia OSI). En el cas d'un xifratge d'enllaç calen equips de xifratge a cada node de la xarxa. En el cas d'un xifratge extrem a extrem, els equips terminals són els encarregats de fer el xifratge i el desxifratge.

El correu electrònic segur,...

... implementat per paquets tan coneguts com *Pretty Good Privacy* (PGP) o *Privacy Enhanced Mail* (PEM), és un exemple de xifratge extrem a extrem a nivell d'aplicació.

2) La **telefonia mòbil** és una altra gran consumidora de criptografia. Els primers sistemes de telefonia mòbil no feien servir xifratge de cap mena, a semblança de la telefonia fixa. La diferència, però, entre ambdós sistemes és que una trucada d'un telèfon mòbil pot ser escoltada sense necessitat de punxar cap cable: n'hi ha prou amb un equip sintonitzador. Actualment, la tecnologia GSM fa servir un algorisme de xifratge en flux que permet xifrar i desxifrar la conversa en temps real.

Els perills dels mòbils

Al seu dia, fou molt comentada la intercepció de converses comprometedores mantingudes per dirigents socialistes espanyols amb telèfons mòbils sense xifratge.

3) La **televisió de pagament** és una aplicació de la criptografia que es pot veure tant des del punt de vista de la seguretat de les comunicacions com des del punt de vista del comerç electrònic. En efecte, cal protegir els continguts televisius respecte d'aquells espectadors que no són abonats. Els descodificadors habituals són en realitat dispositius desxifradors; com en el cas de la telefonia mòbil, es fa servir una xifra en flux que permet xifrar i desxifrar les imatges en temps real.

3.2. Votacions i contractes electrònics

Una **votació electrònica** és una votació en la qual el votant no es desplaça físicament a un col·legi electoral per votar, sinó que vota per mitjà del seu terminal, que està connectat a una xarxa. Els problemes de seguretat que planteja la votació electrònica són complexos:

La criptografia proporciona la seguretat necessària per a fer possible el vot electrònic.

1) Tan sols els votants autoritzats haurien de poder votar i només ho podrien fer un sol cop. En un col·legi electoral, el votant presenta un document acreditatiu, es comprova si apareix a la llista censual i, en cas afirmatiu, es fa constar. Fer aquest procediment de manera electrònica requereix proporcionar una credencial electrònica al votant i mantenir la integritat del cens. Les tècniques criptogràfiques poden satisfer aquests requeriments.

2) El vot ha de ser secret. En un col·legi electoral, el votant diposita el seu vot en un sobre tancat abans de ficar-lo a l'urna. La criptografia pot ajudar a mantenir el secret del vot en un entorn electrònic.

3) No s'hauria de poder duplicar el vot de ningú. A diferència dels vots de paper en una urna, la còpia de vots electrònics és trivial si no s'utilitza la criptografia.

4) El votant ha de poder comprovar que el seu vot s'ha tingut en compte. Quan dipositem un vot de paper a l'urna, sabem que els interventors i la mesa no permetran que es descarti el nostre vot. En un entorn electrònic, hauríem de tenir la mateixa tranquil·litat.

La **signatura electrònica de contractes** planteja uns problemes similars als de la votació electrònica. En efecte, signar un contracte sobre paper de

manera presencial és trivial: dues parts A i B es reuneixen en una sala; A no deixa marxar B fins que A no obté una còpia del contracte signada per B; igualment, B no deixa marxar A fins que B no obté una còpia del contracte signada per A.

Veurem que hi ha tècniques criptogràfiques que permeten signar documents en format electrònic; ara bé, sense la presència física d'ambdues parts interessades a la mateixa sala, quina gosarà signar primer el contracte electrònic? Si no es fa servir una tercera part de confiança (notari electrònic), podria passar que B obtingués una còpia del contracte signada per A, i que en canvi A es quedés sense res. Per a resoldre aquest problema, hi ha protocols criptogràfics que asseguren que ambdues parts es troben en igualtat de condicions durant tot el procés de signatura del contracte.

3.3. Comerç electrònic

Com les votacions i els contractes electrònics, el comerç electrònic és un altre pas cap a la informatització de les relacions socioeconòmiques. Ja hem vist que la pèrdua de presència física en les relacions humanes genera problemes de seguretat complexos. El comerç electrònic no n'és una excepció i tampoc no seria viable sense la criptografia.

El problema bàsic del comerç electrònic és el pagament: com es pot pagar per mitjà d'una xarxa? La manera usual de fer transaccions monetàries per Internet és, avui dia, enviant el número de la targeta de crèdit. Això té inconvenients si ho comparem amb el pagament en efectiu: d'una banda, ens poden cobrar més del que volíem pagar; de l'altra, el pagament no és anònim: el client s'identifica cada cop que fa una compra i per tant el venedor sap qui compra què. En el seu article "Untraceable electronic mail, return addresses, and digital pseudonyms", D. Chaum va suggerir un protocol criptogràfic per a obtenir **diners electrònics** que no suposessin cap inconvenient respecte dels diners convencionals.

Quan la mercaderia objecte de comerç electrònic és informació en format digital (música, llibres, pel·lícules, etc.), apareix un altre problema, el de la **protecció del copyright**. En efecte, si fotocopiar paper ja és fàcil, copiar informació en format digital és trivial, barat i es troba a l'abast de tothom. La criptografia no pot impedir la pirateria informàtica, però sí que pot ajudar a identificar els pirates.

La supressió de la presència física en les relacions socioeconòmiques seria inviable per raons de seguretat sense l'existència de la criptografia.

Resum

Amb l'aparició dels primers ordinadors, la criptografia deixa de ser un art mil·lenari i esdevé una ciència, l'objectiu bàsic de la qual és permetre la comunicació i l'emmagatzematge segur d'informació en presència d'un adversari. Juntament amb aquest **objectiu bàsic de secret o privacitat**, la criptografia moderna permet resoldre el **problema de l'autenticitat o integritat de la informació**.

Una **xifra** o **criptosistema** transforma un text en clar en un text xifrat o criptograma. Els processos de xifratge i de desxifratge són controlats per una o més claus, que solen ser secretes. Les xifres elementals es basen en transposicions i en substitucions.

En funció de la seguretat, les xifres es poden classificar en febles, fortes i incondicionalment segures. La **criptoanàlisi** té com a objectiu trencar una xifra determinant-ne la clau a partir del text en clar i del text xifrat; segons el coneixement que se suposa que té el criptoanalista, hi ha diversos **tipus d'atac criptoanalític**.

A banda dels atacs criptoanalítics, cal tenir en compte els **atacs als sistemes informàtics i de comunicacions**. A diferència dels atacs criptoanalítics, aquesta modalitat d'atacs no es basa en l'explotació de les febleses dels algorismes de xifra. La idea és aprofitar febleses dels sistemes informàtics o de les comunicacions per a recuperar la clau o el text en clar.

Les **aplicacions de la criptografia moderna** van molt més enllà de la seguretat de les comunicacions militars i diplomàtiques. Pel que fa a comunicacions segures, la criptografia permet garantir la seguretat de les xarxes obertes, del correu electrònic i de la telefonia mòbil. Processos com les votacions, la signatura de contractes, etc., també poden ser fets de manera electrònica i sense coincidència física de les parts mitjançant tècniques criptogràfiques. El comerç electrònic és una altra "aplicació estrella" en els nostres dies.

Activitats

1. Identifiqueu en el vostre entorn algun dispositiu que utilitzi el xifratge.

2. Els pirates informàtics es valen d'atacs criptoanalítics o bé d'atacs als sistemes informàtics i de comunicacions?

3. Cerqueu a Internet informació sobre els programes analitzadors de trànsit (en anglès, *sniffers)*, que permeten escoltar el trànsit que circula per una xarxa local a l'usuari d'una estació que hi és connectada.

4. Quins problemes de seguretat planteja efectuar una votació de manera electrònica? Com es resolen aquests problemes en les votacions convencionals?

Glossari

Atac: estratègia o mètode que té per objectiu descobrir la clau de xifratge o bé el text en clar. Els atacs criptoanalítics exploten les febleses dels algorismes de xifra. Els atacs als sistemes informàtics i de comunicacions exploten les vulnerabilitats d'aquests sistemes.

Autenticitat: propietat de trobar-se, en relació amb la informació, en el mateix estat en què va ser produïda, sense modificacions no autoritzades; és sinònim d'integritat.

Autentificació: comprovació de l'autenticitat.

Clau: paràmetre, normalment secret, que controla els processos de xifratge i/o de desxifratge.

Criptoanàlisi: ciència que s'ocupa de trencar xifres, és a dir, descobrir la clau o el text en clar usats com a entrades de la xifra.

Criptografia: ciència i estudi de l'escriptura secreta.

Criptograma: text xifrat.

Criptologia: denominació conjunta de la criptografia i de la criptoanàlisi.

Criptosistema: xifra.

Desxifratge: procés de transformació del text xifrat en text en clar.

Integritat: propietat de no haver sofert, en relació amb la informació, modificacions ni supressions parcials no autoritzades.

Privacitat: dret de les persones a salvaguardar la seva intimitat, especialment pel que fa a les dades de què disposen les entitats públiques o privades.

Xifra: mètode secret d'escriptura, mitjançant el qual un text en clar es transforma en un text xifrat.

Xifra de substitució: xifra basada a canviar els bits o els caràcters del text en clar per substituts.

Xifra de transposició: xifra basada a reordenar els bits o els caràcters del text en clar.

Xifratge: procés de transformació d'un text en clar en un text xifrat.

Bibliografia

Denning, D.E. (1982). *Cryptography and Data Security*. Reading (Massachusetts): Addison-Wesley.

Fúster, A.; De la Guía, D.; Hernández, L.; Montoya, F.; Muñoz, J. (1997). *Técnicas criptográficas de protección de datos*. Madrid: Ra-ma.

Simmons, G.J. (1992). *Contemporary Cryptology: the Science of Information Integrity*. Nova York: IEEE Press.

Fonaments de criptografia

Josep Domingo Ferrer
Jordi Herrera Joancomartí

P03/05024/02260

UOC
www.uoc.edu

Índex

Fonaments de criptografia

Introducció

En aquest mòdul didàctic presentem els fonaments per a entendre les tècniques criptogràfiques modernes, i ho fem seguint l'esquema següent:

1) Comencem tractant detalladament els **criptosistemes històrics**, és a dir, els que es feien servir abans de l'aparició dels ordinadors. En molts casos, aquests criptosistemes han servit de base als actuals.

2) A continuació veurem els **conceptes bàsics de la teoria de la informació**, sobre la qual s'ha construït la criptografia moderna.

En termes de la teoria de la informació formularem les **definicions de secret perfecte i d'autenticitat perfecta**. Veurem que en la pràctica és difícil, si no impossible, aconseguir el secret i l'autenticitat perfectes. No obstant això, aquests conceptes han de servir de guia en el disseny dels criptosistemes.

Per acabar, introduirem dos conceptes bàsics en la criptoanàlisi: la **suposició de Kerckhoff** i la **distància d'unicitat**, i finalment il·lustrarem amb un exemple com es pot criptoanalitzar un criptosistema concret.

Objectius

En els materials didàctics associats a aquest mòdul l'estudiant trobarà les eines i els continguts necessaris per a assolir els objectius següents:

1. Conèixer les tècniques històriques de xifratge.

2. Adquirir un coneixement operatiu dels conceptes bàsics de la teoria de la informació.

3. Fer-se càrrec de la rellevància de la teoria de la informació per a la criptografia i la criptoanàlisi modernes.

4. Copsar els principis elementals de la criptoanàlisi.

5. Adonar-se de la feblesa dels criptosistemes històrics.

6. Entendre les definicions de *secret* i d'*autenticitat perfectes*.

1. Criptosistemes històrics

Occultas seu furtivas notas politioris literature viri eas literas appellant, quae artificio huiusmodi confictae sunt, ut non possint ab alio, quam ab eo, cui literi destinatur, interpretari.
G.B. Porta

Hi ha dues menes de xifres elementals, les basades en el principi de transposició i les basades en el principi de substitució. Doncs bé, totes les xifres històriques (anteriors a la Segona Guerra Mundial) es basen en un d'aquests dos principis, o en una combinació d'ambdós.

Recordeu que hem vist les xifres de transposició i les de substitució en el subapartat 1.1 del mòdul "Introducció a la criptografia" d'aquesta assignatura.

1.1. Xifres de transposició

Les xifres de transposició reordenen els caràcters d'acord amb certes regles.

Normalment, la reordenació dels caràcters es feia amb l'ajut d'alguna figura geomètrica. Aquest xifratge s'efectuava en dos passos:

1) El text en clar s'escrivia a la figura seguint un determinat *camí d'entrada*.

2) Seguint un determinat *camí de sortida*, s'extreia el text xifrat de la figura.

Transposicions espartanes

Els espartans feien servir un bastó com a figura per a fer una xifra de transposició. Es preparaven dos bastons del mateix gruix exactament; l'emissor se'n quedava un i l'altre era lliurat al receptor com a pas previ a l'enviament de missatges. Per a xifrar un missatge, l'emissor enrotllava una cinta de pergamí en espiral al voltant del seu bastó. Tot seguit, escrivia el missatge a la cinta en línies al llarg del bastó, després retirava la cinta i l'enviava al receptor. Pel camí, la cinta de pergamí no era més que una successió de lletres gregues col·locades en un ordre inintel·ligible. Quan el receptor enrotllava la cinta rebuda al voltant del seu bastó, podia llegir el missatge original. Noteu que el gruix del bastó actuava com a clau (amb un bastó més prim o més gruixut no podia recuperar-se el missatge inicial).

Esparta...

... era una ciutat estat de la Grècia clàssica que durant el segle v aC va tenir una llarga rivalitat amb Atenes. Els espartans eren fortament militaristes i empraven el xifratge per a les comunicacions militars.

Moltes xifres de transposició permuten els caràcters del text en clar amb un període fixat d. Siguin $\mathbb{Z}_d$ els enters d'1 fins a d, i sigui $f : \mathbb{Z}_d \to \mathbb{Z}_d$ una permutació sobre $\mathbb{Z}_d$. La clau per a la xifra ve donada per la parella $K = (d,f)$. Els blocs successius de d caràcters es xifren permutant els caràcters segons f. D'aquesta manera, un missatge en clar com ara el següent:

$$M = m_1 \dots m_d\, m_{d+1} \dots m_{2d} \dots$$

queda xifrat com:

$$E_K(M) = m_{f(1)} \dots m_{f(d)}\, m_{d+f(1)} \dots m_{d+f(d)} \dots$$

El desxifratge fa servir la permutació inversa de f.

Transposició amb període fix

Considerem $d = 3$ i prenem la permutació f tal que $f(1) = 2$, $f(2) = 3$ i $f(3) = 1$. Llavors el missatge:

$$M = \text{CRIPTOGRAFIA}$$

queda xifrat com:

$$E_K(M) = \text{ICROPTAGRAFI}.$$

1.2. Xifres de substitució

Hi ha quatre menes de xifres de substitució: la simple, l'homofònica, la polialfabètica i la poligràfica. A continuació expliquem cadascuna d'aquestes xifres.

1.2.1. Substitució simple

Una **xifra de substitució simple** canvia cada caràcter d'un alfabet en clar ordenat, denotat per $\mathcal{A}$, per la lletra corresponent d'un alfabet xifrat, denotat per $\mathcal{C}$.

Més formalment ho podem expressar de la manera següent:

- $\mathcal{A} = \{a_0, a_1, ..., a_{n-1}\}$,
- $\mathcal{C} = \{f(a_0), f(a_1), ..., f(a_{n-1})\}$,

on $f : \mathcal{A} \to \mathcal{C}$ és una aplicació bijectiva que fa correspondre a cada caràcter de $\mathcal{A}$ un caràcter de $\mathcal{C}$. La clau de la xifra és determinada per la funció f, de manera que un missatge en clar com aquest:

$$M = m_1\, m_2\, ...$$

queda xifrat com:

$$E_K(M) = f(m_1)\, f(m_2)\, ...$$

La **xifra de Cèsar** és una xifra de substitució simple, on $\mathcal{A} = \mathcal{C}$ i la clau és $f(a) = (a + k) \bmod 26$, amb k entre 0 i 25. De fet, Cèsar feia servir $k = 3$.

Recordeu que vam veure la xifra del Cèsar al subapartat 1.1 del mòdul "Introducció a la criptografia" d'aquesta assignatura.

1.2.2. Substitució homofònica

Les substitucions simples tenen l'inconvenient que preserven les freqüències del text en clar en el text xifrat, la qual cosa facilita molt els atacs criptoanalítics.

Freqüències del text

Imaginem que volem xifrar missatges en català amb una xifra de Cèsar amb $k = 3$. Llavors la freqüència de la lletra D en el text xifrat és la mateixa que la freqüència de la lletra A en el text en clar. Com que les freqüències de les lletres en català són conegudes, el criptoanalista pot deduir, veient només el text xifrat, que la D es desxifra com a A i que, per tant, la clau és $k = 3$.

Una **xifra de substitució homofònica** té com a objectiu dissimular les freqüències dels caràcters del text en clar. La idea és fer correspondre a cada caràcter a de l'alfabet de text en clar no pas un, sinó un conjunt $f(a)$ de símbols de text xifrat anomenats *homòfons*.

Homòfon

El terme d'arrel grega *homòfon* vol dir 'que té el mateix so'. En criptografia, els símbols homòfons corresponen al mateix caràcter en clar.

Per tant, si fem servir una xifra de substitució homofònica la correspondència entre el text en clar i el text xifrat té la forma $f : \mathcal{A} \rightarrow 2^{\mathcal{C}}$. De manera que un missatge en clar com el següent:

$$M = m_1\, m_2 \ldots$$

queda xifrat com:

$$C = c_1\, c_2 \ldots$$

on cada c_i es tria a l'atzar dins del conjunt d'homòfons $f(m_i)$.

Per a amagar les freqüències del text en clar i evitar atacs criptoanalítics com l'esmentat en parlar de la substitució simple, una bona estratègia és que el conjunt d'homòfons $f(a)$ tingui un cardinal proporcional a la freqüència relativa del caràcter en clar a. D'aquesta manera s'aconsegueix que les freqüències dels símbols en el text xifrat siguin pràcticament uniformes.

Homofonia italiana

El primer ús conegut d'una xifra homofònica a Europa remunta al 1401, en la correspondència entre el Ducat de Màntua i Simeone de Crema.

Durant més d'un segle, criptoanalistes afeccionats varen provar de desxifrar un text que, es deia, indicava la situació d'un tresor enterrat a l'estat americà de Virgínia per un escamot d'aventurers manat per T.J. Beale. Ara se sap que el text era xifrat amb una xifra homofònica anomenada **Xifra de Beale** que té com a clau la declaració d'independència dels Estats Units d'Amèrica. Beale va numerar les paraules de l'esmentada declaració i per a xifrar una lletra del text en clar, la substituïa pel nombre d'alguna paraula que comencés per la lletra que calia xifrar.

Per exemple, la lletra *W* es podia xifrar amb els números 1, 19, 40, 66, 72, 290 i 459 (aquests són els homòfons de *W*). Com que el nombre de paraules que comencen per *W* en un text anglès una mica llarg és aproximadament proporcional a la freqüència relativa de *W* en anglès, la xifra de Beale aconsegueix uniformitzar les freqüències dels símbols del text xifrat si el text en clar és en anglès.

1.2.3. Substitució polialfabètica

Hem vist que la substitució homofònica intenta amagar la distribució de freqüències dels símbols del text en clar assignant diversos símbols de text xifrat a cada caràcter del text en clar.

Itàlia altra vegada

Com la substitució homofònica, la substitució polialfabètica està documentada per primer cop a Itàlia. La primera xifra d'aquesta mena fou publicada per L.B. Alberti l'any 1568.

Doncs bé, la **substitució polialfabètica** persegueix la mateixa finalitat que l'homofònica, però aplica diversos criteris de substitució en comptes d'un de sol.

La majoria de xifres polialfabètiques són xifres de substitució periòdica basades en un període d. Siguin $\mathscr{C}_1, ..., \mathscr{C}_d$ alfabets de text xifrat; sigui $f_i : \mathscr{A} \to \mathscr{C}_i$ una aplicació de l'alfabet de text en clar $\mathscr{A}$ a l'i-èsim alfabet de text xifrat $\mathscr{C}_i$, per a $1 \leq i \leq d$. Un missatge en clar com ara el següent:

$$M = m_1 \ldots m_d \, m_{d+1} \ldots m_{2d} \ldots$$

es xifra repetint la seqüència d'aplicacions $f_1, ..., f_d$ cada d caràcters, amb la qual cosa s'obté:

$$C = E_K(M) = f_1(m_1) \ldots f_d(m_d) \, f_1(m_{d+1}) \ldots f_d(m_{2d}) \ldots$$

L'anomenada **xifra de Vigenère** és una xifra atribuïda indegudament al criptògraf francès del segle XVI Blaise de Vigenère. La clau és determinada per una seqüència de lletres K o, més ben dit, pels números d'ordre d'aquestes lletres:

$$K = k_1 \ldots k_d,$$

on k_i ($i = 1, ..., d$) indica la quantitat de desplaçament a l'i-èsim alfabet, és a dir:

$$f_i(a) = (a + k_i) \bmod n.$$

Exemple de xifratge amb la xifra de Vigenère

Prenem $n = 26$ i $d = 4$. Siguin els alfabets $\mathscr{A}, \mathscr{C}_1, ..., \mathscr{C}_4$ iguals a l'alfabet llatí; si la clau és K = INDI = {8, 13, 3, 8}, el text en clar BALDUFES queda xifrat com JNOLCSHA.

L'anomenada **xifra de Beaufort** s'atribueix a l'almirall anglès Sir Francis Beaufort, però va ser proposada primer per l'italià G. Sestri el 1710. L'única diferència amb la xifra de Vigenère és que les funcions de substitució són:

$$f_i(a) = (a - k_i) \bmod n,$$

per a $i = 1, ..., d$.

Les **màquines de rotors** implementen xifres polialfabètiques amb un període llarg. Una màquina de rotors consisteix en un banc de rotors o rodes. El perímetre de cada rotor té 26 contactes elèctrics (un per a cada lletra) tant a la cara del davant com a la cara del darrere. Cada contacte de la cara del davant es troba connectat a un contacte de la cara del darrere per a implementar-hi una aplicació f_i de lletres del text en clar a lletres del text xifrat. Els rotors poden rodar i col·locar-se en 26 posicions diferents, per tal d'alterar l'aplicació. La cara del darrere del rotor i-èsim es troba connectada a la cara del davant del rotor $i+1$-èsim. Una lletra en clar en forma de senyal elèctric entra al primer rotor i passa tots els rotors en seqüència i surt del darrer rotor.

Durant la Segona Guerra Mundial,...

... els alemanys van fer servir una màquina de rotors anomenada *Enigma* i inventada per A. Scherbius.

El cablejat entre rotors i les posicions inicials dels rotors determinen la clau inicial. Cada cop que es xifra una lletra de text en clar, un o més rotors canvien de posició, amb la qual cosa la clau canvia. Una màquina amb t rotors no torna a la posició inicial fins al cap de $d = 26^t$ xifratges.

La **xifra de Vernam** pot ser vista com una xifra de substitució polialfabètica en què la clau és aleatòria i el període d és més gran que la llargada del text en clar.

1.2.4. Substitució poligràfica

Les tres substitucions anteriors xifren una sola lletra de text en clar cada cop.

Les **xifres de substitució poligràfica** xifren blocs grans de lletres, amb la qual cosa dificulten atacs criptoanalítics basats en les freqüències individuals de les lletres del text en clar.

La clau de la **xifra de Hill** és una matriu K de d files i d columnes. Per a xifrar, s'agafen blocs successius de d caràcters del text en clar com si fossin vectors. Cada vector de text en clar es multiplica per K per a obtenir un vector de text xifrat (també amb d caràcters). Els productes són mòdul n, on n és el cardinal de l'alfabet de text xifrat. El procés de xifratge d'un vector M es pot escriure com:

$$C = E_K(M) = K \cdot M \bmod n.$$

2. Fonaments de la teoria de la informació

L'any 1949 C.E. Shannon va proporcionar un fonament teòric a la criptografia basat en la teoria de la informació que ell mateix havia elaborat l'any anterior. És important conèixer el fonament teòric de la criptografia perquè és allò que l'eleva a la condició de ciència. Per a entendre la formulació de Shannon calen uns conceptes bàsics de la teoria de la informació que donarem en aquest apartat.

Una **variable aleatòria** es pot definir informalment com una variable que adopta un valor entre un conjunt de valors possibles, de manera que cada valor possible té una certa probabilitat no nul·la de ser adoptat. Si el conjunt de valors que pot prendre la variable és discret, llavors es diu que la variable aleatòria és discreta. Així, doncs, tenim que:

- Un missatge M és vist pel criptoanalista com una variable aleatòria discreta que pot prendre com a valors uns textos concrets.

- Una clau K també és vista pel criptoanalista com una variable aleatòria discreta. El criptoanalista voldria saber quin valor pren la variable K, però en principi només pot conèixer (a tot estirar) la distribució de probabilitats de la clau, és a dir, les probabilitats que té la clau d'adoptar cadascun dels valors possibles. Si la clau és aleatòria en el sentit de la xifra de Vernam, la probabilitat de cadascun dels valors possibles és la mateixa.

Per exemple,...

... imaginem un missatge "Comanda Unix" enviat per *telnet*; en un 50% dels casos (probabilitat 0,5) es tracta de la comanda `ls`, en un 20% dels casos (probabilitat 0,2) es tracta de la comanda `vi`, etc.

Sigui X una variable aleatòria discreta i x un dels valors possibles de X. Anomenem **funció de probabilitat de la variable X**, la funció $p(x) = P(X = x)$.

Per a cada valor x, $p(x)$ indica la probabilitat que la variable X prengui el valor x. Si definim $\sup(X) = \{x \mid P(X = x) \neq 0\}$, es compleix:

$$\sum_{x \in \sup(X)} p(x) = 1.$$

A partir de la funció de probabilitat d'una variable aleatòria X, l'**entropia de Shannon** mesura en bits la incertesa sobre X o, dit altrament, la informació que ens aporta el fet de saber que X ha pres tal o tal altre valor. Clarament, no obtenim la mateixa informació en saber el valor que ha pres una variable que només en té un de possible que en saber el que ha pres una altra variable que en pot pendre cinquanta de diferents amb la mateixa probabilitat.

L'**entropia d'una variable aleatòria X**, $H(X)$, és l'esperança matemàtica del logaritme en base 2 de la seva funció de probabilitat, amb el signe canviat:

$$H(X) = -\sum_x p(x) \log_2 p(x),$$

on el sumatori s'estén per a $x \in \sup(X)$. Podem expressar la mateixa relació de manera equivalent:

$$H(X) = \sum_x p(x) \log_2\left(\frac{1}{p(x)}\right). \qquad (2.1)$$

És important recordar que l'entropia es mesura en bits.

Càlcul de l'entropia del camp *sexe*

El camp *sexe* d'una base de dades pot ser vist com una variable aleatòria que pot prendre dos valors: home i dona. Si ambdós valors es consideren equiprobables, el càlcul de l'entropia és el següent:

$$H(sexe) = \frac{1}{2}(\log_2 2) + \frac{1}{2}(\log_2 2) = 1 \text{ bit.}$$

Així, doncs, el camp *sexe* conté 1 bit d'informació sota la hipòtesi que els dos sexes tenen la mateixa probabilitat.

Intuïtivament, cada terme $\log_2 [1/p(x)]$ a l'equació (2.1) representa el nombre de bits necessaris per a codificar el valor x amb una codificació òptima, és a dir, una codificació que minimitzi el nombre esperat de bits per a transmetre o emmagatzemar. Vist d'aquesta manera, $H(X)$ seria la longitud mitjana ponderada de les codificacions òptimes dels valors de X (la ponderació es fa segons la probabilitat de cada valor). Els codis de Huffman serveixen per a construir la codificació òptima esmentada i són emprats en compressió de dades.

Lectura complementària

Podeu veure amb detall els codis de Huffman al llibre: **J. Rifà, L. Huguet** (1991). *Comunicación digital*. Barcelona: Masson.

Càlcul de la informació continguda en un missatge

Suposem un missatge X que pot prendre n valors $x_1, ..., x_n$, tots amb la mateixa probabilitat $p(x_i) = 1/n$, per a $i = 1, ..., n$. Llavors:

$$H(X) = n\left(\frac{1}{n}\log_2 n\right) = \log_2 n \text{ bits.}$$

Així, doncs, saber quin valor ha pres X aporta $\log_2 n$ bits d'informació. També podem dir que la codificació òptima de cadascun dels n possibles valors requereix $\log_2 n$ bits.

Si en aquest exemple fem $n = 1$ i $p(x_1) = 1$, tenim que $H(X) = \log_2 1 = 0$ bits. En una variable que no és tal, sinó que pren un valor constant, no hi ha informació.

A l'exemple anterior hem vist dos casos extrems. De fet, l'entropia màxima es dóna quan hi ha incertesa màxima, és a dir, quan tots els n valors possibles d'una variable són equiprobables i l'entropia mínima es dóna quan només hi ha un valor possible i llavors val 0. Formalment, ho podem recollir en la proposició que expressem tot seguit:

Proposició 1: si X té n possibles valors, llavors $0 \leq H(X) \leq \log_2 n$.

A més de l'entropia, per a l'estudi teòric de la criptografia és rellevant el concepte d'*entropia condicionada*. Aquesta entropia mesura la incertesa que ens queda sobre una variable X quan coneixem el valor pres per una altra variable Y.

L'**entropia de la variable *X* condicionada a la variable *Y*** es denota per $H(X|Y)$ i es calcula de la manera següent:

$$H(X|Y) = -\sum_x \sum_y p(x,y) \log_2 p(x|y) = \sum_y p(y) \sum_x p(x|y) \log_2 \left(\frac{1}{p(x|y)}\right), \qquad (2.2)$$

on $p(x,y) = p(X = x, Y = y)$ és la funció de probabilitat conjunta de X i de Y*, i $p(x|y) = p(X = x|Y = y) = p(X = x, Y = y)/p(Y = y)$ és la funció de probabilitat de X condicionada a Y**.

*** Probabilitat que *X* valgui *x* i alhora *Y* valgui *y*.**
**** Probabilitat que *X* valgui *x* si sabem que *Y* val *y*.**

Càlcul de l'entropia condicionada d'un missatge

Sigui X un missatge que pot prendre quatre valors, tots amb la mateixa probabilitat 1/4; per tant, $H(X) = \log_2 4 = 2$ bits; i sigui Y un missatge que pot prendre quatre valors, també tots amb probabilitat 1/4. Suposem que cada valor de Y restringeix la tria de X a dos dels quatres valors possibles, segons les regles següents:

- Si surt $Y = y_1$, llavors $X = x_1$ o $X = x_2$.
- Si surt $Y = y_2$, llavors $X = x_2$ o $X = x_3$.
- Si surt $Y = y_3$, llavors $X = x_3$ o $X = x_4$.
- Si surt $Y = y_4$, llavors $X = x_4$ o $X = x_1$.

De la primera regla resulta $p(x_1|y_1) = p(x_2|y_1) = 1/2$ i $p(x_3|y_1) = p(x_4|y_1) = 0$. Anàlogament, les altres regles donen lloc a dues probabilitats condicionades iguals a 1/2 i dues de nul·les. Aplicant el darrer membre de l'equació (2.2) en resulta

$$H(X|Y) = 4 \cdot \left[\frac{1}{4} \cdot 2 \cdot \left(\frac{1}{2} \cdot \log_2 2\right)\right] = \log_2 2 = 1.$$

Veiem com el coneixement de Y redueix la incertesa sobre X a un sol bit, mentre que inicialment n'eren dos.

Pot passar que X i Y siguin variables que no tenen res a veure l'una amb l'altra*. En aquest cas, es diu que X i Y són independents i llavors es verifica el següent:

*** Per exemple, la latitud i la longitud d'un punt del planeta triat a l'atzar.**

- $H(X|Y) = H(X)$.
- $H(Y|X) = H(Y)$.

El coneixement del valor pres per una de les variables no redueix la incertesa sobre l'altra. La proposició següent explica la relació general entre $H(X|Y)$ i $H(X)$.

Proposició 2: el coneixement d'una variable Y només pot reduir la incertesa sobre una altra variable X, és a dir:

$$0 \leq H(X|Y) \leq H(X). \qquad (2.3)$$

En el cas que Y determini X, la incertesa sobre X baixa a zero (igualtat esquerra). En el cas que X i Y siguin independents, no es redueix gens la incertesa sobre X, que continua essent $H(X)$ (igualtat dreta).

L'**entropia conjunta de dues variables *X* i *Y***, $H(X,Y)$, és la incertesa sobre la combinació de valors que prendran ambdues variables, és a dir, el valor que prendrà el vector de variables (X,Y):

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y). \tag{2.4}$$

Intuïtivament, la incertesa sobre el comportament conjunt de X i de Y es descompon en la incertesa quant al valor que prendrà X més la incertesa sobre el que prendrà Y sabent quin valor ha pres X (es podria dir el mateix si intercanviéssim X i Y).

Proposició 3: si X, Y i Z són variables aleatòries es compleixen unes quantes igualtats que no ens haurien de sorprendre si tenim en compte que l'entropia de Shannon és bàsicament el logaritme d'una probabilitat:

- $H(X,Y,Z) = H(X) + H(Y|X) + H(Z|X,Y)$.
- $H(X,Y|Z) = H(X|Z) + H(Y|X,Z) = H(Y|Z) + H(X|Y,Z)$.
- $0 \leq H(X|Y,Z) \leq H(X|Y)$.
- $H(X) \leq H(X,Y)$.
- $H(X|Z) \leq H(X,Y|Z)$.

Un concepte important per a la definició de secret perfecte és el d'informació mútua. La **informació mútua entre dues variables aleatòries *X* i *Y***, $I(X,Y)$, es defineix de la manera següent:

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

Tenint en compte la proposició 2, queda clar que $I(X,Y)$ sempre és positiva o nul·la. Intuïtivament, la informació mútua entre X i Y és la reducció en la incertesa que tenim sobre X quan sabem el valor que ha pres Y (es podria dir el mateix si intercanviéssim X i Y).

3. Secret perfecte i autenticitat perfecta

El *secret* i l'*autenticitat* són conceptes diferents, i és per això que els tractarem de manera independent.

3.1. Secret perfecte

Shannon va mesurar el **secret teòric d'una xifra** com la incertesa sobre el text en clar un cop s'ha interceptat el text xifrat corresponent. Si, amb independència de la quantitat de text xifrat interceptat, no es pot saber res del text en clar, llavors la xifra ofereix un secret perfecte. La definició següent formalitza en termes d'entropies això que hem dit.

Si M és un text en clar i C és el text xifrat corresponent obtingut amb un cert criptosistema, diem que el criptosistema proporciona un **secret perfecte** si $H(M|C) = H(M)$; de manera equivalent $I(M,C) = 0$. Intuïtivament, la incertesa sobre el valor del text en clar M quan el criptoanalista veu el text xifrat C és la mateixa que tindria si no l'hagués vist.

El problema del secret perfecte, com en la majoria de coses perfectes, és que és difícil d'aconseguir. Al seu article de 1949, Shannon va fer servir les propietats de l'entropia per a demostrar que si M, C, K són el text en clar, el text xifrat i la clau, respectivament, llavors tenim:

$$H(M|C) \leq H(M,K|C) = H(K|C) + H(M|C,K) = H(K|C) \leq H(K). \qquad (3.1)$$

En la desigualtat anterior s'ha fet servir que $H(M|C,K) = 0$, és a dir, que no hi ha incertesa sobre el text en clar un cop coneguts el text xifrat corresponent i la clau emprada.

Combinant la definició de *secret perfecte* i la desigualtat (3.1) obtenim com a resultat el **teorema de la fita fonamental de Shannon per al secret perfecte**, que diu el següent: en un criptosistema perfectament secret, la incertesa de la clau secreta K ha de ser almenys tan gran com la incertesa del text en clar que pretén amagar. Formalment podem expressar la fita de Shannon de la manera següent:

$$H(M) \leq H(K). \qquad (3.2)$$

Aquesta fita fonamental té conseqüències "tràgiques" pel que fa a la longitud de clau necessària per a mantenir el secret perfecte. En efecte, hem vist anteriorment que un missatge X amb incertesa $H(X)$ requereix com a mínim $H(X)$ bits de mitjana per a codificar-ne els valors possibles.

En resum, la fita de Shannon ens diu que, en un criptosistema perfectament secret, la clau K requereix més bits que el text en clar M, és a dir, la clau secreta ha de ser més llarga que no el text que es vol xifrar.

La **xifra de Vernam** és l'única coneguda que ofereix un secret perfecte. Recordem que:

- La xifra de Vernam consisteix a sumar una clau K al text en clar M per a obtenir el text xifrat C.
- M, C i K prenen valors a $\{0, 1\}$ i la suma és mòdul 2, és a dir, $C = M \oplus K$.
- La clau és aleatòria i només es fa servir una vegada, és a dir, cada bit de text en clar es xifra amb un nou bit de clau elegit a l'atzar.

La darrera condició implica que la clau K és almenys tan llarga com el text en clar, és a dir, $|K| \geq |M|$. Com que K és aleatòria (cada bit val 0 o 1 amb probabilitat 1/2), resulta que $H(K) = |K|$. D'altra banda, $H(M) \leq |M|$ (la igualtat només es produeix si M és un missatge aleatori). Per tant:

$$H(M) \leq |M| \leq |K| = H(K),$$

i es compleix la condició de secret perfecte.

El criptosistema de Vernam pot semblar inútil a primera vista: es pot pensar que, posats a fer arribar al receptor de manera segura una clau secreta d'un sol ús més llarga que el missatge en clar, tant se valdria enviar directament el missatge en clar pel mateix canal segur per on es vol enviar la clau. Això no obstant, cal tenir en compte que si la clau es fa arribar al receptor en circumstàncies més favorables que no les que tindrem a l'hora d'enviar el missatge xifrat, té sentit passar una clau secreta més llarga que el mateix missatge en clar; per exemple, una bona estratègia és passar grans longituds de clau abans d'un conflicte armat per a poder enviar missatges xifrats durant el conflicte.

One-time pad

De fet, en anglès la xifra de Vernam es coneix també com a *one-time pad*, perquè es va fer servir poc abans, durant i després de la Segona Guerra Mundial per part d'espies de diferents països que rebien un full de paper (*pad*) amb la clau secreta aleatòria i la instrucció de fer-la servir només per a un sol xifratge (*one-time*).

3.2. Autenticitat perfecta

La criptografia pretén assegurar el secret i l'autenticitat dels missatges. Però, de fet, només molt recentment hom s'ha adonat que secret i autenticitat són

atributs independents. Si l'Anna comparteix una clau amb en Bernat i l'Anna rep un criptograma que es desxifra bé amb la clau compartida, pot estar segura que el criptograma va ser enviat per en Bernat? La resposta és no. G.J. Simmons va publicar el 1984 l'article "Authentication theory/coding theory", que presenta una teoria de l'autenticitat anàloga en molts aspectes a la teoria del secret publicada per Shannon el 1949.

Com Shannon, Simmons suposa que la clau de xifratge es fa servir un sol cop. A l'escenari suposat per Simmons el criptoanalista enemic es troba entre l'emissor i el receptor i genera un criptograma fraudulent $\widetilde{C}$. Hi ha dos tipus d'atac:

1) Suplantació: el criptoanalista genera un criptograma fraudulent $\widetilde{C}$. L'atac té èxit si el receptor accepta $\widetilde{C}$ com a bo. Anomenem P_{sup} la probabilitat d'èxit d'un atac de suplantació.

2) Substitució: el criptoanalista canvia un criptograma autèntic C enviat per l'emissor per un criptograma fraudulent $\widetilde{C}$. Aquesta mena d'atac té èxit si el receptor accepta $\widetilde{C}$ com a bo i $\widetilde{C} \neq C$. Anomenem P_{sub} la probabilitat d'èxit d'un atac de substitució.

Suposant que el criptoanalista enemic triarà el tipus d'atac que li sigui més favorable, es defineix la **probabilitat d'engany**, P_e, com:

$$P_e = \max(P_{sub}, P_{sup}).$$

Anomenem $\#C$ el nombre de criptogrames c que tenen una probabilitat no nul·la d'aparèixer, és a dir, tals que $P(C = c) \neq 0$; de manera anàloga, anomenem $\#M$ i $\#K$ el nombre de missatges en clar i de claus, respectivament, amb una probabilitat no nul·la d'aparèixer. Llavors per a cada clau k hi ha d'haver $\#M$ criptogrames amb probabilitat no nul·la: són els criptogrames resultants de xifrar amb k els $\#M$ missatges en clar que tenen una probabilitat no nul·la.

Per tant, si el criptoanalista enemic fa un atac de suplantació triant a l'atzar un dels $\#C$ criptogrames que tenen una probabilitat no nul·la, la seva probabilitat d'èxit serà:

$$P_{sup} \geq \frac{\#M}{\#C}. \tag{3.3}$$

Com que $P_e \geq P_{sup}$, la desigualtat (3.3) mostra que no és possible aconseguir una protecció total contra l'engany. Una bona manera de lluitar contra l'engany és que el conjunt de criptogrames possibles sigui molt més gran que el conjunt de textos en clar possibles, és a dir, agafar $\#C \gg \#M$.

Pel fet que la protecció total és impossible, Simmons va definir l'**autenticitat perfecta** com la màxima protecció possible contra l'engany. Cal admetre que

aquesta definició presenta alguns "casos patològics", ja que quan $\#M = \#C$ haurem de considerar perfectament autèntic un sistema amb $P_e = 1$, perquè la desigualtat (3.3) indica que, per força, $P_{sup} = 1$.

La definició d'autenticitat perfecta es fa en termes d'informació mútua. Per a això necessitem conèixer el **teorema de la fita de Simmons**, que diu: si P_{sup} és la probabilitat de suplantació amb èxit, es compleix:

$$P_{sup} \geq 2^{-I(C,K)}. \tag{3.4}$$

Paradoxalment, resulta que per a reduir la fita (3.4) cal que $I(C,K)$ sigui gran, és a dir, que un criptograma doni molta informació sobre la clau emprada. Això té sentit, perquè que $I(C,K)$ sigui gran vol dir que el criptograma C difícilment pot ser produït per algú que no sàpiga la clau K; en altres paraules, per algú diferent de l'emissor legítim. Clarament, secret i autenticitat són, com a mínim, independents.

Lectura complementària

Podeu trobar la demostració del teorema de la fita de Simmons a l'article de J. Massey "An Introduction to Contemporary Cryptology", en l'obra següent:

G.J. Simmons (1992). *Contemporary Cryptology: The Science of Information Integrity.* Nova York: IEEE Press.

De fet, reduir la fita (3.3) fent créixer $\#C$ en relació amb $\#M$ implica també reduir la fita (3.4). En efecte, si $\#M \lll \#C$, vol dir que veure un dels $\#M$ criptogrames vàlids dóna força informació sobre la clau emprada per a produir-lo; per tant, $I(C,K)$ és gran i la fita (3.4) és petita. Per a entendre aquest raonament , pensem en el cas oposat: si $\#M = \#C$, llavors un criptograma pot aparèixer amb qualsevol clau (sigui quina sigui la clau triada, hi ha d'haver $\#M = \#C$ criptogrames vàlids); per tant, veure un criptograma no diu res sobre la clau emprada, amb la qual cosa $I(C,K) = 0$ i la fita (3.4) és màxima ($P_e = P_{sup} = 1$).

Així doncs, un criptosistema té la propietat d'autenticitat perfecta si P_e pren el mínim valor possible, és a dir, si $P_e = 2^{-I(C,K)}$.

3.3. Exemples d'independència entre el secret i l'autenticitat

En aquest subapartat recollim cinc exemples proposats per J. Massey per a il·lustrar l'afirmació que secret i autenticitat són dues propietats independents d'un criptosistema. En els criptosistemes dels exemples, el text en clar és sempre un sol dígit binari M, el criptograma consisteix en dos dígits binaris $C = [C_1, C_2]$ i la clau K és aleatòria i consisteix en un o dos bits. En ser aleatòria la clau, la seva entropia $H(K)$ és igual a la seva llargada.

3.3.1. Criptosistema no secret ni autèntic

Considerem $M = \{0, 1\}$, $K = \{0, 1\}$ i $C = \{00, 01, 10, 11\}$. La transformació de xifratge s'indica a la taula següent:

K	$M = 0$	$M = 1$
0	$C = 00$	$C = 10$
1	$C = 01$	$C = 11$

En aquest cas, clarament no hi ha secret, atès que el primer bit del text xifrat és igual al text en clar.

Analitzem-ne l'autenticitat:

1) La probabilitat de suplantació amb èxit és $P_{sup} = 1/2$, perquè només dos dels quatre criptogrames seran acceptats pel receptor (que sap la clau) i perquè no hi ha cap criptograma que valgui amb totes dues claus.

2) Si el criptoanalista enemic veu un criptograma, sap que pot invertir-ne el primer bit i que el criptograma alterat serà acceptat sigui quina sigui la clau. En efecte, si $K = 0$, valen $C = 00$ i $C = 10$; si $K = 1$, valen $C = 01$ i $C = 11$. Per tant, la probabilitat de tenir èxit en una substitució és $P_{sub} = 1$, amb la qual cosa $P_e = 1$.

3) Calculem la fita de Simmons. Tenim $H(K) = 1$ bit i, d'altra banda, $H(K|C) = 0$, atès que la clau queda determinada pel segon bit del criptograma. Per tant, $I(C,K) = H(K) - H(K|C) = 1$. La fita és $2^{-I(C,K)} = 2^{-1} < P_e$. Així, doncs, el criptosistema no proporciona una autenticitat perfecta.

En aquest cas, l'atac de substitució és més perillos que l'atac de suplantació.

3.3.2. Criptosistema no secret i trivialment autèntic

Considerem $M = \{0, 1\}$, $K = \{0, 1\}$ i $C = \{00, 01, 10, 11\}$. Introduïm un paràmetre R aleatori per tal d'aleatoritzar el xifratge. La transformació de xifratge és determinada per la taula següent:

K	R	$M = 0$	$M = 1$
0	0	$C = 00$	$C = 10$
0	1	$C = 01$	$C = 11$
1	0	$C = 00$	$C = 11$
1	1	$C = 01$	$C = 10$

En aquest cas tampoc no hi ha secret, atès que el primer bit del text xifrat és igual al text en clar.

Analitzem-ne l'autenticitat:

1) La probabilitat de suplantació amb èxit és $P_{sup} = 1$, perquè tots quatre criptogrames són vàlids sigui quin sigui el valor de la clau (el receptor els acceptarà com a bons).

2) Si el criptoanalista enemic veu un criptograma, es troba davant de dues alternatives equiprobables a l'hora d'intentar substituir-lo. Per exemple, no sap si canviar $C = 00$ per $C = 10$ o bé per $C = 11$ (li caldria saber la clau secreta per a decidir-ho amb seguretat). Per tant, la probabilitat de substitució amb èxit és $P_{sub} = 1/2$ i llavors $P_e = \max(P_{sup},P_{sub}) = 1$.

3) Calculem la fita de Simmons. Tenim $H(K) = 1$ bit i, d'altra banda, $H(K|C) = 1$, atès que per a qualsevol text xifrat els dos valors de la clau són equiprobables. Per tant, $I(C,K) = H(K) - H(K|C) = 0$. La fita és $2^{-I(C,K)} = 1 = P_e$. Així, doncs, el criptosistema proporciona una autenticitat perfecta de manera trivial; no obstant això, l'autenticitat perfecta no té gaire gràcia si la probabilitat d'engany és 1.

De fet, aquest exemple i l'anterior mostren que l'atac de substitució pot ser més perillós que el de suplantació o a l'inrevés, en funció de cada cas particular.

3.3.3. Criptosistema no secret i perfectament autèntic

Considerem el mateix esquema que presentàvem al subapartat anterior, llevat que ara K i R són els dos bits de la clau (K_1 i K_2) i, per tant, tots dos seran coneguts pel receptor legítim. Tenim, doncs, que la taula de xifratge és la següent:

K_1	K_2	$M = 0$	$M = 1$
0	0	$C = 00$	$C = 10$
0	1	$C = 01$	$C = 11$
1	0	$C = 00$	$C = 11$
1	1	$C = 01$	$C = 10$

En aquest cas continua sense haver-hi secret, atès que el primer bit del text xifrat és igual al text en clar.

Analitzem-ne l'autenticitat:

1) Per a cadascun dels quatre valors de la clau només hi ha dos criptogrames vàlids i cada criptograma només és vàlid amb dos dels quatre valors de la clau.

Per tant, la probabilitat que un criptograma triat a l'atzar sigui vàlid és 1/2 i llavors $P_{sup} = 1/2$.

2) Si el criptoanalista enemic veu un criptograma, es troba davant de dues alternatives equiprobables a l'hora d'intentar substituir-lo. Per exemple, no sap si canviar $C = 00$ per $C = 10$ o bé per $C = 11$; li caldria saber la clau secreta per a decidir-ho amb seguretat. Per tant, la probabilitat de substitució amb èxit és $P_{sub} = 1/2$ i llavors $P_e = \max(P_{sup},P_{sub}) = 1/2$.

3) Calculem la fita de Simmons. Tenim $H(K) = 2$ bits i, d'altra banda, $H(K|C) = 1$ bit, atès que en veure un text xifrat el criptoanalista dubta només entre dues de les quatre claus (per exemple, $C = 00$ només pot aparèixer amb les claus $K = 00$ i $K = 10$). Per tant, $I(C,K) = H(K) - H(K|C) = 1$. La fita és $2^{-I(C,K)} = 1/2 = P_e$. Així, doncs, el criptosistema proporciona una autenticitat perfecta aconseguida d'una manera no trivial.

3.3.4. Criptosistema perfectament secret i no autèntic

Considerem l'esquema de xifratge següent:

K_1	K_2	$M = 0$	$M = 1$
0	0	$C = 00$	$C = 11$
0	1	$C = 01$	$C = 10$
1	0	$C = 10$	$C = 01$
1	1	$C = 11$	$C = 00$

Si analitzem el secret podem veure que cada missatge en clar possible pot ser xifrat amb la mateixa probabilitat com un dels quatre criptogrames possibles (depenent de la clau); és a dir, $P(C = c|M = m) = 1/4$ per a qualsevol parella (m,c). Una altra manera de dir-ho és que en veure un criptograma, no tenim cap pista sobre quin és el missatge en clar. Per tant, hi ha un secret perfecte.

Analitzem-ne l'autenticitat:

1) Per a cadascun dels quatre valors de la clau només hi ha dos criptogrames vàlids i cada criptograma només és vàlid amb dos dels quatre valors de la clau. Per tant, la probabilitat que un criptograma triat a l'atzar sigui vàlid és 1/2 i llavors $P_{sup} = 1/2$.

2) Si el criptoanalista enemic veu un criptograma, sap que pot invertir-ne els dos bits i que el criptograma resultant serà acceptat com a vàlid pel receptor. Per tant, $P_{sub} = 1$ i llavors $P_e = \max(P_{sup},P_{sub}) = 1$.

3) Calculem la fita de Simmons. Tenim $H(K) = 2$ bits i, d'altra banda, $H(K|C) = 1$ bit, atès que en veure un text xifrat el criptoanalista dubta només entre dues de les quatre claus (per exemple, $C = 00$ només pot aparèixer amb les claus $K = 00$ i $K = 11$). Per tant, $I(C,K) = H(K) - H(K|C) = 1$. La fita és $2^{-I(C,K)} = 1/2 < P_e$. Així, doncs, no hi ha una autenticitat perfecta.

3.3.5. Criptosistema perfectament secret i perfectament autèntic

Considerem l'esquema de xifratge:

K_1	K_2	$M = 0$	$M = 1$
0	0	$C = 00$	$C = 10$
0	1	$C = 01$	$C = 00$
1	0	$C = 11$	$C = 01$
1	1	$C = 10$	$C = 11$

Analitzem-ne el secret. Cada possible missatge en clar pot ser xifrat amb la mateixa probabilitat com un dels quatre criptogrames possibles (segons la clau); és a dir, $P(C = c|M = m) = 1/4$ per a qualsevol parella (m,c). Una altra manera de dir-ho és que, en veure un criptograma, no tenim cap pista sobre quin és el missatge en clar. Per tant, hi ha un secret perfecte.

Analitzem-ne l'autenticitat:

1) Per a cadascun dels quatre valors de la clau només hi ha dos criptogrames vàlids i cada criptograma només és vàlid amb dos dels quatre valors de la clau. Per tant, la probabilitat que un criptograma triat a l'atzar sigui vàlid és 1/2 i llavors $P_{sup} = 1/2$.

2) Si el criptoanalista enemic veu un criptograma, es troba davant de dues alternatives equiprobables a l'hora d'intentar substituir-lo. Per exemple, no sap si canviar $C = 00$ per $C = 10$ o bé per $C = 01$; li caldria saber la clau secreta per a decidir-ho amb seguretat. Per tant, la probabilitat de substitució amb èxit és $P_{sub} = 1/2$ i llavors $P_e = \max(P_{sup},P_{sub}) = 1/2$.

3) Calculem la fita de Simmons. Tenim $H(K) = 2$ bits i, d'altra banda, $H(K|C) = 1$ bit, atès que en veure un text xifrat el criptoanalista dubta només entre dues de les quatre claus (per exemple, $C = 00$ només pot aparèixer amb les claus $K = 00$ i $K = 01$). Per tant, $I(C,K) = H(K) - H(K|C) = 1$. La fita és $2^{-I(C,K)} = 1/2 = P_e$. Així, doncs, hi ha una autenticitat perfecta aconseguida d'una manera no trivial.

4. Criptoanàlisi elemental

En aquest apartat revisarem dos conceptes bàsics en la criptoanàlisi: la suposició de Kerckhoff i la distància d'unicitat. D'altra banda veurem un exemple pràctic de criptoanàlisi trencant una xifra de Vigenère amb la criptoanàlisi de freqüències.

4.1. Suposició de Kerckhoff

Una suposició quasi universal en criptografia és que el criptoanalista enemic té accés al criptograma. Gairebé tan universal és la suposició de Kerckhoff, formulada per l'holandès A. Kerckhoff (1835-1903), segons la qual la seguretat de la xifra ha de residir totalment en la clau secreta.

La **suposició de Kerckhoff** diu que tot el mecanisme de xifratge, excepte el valor de la clau secreta, és conegut pel criptoanalista enemic.

No deixa de sorprendre que la suposició de Kerckhoff sigui sovint ignorada en criptografia militar i fins i tot en aplicacions civils com la telefonia mòbil. Els dissenyadors de criptosistemes cauen fàcilment en la temptació de pensar que si el mecanisme de xifratge es manté secret, la seguretat de la xifra és més alta. Això no és necessàriament veritat per les raons següents:

1) Un mecanisme de xifratge públic pot ser sotmès a l'examen de tota la comunitat científica. D'aquesta manera, se'n pot comprovar la fortalesa o la feblesa. Un mecanisme de xifratge secret només ha passat l'examen d'un grup reduït de criptògrafs que, per la seva mateixa contribució al disseny, corren el risc de passar per alt eventuals deficiències de seguretat.

2) A la llarga, és molt difícil de mantenir en secret un mecanisme de xifratge, fins i tot si els dissenyadors signen un contracte de no-revelació. Com a exemple, actualment es coneixen gairebé del tot els mecanismes de seguretat usats en telefonia mòbil GSM, que en teoria haurien de ser secrets. Per tant, és més realista suposar, com fa Kerckhoff, que l'única cosa secreta és la clau.

Secret de dos

La saviesa popular diu: "Secret de dos és dubtós, secret de tres no val res". Això pot explicar la dificultat de mantenir en secret un mecanisme de xifratge dissenyat per un grup de criptògrafs.

4.2. Redundància i distància d'unicitat

De la suposició de Kerckhoff i de la desigualtat (3.1) es dedueix que pot emprar-se $H(K|C)$ per a mesurar el secret d'una xifra. La magnitud $H(K|C)$ s'anomena **ambigüitat de clau** i representa la incertesa que queda sobre la clau un cop es veu un criptograma. Si $H(K|C) = 0$, no hi ha incertesa i la xifra es pot trencar

en teoria si disposem de prou recursos de càlcul. Normalment, $H(K|C)$ decreix a mesura que augmenta la longitud N del criptograma conegut C.

La **distància d'unicitat d'una xifra** és la menor longitud N de criptograma que fa que $H(K|C)$ sigui propera a zero. Altrament dit, és la quantitat de text xifrat que cal perquè la clau quedi determinada de manera única.

Shannon va anomenar *idealment secrets* aquells criptosistemes que, tot i no proporcionar un secret perfecte, són irrompibles perquè no donen prou informació per a determinar-ne la clau.

La majoria de criptosistemes són massa complexos per a trobar-ne la distància d'unicitat. Shannon va formular un model anomenat de xifra aleatòria que permet aproximar la distància d'unicitat en alguns casos. Vegem com funciona el model, però per a fer-ho necessitem algunes definicions prèvies.

Per a un llenguatge determinat, considerem el conjunt de tots els missatges de llargada N caràcters. La **taxa del llenguatge per a missatges X de longitud N**, r, es defineix com:

$$r = H(X)/N,$$

és a dir, el nombre mitjà de bits d'informació per caràcter.

Segons diversos estudis, la taxa de l'anglès per a N gran és entre 1,0 i 1,5 bits/lletra.

La **taxa absoluta d'un llenguatge**, R, es defineix com el nombre màxim de bits d'informació que poden ser codificats en cada caràcter suposant que totes les seqüències de caràcters són equiprobables. Si hi ha L caràcters al llenguatge, la taxa absoluta és:

$$R = \log_2 L.$$

La taxa absoluta de l'anglès (i del català) és de $R = \log_2 26 = 4{,}7$ bits/lletra.

La **redundància d'un llenguatge** amb taxa r i taxa absoluta R es defineix com $D = R - r$.

Per a l'anglès,...

... $D = 4{,}7 - 1{,}5 = 3{,}2$ bits/lletra. Del quocient D/R es veu que l'anglès és aproximadament redundant en un 79%. Això vol dir que si xifrem un text en clar en anglès, la distància d'unicitat no pot ser gaire gran.

Suposem que cada text en clar i cada missatge xifrat vénen d'un alfabet finit de L símbols. Llavors hi ha 2^{RN} possibles missatges de longitud N, on $R = \log_2 L$, que es poden dividir en un conjunt de 2^{rN} **missatges amb sentit** i un conjunt de $2^{RN} - 2^{rN}$ **missatges sense sentit**, on r és la taxa del llenguatge.

Se suposa que tots els missatges amb sentit tenen la mateixa probabilitat d'aparició $1/2^{rN} = 2^{-rN}$, mentre que els missatges sense sentit tenen probabilitat

zero. Suposarem també que hi ha $2^{H(K)}$ claus, totes equiprobables, on $H(K)$ és l'entropia de la clau (nombre de bits de la clau). La probabilitat de totes les claus k és:

$$P(K = k) = 1/2^{H(K)} = 2^{-H(K)}.$$

Una **xifra aleatòria** és aquella en la qual, per a cada clau k i per a cada text xifrat c, el desxifratge $D_{k(c)}$ és vist pel criptoanalista com una variable aleatòria independent (d'altres $D_{k'}(c')$) i distribuïda uniformement sobre tots els 2^{RN} missatges, amb sentit o sense.

Considerem el text xifrat $c = E_{k(m)}$ per k i m donats. Hi ha un **desxifratge espuri** quan el xifratge sota una altra clau k' pot donar c; és a dir, $c = E_{k'}(m)$ per al mateix missatge m, o bé $c = E_{k'}(m')$ per a un altre missatge amb sentit m'. Un criptoanalista que intercepti c no podrà decidir si la clau correcta és k o bé k'.

Ara bé, per cada desxifratge correcte d'un text xifrat determinat hi ha $2^{H(K)} - 1$ claus restants, cadascuna de les quals té la mateixa probabilitat, q, de produir un desxifratge espuri, que és el nombre de missatges amb sentit dividit pel nombre de missatges possibles:

$$q = 2^{rN}/2^{RN} = 2^{(r-R)N} = 2^{-DN}.$$

Si denotem per F el nombre esperat de desxifratges espuris utilitzant una clau d'entre les restants, tenim:

$$F = (2^{H(K)} - 1) \cdot q = (2^{H(K)} - 1) \cdot 2^{-DN} \approx 2^{H(K)-DN}.$$

A causa del decreixement ràpid que experimenta F quan N creix, es pren $\log_2 F = H(K) - DN = 0$ com el punt on el nombre de solucions falses és prou petit perquè la xifra es pugui trencar. Per tant, la **distància d'unicitat**, N, és a dir, la quantitat de text necessària per a trencar la xifra és la següent:

$$N = \frac{H(K)}{D}.$$

Si el criptoanalista disposa de N caràcters i d'una capacitat de càlcul il·limitada, llavors podrà trobar l'única clau que pot produir els N caràcters xifrats. Noteu que si el text en clar no tingués redundància (és a dir, si consistís en una seqüència aleatòria de bits), la distància d'unicitat fóra infinita.

En la **xifra de Vernam**, per cada N el nombre de claus possibles és tan gran com el nombre de missatges possibles. Llavors, $H(K) = \log_2 (2^{RN}) = RN$ i tenim:

$$H(K) - DN = (R - D)\, N = rN \neq 0,$$

cosa que implica que la xifra de Vernam és irrompible en teoria, com ja haviem esmentat anteriorment.

Seguretat de l'algorisme DES

L'algorisme DES xifra blocs de 64 bits (8 caràcters) i fa servir claus de 56 bits. Aquest algorisme encaixa raonablement amb el model de xifra aleatòria. Si es fa servir per a xifrar un text en clar en anglès, llavors $H(K) = 56$ i $D = 3{,}2$, amb la qual cosa la distància d'unicitat en caràcters és:

$$N = \frac{56}{3{,}2} = 17{,}5.$$

Per tant, en teoria n'hi ha prou amb una mica més de dos blocs de text xifrat perquè la clau DES emprada quedi determinada únicament. Ara bé, que quedi determinada únicament no vol pas dir que al criptoanalista li sigui fàcil trobar-la! A tall d'analogia, podem saber que un atracament ha estat comès per una sola persona, però d'això a identificar-la hi ha molta feina.

Vegeu l'algorisme DES al subapartat 2.1 del mòdul " Xifres de clau compartida: xifres de blocs" d'aquesta assignatura.

4.3. La criptoanàlisi de Kasiski per al xifratge de Vigenère

L'any 1863, l'oficial prussià W.F. Kasiski aconseguí trencar la xifra de Vigenère a partir de l'anàlisi de les repeticions de grups de símbols en el text xifrat i l'anàlisi de freqüències.

Vegeu la descripció de la xifra de Viginière en el subapartat 1.2.3 d'aquest mòdul.

Una criptoanàlisi de Kasiski s'estableix en dues etapes. En la primera etapa se suposa la longitud de la clau, i en la segona se'n calcula el valor exacte.

Calcularem la longitud de la clau de xifratge del criptograma per les repeticions dels grups de caràcters. Per tal d'entendre'n el procés, començarem al revés. Suposarem que la longitud de la clau que volem trobar és 5. Aleshores, escrivim el criptograma format per una seqüència d'm caràcters $C_1C_2...C_{m-1}C_m$ en cinc columnes:

C_1	C_2	C_3	C_4	C_5
C_6	C_7	C_8	C_9	C_{10}
...	...	...	...	...
C_{m-4}	C_{m-3}	C_{m-2}	C_{m-1}	C_m

El mètode de xifratge de Vigenère amb una clau de 5 caràcters xifra cada columna amb el mateix element de la clau. És a dir, els caràcters de cada columna corresponen a una xifra de substitució simple que utilitza el caràcter i-èsim de la clau. Per tant, dos o més caràcters iguals d'una columna provindran de caràcters iguals de text en clar. Com que tots els idiomes presenten una alta redundància, possiblement hi haurà conjunts de caràcters característics (per exemple, en anglès els conjunts *the*, *ing*, *ght*; en castellà: *ando*, *ado*, *ción*; en català: *ant*, *ada*, *ció*) que quedaran xifrats amb la mateixa porció de la clau i donaran lloc a cadenes de text xifrat repetides en el criptograma.

Vegeu les xifres de substitució simple en el subapartat 1.2.1 del mòdul didàctic "Fonaments de criptografia" d'aquesta assignatura.

La probabilitat que aquestes repeticions de cadenes es donin de manera aleatòria és més baixa com més llarga sigui la longitud de la cadena que es repeteix. De fet, conjunts de tres o quatre caràcters repetits més d'una vegada indiquen una alta probabilitat que la distància entre les cadenes sigui un múltiple de la clau utilitzada per xifrar. Per exemple, considerem el missatge:

$$M = \text{TOBEORNOTTOBETHATISTHE...}$$

Si xifrem aquest missatge amb la clau K = HAM, el criptograma resultant serà:

$$C = \textbf{AONL}\text{ODUOF}\textbf{AONL}\text{TTHTTZTTL...}$$

Podem observar que la distància entre les dues seqüències de caràcters AONL és igual a 9, la qual cosa indica que 9 és un múltiple de la longitud de la clau: és a dir, serà 3 o 9. A més, la seqüència TT està separada per 6 espais, cosa que confirma que, efectivament, la longitud de la clau és 3. És a dir, el màxim comú divisor de les distàncies entre conjunts de caràcters iguals ha de ser un múltiple de la longitud de la clau.

Fins aquí hem vist la part de la criptoanàlisi que ens permet obtenir la longitud de la clau a partir del text xifrat. A continuació veurem com podem obtenir el valor concret de la clau, un cop n'hem obtingut la longitud.

Per entendre'n el funcionament, farem la criptoanàlisi d'un text d'exemple. Partirem del següent text xifrat de 1.340 caràcters, corresponent a un text en clar en català:

Simplificacions

... Hem eliminat del text en clar els accents, els signes de puntuació i els espais perquè l'anàlisi sigui més simple.

UUURABMNUPSZIJPPIJPLAXAPPENUUAHEIHEGYOOEUDHEPANMT
PDCDSENJAAXTMDNSDAGFAHTUNROHISÇMTMSBAXLAXPUXSAXA
KVENXAZISZDLCZNKVOBISNSRLZEQDAJQPGMRGISPPTDQEN
DCOI**SZZN**UHOIDDAPAIITANALZEVZIZDLAXENFOHERUVIAXHDLAAP
SJPQPICUZPGSMUYDUQUIYEGXEZMFDFINSNISHDLAUPUHREN
DLUKENTALZENLAZIRAKAMQENUUAQADTEMUUAPIGPAAXDAXT
DZSZZRNUUAHEQSRAREGAEMHRAFOMVEMZNXDMKZSPRIQIRN
MTUVIUJIIDLNHEEZLDSLEZSODBURSLZEXSMARCDPACSSOTRJHUAM
XPRANINNDCDSCPVIJXAAXUIDBUVRANAZMNXVEZZLD
YAOMDAHENDSNSSNICXSMNMEIXTJFANVEXSRMIRPRPUMSUYGA
QOGYCJRECZTUPQPDLHDNLZEIILAQEIYSDRDDXPARSUELAXUIDPG-
DTEDDUVEIDBGDNXDPAVERIMKPEUQBJQBMI**LGIS**DYOQDLGSLAXA
GYEMVAUQBCINOUUATRAREGXOGMBPXCUZNÇSRDYZJUUAROA
BINYEDBPAVQPINJLIÇDMUVUIDUIMVAVSDYAOXEIXEUQBDINOZN
DAEMXIODRDISXSMPRAKPAONANINNIMUVUIDBNZRZINXDRUROB
DGUMRAXDDISGDGAXPUIRUTLARAZICJXSJXQPISUEOXDVARAGXAK
ZNOXLGIGDYSZIPMISNDMDVAZISLZEXIRXDVARLAXLGITMI**SZZN**
GPIVVEJPAHMRUHAXSMKPIXIDUPGPRCJQPURYZMUARQPIAGDUI
MVAVSDYAO**UUURA**MVIVDEGEOIYEHTSAPSAXTPHIURTNXEI**D**

MJVEIXOI**DMJV**SVVEPXIBZGDXSAVSLZEISDPVEIKADVENSNKDRARTA
XINUUAILNTEMQEOINMISKMRUVEIYRAIXUQEIMERDMARQPDNAP
SZMENISBDNBIIRZCNMEGXPMSFAXSJVSNIGPVAHINOINNJED
QIIXUKSRODBGISUVAAPSDPEIFIQDGPDNSDNOYEMVEICCJRQPIRA
MXXDDUDUGDEGXPUXSUHINXONILVDRDRUNZAGQEIYBPMTGISKD
NOD**LGIS**UTACDDAXDAPSJVDDRAZSRNDLNHENTAOBONILNTR
JJENXOMXEIFAMDHDVEXZLGINGISPPTDQENTEMYIIINXISGISNICMI
TUVIAXBPXQPINAPRAJUCMDAPADVEXSNZMCDSNUYIAPSJPSAKUA
MXUAAIGAIYUITAHFAZDMUYI

Hem destacat en negreta algunes de les cadenes llargues que es repeteixen:

- dues cadenes UUURA separades per 882 caràcters
- dues cadenes SZZN separades per 144 caràcters
- dues cadenes LGIS separades per 609 caràcters
- dues cadenes DMJV separades per 9 caràcters

El màxim comú divisor de totes aquestes distàncies hauria de ser un múltiple de la longitud de la clau. Així, en aquest cas: mcd(882,144,609,9) = 3 i, per tant, la clau podria tenir longitud 3.

Cal anar amb compte de no escollir cadenes massa curtes, ja que aquestes es podrien repetir de manera aleatòria. Per exemple, si en el criptograma anterior ens haguéssim fixat en les cadenes JP i DD, que hem subratllat, la primera es repeteix amb una separació de 2 i la segona, amb una distància de 355. Si haguéssim fet els càlculs del màxim comú divisor amb aquestes distàncies, clarament no hauríem obtingut el valor 3 que ens ha sortit amb les altres cadenes de quatre caràcters.

Per trobar el valor exacte de la clau, farem les manipulacions següents:

1) Una vegada calculada la longitud de la clau, que en el nostre exemple val 3, dividirem el text xifrat en tants trossos com longitud té la clau, en aquest cas 3.

2) Anomenarem *subcriptogrames* les cadenes C_A, C_B i C_C resultants. Calcularem aquests subcriptogrames (que no tenen per què tenir la mateixa longitud) de la manera següent: situarem el text xifrat en files de longitud igual a la de la clau:

C_1	C_2	C_3
C_4	C_5	C_6
C_7	C_8	C_9
C_{10}	C_{11}	C_{12}
...	...	...
$C_{1.339}$	$C_{1.340}$	

3) A continuació, prendrem com a subcriptogrames cada una de les columnes resultants, és a dir:

$$C_A = C_1\ C_4\ C_7\ \ldots\ C_{1.336}\ C_{1.339}$$
$$C_B = C_2\ C_5\ C_8\ \ldots\ C_{1.337}\ C_{1.340}$$
$$C_C = C_3\ C_6\ C_9\ \ldots\ C_{1.338}$$

En el nostre exemple, el subcriptogrames quedaran constituïts pels següents caràcters:

C_A = URMPIPPXPUHHYEHPMDSJXDDFTRIMSXXXXVXIDZVISZDQMI
PQDIZHDPINZZDXFEVXLPPIZSYQYXMFSSLPRDKTZLIKQUQTUPPXXZ
ZUHSRAHFVZDZRIMVJDHZSZDRZSRPSTHMRIDSVXXDVNMVZYMHD
SISMXFVSIRMYQYRZPDDZIQYRXREXDDDVDDDVIPQQII
YDSXYVQIUTRXMXZSYURBYBVILDVDMVYXXQIZAXDISRPNII
VDZIDRDMXIDXITRIXXIEDRXZXIYIIDVIZIDRXIIZPVPMHSPIPRQR
MRIDMVYURVDEYTPXHRXDVXDVVXZXVZSVKVSDRXUITQII
MVYIQMDRDPMIDIZMXSXVIVIIJQXSDIVPPFDDDYVCRIMDDDXXH
XIDRZQYMIDDITDXPVRSDHTBITJXXFDVZIIPQTYIIIIIVXXIPJM
PVSMSYPPKMAGYTFDY

C_B = UANSJILAEUEEOUEATCEATNAAUOSTBLPSAEASLNOSREAPRSTECS
NODATAEILEORIHASQCPMDUEEFINHAUELEAEARAEUAEUIADTS
RUEREEROENMSIRTIILELLSBSEMCASRUXANCCIAUBRANELADESS
CMETAERRPSGOCETQLNELESDPSLUPTDEBNPEMEBBLSOLLAEABNU
REOBCNRZUOIEPQNIMUUVSAEEBNNEIRSMAAANMUBRNROGRDS
GPRLACSQSOVAANLGSPSMASERVLLTSNIEARAMIDGCPYUQAUVS
AUAIEOESSTITEMEOMSEIGSSEDEAENRTIULEENSRERXEEMQNSES
NICEPFSSGANNEIURBSASEIGNNEECQRXDUEPSIOLRUAEBTSNLSAD
DSDARLEAOLREOEAHELNSTEEINSSCTIBQNRUDAENCNISSUXA
AUAAMI

C_C = UBUZPJAPNAIGODNNPDNAMSGHNHÇMAAUAKNZZCKBNL
QJGGPDNOZUIAIALVZANHUADAJPUGUUIGZDNIDUHNUNLN
ZAMNADMAGAADZNAQAGMAMMXKPQNUUINEDEOULXADCOJA
PNNDPJAIUAZXZDOANNNXNIJNXMPUUAGJCUPHLIAIDDAUAIGEU
IGXARKUJMGDQGAGMUCOAAGGPUÇDJAANDAPJÇUIIADOIUDO
DMODXPKONNUINZXUBUADGAUUAZJJPUXAGKOGDZMN
DZLXXAAGMZGVJHUXKXUPJUZAPGIADOUMVGIHAAPUNIJII
JVPBDALIPIDNKAANANMOMKUIAUIRAPAZNBBRNGMAJNPHONDIK
OGUADIQPSOMIJPAXUGGUUNNVDNGIPGKOGUCAAJDZNNNON
NJNMIMDXGGPDNMIXGNMUAPPAACADXZDUAJAAUIIIHZU

4) Per a cada un dels subcriptogrames 14 s'aplica una xifra de substitució simple utilitzant com a clau el caràcter corresponent. Per tal d'obtenir el valor de cada element de la clau, aplicarem el **mètode de coincidència múltiple**,

basat en l'observació de les freqüències relatives dels caràcters de cada subcriptograma:

- Es determinen els tres (o més) caràcters amb freqüències més altes del llenguatge que criptoanalitzem i ens fixem en les distàncies entre ells en l'alfabet de l'idioma.

- A continuació, es determinen els tres caràcters amb freqüències més altes per a cada un dels subcriptogrames i també se n'analitzen les distàncies. La clau resultarà de la relació entre aquestes dues distàncies.

5) A continuació, i atès que el text xifrat correspon a un text en clar en català, prendrem com a caràcters més freqüents la A, la E i la S. Si en l'alfabet català la lletra A correspon a la posició $p(\mathrm{A}) = 0$ i la lletra Z a la $p(\mathrm{Z}) = 25$, les distàncies entre aquestes lletres són:

$$\begin{aligned} (p(\mathrm{A}) + 5) \quad & \mod 26 = p(\mathrm{E}) \\ (p(\mathrm{E}) + 14) \quad & \mod 26 = p(\mathrm{S}) \\ (p(\mathrm{S}) + 7) \quad & \mod 26 = p(\mathrm{A}) \end{aligned} \qquad (1)$$

Com que el xifratge dins de cada subcriptograma és de substitució simple, algun caràcter del text xifrat tindrà aproximadament la freqüència característica de la A, un altre, la de la E, i un tercer, la de la S. A més, aquests valors amb aquestes tres freqüències altes hauran d'estar separats per una relació de distàncies constants igual a la descrita en (1). Efectivament, si calculem la taula de freqüències dels diferents caràcters en cada subcriptograma del nostre exemple, tenim que:

Taula de freqüències relatives

	A	B	C	Ç	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z
Valor de la lletra	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
C_A	3	3	1	0	**56**	5	8	1	13	**60**	5	4	4	26	3	0	30	17	29	25	13	9	35	**47**	22	28
C_B	**52**	13	15	0	14	**69**	2	8	3	26	1	0	25	14	33	17	13	8	30	**50**	17	27	4	4	1	1
C_C	**65**	6	6	3	35	3	0	33	9	34	21	11	7	22	**48**	18	27	5	3	2	0	**46**	5	17	0	20

Si ens fixem en els valors més freqüents per a cada subcriptograma (marcats en negreta), es compleixen en bona mesura les distàncies indicades en (1), és a dir:

$$\begin{aligned} (m_A + 5) \mod 26 &= m_E \\ (m_E + 14) \mod 26 &= m_S \\ (m_S + 7) \mod 26 &= m_A \end{aligned} \qquad (2)$$

on m_A, m_E i m_s són les posicions dels caràcters amb freqüència relativa més alta en el subcriptograma.

a) Així, per a calcular el primer caràcter de la clau, utilitzem C_A. L'única solució que compleix amb l'equació (2) és la de les lletres DIX, que tenen freqüències (56, 60, 47). De les equacions del xifratge de substitució simple resulta:

$$\begin{array}{lll} p(A) + K = p(D) & 0 + K = 4 & \\ p(E) + K = p(I) & 5 + K = 9 & K = 4 \\ p(S) + K = p(X) & 19 + K = 23 & \end{array}$$

De manera que és possible que la primera lletra de la clau sigui la D.

b) Per a C_B, la relació de les tres lletres amb freqüència més alta i que compleix (2) correspon a AES, amb freqüències (52, 69, 50), que curiosament correspon a quan el text xifrat coincideix amb el text en clar. Les equacions en aquest cas seran:

$$\begin{array}{lll} p(A) + K = p(A) & 0 + K = 0 & \\ p(E) + K = p(E) & 5 + K = 5 & K = 0 \\ p(S) + K = p(S) & 19 + K = 19 & \end{array}$$

I la clau podria ser la A.

c) Finalment, per a C_C, la relació de les tres lletres amb freqüència més alta i que compleix (2) correspon a la cadena UAN amb freqüències (46, 65, 48). En aquest cas, les equacions determinen:

$$\begin{array}{lll} p(A) + K = p(\mathrm{U}) & 0 + K = 21 & \\ p(E) + K = p(A) & 5 + K = 0 & K = 21 \\ p(S) + K = p(\mathrm{N}) & 19 + K = 14 & \end{array}$$

Per tant, hem obtingut que la clau és K = DAU.

6) Si ara fem servir un algoritme de desxifrat de Vigenère amb el criptograma inicial C i la clau K trobada, obtindrem el text en clar següent:

$M =$ QUAN A FINALS DE JULIOL LES AULES QUEDEN DEL TOT BUIDES, LA SITUACIÓ ES FA ESTRANYA. AL CAMPUS NOMÉS HI TROBES LES PASSES APRESSADES D'ALGUN PROFESSOR QUE VA A OMPLIR LES ÚLTIMES ACTES, D'UNA DONA DE LA NETEJA QUE BUIDA LES ESCOMBRARIES. HI HA EL SOL QUE CAU, PLOMAT, DAMUNT ELS EDIFICIS ON NO HI HA ALUMNES, A LA GESPA QUE S'HA DE REGAR MÉS QUE MAI, PERQUÈ L'ILLA ÉS D'ESTIUS DURS QUE DEVOREN EL VERD. RECÓRRER UN CAMPUS UNIVERSITARI A FINALS DE JULIOL, JUST ABANS QUE COMENCI L'AGOST, PRODUEIX UNA SENSACIÓ CURIOSA. ÉS UNA BARREJA D'INCREDULITAT I DE DESASSOSSEC, COM SI ENS TOCÀS RECÓRRER UN PAISATGE MOLT CONEGUT AL QUAL MANQUEN ELEMENTS INDISPENSABLES: UNA PLATJA

D'ARENA BLANCA, PER EXEMPLE, AMB OMBREL·LES I TOVALLOLES AL TERRA, AMB GENT QUE PREN EL SOL I BUSCA UN HORITZÓ QUE NO EXISTEIX, PERQUÈ NO HI HA MAR. UNA UNIVERSITAT SENSE AMBIENT UNIVERSITARI ÉS COM UNA PLATJA SENSE MAR. UN ABSURD. ENCARA NO FA GAIRES DIES, LA GESPA ERA PLENA DE COSSOS QUE S'ABOCAVEN ALS APUNTS LLEGITS DE PRESSA. MIRADES QUE CERCAVEN LES LLETRES D'UN LLIBRE O LA MIRADA CÒMPLICE D'ALGUN COMPANY. DIUEN QUE, A LA UNIVERSITAT, QUAN ARRIBA EL BON TEMPS, ELS ESTUDIANTS S'ENAMOREN. SÓN AMORS BREUS I FUGISSERS QUE NO DUREN GAIRE. SÓN PARÈNTESIS QUE ELS PERMETEN RESPIRAR, ENTRE EXAMEN I EXAMEN, QUAN ELS DIES ES FAN FEIXUCS I ELS PROFESSORS, SEGURAMENT, ENS FEM INSUPORTABLES. ARA, EL SILENCI VA GUANYANT TERRENY. CONQUEREIX CADA AULA, ELS PASSADISSOS, EL BAR INUSUALMENT BUIT, LES PANTALLES APAGADES DELS ORDINADORS. ALS DESPATXOS, ELS PROFESSORS ENCARA HI RECULLEN LES ÚLTIMES PERTINENCES, LES SECRETÀRIES BUSQUEN EL REFUGI DE L'AIRE CONDICIONAT, I EL SOL SEGUEIX AVANÇANT UN PAM CADA MATÍ.

Com acabem de veure, el mètode de criptoanàlisi de Kasiski és estadístic i, per tant, no és ni de bon tros exacte o infal·lible. Per exemple, en buscar la longitud de la clau pot succeir que es repeteixin per atzar cadenes de caràcters que facin que la seva separació no sigui un múltiple de la clau, o pitjor, que la longitud de la clau sigui un nombre primer i el mcd sigui 1. Per tal d'evitar aquestes situacions, necessitarem criptogrames de textos llargs (d'uns quants centenars de caràcters) i hi buscarem repeticions de cadenes d'almenys tres elements que apareguin més de dues vegades. D'altra banda, quan els subcriptogrames són petits, pot passar que no es conservi la rotació modular de les lletres amb més freqüència i, per tant, que la clau no sigui tan simple d'obtenir amb la primera anàlisi.

Resum

En aquest mòdul didàctic hem presentat els fonaments de la criptografia estudiant-ne quatre aspectes diferents:

1) Els **criptosistemes històrics**, basats en els principis de **transposició** i de **substitució**, que es feien servir abans de l'aparició dels ordinadors. De tota manera, les xifres de transposició i de substitució es continuen emprant com a blocs constituents de moltes de les xifres actuals.

2) Els **fonaments de la teoria de la informació**, i d'una manera especial el concepte d'**entropia de Shannon**, que proporcionen un marc teòric per a quantificar la seguretat dels criptosistemes, tant pel que fa al secret com a l'autenticitat.

3) El **secret perfecte** i l'**autenticitat perfecta**, que són dues propietats independents i difícilment assolibles en la pràctica. Serveixen com a guia d'allò que s'hauria de tendir a aconseguir quan es dissenyen criptosistemes. La teoria de la informació permet formular ambdues propietats amb precisió.

4) Els **fonaments de la criptoanàlisi elemental**, dels quals hem vist dos conceptes:

a) La **suposició de Kerckhoff**, segons la qual els algorismes de xifratge i de desxifratge han de ser públics i l'únic paràmetre secret ha de ser la clau.

b) La **distància d'unicitat**, que està relacionada amb la redundància del text en clar i és la quantitat de text xifrat que ha d'obtenir un criptoanalista perquè hi hagi una sola clau que pugui produir aquest text xifrat a partir d'un text en clar amb sentit. Amb tot, que la clau quedi determinada de manera única no vol dir que sigui computacionalment fàcil de trobar.

Activitats

1. Implementeu la xifra de Beale fent servir com a clau qualsevol text prou llarg.

2. Escriviu un programa que simuli una màquina de 10 rotors.

3. Agafeu un text en català prou llarg i intenteu determinar-hi la taxa r i la redundància D del català. Els passos que heu de seguir són els següents:
a) Considereu només les lletres del text i no distingiu entre majúscules i minúscules. Compteu la lletra "ç" com si fos una "c". No compteu espais ni signes de puntuació.
b) Escriviu un programa que calculi la freqüència relativa de cadascuna de les 26 lletres en el text.
c) Calculeu l'entropia d'un caràcter del text, fent servir les freqüències calculades com si fossin probabilitats d'aparició de les lletres en català. Aquesta entropia us dóna una aproximació de r.
d) Obteniu $D = R - r = 4{,}7 - r$. Calculeu la proporció de redundància com a D/R.

Exercicis d'autoavaluació

1. Imagineu que la freqüència relativa d'aparició de la lletra "a" en un text en clar és 0,1. Si es fa servir una xifra de transposició, hi haurà necessàriament alguna lletra en el text xifrat que tingui la mateixa freqüència? Si el digrama "ny" apareix amb una freqüència 0,02 al text en clar, hi haurà algun digrama en el text xifrat amb la mateixa freqüència?

2. Desxifreu el text xifrat següent amb la xifra de Cèsar fent servir $k = 3$:

VLWXYDVDOFHO.

3. Considereu els textos xifrats següents:
- XXXXX.
- VWXYZ.
- RKTIC.
- JZQAT.

Quines d'aquestes paraules podrien ser el resultat de xifrar paraules catalanes de cinc lletres fent servir:
a) Una xifra de substitució simple, no necessàriament del tipus Cèsar.
b) Qualsevol xifra de transposició.

4. Sigui X una variable aleatòria entera representada per 32 bits. Suposeu que la probabilitat que X caigui a l'interval $[0,2^8 - 1]$ és 1/2, amb tots els valors de l'interval equiprobables; la probabilitat que X caigui a l'interval $[2^8,2^{32} - 1]$ és també 1/2 i tots els valors de l'interval són equiprobables també. Amb aquesta informació calculeu $H(X)$.

5. Sigui M un dels sis missatges A, B, C, D, E, F on $p(A) = p(B) = p(C) = 1/4$, $p(D) = 1/8$, $p(E) = p(F) = 1/16$. Calculeu $H(M)$.

6. Demostreu que si M pot prendre dos valors, $H(M)$ és màxima per a $p_1 = p_2 = 1/2$. Generalitzeu-ho a qualsevol n, és a dir, proveu que si M pot prendre n valors, $H(M)$ és màxima quan tots els valors són equiprobables.

7. Expliqueu per què una xifra pot ser segura computacionalment encara que tingui una distància d'unicitat petita.

Solucionari

Exercicis d'autoavaluació

1. En una xifra de transposició es conserva la distribució de freqüències. Així, doncs, la lletra xifrada corresponent a la "a" tindrà exactament la mateixa freqüència 0,1. En canvi, dues lletres contigües en el text en clar no tenen per què ser-ho en el text xifrat. En particular, el digrama "ny" no tindrà cap digrama corresponent al text xifrat i, per tant, no se'n conservarà la freqüència.

2. El resultat del desxifratge és el text en clar següent:

SITUVASALCEL.

3. La paraula xifrada XXXXX no pot ser obtinguda com a resultat de transposar les lletres d'una paraula catalana (no hi ha cap paraula al diccionari que consti de cinc lletres iguals). Per la mateixa raó, XXXXX no pot ser obtinguda amb una xifra de substitució simple. La resta de paraules es poden obtenir, però no per substitució simple per transposició.

4. Cadascun dels 2^8 valors de l'interval $[0,2^8 - 1]$ es prenen amb probabilitat 2^{-9}. Cadascun dels $2^{32} - 2^8$ valors restants es prenen amb probabilitat:

$$\frac{1}{2 \cdot (2^{32} - 2^8)} = \frac{1}{2^{33} - 2^9}.$$

Per tant, l'entropia de X serà:

$$H(X) = 2^8 \cdot (2^{-9} \log 2^9) + (2^{32} - 2^8) \cdot \left(\frac{1}{2^{33} - 2^9} \cdot \log(2^{33} - 2^9)\right) \approx \frac{9}{2} + \frac{1}{2}\, 33 = 21 \text{ bits.}$$

Així, doncs, veiem que el contingut informatiu de la variable és bastant menor que els 32 bits usats per a representar-la.

5. Tenim que l'entropia de M es la següent:

$$H(M) = 3 \cdot \frac{1}{4} \cdot \log 4 + \frac{1}{8} \cdot \log 8 + 2 \cdot \frac{1}{16} \cdot \log 16 = 6/4 + 3/8 + 8/16 = 2.375 \text{ bits.}$$

6. Amb dos valors possibles per a X, la probabilitat del segon valor és $p_2 = 1 - p_1$. Per tant, tenim el següent:

$$H(X) = H(p_1) = p_1 \log\left(\frac{1}{p_1}\right) + (1 - p_1) \log\left(\frac{1}{1 - p_1}\right).$$

Si derivem segons p_1 i igualem a zero, obtenim que $H(p_1)$ té un màxim a $p_1 = 1/2$. En el cas de n valors possibles, es tracta de cercar el màxim de la funció $H(p_1,...,p_n)$ subjecta a la restricció següent:

$$\sum_{i=1}^{n} p_i = 1.$$

Fent servir el mètode dels multiplicadors de Lagrange, obtenim que el màxim es produeix quan $p_1 = ... = p_n = 1/n$.

7. Si la distància val d i donem d caràcters de text xifrat com a entrada en un ordinador, més tard o més d'hora l'ordinador trobarà una única clau que pot produir els d caràcters. Ara bé, encara que d sigui petita, pot passar que l'ordinador necessiti un temps molt i molt llarg (anys o fins i tot segles, en funció de l'algorisme de xifratge).

Glossari

Ambigüitat de clau: incertesa que queda sobre el valor de la clau un cop es coneix un criptograma.

Autenticitat perfecta: propietat que té una xifra quan la probabilitat d'engany amb èxit del receptor per part d'un criptoanalista és mínima.

Criptosistema històric: criptosistema emprat abans de l'aparició dels ordinadors.

Desxifratge espuri: desxifratge que, donat el text xifrat $c = E_k(m)$, existeix quan el xifratge sota una altra clau k' pot donar c; és a dir, $c = E_{k'}(m)$ per al mateix missatge m, o bé $c = E_{k'}(m')$ per a un altre missatge amb sentit m'.

Distància d'unicitat: nombre mínim de caràcters de text xifrat tal que existeix una única clau que produeix aquests caràcters de text xifrat a partir d'un text en clar amb sentit.

Entropia de Shannon: entropia d'una variable aleatòria, que mesura, en bits, la incertesa sobre el valor que prendrà la variable.

Probabilitat d'engany: probabilitat que un criptoanalista enemic aconsegueixi enganyar el receptor fent-li acceptar com a bo un missatge modificat o un missatge inserit.

Suposició de Kerckhoff: suposició segons la qual els algorismes de xifratge i de desxifratge d'un criptosistema són públics i el secret queda restringit a la clau emprada.

Teoria de la informació: teoria introduïda per Shannon que mesura la informació des d'un punt de vista quantitatiu.

Variable aleatòria: variable que pren un valor entre una colla de valors possibles, de tal manera que cada valor possible té una certa probabilitat no nul·la de ser adoptat.

Bibliografia

Bibliografia bàsica

Denning, D.E. (1982). *Cryptography and Data Security*. Reading (Massachusetts): Addison-Wesley.

Kahn, D. (1967). *The Codebreakers*. Nova York: Macmillan.

Rifà, J.; Huguet, L. (1991). *Comunicación digital*. Barcelona: Masson.

Simmons, G.J. (1992). *Contemporary Cryptology: The Science of Information Integrity*. Nova York: IEEE Press.

Stinson, D. (1995). *Cryptography: Theory and Practice*. Boca Raton: CRC Press.

Referències bibliogràfiques

Porta, G.B. (1593). *De Occultis Literarum* (llibre 1, capítol 1). Edició facsimil. Universidad de Zaragoza, 1996.

Xifres de clau compartida: xifres de flux

Jordi Herrera Joancomartí

P03/05024/02261

UOC
www.uoc.edu

Índex

Xifres de clau compartida: xifres de flux

Introducció

Hem vist en altres mòduls didàctics que un criptosistema incondicionalment segur necessita, com a mínim, tants bits de clau com bits de text per a xifrar i que el xifratge de Vernam és l'únic que aconsegueix aquesta seguretat incondicional, però el preu que paga per això és la ineficiència del xifratge. Aquesta ineficiència recau justament en el fet que la clau ha de tenir la mateixa longitud que el text que s'ha de xifrar. Això comporta que la llargària de les claus sigui molt gran i, per tant, sigui més difícil guardar-les en secret. A més, es dóna la paradoxa que si tenim un canal segur per a intercanviar les claus també el podem utilitzar per a intercanviar els missatges, ja que tenen la mateixa llargada.

Vegeu el mòdul "Fonaments de criptografia" d'aquesta assignatura.

Les **xifres de flux** sorgeixen com una aproximació optimitzada del xifratge de Vernam. La idea és construir una clau prou llarga, si més no de la llargada del missatge, a partir d'una clau inicial curta mitjançant el que s'anomena *generador pseudoaleatori*. Aquest generador expandeix una clau curta, anomenada *llavor*, per a obtenir-ne una de molt més llarga. A més, l'operació d'expansió ha de tenir unes característiques determinades, ja que la seqüència que en resulta es fa servir per a xifrar el text en clar.

Cal, doncs, analitzar les propietats que han de complir les seqüències esmentades i estudiar quins tipus de generadors hi ha per a obtenir-les.

Objectius

En els materials didàctics facilitats en aquest mòdul, l'estudiant trobarà les eines necessàries per a assolir els objectius següents:

1. Comprendre l'esquema general de les xifres de flux.

2. Assimilar les característiques que ha de complir necessàriament una seqüència pseudoaleatòria perquè sigui utilitzable en un esquema de xifratge de flux.

3. Entendre el funcionament de diferents generadors pseudoaleatoris.

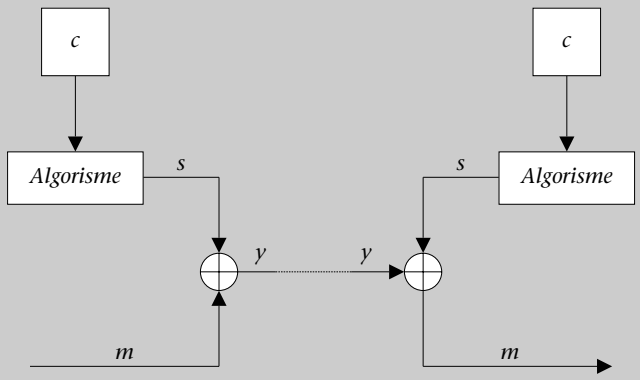
1. Requisits de les seqüències de xifratge de flux

Les xifres de flux s'inclouen dins els anomenats *criptosistemes de clau compartida*. Els **criptosistemes de clau compartida** són aquells en els quals l'emissor i el receptor comparteixen una mateixa clau per a xifrar i desxifrar missatges.

Així, doncs, en aquests criptosistemes, la clau que s'utilitza per a xifrar el missatge és la mateixa que es fa servir per a desxifrar-lo; per tant, emprant sempre la mateixa clau, en qualsevol moment l'emissor pot passar a fer de receptor i a l'inrevés.

D'una manera esquemàtica, un criptosistema de flux es pot representar com mostra la figura següent:

Esquema general de flux



Els elements que intervenen...

... en el xifratge de flux són: *c*, la clau compartida; *Algorisme*, el generador pseudoaleatori; *m*, el missatge en clar; *s*, la seqüència de xifratge, i *y*, la seqüència xifrada.

Tant l'emissor com el receptor disposen d'una mateixa clau *c*, anomenada **llavor del generador**, i d'un mateix algorisme determinista, anomenat **generador pseudoaleatori**. En proporcionar la clau com a entrada a l'algorisme, aquest genera en la sortida una seqüència anomenada **seqüència de xifratge.**

Per a xifrar el missatge, l'emissor va sumant cada bit del missatge en clar, amb cada bit de la seqüència de xifratge. Quan el receptor rep el missatge xifrat fa servir el mateix algorisme determinista, que en la figura anterior hem denotat com *Algorisme*, i la clau *c*, que comparteix amb l'emissor, per a obtenir la mateixa seqüència de xifratge. Així, sumant bit a bit el missatge que li arriba, *y*, amb la seqüència resultant de l'algorisme, *s*, el receptor obté el text en clar *m* enviat per l'emissor.

Totes les seqüències amb què treballarem en aquest mòdul didàctic seran binàries, i les operacions a què ens referirem seran totes mòdul 2.

Perquè aquest criptosistema sigui segur, és fonamental que la seqüència de xifratge no sigui coneguda; és a dir, que no es pugui saber en cap moment quin serà el bit de sortida següent. Idealment, el que es necessita per a garantir la seguretat incondicional és que la clau, en aquest cas la seqüència de xifratge, sigui completament aleatòria.

En el nostre esquema no es pot donar aquesta condició, ja que el generador que utilitzem ha de ser determinista perquè l'emissor i el receptor obtinguin la mateixa seqüència quan donen com a entrada la mateixa clau secreta.

Així, doncs, la seqüència de xifratge tindrà propietats molt properes a les que té una seqüència completament aleatòria i s'anomenarà **seqüència pseudoaleatòria**.

Concretament, si una seqüència no és aleatòria a partir d'un cert moment es repeteix. Aquesta subseqüència que es va repetint s'anomena *període*. És important, doncs, que aquesta subseqüència, el període, sigui indistingible d'una seqüència completament aleatòria de la mateixa longitud, i per a aconseguir-ho la seqüència ha de complir unes propietats determinades que veurem més endavant.

Vegeu els postulats de Golomb al subapartat 1.2 d'aquest mòdul didàctic.

Cal no oblidar que els criptosistemes de clau compartida basen la seguretat en el fet que la clau emprada per a xifrar i desxifrar només la coneixen l'emissor i el receptor. En el xifratge de flux, si bé la clau no es fa servir directament per a xifrar, cal igualment que no es faci pública, ja que l'algorisme determinista és conegut i es podria obtenir la seqüència de xifratge a partir d'aquest i de la clau.

1.1. Període

Hem vist que per a implementar un criptosistema de xifratge de flux necessitem un algorisme que ens doni com a sortida la seqüència de xifratge. El fet que aquest algorisme sigui determinista implica que la seqüència que en resulta no sigui completament aleatòria i, per consegüent, que a partir d'un cert moment es repeteixi. Ja hem dit que aquesta subseqüència que es va repetint és el *període*.

Formalment, podem definir el període de la manera següent: sigui $\{s_i\}_{i\geq 0}$ una seqüència periòdica, el **període** p és l'enter més petit tal que $s_{i+p} = s_i$ per a tot $i \geq 0$.

Tenint en compte que el període es repeteix, una vegada conegut és possible determinar exactament tota la seqüència de xifratge i trencar el criptosistema. Per això, les seqüències que s'utilitzen per al xifratge de flux cal que tinguin un període molt llarg, ja que d'aquesta manera triguen molt a repetir-se i és més difícil predir-ne la sortida.

El concepte de *període llarg*...

... es refereix al xifrador i a l'aplicació. Un període de 2^{32} pot no ser prou llarg per a un xifrador que xifri a 1 megabyte per segon, ja que a aquesta velocitat el període es repeteix només cada 8,5 minuts.

1.2. Postulats de Golomb

Per a aconseguir una seqüència que ens serveixi per al xifratge de flux no n'hi ha prou que el període sigui llarg. Cal també que la distribució dels zeros i els uns que la formen tingui una certa uniformitat. En concret, perquè una seqüència es pugui considerar pseudoaleatòria ha de complir els tres postulats de Golomb. Abans d'estudiar els postulats, però, haurem d'introduir la terminologia que emprarem.

Definirem una **ràfega** com un conjunt de bits consecutius iguals; és a dir, una ràfega de longitud k és el conjunt dels elements s_t, ..., s_{t+k-1} tals que $s_{t-1} \neq s_t = \ldots = s_{t+k-1} \neq s_{t+k}$.

La **funció d'autocorrelació**, $AC(k)$, d'una seqüència periòdica $\{s_i\}_{i\geq 0}$ amb un període p es defineix com:

$$AC(k) = \frac{A - D}{p},$$

en què A i D són, respectivament, el nombre de coincidències i de no-coincidències de tot el període entre la successió $\{s_i\}_{i\geq 0}$ i la mateixa successió desplaçada k posicions, $\{s_{i+k}\}_{i\geq 0}$. És a dir:

- $A = |\{0 \leq i < p \text{ tal que } s_i = s_{i+k}\}|$.
- $D = |\{0 \leq i < p \text{ tal que } s_i \neq s_{i+k}\}|$.

Fixeu-vos que si k és un múltiple de p, llavors $AC(k) = 1$.

Els postulats de Golomb, que ha de verificar tota seqüència pseudoaleatòria, diuen el següent:

1) Dins el període d'una seqüència pseudoaleatòria, el nombre de zeros i d'uns ha de ser el mateix o ha de diferir com a màxim d'una unitat; és a dir, ha de ser $p/2$ si p és parell i $(p \pm 1) / 2$ si és senar.

2) El nombre total de ràfegues de longitud k en un període ha de valer com a mínim $n/2^k$, essent n el nombre total de ràfegues del període.

3) La funció d'autocorrelació $AC(k)$ és bivaluada, és a dir, només pren dos valors: 1 si k és múltiple de p, i un altre valor constant si p no divideix k.

Cal afegir que tot i que aquests postulats donen informació sobre l'aleatorietat de les seqüències, un cop s'han verificat cal recórrer, a més, a diferents tests estadístics, com ara el de la χ^2 o els tests espectrals.

1.3. Complexitat lineal

Per a utilitzar una seqüència pseudoaleatòria per al xifratge de flux, cal que cada bit depengui el mínim possible de l'anterior. És a dir, la seqüència de xifratge ha de tenir un grau elevat d'impredictibilitat. Cal que la probabilitat que surti un 0 o bé un 1 sigui propera a 0,5 i que depengui en mínima mesura del bit de sortida anterior.

El concepte de **complexitat lineal** ens mesura aquesta impredictibilitat, es a dir, ens informa de quina part de la seqüència ens cal conèixer per a poder-la predir totalment.

Per a calcular la complexitat lineal farem servir l'algorisme de Massey, que detallarem més endavant.

Vegeu la referència a l'algorisme de Massey al subapartat 2.3 d'aquest mòdul.

1.4. Implementabilitat

Si ens fixem en l'esquema de xifratge de flux de la figura anterior, veiem que per a obtenir el text xifrat que enviem al receptor hem d'anar sumant el text en clar amb la seqüència de xifratge que produeix el generador pseudoaleatori. Això significa que la velocitat de transmissió de les dades entre l'emissor i el receptor la determina el valor mínim entre la velocitat de generació del missatge i la velocitat de generació de la seqüència de xifratge. Així, doncs, haurem de tenir en compte aquest fet quan estudiem els possibles generadors pseudoaleatoris, ja que d'acord amb la implementació que se n'hagi fet (ja sigui en maquinari o en programari), obtindrem una velocitat o una altra.

Els telèfons mòbils...

... amb tecnologia GSM incorporen un xifrador de flux. Seria impensable que el cost econòmic del xifrador incrementés el preu del telèfon mòbil. Tampoc no seria admissible que la velocitat de la comunicació es veiés afectada per la velocitat del xifrador emprat.

Cal que l'algorisme que generi la seqüència de xifratge sigui fàcil d'implementar, tant pel que fa a la complexitat com al cost econòmic.

2. Generadors lineals

En l'apartat anterior hem estudiat les propietats que han de tenir les seqüències de xifratge per a poder-les utilitzar en criptosistemes de xifratge de flux. Analitzem ara com han de ser els algorismes deterministes que generen aquests tipus de seqüències. Des d'un punt de vista general tenim dos tipus de generadors:

1) Els **generadors lineals** són aquells que només executen operacions lineals sobre els elements d'entrada per a obtenir la seqüència de sortida.

2) Els **generadors no lineals** són els que executen, a més, operacions no lineals, com ara permutacions.

2.1. Generadors congruents

Els generadors congruents es basen en equacions modulars recurrents del tipus: $x_n = (a x_{n-1} + b) \bmod m$.

En aquest cas, el valor x_0 seria la llavor de la seqüència de xifratge. Un criptosistema que empri un generador d'aquest tipus ha de tenir com a clau secreta els valors $\{x_0, a, b, m\}$, i perquè el període sigui màxim s'ha de complir que $\text{mcd}(a,m) = 1$.

Val a dir, però, que aquests tipus de generadors pseudoaleatoris no són segurs des d'un punt de vista criptogràfic, ja que s'ha pogut demostrar que amb pocs valors x_i coneguts ja es poden esbrinar els paràmetres secrets $\{x_0, a, b, m\}$. Fins i tot coneixent només una part dels bits que formen els x_i (però, això sí, coneixent els paràmetres $\{a, b, m\}$) es pot arribar a determinar el valor de la llavor x_0.

Malgrat això, aquests tipus de generadors són molt utilitzats en sistemes informàtics per a aplicacions no criptogràfiques.

Exemples de generadors congruents

La funció `rand()` del sistema UNIX fa servir el generador congruent afí següent:

$$x_n = (1.103.515.245\ x_{n-1} + 12.345) \bmod 2^{31}.$$

La mateixa funció `rand()` de TurboPascal obté el resultat a partir de la congruència següent:

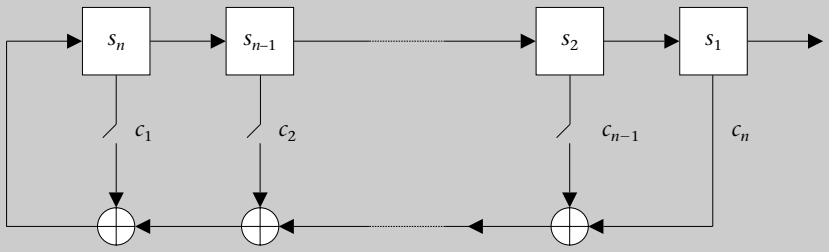
$$x_n = (129\ x_{n-1} + 907.633.385) \bmod 2^{32}.$$

2.2. Registres de desplaçament realimentats linealment

Un registre de desplaçament realimentat linealment, LFSR*, de longitud n, és un dispositiu físic o lògic format per n cel·les de memòria i n portes lògiques, tal com mostra la figura següent:

*** LFSR és la sigla del terme anglès *Linear Feedback Shift Register.***

Esquema general d'un LFSR



Inicialment, les cel·les contenen els valors d'entrada, i a cada impuls de rellotge el contingut de la cel·la s_i es desplaça a la cel·la s_{i-1} fent les operacions associades. D'aquesta manera es genera un nou element, s_{n+1}, que és determinat per:

$$s_{n+1} = c_1 s_n + \ldots + c_n s_1, \qquad (2.1)$$

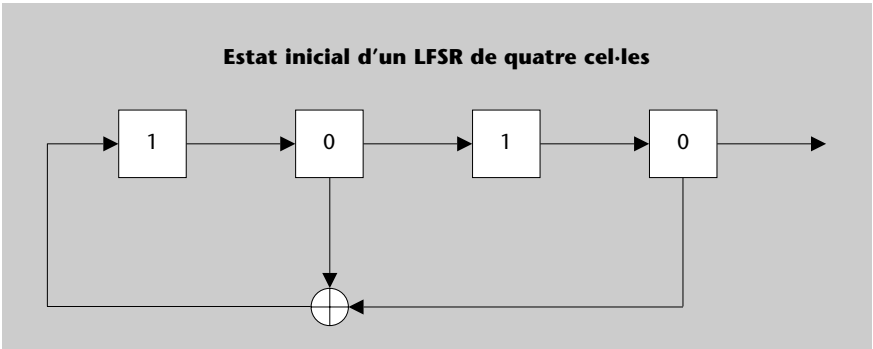
en què els $c_i \in \{0, 1\}$ corresponen als valors de les portes lògiques de l'esquema. És a dir, els coeficients seran 1 si hi ha una connexió, i 0 si no n'hi ha. Aquest nou element, s_{n+1}, se situa a la cel·la s_n, que ha quedat buida a causa del desplaçament.

El conjunt de valors continguts en cada cel·la en un instant de temps s'anomena **estat**. L'**estat inicial** és l'estat en què es troba l'LFSR en el moment de començar el procés.

Funcionament d'un LFSR de quatre cel·les

Vegem ara el funcionament d'un LFSR de quatre cel·les:

Estat inicial d'un LFSR de quatre cel·les



Podem observar que aquest registre de desplaçament té com a estat inicial 1010, que correspon a l'impuls de rellotge $t = 0$. L'evolució de l'estat en cada cicle del processador es mostra a la taula següent:

Impuls de rellotge (t)	Estat				Sortida
0	1	0	1	0	0
1	0	1	0	1	1
2	0	0	1	0	0
3	0	0	0	1	1
4	1	0	0	0	0
5	0	1	0	0	0
6	1	0	1	0	0
7	0	1	0	1	1

En l'impuls $t = 1$ el conjunt de les cel·les es desplaça però no varia, ja que en la cel·la s_1 i en la s_3 per $t = 0$ hi ha un 0. En canvi, per $t = 2$ ja hi ha hagut modificacions. En concret, $s_4 = 0$, ja que $s_4(t = 2) = s_1(t = 1) \oplus s_3(t = 1) = 1 \oplus 1 = 0$. En general, per $i \geq 1$ tenim:

- $s_4(t = i) = s_3(t = i - 1) \oplus s_1(t = i - 1)$,
- $s_3(t = i) = s_4(t = i - 1)$,
- $s_2(t = i) = s_3(t = i - 1)$,
- $s_1(t = i) = s_2(t = i - 1)$.

Fem notar que en l'impuls de rellotge $t = 6$ tornem a tenir l'estat inicial, i a partir d'aquí la seqüència es torna a repetir. Aquesta seqüència, doncs, té període 6.

Un cop definit què és un LFSR en podem fer un estudi una mica més exhaustiu per a determinar-ne les característiques més importants. L'avantatge principal dels LFSR és que tenen una formulació matemàtica molt simple, com veurem a continuació, i, per tant, es poden estudiar de manera força clara i completa. A més, com que es defineixen per mitjà de cel·les i portes lògiques, s'implementen fàcilment en el maquinari, fet que permet obtenir generadors de gran velocitat.

Primerament cal fer notar que l'estat inicial d'un LFSR no pot ser tot de zeros. Si fos així, la seqüència que produiria seria també de zeros, ja que totes les operacions són lineals. Es diu que l'estat que tan sols té zeros és un **estat absorbent**. També convé destacar que el període màxim d'un LFSR és $2^n - 1$, valor i que aquest s'obté de considerar tots els estats possibles 2^n i eliminar-ne l'estat nul.

Si observem l'expressió 2.1, ens adonarem que tota seqüència generada per un LFSR és determinada per l'estat inicial $\{s_1, ..., s_n\}$ i per la relació:

$$s_{n+k} = \sum_{i=1}^{n} c_i s_{n+k-i}, \qquad (2.2)$$

on $c_i \in \{0, 1\}$ per $1 \leq i \leq n$.

El **polinomi de connexions d'un LFSR** de longitud n és el polinomi de grau n, que té la forma següent:

$$C(x) = 1 + c_1x^1 + c_2x^2 + ... + c_nx^n,$$

on els $c_i \in \{0, 1\}$ corresponen als valors de les portes lògiques de la figura de l'esquema general d'un LFSR.

Polinomi de connexions de l'LFSR de quatre cel·les

El polinomi de connexions corresponent a l'LFSR de l'exemple anterior és el següent:

$C(x) = 1 + x^2 + x^4$.

Un LFSR es determina pel seu polinomi de connexions, i una seqüència, pel polinomi de connexions i per l'estat inicial.

Definit el polinomi de connexions, ja podem determinar les característiques de l'LFSR d'acord amb les del seu polinomi de connexions:

1) Polinomi de connexions factoritzable: els LFSR que tenen com a polinomis de connexions polinomis factoritzables generen seqüències que depenen de l'estat inicial. A més, el període d'aquestes seqüències és sempre més curt que el període màxim que pot tenir un LFSR, que és $2^n - 1$.

2) Polinomi de connexions irreductible: les seqüències generades pels LFSR que tenen polinomis de connexions irreductibles no depenen de l'estat inicial, sinó que simplement queden desplaçades. En aquest cas, el període serà un divisor de $2^n - 1$.

3) Polinomi de connexions primitiu: un LFSR amb polinomi de connexions primitiu té la seqüència de sortida de període màxim, $2^n - 1$. Aquesta seqüència de període màxim s'obté per a qualsevol estat inicial, llevat de l'estat absorbent.

Considerant les propietats de les seqüències segons els seus polinomis de connexions, veiem que per a esquemes de xifratge de flux és aconsellable fer servir la que determina el període màxim. A més, les seqüències generades per l'LFSR amb un polinomi de connexió primitiu compleixen els tres postulats de Golomb, gràcies a la longitud màxima del seu període i al fet que han de passar per tots els estats possibles.

2.3. Limitacions dels generadors lineals

Recordeu

No hem d'oblidar que el nombre de polinomis primitius de grau n el dóna l'expressió $\phi(2^n - 1) / n$ on ϕ és la funció totient d'Euler.

Ja hem posat en relleu que els LFSR es comporten molt bé en termes de facilitat d'anàlisi, d'implementació i de velocitat. A més, en el cas que el polinomi de connexions sigui primitiu, també compleixen tots els postulats de Golomb. Ara bé, un inconvenient d'aquests generadors és que, perquè el període $2^n - 1$ sigui llarg, cal que la longitud de l'LFSR també sigui llarga.

Això pot representar un problema, ja que el cost de trobar polinomis primitius de grau alt és força elevat.

Malgrat els avantatges i inconvenients dels generadors lineals, la raó principal per la qual no serveixen per a implementar sistemes de xifratge de flux és que són fàcilment predictibles.

En efecte, suposem que coneixem $2n$ bits consecutius d'una seqüència de xifratge, s_{k+1}, s_{k+2}, ..., s_{k+2n}. Aleshores podem determinar els coeficients del polinomi de realimentació, c_i, i, per tant, tota la seqüència. Per a fer-ho només cal que ens basem en l'expressió 2.2 i que plantegem el sistema d'equacions següent:

$$\begin{bmatrix} s_{k+1} & s_{k+2} & \cdots & s_{k+n} \\ s_{k+2} & s_{k+3} & \cdots & s_{k+n+1} \\ \vdots & \vdots & \cdots & \vdots \\ s_{k+n} & s_{k+n+1} & \cdots & s_{k+2n-1} \end{bmatrix} \begin{bmatrix} c_n \\ c_{n-1} \\ \vdots \\ c_1 \end{bmatrix} = \begin{bmatrix} s_{k+n+1} \\ s_{k+n+2} \\ \vdots \\ s_{k+2n} \end{bmatrix}.$$

Tenim ara un sistema d'n equacions amb n incògnites, c_i, per $1 \leq i \leq n$, amb el qual podem determinar tots els coeficients.

Així, doncs, a l'hora d'utilitzar un generador per a un procés de xifratge de flux cal que ens fixem també en la predictibilitat que té, és a dir, en el que s'anomena *complexitat lineal*.

Atès que qualsevol seqüència periòdica es pot generar amb un LFSR no singular, J.L. Massey va definir la **complexitat lineal d'una seqüència** com el nombre de cel·les de l'LFSR més curt que és capaç de generar-la.

Per tant, una seqüència generada per un LFSR de longitud n té, òbviament, complexitat lineal n, una complexitat molt baixa comparada amb el període, $2^n - 1$. El mateix Massey va proposar un algorisme que a partir d'una seqüència determina l'LFSR mínim que la genera amb l'estat inicial corresponent.

Per a disminuir la predictibilitat de la seqüència de xifratge cal, doncs, augmentar la complexitat lineal de la seqüència de xifratge, que convindria que fos de llargada propera a la del període. Una manera de fer-ho és basant-se en operacions no lineals, tal com veurem més endavant.

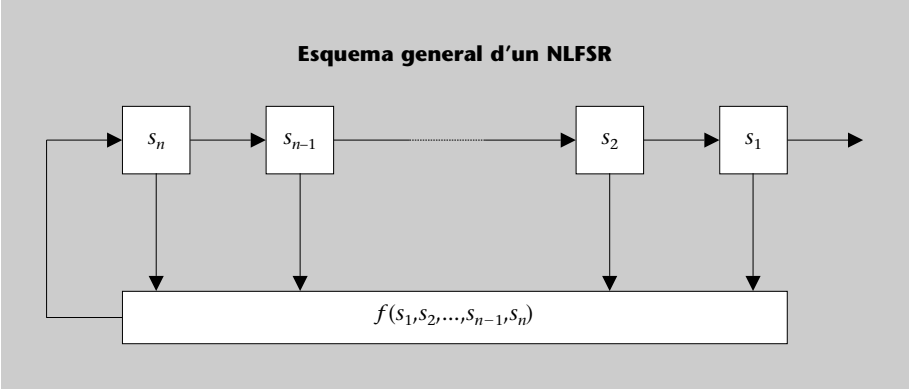
3. Generadors no lineals

A continuació analitzarem alguns del generadors no lineals destinats a augmentar la complexitat lineal de les seqüències de flux.

3.1. Registres de desplaçament realimentats no linealment

Els registres de desplaçament realimentats no linealment, NLFSR*, tenen un esquema molt semblant als LFSR, amb la variant que la funció que els realimenta no és lineal, tal com mostra la figura següent:

* NLFSR és la sigla del terme anglès *Non-Linear Feedback Shift Register.*



El fet que la funció de realimentació no sigui lineal presenta el problema que es fa molt complicat estudiar-la i especificar-la. En particular, és difícil trobar NLFSR amb períodes llargs*. Per aquesta raó, els NLFSR no s'estudien exhaustivament, tot i que s'intenta obtenir-ne propietats a partir d'algunes funcions de realimentació.

* Un exemple el tenim en les seqüències de De Bruijin, generades amb NLFSR i de període màxim.

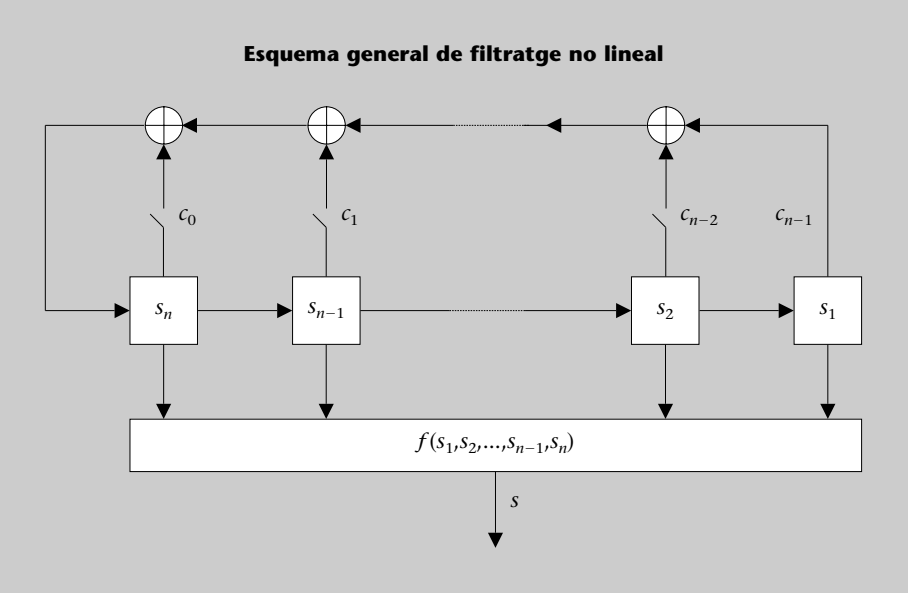
3.2. Filtratge no lineal

Una manera alternativa de generar seqüències és com a sortida d'una funció no lineal que té per entrada l'estat d'un LFSR. Una representació esquemàtica de la funció de xifratge que fa servir un filtratge no lineal és la que podem veure a la figura de la pàgina següent.

Aquest tipus de generadors són difícils d'analitzar, alhora que la complexitat lineal de la seqüència que generen també resulta difícil de calcular. De fet, el que s'intenta no és calcular aquesta complexitat lineal, sinó fitar-la.

Tal com es mostra a la figura següent, generalment els filtres no lineals consten d'un LFSR a fi d'aconseguir que una part de l'estudi sigui senzilla i tingui

unes característiques òptimes. Convé que el polinomi de connexions d'aquest LFSR sigui primitiu, perquè, com ja hem vist, d'aquesta manera el període de la seqüència resultant serà màxim.



La part més complicada d'aquests generadors és triar la funció de filtratge f, perquè cal que sigui una funció que permeti obtenir una complexitat elevada, però que sigui fàcil de calcular.

3.3. Combinadors no lineals

Els combinadors no lineals, com indica el seu nom, basen el funcionament en la combinació d'un cert nombre d'LFSR.

Segons com utilitzem i combinem aquests LFSR podem distingir bàsicament els tres tipus de generadors següents:

1) Generadors basats en combinacions no lineals d'LFSR, que fan servir les sortides de diferents LFSR com a entrada d'una funció no lineal que generarà la seqüència final.

2) Generadors basats en un control pas per pas, que utilitzen diferents LFSR, alguns dels quals regulen els impulsos de rellotge d'altres.

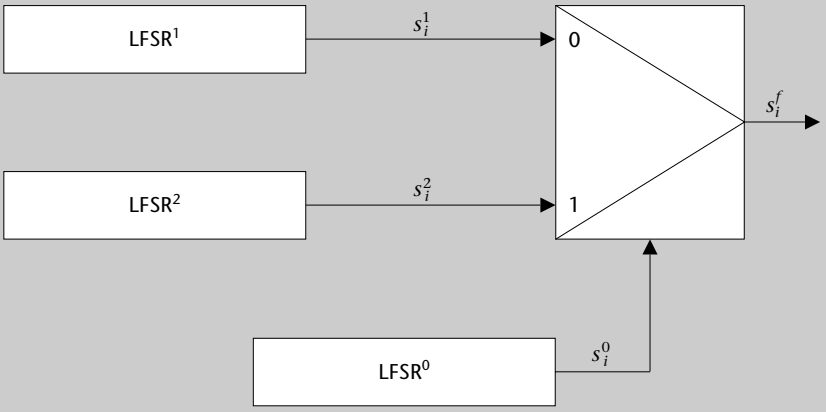
3) Generadors de seqüència multirellotge, formats per diferents LFSR regits per cicles de rellotge de velocitats diferents.

A continuació descrivim el funcionament de diferents combinadors no lineals per a entendre clarament la diferència entre els tres grups que hem presentat.

3.3.1. Generador de Geffe

El generador de Geffe és un generador pseudoaleatori format per tres LFSR relacionats, tal com mostra la figura següent:

Esquema d'un generador de Geffe



Notació

En aquest apartat els superíndex indiquen l'LFSR a què fa referència el valor.

Observem que hi ha dos LFSR que generen dues seqüències i un tercer LFSR que determina la funció de sortida. D'aquesta manera, a cada impuls de rellotge $t = i$ tenim dos valors dels LFSR^1 i LFSR^2 i un altre de l'LFSR^0, que és el que determina quina de les dues sortides prenem. Si la sortida de l'LFSR^0, s_0, és un 0, es pren la sortida de l'LFSR^1, $s_i^f = s_i^1$, mentre que si $s_i^0 = 1$, es pren la sortida de l'LFSR^2, $s_i^f = s_i^2$.

Analíticament el valor de sortida del generador per a l'impuls de rellotge $t = i$ es pot expressar de la manera següent:

$$s_i^f = (1 - s_i^0)s_i^1 \oplus s_i^0 s_i^2$$

on s_i^0, s_i^1, s_i^2, són, respectivament, els valors que els LFSR 0, 1 i 2 generen per a l'impuls de rellotge $t = i$.

En aquest tipus de generadors la complexitat lineal augmenta respecte a un generador format per un simple LFSR. En concret, siguin n_1, n_2, n_0 els graus dels polinomis de connexions dels tres LFSR, la complexitat lineal és $CL = n_1 + n_1 n_0 + + n_2 n_0$. D'altra banda, el període resultant és $p = \text{mcm}(2^{n_0} - 1, 2^{n_1} - 1, 2^{n_2} - 1)$.

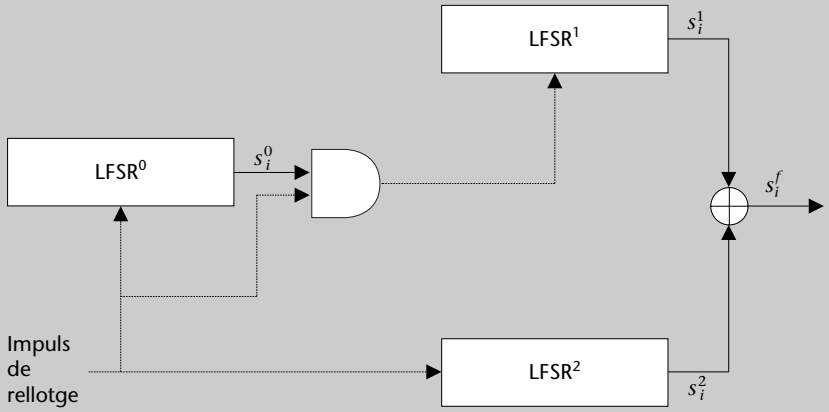
Tot i l'elevada complexitat lineal que s'obté amb aquest generador, tampoc no és útil per a aplicacions de xifratge de flux. El problema que té és la gran probabilitat de coincidència entre la seqüència final, s^f, i les seqüències de sortida intermèdies dels LFSR, s^1 i s^2:

- $P(s^f = s^1) = 3/4$,
- $P(s^f = s^2) = 3/4$.

3.3.2. Generador de Beth-Piper

Aquest generador es troba dins el grup de combinadors de control pas per pas i el seu funcionament queda reflectit a la figura següent:

Esquema d'un generador de Beth-Piper



L'esquema mostra com l'LFSR^0 controla el ritme del rellotge de l'LFSR^1. Així, LFSR^1 només canvia l'estat en l'instant t si $s^0_{t-1} = 1$; és a dir, si en l'instant $t - 1$ LFSR^0 dóna com a sortida un 1.

La seqüència de sortida final té una complexitat lineal $CL = (2^{n_0} - 1)\, n_1 + n_2$ i el període $p = (2^{n_0} - 1)(2^{n_1} - 1)(2^{n_2} - 1)$, en què, com abans, n_0, n_1, n_2 són els graus dels polinomis de connexions d'LFSR^0, LFSR^1 i LFSR^2, respectivament.

Igual que en el generador de Geffe, tot i la seva elevada complexitat, el generador de Beth-Piper no oferix una seguretat òptima, ja que la probabilitat que la seqüència final, s^f, i la seqüència generada per l'LFSR^2 coincideixin és molt elevada:

$$P(s_t^f \oplus s_{t+1}^f = s_t^2 \oplus s_{t+1}^2) = \frac{3}{4}$$

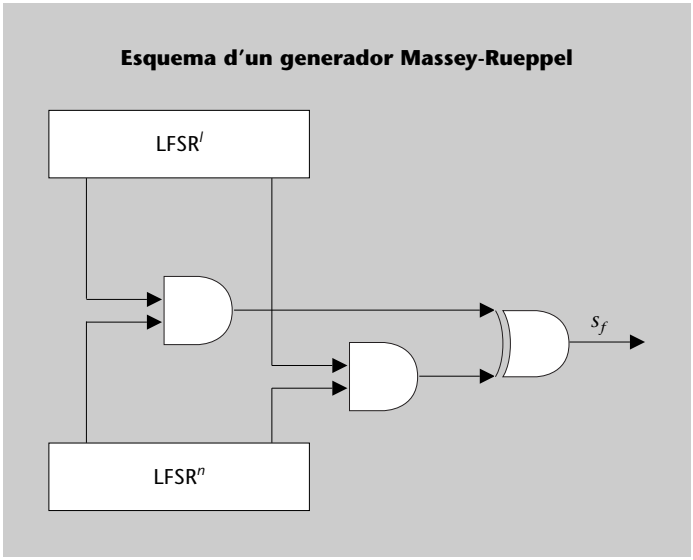
Una versió més segura...

... del generador de Beth-Piper és el generador en cascada de Gollmann, que també es basa en un combinador de control pas per pas.

3.3.3. Generador multivelocitat de Massey-Rueppel

A la figura de la pàgina següent es pot veure el funcionament del generador multivelocitat de Massey-Rueppel. Aquest generador fa servir dos LFSR que funcionen a una velocitat diferent. L'LFSR^n, que té n cel·les, va d vegades ($d \geq 2$) més de pressa que l'LFSR^l, de l cel·les (on $n \geq l$). L'expressió algèbrica de la seqüència de sortida és la següent:

$$s_t^f = \sum_{i=1}^{f} s_{dt+i}^n \bullet s_{t+i}^n$$



El factor de velocitat d és variable i es fa servir com una part de la clau de xifratge.

Pel que fa a la complexitat lineal de les seqüències generades, val $CL = nl$ sempre que els dos generadors lineals tinguin polinomis de connexions primitius, mcd(l,n) = 1 i mcd(d,$2^n - 1$) = 1. El període de la seqüència generada val: $p = (2^n - 1)(2^l - 1)$.

Resum

En aquest mòdul didàctic hem descrit el funcionament del xifratge de flux i hem estudiat les propietats que ha de tenir una seqüència aleatòria perquè es pugui utilitzar com a seqüència de xifratge.

Hem presentat igualment diferents tipus de generadors per a obtenir seqüències pseudoaleatòries. Hem assenyalat que els registres de desplaçament realimentats linealment (LFSR) eren els més interessants perquè són fàcils d'estudiar, tot i que, com ja hem apuntat, no n'aconsellem l'aplicació en criptografia perquè la seva criptoanàlisi és força senzilla.

Finalment, hem estudiat diferents tipus de generadors que fan servir com a base els registres de desplaçament realimentats linealment (LFSR), per exemple els filtres i els combinadors, encara que no tots complien els requisits necessaris per a generar seqüències segures per a la criptografia.

Activitats

1. Genereu una seqüència pseudoaleatòria de 64 bits amb el generador de nombres aleatoris de la vostra calculadora. Per a generar un bit, pitgeu la tecla RAN# i arrodoniu el resultat per a obtenir un 0 o un 1. Sobre la seqüència obtinguda, feu les operacions següents:
a) Calculeu-ne el període.
b) Comproveu si es compleixen els tres postulats de Golomb.
c) Avalueu-ne la imprevisibilitat.

2. Simuleu el generador de Geffe amb els LFSR de quatre cel·les amb polinomis de connexions dels LFSR primitius, i comproveu que es compleix la relació:

$$P(s^f = s^1) = P(s^f = s^2) = \frac{3}{4}.$$

Exercicis d'autoavaluació

1. Donada la seqüència de xifratge s = 000111101011001000111 amb un període p = 15, verifiqueu si compleix els tres postulats de Golomb.

2. Partint d'un LFSR de set cel·les amb un polinomi de connexions primitiu, calculeu:
a) el període de la seqüència resultant;
b) la complexitat lineal;

3. Trobeu el polinomi de connexions d'un LFSR de sis cel·les sabent que els primers dotze dígits de la seqüència de sortida són 110011011101.

4. Indiqueu els paràmetres que han de tenir els generadors següents:
a) Un generador de Beth-Piper, per a generar una seqüència de període 32.426.527.
b) Un generador de Massey-Rueppel, per a generar-ne una altra amb període 1.046.017.

Solucionari

Exercicis d'autoavaluació

1. Com que sabem que el període val 15, només ens cal prendre els primers 15 bits, que són 000111101011001.

Comprovem que la seqüència s sí que compleix els tres postulats de Golomb:

- El primer postulat es compleix, ja que al període p hi ha 7 zeros i 8 uns.
- Hi ha quatre ràfegues de longitud 1, dues ràfegues de longitud 2, una ràfega de longitud 3 i una ràfega de longitud 4, per la qual cosa es compleix el segon postulat.
- La funció d'autocorrelació de la seqüència per a tos els valor de k no múltiples de 15 val:

$$AC(k) = \frac{7 - 8}{15} = \frac{-1}{15},$$

que és la condició necessària perquè es compleixi el tercer postulat.

Verifiquem per a algunes de les seqüències desplaçades k bits, s_k, el nombre de bits coincidents i no coincidents amb la seqüència inicial s = 000111101011001:

- $k = 1$; s_1 = 001111010110010. Per a trobar el nombre de bits coincidents amb la seqüència inicial calculem: $s \oplus s_1$ = 001000111101011.
- $k = 2$; s_2 = 011110101100100, $s \oplus s_2$ = 011001000111101.
- $k = 4$; s_4 = 111010110010001, $s \oplus s_4$ = 111101011001000.
- $k = 7$; s_7 = 010110010001111, $s \oplus s_7$ = 010001111010110.

2.

a) Tenint en compte que el polinomi de connexions de l'LFSR és primitiu, el valor del període és màxim i val $2^7 - 1$.

b) La complexitat lineal és el nombre de cel·les de l'LFSR més curt capaç de generar-lo. Per tant, $CL = 7$.

3. Per a obtenir el polinomi de connexions que correspon a l'LFSR amb les *2n* dades de què disposem, només cal plantejar el sistema d'equacions corresponent. En aquest cas:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_6 \\ c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

Un cop resolt, veiem que el polinomi de connexions val:

$$f(x) = 1 + x + x^6.$$

4.

a) En el cas de Beth-Piper, el període quedava determinat per l'expressió:

$$p = (2^{n_0} - 1)(2^{n_1} - 1)(2^{n_2} - 1),$$

i així tenim:

$$32.426.527 = (2^{n_0} - 1)(2^{n_1} - 1)(2^{n_2} - 1).$$

Si calculem quins períodes donen els LFSR amb un polinomi primitiu de grau petit tenim:

n	$2^n - 1$
3	7
4	15
5	31
6	63
7	127

(Continua a la pàgina següent.)

n	$2^n - 1$
8	255
9	511
10	1.023
11	2.047

Si descomponem 32.426.527 en factors primers tenim que 32.426.527 = 7 · 23 · 31 · 73 · · 89. Observeu que els podem agrupar de la manera següent: 31 · (7 · 73) · (23 · 89) = 31 · · 511 · 2.047. Per tant, si prenem un generador de Beth-Piper amb polinomis de grau 5, 9 i 11 obtindrem el període que ens demanen.

b) En el cas d'un generador de Massey-Rueppel ocorre el mateix, però en aquest cas l'expressió és:

$$p = (2^n - 1)(2^l - 1),$$

i substituint les dades conegudes obtenim:

$$1.046.017 = (2^n - 1)(2^l - 1).$$

Descomponent p i agrupant convenientment els factors que en resulten, trobem:

$$1.046.017 = 7 \cdot 23 \cdot 73 \cdot 89 = (7 \cdot 73) \cdot (23 \cdot 89) = 511 \cdot 2.047.$$

Per tant, tenim $n = 11$ i $l = 9$.

Recordem, però, que l'expressió anterior només és vàlida si $\text{mcd}(l,n) = 1$ i $\text{mcd}(d,2^n - 1) = = 1$. En aquest cas es compleix que $\text{mcd}(9,11) = 1$, i només cal prendre $d \geq 2$ tal que $\text{mcd}(d,2.047) = 1$. Així, per exemple, per $n = 11$, $l = 9$ i $d = 15$ obtenim el generador demanat.

Glossari

Complexitat lineal d'una seqüència: nombre de cel·les de l'LFSR més curt que és capaç de generar una seqüència.

Criptosistema de clau compartida: criptosistema en què tant l'emissor com el receptor comparteixen una sola clau que fan servir tant per a xifrar com per a desxifrar.

Criptosistema de flux: sistema de xifratge que utilitza un generador pseudoaleatori per a xifrar un missatge, sumant bit a bit el text en clar amb la seqüència pseudoaleatòria que resulta del generador.

Estat d'un LFSR: conjunt de valors continguts en cada cel·la d'un LFSR en un instant de temps.

Funció d'autocorrelació d'una seqüència periòdica: nombre de coincidències menys nombre de no-coincidències entre la successió original i la mateixa successió desplaçada k posicions, dividit pel període de la seqüència original.

Generador lineal: generador de seqüències de bits que només executa operacions lineals sobre els elements d'entrada per a obtenir la seqüència de sortida.

Generador no lineal: generador de seqüències de bits que executa operacions no lineals, com ara permutacions, sobre els elements d'entrada per a obtenir la seqüència de sortida; a més, pot fer servir també operacions lineals.

Generador pseudoaleatori: procés determinista capaç de generar una seqüència pseudoaleatòria.

LFSR: registre de desplaçament realimentat linealment.

NLFSR: registre de desplaçament realimentat no linealment.

Període: enter més petit p tal que $s_{i+p} = s_i$ per a tot $i \geq 0$, en què $\{s_i\}_{i\geq 0}$ és una seqüència periòdica.

Polinomi de connexions d'un LFSR: polinomi que determina o que és determinat per la funció lineal de realimentació de l'LFSR.

Ràfega: conjunt de bits consecutius iguals dins una seqüència.

Registre de desplaçament realimentat linealment: dispositiu físic o lògic format per *n* cel·les de memòria i una funció de realimentació lineal.

Seqüència de xifratge: seqüència que resulta del generador pseudoaleatori en un criptosistema de flux.

Seqüència pseudoaleatòria: seqüència que compleix els tres postulats de Golomb.

Bibliografia

Bibliografia bàsica

Beker, H.; Piper, F. (1982). *Cipher systems, the protection of the communications*. Londres: Northwood Books.

Beth, T.; Piper, F. (1984). "The stop-and-go generator". *Advances in Cryptology - Eurocrypt'84* (pàg. 88-92). Berlín: Springer-Verlag.

Fúster, A.; de la Guía, D.; Hernández, L.; Montoya, F.; Muñoz, J. (1997). *Técnicas criptográficas de protección de datos*. Madrid: Ra-ma.

Geffe, P.R. (1973). "How to protect data with ciphers that are really hard to break". *Electronics* (pàg. 99-101).

Golomb, S.W. (1967). *Shift Register Sequences*. San Francisco: Holden-Day.

Massey, J.L. (1969). "Shift-register synthesis and BCH decoding". *IEEE Transactions on Information Theory* (IT-15, pàg.122-127).

Rueppel, R.A. (1986). *Analysis and design of streamciphers*. Berlín: Springer-Verlag.

Bibliografia complementària

Coveyou, R.R.; Mcpherson, R.D. (1967). "Fourier analysis of uniform random number generators". *J. Assoc. Comput. Mach* (núm. 14, pàg. 100-119).

Fredricksen, H. (1982). "A survey of full length nonlinear shift register cycle algorithms". *SIAM Journal on Applied Mathematics* (núm. 24(2), pàg. 195-221).

Gollmann, G., Chambers, W.G. (1989). "Clock-controlled shift registers: a review". *IEEE Journal on Selected Areas in Communications* (núm. 7(4), pàg. 525-533).

Knuth, D. (1981). *The art of Computer Programming*. Reading (Massachusetts): Addison-Wesley.

Maurer, U.M. (1991). "A universal statistical test for random bit generators". *Advances in Cryptology - Crypto '90* (pàg. 409-420). Nova York: Springer-Verlag.

Ronce, C.A. (1984). "Feedback Shift Registers". *LNCS 169*. Berlín: Springer-Verlag.

Xifres de clau compartida: xifres de bloc

Jordi Herrera Joancomartí

P03/05024/02262

UOC
www.uoc.edu

Índex

Introducció

Una alternativa al xifratge de flux és el que s'anomena **xifratge de bloc**. Aquest tipus de xifratge s'inclou també dins el grup de criptosistemes de clau compartida, ja que la clau que s'utilitza per a xifrar i desxifrar és la mateixa i la comparteixen emissor i receptor. La diferència bàsica entre el xifratge de flux i el xifratge de bloc és la utilització de memòria en els algorismes de xifratge. Es podria fer una analogia entre la criptografia i la codificació, en la qual el xifratge de flux seria l'equivalent als codis convolucionals, mentre que el xifratge de bloc correspondria a la codificació en bloc.

El xifratge de flux s'estudia al mòdul "Xifres de clau compartida: xifres de flux" d'aquesta assignatura.

El xifratge de flux empra una clau diferent per a cada bit d'informació. Aquesta clau depèn no solament de l'estat inicial del generador, sinó també de l'estat del generador en el moment de xifrar un bit concret. Per consegüent, dos bits iguals es podran xifrar de maneres diferents en funció de l'estat en què es trobi el generador. En canvi, en el xifratge de bloc això no passa, perquè les xifres de bloc actuen sense memòria i el text xifrat només pot dependre del text en clar i de la clau. Així, doncs, en el xifratge de bloc dos textos en clar iguals es xifren sempre de la mateixa manera quan s'utilitza la mateixa clau.

Cal estudiar aquest fet en detall, perquè els sistemes de xifratge que en resulten són força vulnerables i, si no es corregeixen, es podrien inserir o esborrar blocs de text xifrat sense que es pogués detectar. A més, el fet que dos blocs de text en clar quedin xifrats de la mateixa manera pot donar pistes en una possible criptoanàlisi de tipus estadístic.

Pel que fa a la utilització, els xifradors de bloc són força emprats, ja que aconsegueixen una velocitat acceptable de xifratge. En concret, el xifrador de bloc més utilitzat és el Data Encryption Standard (DES), pel fet que s'ha establert com a estàndard. Actualment, però, ha estat seleccionat l'Advanced Encryption Standard (AES), ja que en molts àmbits es considera que el DES (que va ser creat al final dels anys 70) no ofereix les mateixes garanties de seguretat que presentava quan va ser creat, tenint en compte la rapidesa amb què s'ha desenvolupat la tecnologia informàtica aquests darrers anys.

Objectius

En els materials didàctics facilitats en aquest mòdul l'estudiant trobarà les eines necessàries per a assolir els objectius següents:

1. Comprendre l'esquema general de les xifres de bloc i les característiques principals que els són comunes.

2. Entendre els principis de confusió i difusió, i també els diferents modes de xifratge de bloc.

3. Conèixer el funcionament dels criptosistemes de bloc més rellevants.

4. Obtenir uns coneixements bàsics en criptoanàlisi de sistemes de xifratge de bloc.

1. Estructura del xifratge de bloc

1.1. Els principis de confusió i de difusió de Shannon

Perquè un criptosistema sigui segur, cal que la informació que proporciona el text xifrat sigui la mínima possible. Això comporta, en particular, aconseguir que les propietats estadístiques del text en clar no es mantinguin en el text xifrat.

En un treball de formalització de la teoria de la informació, Shannon va proposar les dues tècniques que han de seguir els criptosistemes de xifratge de bloc si volen evitar eventuals atacs basats en mètodes estadístics.

Les dues tècniques proposades per Shannon en el seu treball són les que explicitem tot seguit:

- La **confusió**, que inclou substitucions per tal que la relació entre la clau i el text xifrat sigui tan complicada com es pugui.
- La **difusió**, que utilitza transformacions que dissipen les propietats estadístiques del text en clar entre el text xifrat.

1.2. Característiques comunes

Des d'un punt de vista general, tots els esquemes de xifratge de bloc tenen la mateixa estructura bàsica de funcionament. Tots aquests, a partir d'un bloc de text en clar, B, d'una longitud fixada, i una clau, K, executen determinades operacions més o menys complicades fins a obtenir el text xifrat corresponent, Y, de manera que:

$$Y = E_K(B).$$

A més, les operacions anteriors han de permetre que es pugui tornar a obtenir el mateix text en clar a partir del text xifrat Y i la clau K, si es porta a terme el procés de desxifratge següent:

$$B = D_K(Y).$$

Així mateix, les operacions que s'executen sobre el bloc que cal xifrar només combinen els mateixos elements que hi ha en el bloc i en la clau. Això ocorre

perquè aquests mètodes de xifratge no incorporen memòria, de manera que, en el cas que la clau K sigui la mateixa, dos blocs que siguin idèntics produiran textos xifrats idèntics.

Exemples de criptosistemes de bloc...

... són les xifres de substitució, ja sigui aquesta simple o homofònica, i les xifres de transposició.

Normalment, l'estructura bàsica dels criptosistemes de bloc moderns consta de dues transformacions, una d'entrada i una altra de sortida, entre les quals hi ha un nombre determinat d'iteracions d'una certa funció f, no lineal, que combina els elements que formen part del bloc de text en clar amb els elements que formen part de la clau. De fet, generalment la clau, K, s'utilitza per a generar un seguit de subclaus, K_i, a partir d'una certa funció f_K prou complicada, i són aquestes subclaus les que actuen en cada iteració.

El mode bàsic de funcionament d'un criptosistema de bloc és el que es coneix amb la sigla **ECB***. Com que la majoria de vegades el text que s'ha de xifrar té una llargada superior a la longitud del bloc amb el qual treballa el criptosistema, el que es fa és partir el text que cal xifrar, M, en diversos blocs, M_1, M_2, ..., cada un dels quals té la llargada corresponent al bloc per a xifrar. D'aquesta manera es xifra cada un dels blocs amb una mateixa clau, K, per a obtenir el text xifrat resultant: $Y = Y_1Y_2$ ...

*** ECB és la sigla d'*Electronic Code Book*.**

El nom d'ECB no és casual, sinó que es refereix al fet que, una vegada fixada una clau, K, a cada bloc de text en clar correspon un bloc concret de text xifrat, com si tinguéssim un diccionari per a cada clau K i anéssim buscant quina paraula (bloc) de text xifrat correspon a cada paraula de text en clar.

Encara que el mode ECB és el que sembla més natural d'utilitzar, des de la perspectiva de la seguretat té força inconvenients: el fet que el xifratge de dos blocs sigui totalment independent fa que sigui vulnerable a determinats atacs. Per exemple:

a) Podria ser que un atacant esborrés blocs de text xifrat, i com que això no afectaria el procés de desxifratge de la resta, el receptor no ho podria detectar.

b) L'atacant podria inserir blocs de text xifrat.

c) El fet que els mateixos blocs quedin xifrats sempre de la mateixa manera pot facilitar els atacs de tipus estadístic per a obtenir la clau K.

1.3. Modes de xifratge de bloc

Pel que hem vist fins ara, els xifratges de bloc poden ser útils per a xifrar informacions curtes, com ara identificadors, contrasenyes, claus, etc.; en definiti-

va, els missatges en què la llargada del text que es vol xifrar no sobrepassa la llargada del bloc.

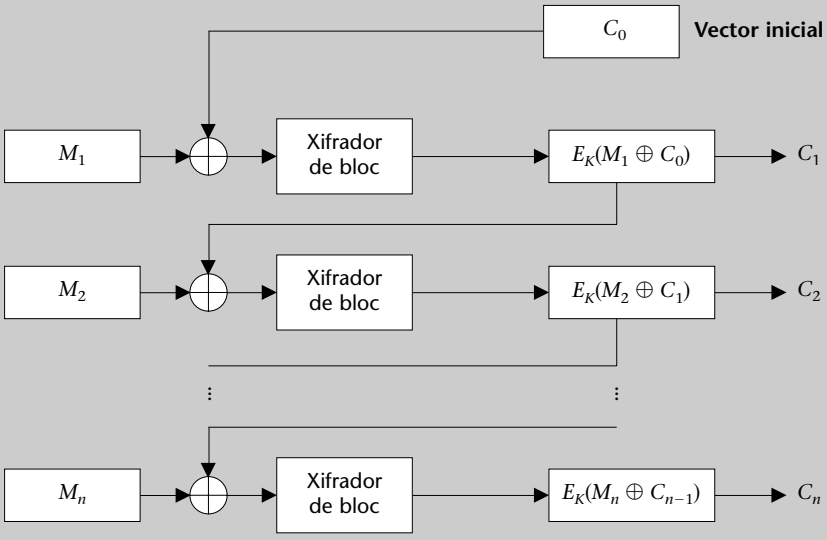
En canvi, la utilització del xifratge de bloc mitjançant el mode ECB sobre missatges llargs i, en especial, en textos que repeteixen determinats patrons, es desaconsella totalment. Així, doncs, si volem usar els xifratges de bloc en aquest tipus de textos caldrà aplicar-hi algun dels modes de xifratge que explicarem a continuació.

Mode CBC

El CBC* consisteix en l'encadenament dels blocs per al xifratge, de manera que es creï una dependència del xifratge de cada bloc amb l'immediat anterior, tal com mostra la figura següent:

*** CBC és la sigla de *Cipher Block Chaining*.**

Esquema del mode CBC



Suposem un xifratge de bloc amb una clau K, una funció de xifratge E i una de desxifratge D. Si $M_1, ..., M_n$ són els blocs de text en clar que cal xifrar, mitjançant el sistema CBC el xifratge del bloc M_i es porta a terme de la manera següent:

$$C_i = E_K(M_i \oplus C_{i-1}).$$

Per a fer-ne el desxiframent també ens cal partir del text xifrat anterior, i aleshores hem d'executar l'operació següent:

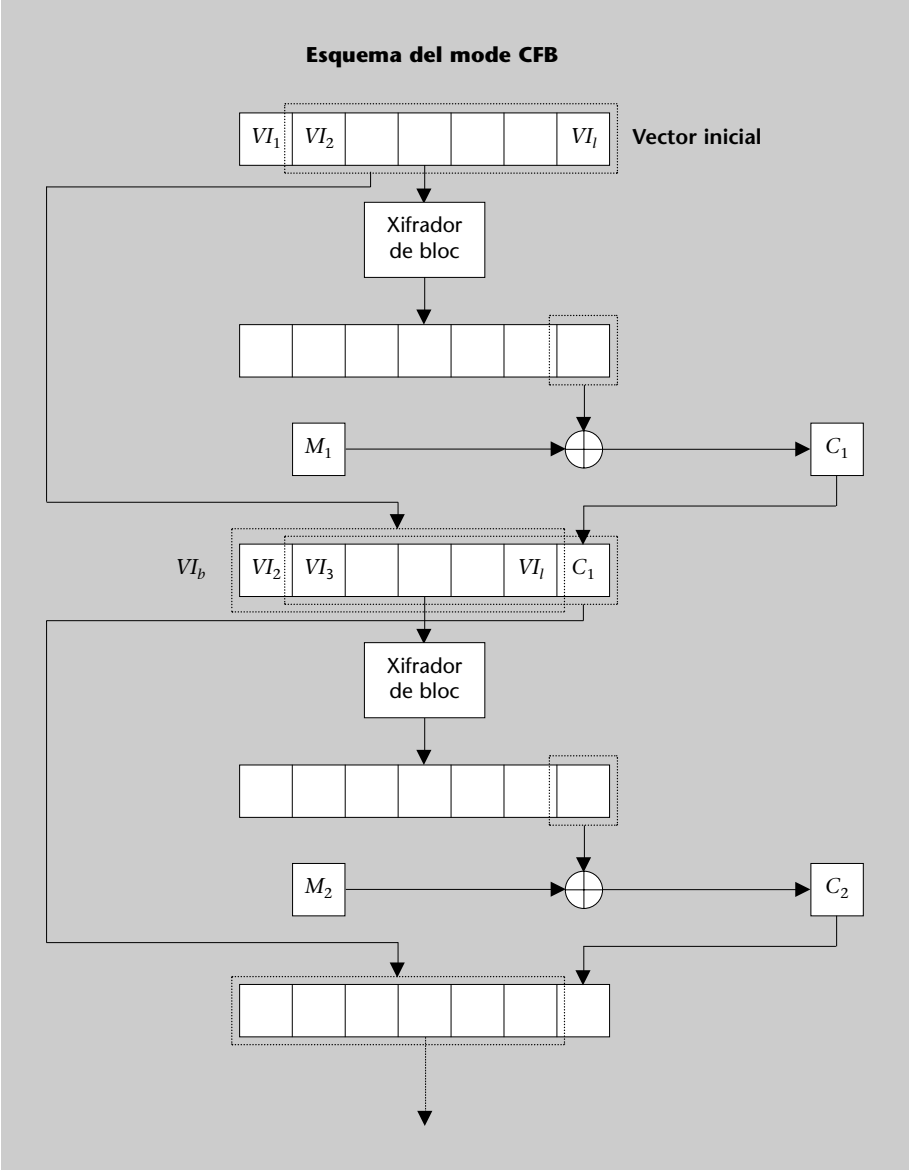
$$D_K(C_i) \oplus C_{i-1} = D_K(E_K(M_i \oplus C_{i-1})) \oplus C_{i-1} =$$
$$= (M_i \oplus C_{i-1}) \oplus C_{i-1} = M_i.$$

Per a xifrar el primer bloc necessitarem un bloc inicial aleatori, C_0, que no cal que sigui secret. Aleshores, incloent aquest nou vector inicial en el xifratge podrem obtenir dos textos en clar iguals però xifrats de manera diferent; així, encara que emprem la mateixa clau, K, només ens caldrà canviar el vector inicial, C_0, que, a més, pot incorporar una marca temporal.

Mode CFB

El mode de xifratge CFB* utilitza indirectament el xifrador de bloc, com veurem a continuació. Per això, la llargada dels blocs que s'han de xifrar no cal que sigui la mateixa que la dels blocs del criptosistema amb què actua, sinó que pot ser més petita. L'esquema general de funcionament d'aquest mètode es mostra a la figura que presentem tot seguit:

* CFB és l'acrònim de *Cipher Feedback*.



Donat $M = M_1 M_2 ...$, en què M és el missatge de text en clar, i M_1, M_2, ... representen els blocs de longitud n que formen el missatge, si considerem el vector

inicial *VI* com una concatenació de *l* blocs de longitud *n*, és a dir, $VI = VI_1\ VI_2\ ...\ VI_l$, on VI_i té *n* bits de llargada, podrem calcular el xifratge del vector *VI*, *E*(*VI*), mitjançant el criptosistema de bloc.

El resultat tindrà la mateixa llargada que *VI* i, per tant, el podrem descompondre de la mateixa manera que aquell:

$$E(VI) = E(VI)_1\ E(VI)_2\ ...\ E(VI)_l.$$

Finalment, ja podrem xifrar el primer bloc de text en clar, M_1, fent la suma bit a bit amb el darrer bloc, $E(VI)_l$:

$$C_1 = M_1 \oplus E(VI)_l;$$

obtenim així el primer bloc xifrat de longitud *n*, C_1.

Per a xifrar el segon bloc, M_2, tornarem a fer el mateix procés, però aquesta vegada prendrem com a vector inicial el vector format pels fragments següents:

$$VI_b = VI_2\ VI_3\ ...\ VI_l\ C_1,$$

és a dir, hem desplaçat els blocs de *n* bits cap a l'esquerra per afegir-hi el bloc C_1 i descartar-ne el VI_1. D'aquesta manera, el segon bloc de text xifrat l'obtenim fent l'operació següent:

$$C_2 = E(VI_b)_l \oplus M_2.$$

El procés es repeteix al llarg dels blocs de text que es vol xifrar: per al bloc següent es desplacen els blocs del vector inicial anterior, VI_b, ... a l'esquerra per afegir-hi el darrer bloc de text xifrat obtingut i anar aplicant el que ja hem descrit anteriorment.

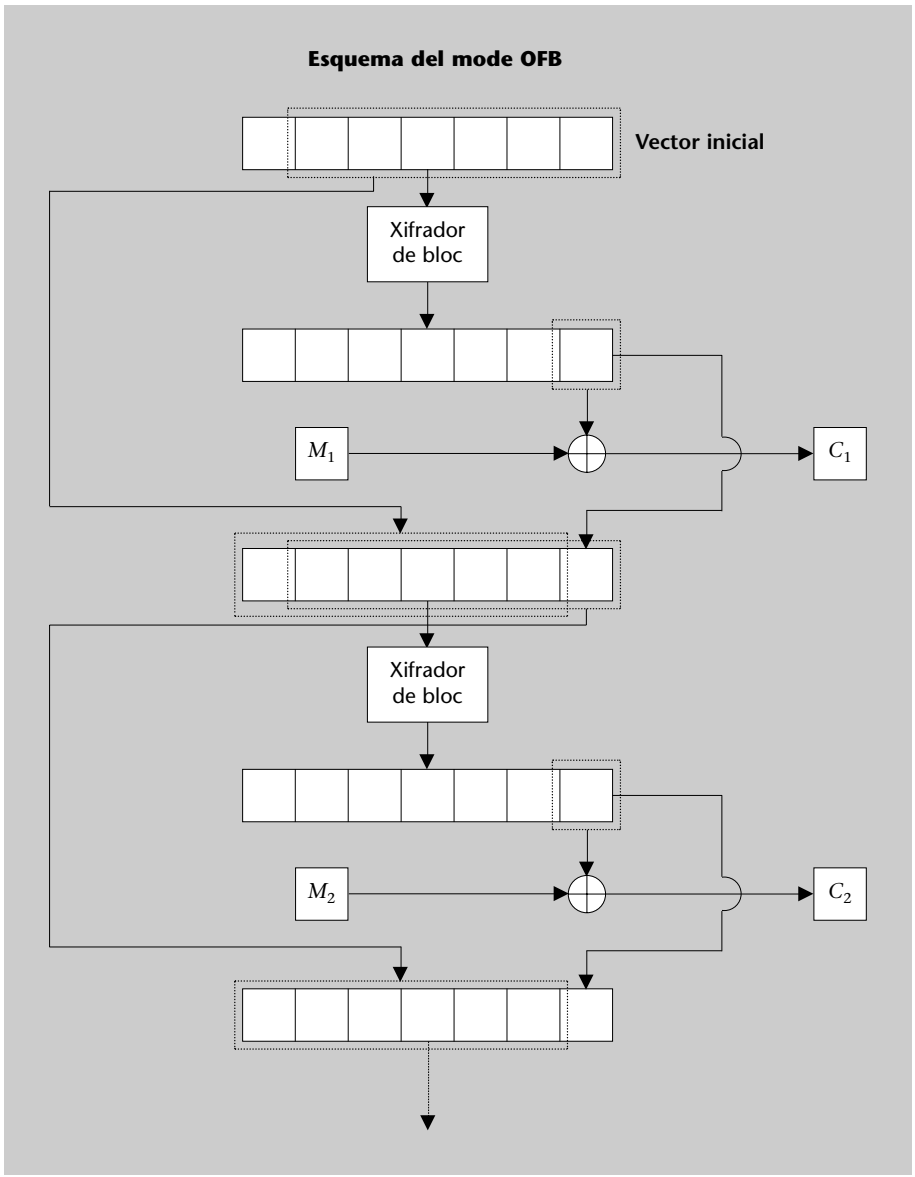
Mode OFB

El mode de xifratge OFB* utilitza el criptosistema de bloc com a generador pseudoaleatori. És un sistema molt semblant a l'anterior; l'única diferència que presenta és que el vector inicial es realimenta directament amb el resultat del xifratge de bloc abans de fer la suma bit a bit amb el bloc de text en clar, com es pot veure a la figura de la pàgina següent.

*** OFB és l'acrònim d'*Output Feedback*.**

Com que el xifrador de bloc actua com un generador pseudoaleatori, cal que els criptosistemes de bloc que emprem amb el mode OFB compleixin les característiques requerides per als generadors pseudoaleatoris, tant pel que fa a la impredictibilitat de la seqüència resultant com a la complexitat lineal.

Vegeu els generadors pseudoaleatoris a l'apartat 1 del mòdul "Xifres de clau compartida: xifres de flux" d'aquesta assignatura.


Esquema del mode OFB
Vector inicial
Xifrador de bloc
M_1
C_1
Xifrador de bloc
M_2
C_2

2. Criptosistemes de xifratge de bloc

2.1. L'estàndard DES

L'any 1977, el National Bureau of Standards (NBS), una secció del Departament de Defensa dels EUA, va publicar un criptosistema estàndard creat amb la finalitat de protegir qualsevol tipus de dades: el DES*. En el desenvolupament d'aquest sistema van participar l'empresa IBM i la National Security Agency (NSA).

*** DES és la sigla de *Data Encryption Standard.***

El **criptosistema DES** és un criptosistema de xifratge de bloc que xifra blocs de dades de 64 bits de llargada per mitjà d'una clau de 56 bits. El DES aconsegueix complir tant el principi de confusió com el de difusió, gràcies a les accions de les caixes *S* que conté.

Vegeu la descripció de les caixes *S* al subapartat 2.1.2 d'aquest mòdul didàctic.

2.1.1. Descripció del funcionament

El funcionament de l'algorisme de xifratge i desxifratge DES queda determinat a la figura de la pàgina següent. Com podem observar, se subministra a l'algorisme un bloc d'entrada, *M*, sobre el qual s'aplica una permutació inicial, σ, d'on s'obté $T_0 = \sigma(M)$. Quan ja s'han fet les 16 iteracions de la funció f, que explicarem a continuació, el resultat es transposa per mitjà de la permutació de sortida σ^{-1}. Les permutacions σ i σ^{-1} es recullen en dues taules:

La interpretació d'aquestes taules...

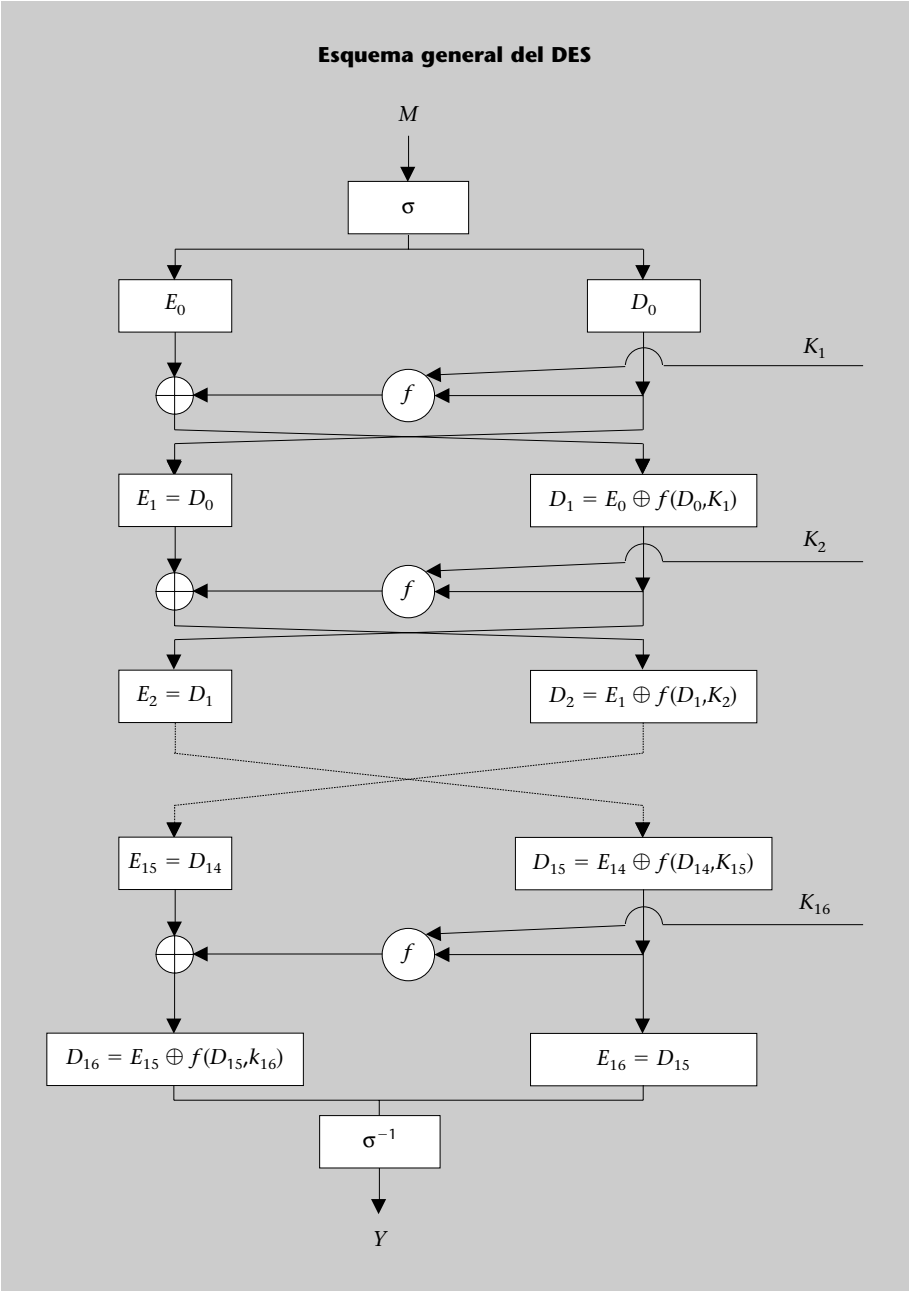
... és molt senzilla.
Per exemple, la taula corresponent a la permutació σ ens indica que el primer bit de sortida correspon al bit 58 d'entrada, el segon, al 50 d'entrada, al tercer al 42, etc.

- Taula de permutacions σ:

Taula de permutacions σ							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

- Taula de permutacions σ^{-1}:

Taula de permutacions σ^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25



2.1.2. Detall d'una iteració

Entre les dues permutacions, σ i σ^{-1}, l'algorisme efectua 16 iteracions de la funció f que combinen substitucions i transposicions.

Si suposem que T_i és el resultat de la i-èsima iteració, T_i té 64 bits de llargada i, en conseqüència, el podem dividir en dues meitats iguals de 32 bits de manera que $T_i = E_i\, D_i$, en què la part esquerra és $E_i = t_1 \ldots t_{32}$ i la dreta és $D_i = t_{33} \ldots t_{64}$; aleshores, per a cada iteració es verificaran les condicions que presentem a continuació:

- $E_i = D_{i-1}$,

- $D_i = E_{i-1} \oplus f(D_{i-1},K_i)$,

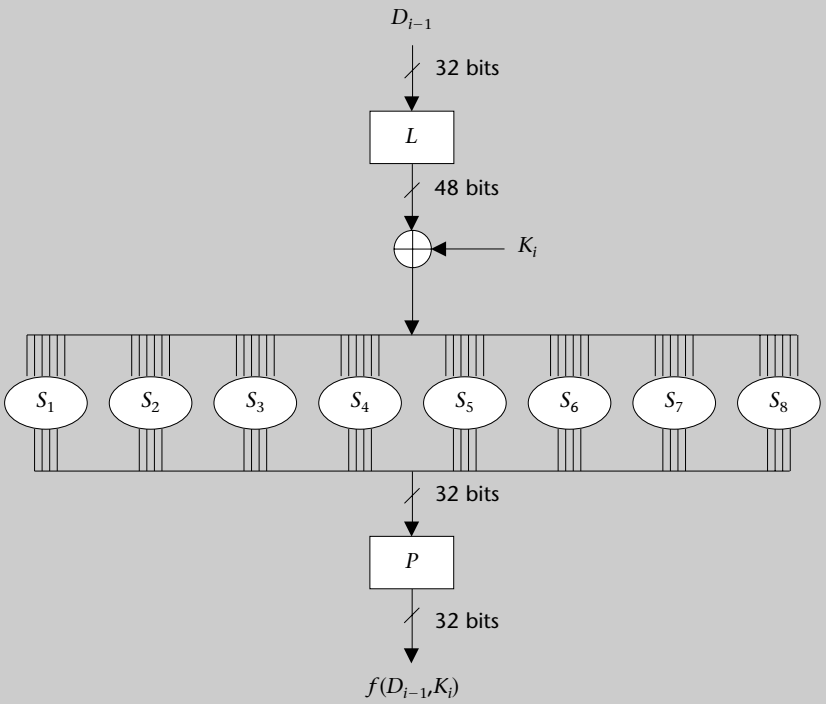
Podeu veure com es calcula K_i al subapartat 2.1.3 d'aquest mòdul didàctic.

en què K_i és una clau de 48 bits de llargada, que descriurem més endavant.

Fem notar que en l'esquema general del DES no s'intercanvien els costats de les parts en la darrera iteració, sinó que es fa la permutació inversa de la inicial, σ^{-1}, per a obtenir la sortida final, Y. D'aquesta manera, el mateix algorisme serveix alhora per a xifrar i per a desxifrar.

El valor que s'obté de la funció, $f(D_{i-1},K_i)$, es pot entendre amb la figura que presentem a continuació:

Descripció de la funció *f*



En primer lloc es transforma D_{i-1} de manera que tingui una llargada de 48 bits*. Per a estendre D_{i-1} s'han repetit alguns dels seus bits per mitjà de la transformació d'expansió, L, que queda determinada a la taula següent:

*** Recordeu que D_{i-1} és la meitat dreta de T_{i-1} i, per tant, només té 32 bits.**

Taula de transformació *L*

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

A la taula...

... el primer bit de sortida correspon al bit 32 d'entrada; el segon bit de sortida, a l'1 d'entrada, etc.

A continuació, seguint l'esquema del DES, fem una suma bit a bit de la meitat expandida, $L(D_{i-1})$, amb la clau K_i.

El resultat d'aquesta operació el descomponem en vuit blocs de 6 bits cada un, és a dir:

$$L(D_{i-1}) \oplus K_i = B_1\, B_2\, ...\, B_8.$$

Cada bloc, B_j, s'usa com a entrada del que s'anomena una caixa S. Cada caixa, S_j, rep el seu bloc corresponent de 6 bits $B_j = b_1\, b_2\, b_3\, b_4\, b_5\, b_6$, i en retorna un de 4 bits de llargada, d'acord amb la taula següent:

		Columna															
Caixa *S*	Fila	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_1	**0**	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	**0**	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	12	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	**0**	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	**0**	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	**0**	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	**0**	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

(Continua a la pàgina següent.)

		Columna															
Caixa *S*	Fila	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_7	**0**	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	**0**	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

El criteri per mitjà del qual s'assigna una fila i una columna d'una caixa a B_j és el següent: l'índex j fixa la caixa S_j, l'enter corresponent a $b_1\ b_6$ selecciona la fila i l'enter que correspon a $b_2\ b_3\ b_4\ b_5$ determina la columna.

Exemple de transformació en bloc

Per a transformar un bloc donat, $B_1 = 011001$, S_1 retornarà el valor de la fila 1 (ja que $01_2 = 1$) i la columna 12 (ja que $1100_2 = 12$). Si ens fixem en la taula de les caixes *S* hi ha un 9, que en binari és 1001.

A continuació es prenen els blocs resultants de les caixes *S* i es concatenen fins a obtenir un bloc de 32 bits de llargada. Finalment, s'aplica sobre aquest darrer bloc la permutació *P* definida a la taula següent per tal d'obtenir el valor $f(D_{i-1},K_i)$.

Taula de permutacions *P*			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

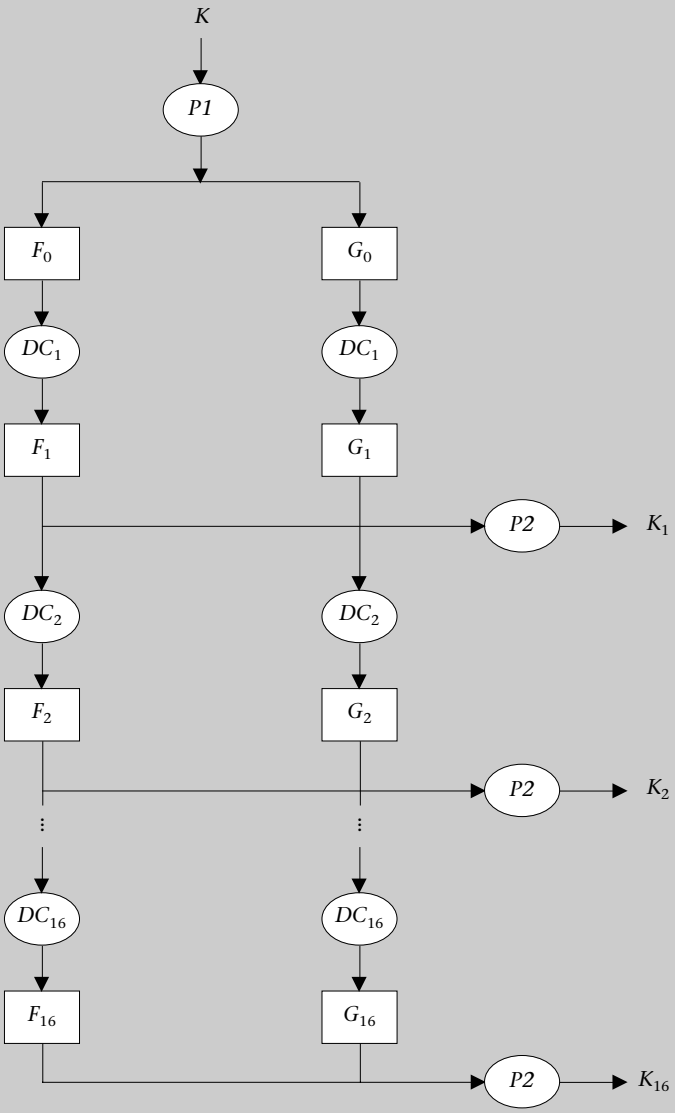
2.1.3. Generació de subclaus

En començar la descripció del criptosistema DES hem assenyalat que operava amb una clau, *K*, de 56 bits de llargada. En canvi, l'algorisme que hem descrit utilitza 16 subclaus K_i de 48 bits. Vegem, doncs, com es poden obtenir aquestes subclaus a partir de la clau principal. L'algorisme de generació de subclaus s'il·lustra a la figura de la pàgina següent.

A la clau inicial, *K*, de 56 bits s'hi han afegit 8 bits més en les posicions 8, 16, ..., 64, que corresponen a les paritats. La permutació *P1* descarta els bits de paritat i transposa els 56 restants.

El resultat de fer la permutació *P1*(*K*) és dividir *K* en dues meitats, F_0 i G_0, de 28 bits cadascuna. Els blocs F_0 i G_0 es desplacen cap a l'esquerra per a obtenir

Esquema de la generació de claus DES



Taula de permutacions *P1*

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Nota

En aquesta taula no hi figuren les posicions afegides 8, 16, ..., 64.

cada subclau K_i. Si representem per F_i i G_i els valors emprats per a obtenir K_i, tenim:

- $F_i = DC_i\,(F_{i-1})$,
- $G_i = DC_i\,(G_{i-1})$,

en què DC_i és un desplaçament circular cap a l'esquerra de tantes posicions com determina la taula de desplaçament següent:

Taula de desplaçaments *DC*	
Nombre d'iteració *i*	**Desplaçaments *i* a l'esquerra**
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Finalment, la subclau K_i s'obté per mitjà de l'expressió:

$$K_i = P2(F_i\ G_i),$$

en què *P2* és la permutació especificada a la taula que presentem a continuació:

Taula de permutacions *P2*					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

2.1.4. Desxifratge

Fins aquí hem descrit el procés de xifratge del criptosistema DES; ara, per a desxifrar les dades d'un bloc *Y* xifrat amb una clau *K* farem servir el mateix

algorisme. En aquest cas, però, una vegada generades les subclaus de la mateixa manera com hem fet abans, les farem servir, però en ordre invers, en les iteracions que es mostren a la figura de l'esquema general del DES. Així, en la primera iteració s'utilitza K_{16}, en la segona K_{15}, fins a arribar a K_1, emprada en la darrera iteració.

2.1.5. Particularitats del criptosistema

El criptosistema DES es pot implementar tant en programari com en maquinari. Naturalment, la implementació en maquinari li confereix una velocitat de xifratge i de desxifratge molt superior a les implementacions fetes en programari.

Pel que fa als modes de xifratge de blocs, cal destacar que el DES pot treballar amb tots els modes (CBC, CFB, OFB) gràcies a la complexitat de les seves operacions, que permeten que pugui actuar com un generador pseudoaleatori.

Vegeu els modes de xifratge de bloc CBC, CFB i OFB al subapartat 1.3 d'aquest mòdul didàctic.

Des de l'aparició del DES l'any 1976, fins i tot abans que s'estandarditzés, ja van sorgir diferents veus crítiques sobre aquest criptosistema. Les dues febleses més greus que se li atribueixen són la poca llargada de la clau, de només 56 bits, i l'arbitrarietat de les caixes *S*, que podria obeir a l'existència d'una clau mestra que permetés desxifrar qualsevol missatge sense tenir-ne la clau original.

Pel que fa a la llargada de la clau, Diffie i Hellman van considerar que es podria construir un ordinador amb una arquitectura de processadors paral·lels que arribés a trencar el DES en un dia fent una cerca exhaustiva de claus. Aquestes estimacions, i d'altres fetes al començament dels anys vuitanta, sembla que no es van complir, si més no en medis amplis, ja que el cost que suposava construir un ordinador d'aquestes característiques era molt elevat. No obstant això, el temps ha passat i els avenços en matèria de tecnologia informàtica han estat molt importants, tant que han permès trencar sistemes, com ara el DES en mode ECB en 22 hores i 15 minuts.

Fragilitat de la xifra de 56 bits

Actualment, gairebé a l'entrada del nou segle, ha quedat demostrada la fragilitat dels sistemes que treballen amb claus de 56 bits, com ara el DES, ja que s'ha aconseguit trencar-lo en 22 hores i 15 minuts gràcies a un gran supercomputador i, això sí, amb l'ajut afegit d'uns cent mil ordinadors personals connectats per mitjà d'Internet. La clau del missatge desxifrat era 8558891AB0C851B6, i el missatge que amagava deia: "*Strong cryptography makes the world a safer place*" ('la criptografia forta fa del món un lloc més segur').

Per a resoldre el problema de la llargada de la clau i continuar usant el DES com a criptosistema segur, a l'espera que s'homologui un nou estàndard, s'utilitza el xifratge triple, que en el cas del DES és conegut com a *triple DES*.

Vegeu les característiques del xifratge triple al subapartat 3.2 d'aquest mòdul didàctic.

2.2. El criptosistema IDEA

El criptosistema IDEA* va ser desenvolupat per J. Massey i X. Lai l'any 1990. És un sistema que xifra blocs de text en clar de 64 bits de llargada per mitjà d'una clau de 128 bits. El seu funcionament es basa en vuit iteracions idèntiques seguides d'una transformació de sortida.

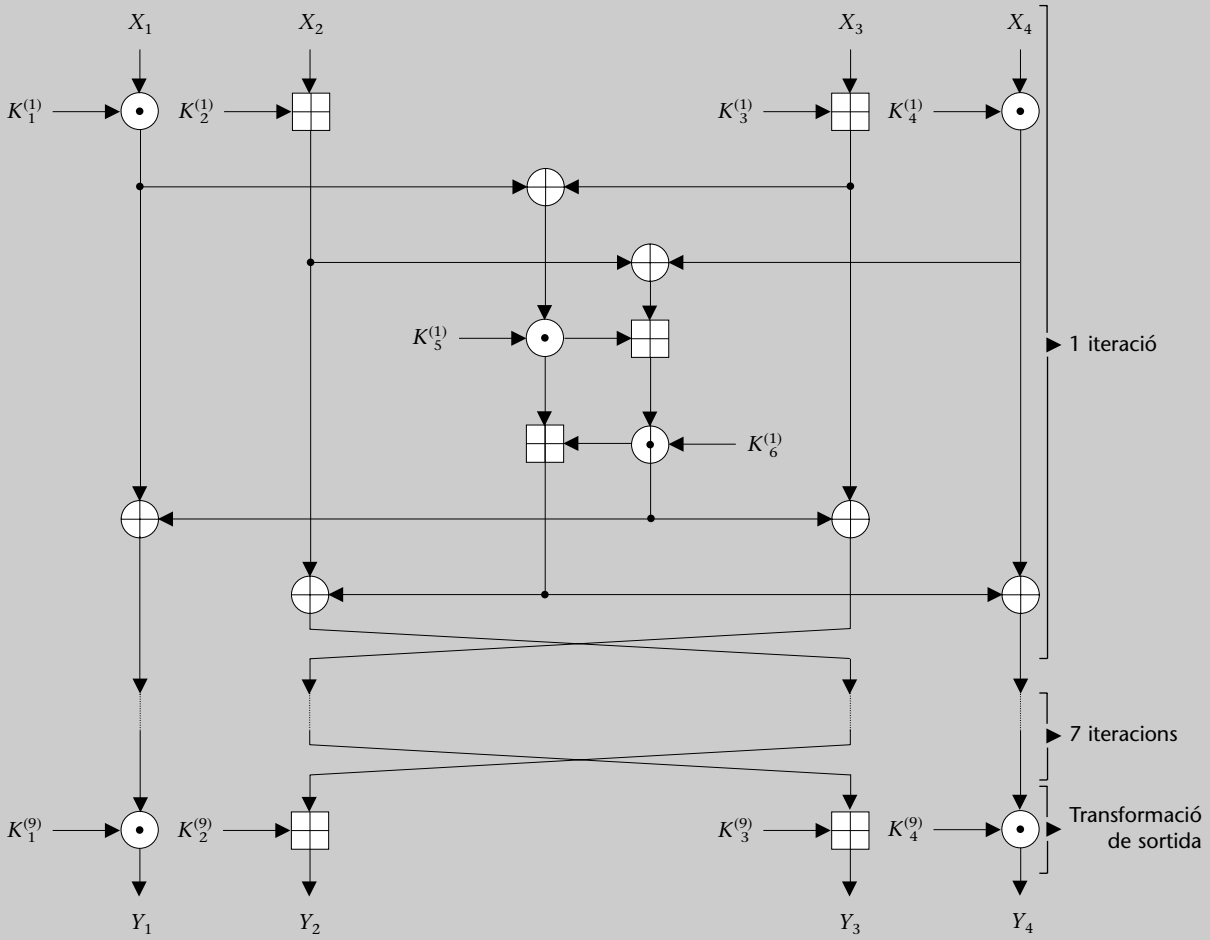
*** IDEA és la sigla d'*International Data Encryption Algorithm.***

En aquest criptosistema queden garantits els principis de confusió i difusió. Pel que fa als processos de xifratge i desxifratge, l'algorisme que s'hi utilitza és el mateix, però l'obtenció de les subclaus amb què operen és diferent.

2.2.1. Descripció del funcionament

L'esquema general de xifratge de l'IDEA el podem veure a la figura següent:

Esquema general de l'IDEA



Perquè l'IDEA sigui segur, s'executen tres operacions bàsiques de grups algèbrics diferents que actuen sobre blocs de 16 bits:

1) Operació XOR bit a bit de dos subblocs de 16 bits, que a la figura anterior es representa per $\oplus$.

2) Suma d'enters mòdul 2^{16}, on cada subbloc de 16 bits és tractat com un enter amb representació binària. En l'esquema, aquesta operació es representa per $\boxplus$.

3) Multiplicació d'enters mòdul $2^{16} + 1$. En aquest cas, el subbloc també s'interpreta com l'expressió en base dos d'un enter, excepte en el cas del subbloc format per tot zeros, que correspon al valor enter 2^{16}. L'operació es representa per $\odot$.

Operacions amb 2^{16} mod ($2^{16} + 1$)

Fixem-nos en el resultat de l'operació següent:

$$(0,...,0) \odot (1,0,...,0) = (1,0,...,0,1).$$

Aquest resultat es deu al fet que:

$$2^{16}\, 2^{15} \bmod (2^{16} + 1) = 2^{15} + 1.$$

2.2.2. Detall d'una iteració

Com es mostra a l'esquema de xifratge de l'IDEA, els 64 bits d'entrada se separen en quatre blocs de 16 bits de llargada cadascun: X_1, X_2, X_3, X_4. Aquests blocs, combinats amb les claus $K_1^{(i)}$, ..., $K_6^{(i)}$, en la i-èsima iteració corresponent acaben donant els blocs de text xifrat Y_1, Y_2, Y_3, Y_4.

2.2.3. Generació de subclaus

En el criptosistema IDEA, com en molts criptosistemes de bloc, es generen subclaus a partir de la clau inicial. En aquest cas, donats els 128 bits de la clau inicial, generarem 52 subclaus de 16 bits. Per a cada iteració de xifratge utilitzarem 6 subclaus, i 4 subclaus més per a la transformació de sortida.

Els processos de xifratge i desxifratge són essencialment els mateixos, ja que només varien les subclaus emprades en cada procés. Descriurem ara com s'obtenen les claus per al xifratge, i posteriorment obtindrem les del desxifratge a partir d'aquestes:

1) Per a generar les subclaus de xifratge, la clau de l'usuari, K, s'expandeix i s'obté una clau, K_{ext}, de 832 bits de llargada, que serà la concatenació de les claus que necessitem, és a dir:

$$K_{ext} = K_1^{(1)}, K_2^{(1)}, ..., K_6^{(1)}, K_1^{(2)}, ..., K_6^{(2)}, ..., K_1^{(8)}, ..., K_6^{(8)}, K_1^{(9)}, K_2^{(9)}, K_3^{(9)}, K_4^{(9)}.$$

L'expansió de la clau K a la clau K_{ext} es fa de la manera següent:

a) La clau inicial de 128 bits, K, forma íntegrament els primers 128 bits de la clau estesa. D'aquesta manera s'obtenen les primeres vuit subclaus de 16 bits:

$$K_1^{(1)}, K_2^{(1)}, ..., K_6^{(1)}, K_1^{(2)}, K_2^{(2)}.$$

b) Els 128 bits següents de K_{ext} s'obtenen fent una rotació cap a l'esquerra de 25 posicions de la clau inicial K; d'aquesta manera queden generades les vuit subclaus següents:

$$K_3^{(2)}, \ldots, K_6^{(2)}, K_1^{(3)}, \ldots, K_4^{(3)}.$$

c) Per a anar obtenint la resta de subclaus només cal anar fent rotacions de 25 posicions cap a l'esquerra del bloc de 128 bits obtingut de la rotació anterior. L'operació s'ha d'anar repetint fins a obtenir la darrera subclau, $K_4^{(9)}$.

2) Les claus utilitzades per al desxifratge s'obtenen a partir de les relacions amb les claus del xifratge, que es mostren a la taula següent:

Claus de xifratge	Claus de desxifratge
$K_1^{(1)}, K_2^{(1)}, K_3^{(1)}, K_4^{(1)}, K_5^{(1)}, K_6^{(1)}$	$K_1^{(9)^{-1}}, -K_2^{(9)}, -K_3^{(9)}, K_4^{(9)^{-1}}, K_5^{(8)}, K_6^{(8)}$
$K_1^{(2)}, K_2^{(2)}, K_3^{(2)}, K_4^{(2)}, K_5^{(2)}, K_6^{(2)}$	$K_1^{(8)^{-1}}, -K_3^{(8)}, -K_2^{(8)}, K_4^{(8)^{-1}}, K_5^{(7)}, K_6^{(7)}$
$K_1^{(3)}, K_2^{(3)}, K_3^{(3)}, K_4^{(3)}, K_5^{(3)}, K_6^{(3)}$	$K_1^{(7)^{-1}}, -K_3^{(7)}, -K_2^{(7)}, K_4^{(7)^{-1}}, K_5^{(6)}, K_6^{(6)}$
$K_1^{(4)}, K_2^{(4)}, K_3^{(4)}, K_4^{(4)}, K_5^{(4)}, K_6^{(4)}$	$K_1^{(6)^{-1}}, -K_3^{(6)}, -K_2^{(6)}, K_4^{(6)^{-1}}, K_5^{(5)}, K_6^{(5)}$
$K_1^{(5)}, K_2^{(5)}, K_3^{(5)}, K_4^{(5)}, K_5^{(5)}, K_6^{(5)}$	$K_1^{(5)^{-1}}, -K_3^{(5)}, -K_2^{(5)}, K_4^{(5)^{-1}}, K_5^{(4)}, K_6^{(4)}$
$K_1^{(6)}, K_2^{(6)}, K_3^{(6)}, K_4^{(6)}, K_5^{(6)}, K_6^{(6)}$	$K_1^{(4)^{-1}}, -K_3^{(4)}, -K_2^{(4)}, K_4^{(4)^{-1}}, K_5^{(3)}, K_6^{(3)}$
$K_1^{(7)}, K_2^{(7)}, K_3^{(7)}, K_4^{(7)}, K_5^{(7)}, K_6^{(7)}$	$K_1^{(3)^{-1}}, -K_3^{(3)}, -K_2^{(3)}, K_4^{(3)^{-1}}, K_5^{(2)}, K_6^{(2)}$
$K_1^{(8)}, K_2^{(8)}, K_3^{(8)}, K_4^{(8)}, K_5^{(8)}, K_6^{(8)}$	$K_1^{(2)^{-1}}, -K_3^{(2)}, -K_2^{(2)}, K_4^{(2)^{-1}}, K_5^{(1)}, K_6^{(1)}$
$K_1^{(9)}, K_2^{(9)}, K_3^{(9)}, K_4^{(9)}$	$K_1^{(1)^{-1}}, -K_2^{(1)}, -K_3^{(1)}, K_4^{(1)^{-1}}$

K^{-1} representa l'invers multiplicatiu mòdul $2^{16} + 1$ de K*, $-K$ és l'invers additiu**, i $K_1^{(i)}, \ldots, K_6^{(i)}$ són les claus del xifratge de la iteració i-èsima.

*** Això significa que $K \odot K^{-1} = 1$.**
**** Això significa que $-K \boxplus K = 0$.**

2.2.4. Particularitats del criptosistema

Les característiques més importants del criptosistema IDEA són les següents:

1) La llargada de la clau (128 bits) del criptosistema impossibilita una criptoanàlisi per cerca exhaustiva.

2) No utilitza res que s'assembli a les tan criticades caixes S que fa servir el DES, sinó que simplement executa operacions algèbriques de grups diferents. A més, com que algunes de les operacions que s'efectuen són modulars, pot ser més ràpid que no altres criptosistemes que actuen per mitjà d'operacions bit a bit.

3) Des del punt de vista de la seguretat, no es coneix cap atac capaç de trencar el criptosistema, el qual és resistent a la criptoanàlisi diferencial (només s'han pogut trencar per mitjà d'aquesta tècnica les dues primeres iteracions).

4) El criptosistema s'inclou en el programari PGP, que permet enviar correu electrònic d'una manera segura.

2.3. Propostes d'AES

Davant l'augment de la potència de càlcul dels ordinadors, i, d'altra banda, a causa de la controvèrsia que hi ha sobre la seguretat del DES en relació amb la llargada de la seva clau, es feia necessari substituir-lo. El National Institute of Standards and Technology (NIST) del govern nord-americà va fer una crida el setembre de 1997 perquè es presentessin propostes per a un nou estàndard de xifratge per tal de seleccionar l'Advanced Encription Standard (AES).

En la petició de propostes, el NIST va demanar un algorisme de xifratge de bloc que fos utilitzable tant en entorns governamentals com en entorns comercials. Es va especificar que hauria de suportar els modes estàndards CBC, CFB, OFB, caldria que fos "significativament" més eficient que el triple DES, i que la clau hauria de ser de longitud variable (128, 192 o 256 bits, com a mínim), i els blocs de xifratge, de longitud 128 bits.

El 20 d'agost del 1998, el NIST va fer pública la primera llista de quinze candidats a AES. Esmentem a continuació les quinze propostes seleccionades, especificant-ne el nom dels autors i algunes de les característiques més rellevants seleccionades:

Nom de la xifra	Fabricant i característiques
CAST-256	**Entrust Technologies, Inc.** Clau de 256 bits de longitud Xifratge de blocs de 128 bits 6 iteracions
Crypton	**Future Systems, Inc.** Clau variable de fins a 256 bits Xifratge de blocs de 128 bits 12 iteracions
DEAL	**Richard Outerbridge i Lars Knudsen** Clau variable de 128, 192 o 256 bits Xifratge de blocs de 128 bits 16 iteracions
DFC	**Centre National pour la Recherche Scientifique (CNRS)** Clau variable de fins a 256 bits Xifratge de blocs de 128 bits 8 iteracions
E2	**Nippon Telegraph and Telephone Corporation (NTT)** Clau de 128, 192 i 256 bits Xifratge de blocs de 128 bits 12 iteracions
Frog	**TecApro Internacional S.A.** Clau de 40 a 1.000 bits (en múltiples de 8) Xifratge de blocs des de 64 fins a 1.024 bits 8 iteracions
HPC	**Rich Schroeppel** Longitud de la clau de qualsevol nombre de bits Xifratge de blocs de qualsevol mida de bits Nombre variable d'iteracions

(Continua a la pàgina següent.)

Nom de la xifra	Fabricant i característiques
Loki97	**Lawrie Brown, Josef Pieprzyk i Jennifer Seberry** Clau de 128, 192 o 256 bits de longitud Xifratge de blocs de 128 bits 16 iteracions
Magenta	**Deutsche Telekom AG** Xifratge de blocs de 128 bits 6 iteracions
Mars	**IBM** Clau de longitud variable Xifratge de blocs de 128 bits
RC6	**Ronald Rivest, M.J.B. Robshaw, R. Sidney, Y.L. Yin** Clau de 16, 24 o 32 bytes Xifratge en blocs de 128 bits 20 iteracions
Rijndael	**Joan Daemen i Vincent Rijmen** Clau variable de 128, 192 o 256 bits Xifratge de blocs de 128, 192 o 256 bits El nombre d'iteracions depèn de la longitud de la clau i dels blocs. Varia entre 10 i 14 iteracions
SAFER+	**James L. Massey** Clau de 128, 192 o 256 bits Xifratge en blocs de 128, 192 o 256 bits El mateix nombre d'iteracions que SAFER
Serpent	**Ross Anderson, Eli Biham i Lars Knudsen** Clau de 256 bits Xifratge de blocs de 128 bits 32 iteracions
Twofish	**Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall i Niels Ferguson** Clau de 128, 192 o 256 bits Xifratge de blocs de 128 bits

L'agost del 1999, el NIST va fer pública la llista dels cinc finalistes entre les quinze propostes. Eren el Mars, RC6, Rijndael, Serpent i Twofish.

El 2 d'octubre del 2000 el NIST va fer pública la decisió de seleccionar el criptosistema Rijndael com a *Advanced Encryption Standard*. Els motius que el NIST va donar per a seleccionar l'AES eren la combinació de seguretat, rendiment, eficiència, facilitat d'implementació i flexibilitat. En particular, en van destacar l'eficiència, tant en implementacions de maquinari com de programari.

Finalment, el 26 de maig del 2002 el FIPS va anunciar l'aprovació de l'*Advanced Encryption Standard* sota el codi FIPS-197. Aquest estàndard especifica el Rijndael com a algorisme simètric de xifratge que les organitzacions del govern dels EUA (i altres) poden fer servir per a protegir informació sensible.

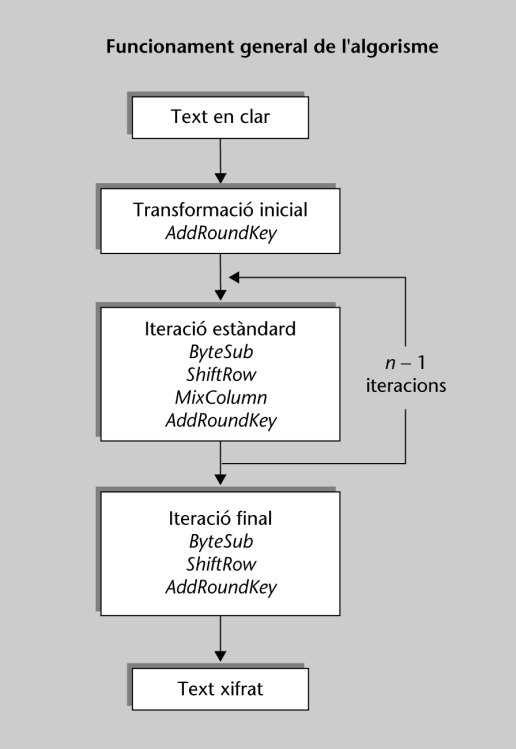
2.4. El criptosistema de Rijndael

L'any 1998, els criptògrafs belgues Vincent Rijmen i Joan Daemen van desenvolupar l'algorisme anomenat (en reconeixement dels autors) criptosistema de Rijndael. Aquest criptograma és el triat pel NIST com a AES.

El **criptosistema de Rijndael** xifra blocs de text en clar de 128, 192 o 256 bits de longitud, encara que l'estàndard aprovat pel NIST només fa servir blocs de 128 bits. La longitud de les claus de xifratge que aquest criptosistema empra també pot variar entre 128, 192 o 256 bits. Les operacions criptogràfiques es basen en un grup finit d'ordre 2^8.

2.4.1. Descripció del funcionament

El funcionament del criptosistema de Rijndael es mostra en la figura següent. Es basa en una transformació inicial seguida d'un nombre d'iteracions que varien entre 10 i 14, segons la longitud tant del bloc que es xifra com de la clau.



El nombre d'iteracions

El nombre d'iteracions que es mostren en el gràfic és $n - 1$ perquè la iteració final, tot i que es considera iteració, no conté la funció *mixColumns*.

La taula següent mostra el nombre exacte d'iteracions N en funció del nombre de paraules de 32 bits que tenen els blocs de text en clar (N_b), i de la clau (N_k):

		Longitud del bloc (N_b)		
		4	**6**	**8**
Longitud de la clau N_k	**4**	10	10	14
	6	12	12	14
	8	14	14	14

Blocs de 128 i 256 bits

Per a xifrar blocs de 128 bits amb claus de 128 bits s'hauran de fer 10 iteracions, mentre que per a xifrar blocs de 256 bits amb claus de 192 bits el nombre d'iteracions serà de 14.

La unitat bàsica d'informació amb què treballa el criptosistema de Rijndael és el byte. Totes les cadenes de bits (textos en clar i claus) es representen amb matrius de bytes. Per exemple, una cadena de 128 bits de text en clar:

$$input_0\ input_1\ input_2\ \ldots\ input_{126}\ input_{127}$$

es representarà amb 16 bytes de la manera següent:

$a_0 = [input_0\ input_1\ input_2\ input_3\ input_4\ input_5\ input_6\ input_7]$
$a_1 = [input_8\ input_9\ input_{10}\ input_{11}\ input_{12}\ input_{13}\ input_{14}\ input_{15}]$
...
$a_{15} = [input_{120}\ input_{121}\ input_{122}\ input_{123}\ input_{124}\ input_{125}\ input_{126}\ input_{127}]$

Aquests *bytes* es poden expressar en forma matricial:

$$\begin{bmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{bmatrix}$$

Les diferents funcions que executa el criptosistema de Rijndael (per exemple, *AddRoundKey*, *ByteSub*, etc.) tenen com a entrada i com a sortida una matriu de bytes com l'anterior.

Les matrius intermèdies amb què treballa el criptosistema de Rijndael s'anomenen *matrius d'estat*. Les **matrius d'estat** tenen 4 files i N_b columnes i cada element de la matriu és un byte. Els elements de cada estat es denoten per s_{ij}, on i determina la fila i j la columna.

Les operacions "suma" i "producte" de bytes que executa el criptosistema Rijndael no són les operacions convencionals que coneixem. En concret, l'algorisme Rijndael considera els bytes en una representació de polinomi. Cada byte b es pot representar amb 8 bits:

$$b = [b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0], \text{ on } b_i \in \{0, 1\}$$

Aquest conjunt de bits es pot expressar com els coeficients d'un polinomi de grau 7:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_i\, x^i$$

Exemple de representació polinòmica

El byte 01100011 té com a representació el polinomi $x^6 + x^5 + x + 1$.

Per tal de simplificar la notació, representarem els bytes en notació hexadecimal. Així, l'element 01100011 en base binària es representarà per un 63 en base hexadecimal, ja que $0110\ 0011_{(2} = 99_{(10} = 63_{(16}$.

Donades aquestes representacions, considerem que la "suma" i el "producte" es defineixen de la manera següent.

Siguin les representacions binàries dels bytes $x = (x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ i $y = (y_7, y_6, y_5, y_4, y_3, y_2, y_1, y_0)$.

definim l'**operació suma**:

$$x \oplus y = (x_7 \oplus y_7, x_6 \oplus y_6, x_5 \oplus y_5, x_4 \oplus y_4, x_3 \oplus y_3, x_2 \oplus y_2, x_1 \oplus y_1, x_0 \oplus y_0)$$

on $\oplus$ denota l'operació XOR bit a bit.

D'altra banda, definim l'**operació producte**:

$$x \otimes y = (b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0) \cdot (b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0) \bmod (x^8 + x^4 + x^3 + x + 1).$$

L'operació XOR

Recordeu que l'operació XOR queda definida per:

$1 \oplus 0 = 0 \oplus 1 = 1$,
$1 \oplus 1 = 0 \oplus 0 = 0$.

Aritmetica modular

La notació *mod* utilitzada en la definició de l'operació "suma" es deu al fet de treballar amb cossos finits. En el subapartat 1.1 del mòdul de "Xifres de clau pública" hi ha una descripció d'aritmètica modular per a elements enters. En aquest cas fem el mateix, però en comptes de treballar amb nombres enters, treballem amb polinomis. De cara als càlculs que trobareu en aquest subapartat, podeu prescindir dels conceptes teòrics i fer servir qualsevol eina informàtica de càlcul simbòlic, que permet treballar directament amb aritmètica modular.

Exemple de càlcul de "suma" i "producte":

Donats els bytes x i y:

$$x = 57 = 01010111 = x^6 + x^4 + x^2 + x + 1$$
$$y = 83 = 10000011 = x^7 + x + 1,$$

calculem la suma i el producte de bytes:

- $x \oplus y = 57 \oplus 83 = D4$, ja que: $01010111 \oplus 10000011 = 11010100 = D4$.
- $x \otimes y = 57 \otimes 83 = C1$, ja que:

$$(x^6 + x^4 + x^2 + x + 1) \cdot (x^7 + x + 1) \bmod (x^8 + x^4 + x^3 + x + 1) =$$
$$= (x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1) \bmod (x^8 + x^4 + x^3 + x + 1) =$$
$$= x^7 + x^6 + 1 = 11000001 = C1.$$

Els exemples d'aquest apartat

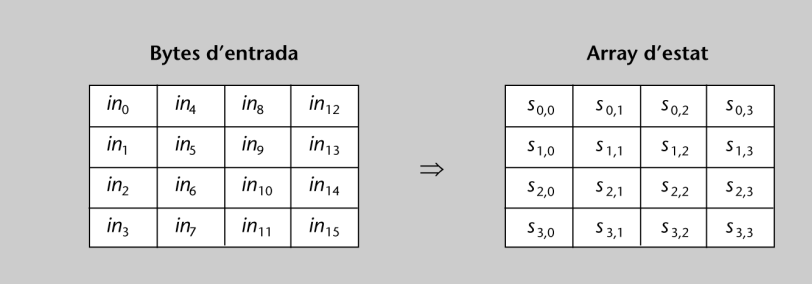
Per a tots els exemples de les funcions del criptosistema de Rijndael d'aquest mòdul suposarem el cas $N_b = 4$, és a dir que els blocs de text per xifrar són de 128 bits. De fet, aquesta longitud és la que especifica l'AES.

Un cop vistes aquestes representacions, ja podem passar a veure el funcionament de l'algorisme.

Abans de la primera iteració, el text en clar s'ha de convertir en una matriu d'estat. A continuació, el criptosistema de Rijndael executa una transformació anomenada *AddRoundKey*.

Transformació inicial

En el gràfic següent es mostra la transformació del text en clar en una matriu d'estat per a un cas amb $N_b = 4$ (és a dir, blocs per xifrar de 128 bits):



Sobre aquesta matriu d'estat s'aplica la funció *AddRoundKey*.

La funció ***AddRoundKey*** fa una suma XOR de la matriu d'estat amb cada byte de la subclau $K(i)$ corresponent. En el cas de la transformació inicial tenim, $i = 0$; per tant, utilitzem la primera subclau $K(0)$.

Les subclaus

K(i) denota la subclau de $(32 \cdot N_b)$ bits que es fa servir en la *i*-èsima iteració tenint en compte que $K(0)$ serà la subclau que es farà servir per a la transformació inicial. Podeu trobar la descripció de com s'obtenen les subclaus a partir de la clau inicial de xifratge en el subapartat 2.4.3 d'aquest mòdul.

Exemple de càlcul de la funció *AddRoundKey*

Considerem la subclau $K(0)$ i la matriu d'estat S:

$$K(0) = \text{B6 92 CF 0B 64 3D BD F1 BE 9B C5 00 68 30 B3 FE}$$

$$S = \begin{bmatrix} 9D & 28 & 91 & 00 \\ F7 & 7F & 78 & A6 \\ 39 & C1 & 6C & C6 \\ 3C & AA & 25 & A5 \end{bmatrix}$$

El resultat d'aplicar la funció *AddRoundKey* serà:

$$AddRoundKey(S, K(0)) = \begin{bmatrix} 9D & 28 & 91 & 00 \\ F7 & 7F & 78 & A6 \\ 39 & C1 & 6C & C6 \\ 3C & AA & 25 & A5 \end{bmatrix} \oplus \begin{bmatrix} B6 & 64 & BE & 68 \\ 92 & 3D & 9B & 30 \\ CF & BD & C5 & B3 \\ 0B & F1 & 00 & FE \end{bmatrix} = \begin{bmatrix} 2B & 4C & 2F & 68 \\ 65 & 42 & E3 & 96 \\ F6 & 7C & A9 & 75 \\ 37 & 5B & 25 & 5B \end{bmatrix}$$

Fixeu-vos que la suma XOR de les matrius correspon a la suma XOR de cada una de les seves entrades. Així, per exemple, la primera posició de la transformació val 2B, ja que 9D $\oplus$ B6 = 10011101 $\oplus$ 10110110 = 2B.

2.4.2. Detall d'una iteració

En el gràfic del funcionament general de l'algorisme es mostra que les $n - 1$ primeres iteracions executen les funcions *ByteSub*, *ShiftRow*, *MixColumn* i *AddRoundKey*, mentre que l'última iteració executa les primeres tres funcions, però no executa la funció *MixColumn*.

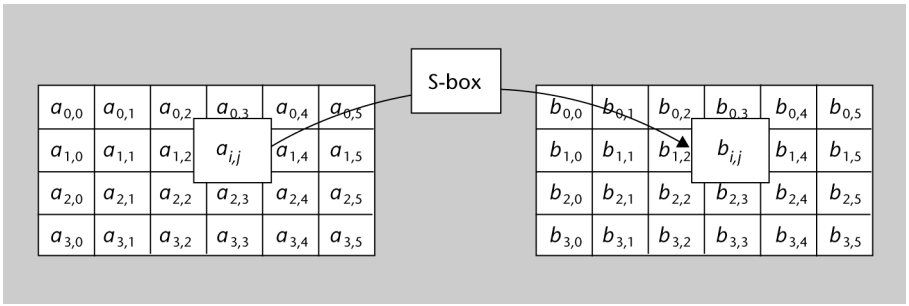
Passem a descriure cada una de les funcions que s'executen en cada iteració.

La funció *ByteSub* aplica una substitució no lineal dels bytes de la matriu d'estat.

Notació

La funció *ByteSub* apareix amb aquesta denominació a la proposta inicial del criptosistema de Rijndael. A la publicació de l'AES en l'estàndard Fip-197, la funció s'anomena *SubBytes*. Sigui quin sigui el nom que s'hi doni, en els dos casos és la mateixa funció.

La funció rep com a entrada una matriu d'estat **A**, hi aplica una transformació **S** i obté una altra matriu d'estat **B**, de manera que $b_{ij} = S(a_{ij})$, tal i com mostra l'esquema següent:



Més concretament, la taula de substitució *S*, que actua per a cada bit de la matriu d'estat, és invertible i es compon de dues transformacions:

1) Donat el byte x, s'expressa en forma polinòmica i se'n calcula l'invers multiplicatiu del polinomi resultant en l'anell de polinomis mòdul $(x^8 + x^4 + x^3 + x + 1)$. És a dir, es calcula el polinomi x^{-1}, tal que:

$$x \cdot x^{-1} = 1 \bmod (x^8 + x^4 + x^3 + x + 1).$$

Si $x = 0$, considerarem $x^{-1} = 0$.

2) El polinomi x^{-1} resultant es transforma a la notació de byte i s'hi aplica la transformació afí següent:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} x^{-1} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Exemple de càlcul de la funció *ByteSub*

Suposem que la matriu d'estat val:

$$S = \begin{bmatrix} B5 & B1 & B9 & B5 \\ C9 & CC & C5 & C8 \\ 17 & 11 & 1B & 15 \\ 9E & 99 & 92 & 9D \end{bmatrix}$$

Calculem la transformació de la primera entrada de la matriu, S_{11} = B5. El byte B5 expressat en bits queda: 10110101. A continuació el passem a l'expressió polinòmica: $x^7 + x^5 + x^4 + x^2 + 1$. Calculem l'invers d'aquest polinomi mòdul $x^8 + x^4 + x^3 + x + 1$, és a dir, el polinomi $b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$ tal que:

$$(x^7 + x^5 + x^4 + x^2 + 1) \cdot (b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0) = \\ = 1 \bmod (x^8 + x^4 + x^3 + x + 1)$$

Això ens dóna el polinomi $x^6 + x^5 + x^4 + x^2 + 1$, que té una representació en binari 01110101. Si ara apliquem el producte de la matriu de substitució:

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

Observeu que...

... la posició del valor x^{-1} com a vector columna té el bit menys significatiu a dalt de tot.

La transformació de la primera entrada de la matriu d'estat per la funció *ByteSub* val 11010101. Si passem a notació hexadecimal ens queda D5.

El resultat d'aplicar els càlculs a cada entrada de la matriu d'estat ens dóna la matriu següent:

$$ByteSub(S) = \begin{bmatrix} D5 & C8 & 56 & D5 \\ DD & 4B & A6 & E8 \\ F0 & 82 & AF & 59 \\ 0B & EE & 4F & 5E \end{bmatrix}$$

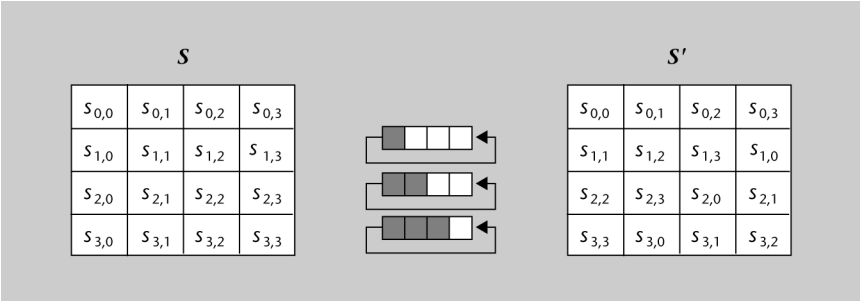
La funció ***ShiftRow*** desplaça les files de la matriu d'estat de manera que la fila zero es deixa igual, la fila 1 es desplaça $C1$ bytes a la dreta, la fila 2 es desplaça $C2$ bytes a la dreta i la fila 3, $C3$ bytes a la dreta. Els valors $C1$, $C2$ i $C3$ depenen de la longitud del block N_b.

Aquests valors s'especifiquen a la taula següent:

N_b	$C1$	$C2$	$C3$
4	1	2	3
6	1	2	3
8	1	3	4

Exemple de càlcul de la funció *ShiftRow*

Suposem que la longitud de la clau és de 128 bits, és a dir, $N_k = 4$. Com sempre, considerem que la longitud dels blocs és també de 128 bits, és a dir $N_b = 4$. Això vol dir que $C1 = 1$, $C2 = 2$ i $C3 = 3$ i, per tant, la transformació de l'estat per la funció *ShiftRow* es pot expressar gràficament com:



Per tant, la matriu d'estat S val:

$$S = \begin{bmatrix} D5 & C8 & 56 & D5 \\ DD & 4B & A6 & E8 \\ F0 & 82 & AF & 59 \\ 0B & EE & 4F & 5E \end{bmatrix}$$

La matriu d'estat transformada per la funció *ShiftRow* serà:

$$ShiftRow(S) = \begin{bmatrix} D5 & C8 & 56 & D5 \\ 4B & A6 & E8 & DD \\ AF & 59 & F0 & 82 \\ 5E & 0B & EE & 4F \end{bmatrix}$$

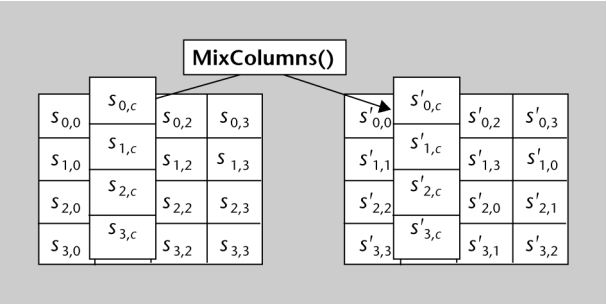
La funció ***MixColumns*** barreja les columnes de la matriu d'estat a partir d'operacions polinomials.

Concretament, aquesta funció considera les columnes de la matriu d'estat com polinomis de grau 3. Cada columna es multiplica pel polinomi $c(x) = '03'x^3 + + '01'x^2 + '01'x + '02'$ i el resultat es redueix mòdul $x^4 + 1$. El polinomi $c(x)$ és coprimer amb $x^4 + 1$ i, per tant, invertible. Aquest producte dels polinomis es pot escriure com un producte de matrius:

$$\begin{bmatrix} s'_{0j} \\ s'_{1j} \\ s'_{2j} \\ s'_{3j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0j} \\ s_{1j} \\ s_{2j} \\ s_{3j} \end{bmatrix} \qquad \forall 0 \leq j \leq N_b$$

Tingueu en compte que les operacions "suma" i "producte" entre els elements de la matriu i els del vector columna són les operacions $\oplus$ i $\otimes$ defini-

des en el subapartat anterior. Gràficament, la funció *MixColumns* fa la transformació següent:



Atenció amb el producte!

Adoneu-vos que aquest producte no és el producte estàndard de polinomis, sinó el definit en el subapartat 2.4.1 d'aquest mòdul.

Exemple de càlcul de la funció *MixColumns*

Suposem que la longitud dels blocs és de 128 bits, és a dir $N_b = 4$ i que la longitud de la clau és també de 128 bits, és a dir $N_k = 4$. A més, la matriu d'estat val:

$$S = \begin{bmatrix} D5 & C8 & 56 & D5 \\ 4B & A6 & E8 & DD \\ AF & 59 & F0 & 82 \\ 5E & 0B & EE & 4F \end{bmatrix}$$

Per a obtenir la transformació de la primera columna calcularem:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} D5 \\ 4B \\ AF \\ 5E \end{bmatrix}$$

Això ens donarà un vector columna de quatre bytes determinats pels valors següents:

$$\begin{bmatrix} (02 \otimes D5) \oplus (03 \otimes 4B) \oplus (01 \otimes AF) \oplus (01 \otimes 5E) \\ (01 \otimes D5) \oplus (02 \otimes 4B) \oplus (03 \otimes AF) \oplus (01 \otimes 5E) \\ (01 \otimes D5) \oplus (01 \otimes 4B) \oplus (02 \otimes AF) \oplus (03 \otimes 5E) \\ (03 \otimes D5) \oplus (01 \otimes 4B) \oplus (01 \otimes AF) \oplus (01 \otimes 5E) \end{bmatrix}$$

Per exemple, vegem quant val la segona posició del vector columna:

$$(01 \otimes D5) \oplus (02 \otimes 4B) \oplus (03 \otimes AF) \oplus (01 \otimes 5E)$$

Si passem els valors hexadecimals a representació polinòmica (passant per la seva representació binària) tenim:

Hexadecimal	Binari	Polinomi
01	00000001	1
D5	11010101	$x^7 + x^6 + x^4 + x^2 + 1$
02	00000010	x
4B	01001011	$x^6 + x^3 + x + 1$
03	00000011	$x + 1$
AF	10101111	$x^7 + x^5 + x^3 + x^2 + x + 1$
5E	01011110	$x^6 + x^4 + x^3 + x^2 + x$

Si ara fem els càlculs, resulta:

('01' ⊗ 'D5') =
$= (1)(x^7 + x^6 + x^4 + x^2 + 1) \bmod x^8 + x^4 + x^3 + x + 1 = x^7 + x^6 + x^4 + x^2 + 1 \Rightarrow$ 11010101

('02' ⊗ '4B') =
$= (x)(x^6 + x^3 + x + 1) \bmod x^8 + x^4 + x^3 + x + 1 = x^7 + x^4 + x^2 + x \Rightarrow$ 10010110

('03' ⊗ 'AF') =
$= (x + 1)(x^7 + x^5 + x^3 + x^2 + x + 1) \bmod x^8 + x^4 + x^3 + x + 1 = x^7 + x^6 + x^5 + x^3 + x \Rightarrow$
$\Rightarrow$ 11101010

('01' ⊗ '5E') =
$= (1)(x^6 + x^4 + x^3 + x^2 + x) \bmod x^8 + x^4 + x^3 + x + 1 = x^6 + x^4 + x^3 + x^2 + x \Rightarrow$ 01011110

Finalment, fem la XOR:

$$11010101 \oplus 10010110 \oplus 11101010 \oplus 01011110 = 11110111 \Rightarrow \text{F7}$$

Concretament, el resultat de tots els elements de la primera columna és:

$$\begin{bmatrix} (02 \otimes D5) \oplus (03 \otimes 4B) \oplus (01 \otimes AF) \oplus (01 \otimes 5E) \\ (01 \otimes D5) \oplus (02 \otimes 4B) \oplus (03 \otimes AF) \oplus (01 \otimes 5E) \\ (01 \otimes D5) \oplus (01 \otimes 4B) \oplus (02 \otimes AF) \oplus (03 \otimes 5E) \\ (03 \otimes D5) \oplus (01 \otimes 4B) \oplus (01 \otimes AF) \oplus (01 \otimes 5E) \end{bmatrix} = \begin{bmatrix} 9D \\ F7 \\ 39 \\ 3C \end{bmatrix}$$

I el resultat de la funció *MixColumns* sobre tota la matriu d'estat és:

$$MixColumns(S) = \begin{bmatrix} 9D & 28 & 91 & 00 \\ F7 & 7F & 78 & A6 \\ 39 & C1 & 6C & C6 \\ 3C & AA & 25 & A5 \end{bmatrix}$$

Per acabar, l'última funció que s'aplica en cada iteració és la funció *AddRoundKey*, que ja hem descrit.

2.4.3. Generació de subclaus

A l'igual que la majoria de criptosistemes en bloc, l'algorisme de Rijndael treballa amb diferents subclaus en cada iteració. Aquestes subclaus s'obtenen per l'aplicació d'una funció d'ampliació a la clau de xifratge inicial.

La funció d'ampliació genera una clau ampliada $W = (W_0, W_1, ..., W_{N_b(N_r+1)-1})$ a partir de les N_K paraules de 32 bits de clau de xifratge $K = (K_0, K_1, ..., K_{N_k-1})$, que conté $N_b(N_r + 1)$ paraules de 32 bits. Cada $K(i)$, que denota cada una de les subcadenes de W de N_b paraules de 32 bytes, és la subclau que es fa servir en la i-èsima iteració. Gràficament, les subclaus de cada iteració en relació amb la clau ampliada es poden expressar de la manera següent:

$$W = (\underbrace{W_0, W_1, ..., W_{N_b-1}, W_{N_b}}_{K(0)}, \underbrace{W_{N_b+1}, ..., W_{2N_b-1}}_{K(1)}, ..., \underbrace{W_{N_b(Nr)}, ..., W_{N_b(N_r+1)-1}}_{K(N_r)})$$

En la transformació inicial s'utilitza la subclau $K(0)$, formada per les primeres N_b paraules de W i en cada una de les N_r iteracions es fan servir N_b paraules.

Més concretament, la funció d'ampliació de claus ***KeyExpansion*** ve definida per l'algorisme següent, en funció de si $N_k \leq 6$ o $N_k > 6$:

a) Cas $N_k \leq 6$. L'algorisme consta de dues etapes:

- **Inicialització**, en què la clau de xifratge es copia íntegrament a les primeres posicions de la clau ampliada; és a dir: $W_i = K_i$ per a $i = 0, ..., N_k - 1$.

- **Fase d'ampliació**, en què s'agafa l'última paraula calculada i s'amplia. Aquesta fase té dos passos:

$temp = W_{i-1}$
si $i = 0 \pmod{N_k}$ **aleshores**
    $temp = SubWord(RotWord(temp)) \oplus Rcon[i/N_k]$
fisi
$W_i = W_{i-N_k} \oplus temp$

b) Cas $N_k > 6$. L'algorisme consta de dues etapes:

- **Inicialització**, en què la clau de xifratge es copia íntegrament a les primeres posicions de la clau ampliada; és a dir: $W_i = K_i$ per a $i = 0, ..., N_k - 1$.

- **Fase d'ampliació**, en què s'agafa l'última paraula calculada i s'amplia. Aquesta fase té dos passos:

$temp = W_{i-1}$
si $i = 0 \pmod{N_k}$ **aleshores**
    $temp = SubWord(RotWord(temp)) \oplus Rcon[i/N_k]$
sino
    si $i = 4 \pmod{N_4}$ **aleshores**
        $temp = SubWord(temp)$
    fisi
fisi
$W_i = W_{i-N_k} \oplus temp$

En ambdós casos, la fase d'ampliació fa servir dues subfuncions, la *SubWord* i la *RotWord*. La funció ***SubWord*** és la mateixa funció que *ByteSub* (definida en el subapartat anterior). La funció ***RotWord*** simplement fa una permutació cíclica a la paraula de 4 bytes, és a dir, si té una entrada $[a_0, a_1, a_2, a_3]$, la sortida serà: $[a_1, a_2, a_3, a_0]$. D'altra banda, també es fa servir la constant $Rcon[i]$, que val $Rcon[i] = [x^{i-1}, '00', '00', '00']$. Recordeu que x en hexadecimal val '02', ja que correspon a la representació en binari de 00000010.

Exemple de generació de subclaus

Suposem que la longitud dels blocs és de 128 bits, és a dir $N_b = 4$ i que la clau de xifratge, en hexadecimal, val:

$$K = 00010203 \quad 04050607 \quad 08090A0B \quad 0C0D0E0F$$

$$K_0 \qquad K_1 \qquad K_2 \qquad K_3$$

És a dir, la longitud de la clau és també de 128 bits, $N_k = 4$ (quatre paraules de 32 bits).

Amb aquests paràmetres i donada la taula del subapartat 2.4.1, el nombre d'iteracions serà $N_r = 10$. Això vol dir que la clau ampliada W tindrà $4 \cdot (10 + 1) = 44$ paraules de 32 bits. On $K(i)$ denota la clau que es fa servir a l'i-èsima iteració.

Els primers bytes de la clau ampliada són els mateixos que els de la clau de xifratge:

$$W_0 = 00\ 01\ 02\ 03$$
$$W_1 = 04\ 05\ 06\ 07$$
$$W_2 = 08\ 09\ 0A\ 0B$$
$$W_3 = 0C\ 0D\ 0E\ 0F$$

Per tant: $K(0) = W_0W_1W_2W_3 = 00010203\ \ 04050607\ \ 08090A0B\ \ 0C0D0E0F = K$.

Aquests quatre bytes són els que es fan servir en la transformació inicial de l'algorisme.

La segona subclau serà:

$W_4 = W_0 \oplus SubWord(RotWord(W_3)) \oplus Rcon[1]$
$RotWord(W_3) = RotWord(0C\ 0D\ 0E\ 0F) = 0D\ 0E\ 0F\ 0C$
$SubWord(0D\ 0E\ 0F\ 0C) = (D7\ AB\ 76\ FE)$
$W_4 = 00\ 01\ 02\ 03 \oplus D7\ AB\ 76\ FE \oplus 01\ 00\ 00\ 00 = D6\ AA\ 74\ FD$
$W_5 = W_1 \oplus W_4 = 04\ 05\ 06\ 07 \oplus D6\ AA\ 74\ FD = D2\ AF\ 72\ FA$
$W_6 = W_2 \oplus W_5 = 08\ 09\ 0A\ 0B \oplus D2\ AF\ 72\ FA = DA\ A6\ 78\ F1$
$W_7 = W_3 \oplus W_6 = 0C\ 0D\ 0E\ 0F \oplus DA\ A6\ 78\ F1 = D6\ AB\ 76\ FE$

Per tant, la subclau $K(1)$ = D6 AA 74 FD D2 AF 72 FA DA A6 78 F1 D6 AB 76 FE

La resta de la clau ampliada es calcula de la mateixa manera.

2.4.4. Desxifratge

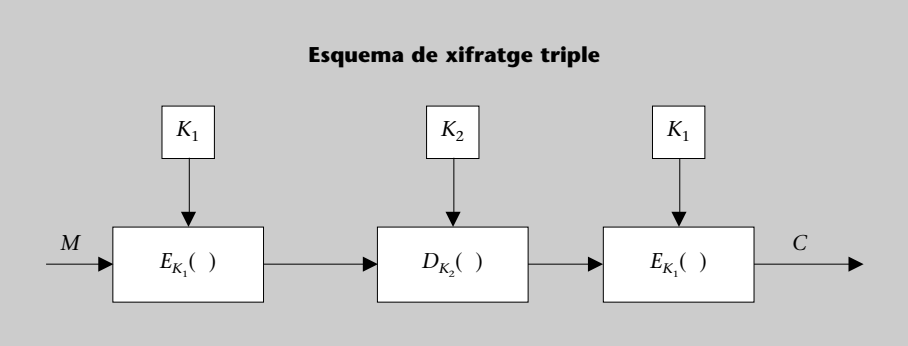
En el subapartat anterior hem definit amb tot detall les operacions de xifratge del criptosistema de Rijndael. Totes les funcions que s'utilitzen en el procés de xifratge (*ByteSub*, *ShiftRow*, *MixColumn* i *AddRoundKey*) són invertibles i, per tant, se'n pot definir la corresponent funció inversa.

Si les funcions definides en el xifratge s'apliquen en l'ordre oposat al que s'executen en el procés de xifratge, obtenim el procés de desxifratge del criptosistema.

3. Atacs a les xifres de bloc

3.1. Atac del punt intermedi

Per tal d'augmentar l'espai de les claus amb la finalitat de frustrar una criptoanàlisi de cerca exhaustiva, una alternativa és encadenar diferents xifratges de bloc en què cadascun faci servir una clau K_i diferent, com mostra la figura següent:



Aquesta tècnica de xifratge múltiple requereix la utilització d'un criptosistema, com ara el DES, que no formi un grup algèbric. Necessitem aquesta darrera condició perquè si la funció de xifratge formés un grup algèbric, l'operació de xifrar un missatge amb una clau K_1 i tornar a xifrar el resultat amb una altra clau K_2 seria el mateix que xifrar el missatge només una sola vegada amb una tercera clau K_3.

Encara que sembla que el xifratge múltiple incrementa molt l'espai de les claus, en realitat no ho fa tant com sembla. S'ha demostrat que un xifratge de dues etapes del DES es pot trencar amb una cerca exhaustiva de només 2^{n+1} iteracions, en comptes del que semblaria lògic pensar, 2^{2n}. Aquest fet es deu a l'aplicació de la criptoanàlisi coneguda com a *atac del punt intermedi*.

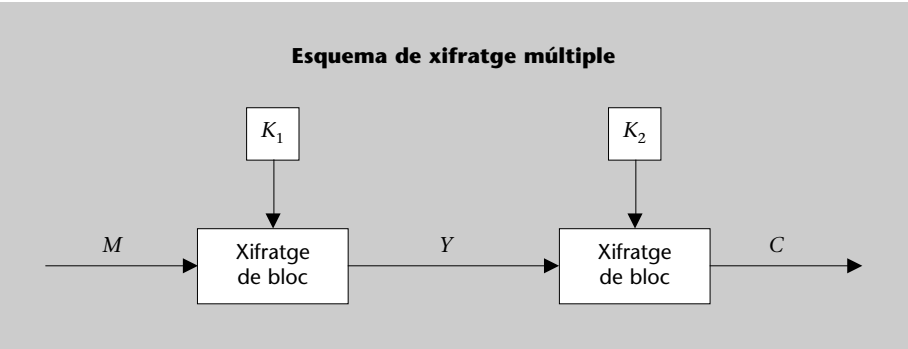
Suposeu que K_1 i K_2 són claus de n bits; l'atac del punt intermedi calcula $D_{K_2}(C)$ per als 2^n valors possibles de K_2. Tot seguit, es calcula $E_{K_1}(M)$ per als 2^n valors possibles de K_1. Forçosament, hi haurà un valor de K_1 i un de K_2 tals que $E_{K_1}(M) = Y = D_{K_2}(C)$. Aquests són els valors correctes de les claus.

Per tant, sumant les operacions fetes en el primer pas i en el segon tenim que, en total, hem efectuat $2^n + 2^n = 2^{n+1}$ operacions.

3.2. Xifratge triple

Per a evitar atacs com l'atac del punt intermedi que acabem de descriure i aconseguir alhora una duplicació efectiva de l'espai de les claus, s'utilitza el que es coneix com a *xifratge triple*.

El **xifratge triple** consisteix a combinar la funció de xifratge de bloc amb la de desxifratge del mateix bloc per mitjà de claus diferents, com mostra la figura següent:



Cada caixa representa el xifrador de bloc. En els dos extrems, el xifrador aplica al missatge que li arriba la funció de xifratge amb la clau, E_{K_1}, mentre que la caixa del mig aplica la funció de desxifratge, D_{K_2}. Les equacions corresponents són:

- $C = E_{K_1}(D_{K_2}(E_{K_1}(M)))$,
- $M = D_{K_1}(E_{K_2}(D_{K_1}(C)))$.

D'aquesta manera, doncs, s'aconsegueix doblar el nombre de bits de l'espai de la clau del criptosistema i evitar al mateix temps l'atac del punt intermedi.

Actualment, com que la llargada de la clau del DES (56 bits) es considera massa curta, gairebé totes les aplicacions implementen el que es coneix com a *triple DES*, que no és més que prendre el DES com a xifrador de l'esquema de xifratge triple que acabem d'explicar.

3.3. Criptoanàlisi diferencial

Una dada

Sembla que la criptoanàlisi diferencial, malgrat que no es va fer pública fins al 1991, ja era coneguda pels investigadors d'IBM que van desenvolupar el DES, perquè curiosament aquest criptosistema, que va ser creat durant els anys setanta, resisteix l'atac d'aquella.

La criptoanàlisi diferencial és un atac estadístic de text en clar conegut que es fa servir contra sistemes que es basen en iteracions d'una transformació criptogràficament feble. L'objectiu és reduir el nombre d'operacions que cal executar per a trencar el criptosistema mitjançant una cerca exhaustiva. Cada criptosistema requereix un esquema d'atac concret.

La idea general de la criptoanàlisi diferencial és veure com es propaga una diferència coneguda de text en clar per mitjà de l'algorisme de xifratge. Donats dos textos en clar, M i M', amb una diferència coneguda, $D = M \oplus M'$, s'obtenen els textos xifrats corresponents, $C = E_K(M)$ i $C' = E_K(M')$, i s'estudia la relació entre D i $D_c = C \oplus C'$.

4. Gestió de claus

La característica més important del xifratge amb clau compartida és que la clau que serveix per a xifrar és la mateixa que s'utilitza per a desxifrar, i la comparteixen l'emissor i el receptor. Això suposa un cert risc, perquè ja se sap que el secret més ben guardat és aquell que no es comparteix. Per això, cal dissenyar sistemes de gestió de claus que es basin en estructures jeràrquiques en les quals hi hagi claus que s'usen per a xifrar altres claus d'un nivell inferior. Els **principals tipus de claus** que hi ha, d'acord amb la seva importància i la funció que tenen assignada, són els següents:

1) Clau mestra. Clau de jerarquia superior que no s'utilitza mai per a xifrar missatges, sinó solament per a xifrar claus primàries i secundàries. Aquesta clau es guarda en clar en un dispositiu segur.

2) Clau primària o secundària. Claus de jerarquia inferior a la clau mestra i que es guarden xifrades amb la clau mestra. Els tipus més importants de claus primàries i secundàries són:

a) La **clau de generació de claus**, que es fa servir per a generar aleatòriament claus de sessió o vectors d'inicialització.

b) La **clau de xifratge de claus**, que s'utilitza per a xifrar altres claus amb la finalitat de protegir-les.

3) La **clau de sessió o comunicació**, que s'envia a través del canal en començar una comunicació, xifrada amb una clau secundària de xifratge de claus. Es caracteritza perquè xifra aquella comunicació i només aquella, i quan la comunicació ha acabat la clau s'esborra.

4) La **clau de xifratge d'arxius**, que es pot considerar de sessió, usada per a xifrar arxius complets. Es caracteritza perquè es guarda a la capçalera de l'arxiu que xifra.

No cal dir que en un sistema jeràrquic com el que s'ha descrit, la pedra angular de tota la seguretat recau en la clau mestra, que és la que xifra les altres claus de rang inferior, per la qual cosa cal que es generi i s'emmagatzemi de manera segura. Normalment, tant la clau mestra com el mateix algorisme de xifratge de claus s'emmagatzemen en un mòdul de seguretat de memòria no volàtil (com ara, els circuits integrats), que pot incorporar també altres mesures de seguretat, com, per exemple, l'autodestrucció en el cas que s'intenti manipular el sistema.

Resum

En aquest mòdul didàctic hem descrit el funcionament general de les xifres de bloc actuals. Per a fer-ho, hem tractat dels aspectes següents:

1) Hem definit els conceptes de confusió i de difusió i també els mètodes de xifratge possibles d'un criptosistema de bloc.

2) Hem detallat el funcionament de quatre modes de xifratge: l'ECB, el CBC, el CFB i l'OFB.

3) Hem descrit el funcionament de tres criptosistemes de bloc: DES, IDEA i l'AES. N'hem esmentat les característiques principals, hem explicat els processos de xifratge i desxifratge, la generació de les subclaus i hem detallat les iteracions que porten a terme.

4) Hem fet referència a algunes tècniques de criptoanàlisi del xifratge de bloc i a la manera d'evitar-ne alguna.

5) Finalment, hem explicat el problema de la gestió de les claus i els esquemes teòrics que hi ha per a resoldre-la.

Activitats

1. Implementeu el DES en mode CBC.

2. Implementeu l'IDEA amb vuit voltes de xifratge.

3. Implementeu l'AES.

Exercicis d'autoavaluació

1. Suposem que la clau principal de 64 bits d'un xifrador DES és la següent:

01010000 01010000 01010101 01000101 01000010 01000001 01001100 01001111.

Trobeu la primera subclau de 48 bits, és a dir, K_1.

2. Suposeu que les claus usades amb el DES consisteixen només en les lletres A-Z i tenen vuit lletres de llargada. Doneu una aproximació del temps que es tardaria a provar totes les claus fent servir una cerca exhaustiva, suposant que cada clau pot ser provada en un microsegon. Feu el mateix per claus que continguin vuit caràcters restringits a majúscules, minúscules i dígits decimals.

3. Suposem que la clau principal de 128 bits d'un xifrador IDEA és la següent:

01001001 01000100 01000101 01000001 00100000 01100101 01110011 00100000
01101100 01100001 00100000 01100011 01101100 01100001 01110110 01100101.

Trobeu els 16 bits que formen la segona clau de la quarta iteració, és a dir, $K_2^{(4)}$.

4. Per a la mateixa clau inicial de l'exercici d'autoavaluació 3, trobeu quina és la tercera clau de desxifratge de la sisena iteració.

5. Suposem que la clau de xifratge de 192 bits d'un xifrador Rijndael expressada en hexadecimal és la següent:

8E 73 B0 F7 DA 0E 64 52 C8 10 F3 2B 80 90 79 E5 62 F8 EA D2 52 2C 6B 7B.

Doneu-ne les dues primeres subclaus, és a dir, $K(0)$ i $K(1)$

6. Donat un xifrador Rijndael amb clau de xifratge K i un bloc de text per xifrar B:

K = 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
B = 32 43 F6 A8 88 5A 30 3D 31 31 98 A2 E0 37 07 34

Quantes iteracions cal fer per xifrar aquest bloc de text en clar amb aquesta clau? Quina és la matriu d'estat a l'inici de la segona iteració?

Solucionari

Exercicis d'autoavaluació

1. En primer lloc, hem d'aplicar la permutació *P1* a la clau d'entrada. Això ens dóna la seqüència de 56 bits següent:

0000 0000 1111 1111 0000 0000 0000 1001 0000 1100 1100 1100 0000 0111.

Per tant, en dividir-la en dues parts tenim:
- F_0 = 0000 0000 1111 1111 0000 0000 0000.
- G_0 = 1001 0000 1100 1100 1100 0000 0111.

Si apliquem el desplaçament circular DC_1, que, tal com ens mostra la taula *DC*, suposa un desplaçament cap a l'esquerra, tindrem:
- F_1 = 0000 0001 1111 1110 0000 0000 0000.
- G_1 = 0010 0001 1001 1001 1000 0000 1111.

Finalment, aplicant-hi la permutació *P2* acabem obtenint:

K_1 = 101000 001001 001001 000010 101101 010100 100101 000100.

2. Partint d'un alfabet de 26 lletres, si cada clau està formada per vuit lletres el nombre total de claus diferents que hi ha és: $26^8 = 2{,}088270 \cdot 10^{11}$. Per tant, si suposem que cada clau es pot provar en un microsegon, 10^{-8} segons, es tardarien uns 20.882 segons, és a dir, unes 5 hores i 48 minuts per a fer una cerca exhaustiva.
Si, a més de les majúscules, hi afegim les minúscules i els dígits decimals, els valors que obtenim són: $(26 + 26 + 10)^8 = 2{,}183401 \cdot 10^{14}$ claus. Llavors, necessitarem 21.834.011 segons, o 6.065 hores o 252 dies de temps, de cerca exhaustiva.
Com podem comprovar, l'espai de claus no es pot restringir a caràcters amb sentit.

3. Com que en cada volta del criptosistema IDEA s'utilitzen sis claus, la segona de la quarta volta serà la vintena clau ($3 \cdot 6 + 2$). Els primers 128 bits de la clau principal generen les vuit primeres claus. Per tant, la vintena clau és generada per les posicions 49, 50, ..., 64 del tercer grup de blocs de 128 bits de clau inicial. Com que cada bloc queda desplaçat 25 posicions a la dreta, tenim que el segon bloc comença a la posició 26 i el tercer, a la 51. Per tant, la segona clau de la quarta iteració la formen els bits de les posicions 99, 100, 101, ..., 114 de la clau inicial, és a dir, 10110001 10000101 (el valor 99 s'obté de 51 + 48).

4. Si ens fixem en la taula d'obtenció de les claus de desxifratge, veiem que en la sisena iteració les claus les dóna $K_1^{(4)^{-1}}$, $-K_3^{(4)}$, $-K_2^{(4)}$, $K_4^{(4)^{-1}}$, $K_5^{(3)}$, $K_6^{(3)}$, en què el superíndex *i* representa la *i*-èsima iteració de xifratge. Per consegüent, la tercera clau que ens demanen és $-K^{(4)}$. Tenint en compte el resultat de l'exercici anterior, 10110001 10000101, el que busquem és el seu invers additiu mòdul $2^{16} + 1$, que és 10110001 10000101.

5. Atès que la clau de xifratge és de 192 bits, el nombre de paraules de 32 bits de la clau val $N_K = 6$; per tant, haurem d'aplicar l'algorisme per al cas $N_K \leq 6$

Els primers bits de la clau estesa són exactament els mateixos bits de la clau de xifratge:

W_0 = 8E 73 B0 F7
W_1 = DA 0E 64 52
W_2 = C8 10 F3 2B
W_3 = 80 90 79 E5
W_4 = 62 F8 EA D2
W_5 = 52 2C 6B 7B

Per tant:

$K(0) = W_0W_1W_2W_3W_4W_5 =$
= 8E73B0F7 DA 0E 64 52 C810F32B 809079E5 62F8EAD2 522C6B7B =
= K

Si apliquem l'algorisme per al cas $N_K \leq 6$ amb els valors W_i anteriors obtenim:

$W_6 = W_0 \oplus SubWord(RotWord(W_5)) \oplus Rcon[1]$
$RotWord(W_5) = RotWord(52\ 2C\ 6B\ 7B) = 2C\ 6B\ 7B\ 52$
$SubWord(2C\ 6B\ 7B\ 52) = (71\ 7F\ 21\ 00)$
W_6 = 8E 73 B0 F7 $\oplus$ 71 7F 21 00 $\oplus$ 01 00 00 00 =
= 8E 73 B0 F7 $\oplus$ 70 7F 21 00 = FE 0C 91 F7
$W_7 = W_1 \oplus W_6$ = DA 0E 64 52 $\oplus$ FE 0C 91 F7 = 24 02 F5 A5
$W_8 = W_2 \oplus W_7$ = C8 10 F3 2B $\oplus$ 24 02 F5 A5 = EC 12 06 8E

$W_9 = W_3 \oplus W_8$ = 80 90 79 E5 $\oplus$ EC 12 06 8E = 6C 82 7F 6B
$W_{10} = W_4 \oplus W_9$ = 62 F8 EA D2 $\oplus$ 6C 82 7F 6B = 0E 7A 95 B9
$W_{11} = W_5 \oplus W_{10}$ = 52 2C 6B 7B $\oplus$ 0E 7A 95 B9 = 5C 56 FE C2

Per tant, la subclau:

$K(1)$ = FE 0C 91 F7 24 02 F5 A5 EC 12 06 8E 6C 82 7F 6B 0E 7A 95 B9 5C 56 FE C2

6. Caldrà fer deu iteracions per a xifrar aquest bloc de text en clar, ja que tant la longitud del bloc de xifratge com la longitud de la clau és de 16 bytes; per tant, $N_b = 4$ i $N_k = 4$.

En la transformació inicial s'aplica la transformació *addRoundKey*. En el nostre cas:

$$AddRoundKey(S, K(0)) = \begin{bmatrix} 32 & 88 & 31 & E0 \\ 43 & 5A & 31 & 37 \\ F6 & 30 & 98 & 07 \\ A8 & 8D & A2 & 34 \end{bmatrix} \oplus \begin{bmatrix} 2B & 28 & AB & 09 \\ 7E & AE & F7 & CF \\ 15 & D2 & 15 & 4F \\ 16 & A6 & 88 & 3C \end{bmatrix} = \begin{bmatrix} 19 & A0 & 9A & E9 \\ 3D & F4 & C6 & F8 \\ E3 & E2 & 8D & 48 \\ BE & 2B & 2A & 08 \end{bmatrix} = S1$$

El resultat de la primera iteració correspondrà a executar les funcions *ByteSub, ShiftRow, MixColumns* i *AddRoundKey*. El resultat de la funció *ByteSub* sobre la matriu d'estat *S1* és:

$$ByteSub \begin{bmatrix} 19 & A0 & 9A & E9 \\ 3D & F4 & C6 & F8 \\ E3 & E2 & 8D & 48 \\ BE & 2B & 2A & 08 \end{bmatrix} = \begin{bmatrix} D4 & E0 & B8 & 1E \\ 27 & BF & B4 & 41 \\ 11 & 98 & 5D & 52 \\ AE & F1 & E5 & 30 \end{bmatrix} = S2$$

El resultat de la funció *ShiftRow* sobre la matriu d'estat *S2* és:

$$ShiftRow \begin{bmatrix} D4 & E0 & B8 & 1E \\ 27 & BF & B4 & 41 \\ 11 & 98 & 5D & 52 \\ AE & F1 & E5 & 30 \end{bmatrix} = \begin{bmatrix} D4 & E0 & B8 & 1E \\ BF & B4 & 41 & 27 \\ 5D & 52 & 11 & 98 \\ 30 & AE & F1 & E5 \end{bmatrix} = S3$$

El resultat de la funció *MixColumns* sobre la matriu d'estat *S3* és:

$$MixColumns \begin{bmatrix} D4 & E0 & B8 & 1E \\ BF & B4 & 41 & 27 \\ 5D & 52 & 11 & 98 \\ 30 & AE & F1 & E5 \end{bmatrix} = \begin{bmatrix} 04 & E0 & 48 & 28 \\ 66 & CB & F8 & 06 \\ 81 & 19 & D3 & 26 \\ E5 & 9A & 7A & 4C \end{bmatrix} = S4$$

Calculem el valor de la clau de la segona iteració:

$$\begin{bmatrix} A0 & 88 & 23 & 2A \\ FA & 54 & A3 & 6C \\ FE & 2C & 39 & 76 \\ 17 & B1 & 39 & 05 \end{bmatrix}$$

Finalment, el resultat de la funció *AddRoundKey* sobre la matriu d'estat S4 resulta:

$$\begin{bmatrix} 04 & E0 & 48 & 28 \\ 66 & CB & F8 & 06 \\ 81 & 19 & D3 & 26 \\ E5 & 9A & 7A & 4C \end{bmatrix} \oplus \begin{bmatrix} A0 & 88 & 23 & 2A \\ FA & 54 & A3 & 6C \\ FE & 2C & 39 & 76 \\ 17 & B1 & 39 & 05 \end{bmatrix} = \begin{bmatrix} A4 & 68 & 6B & 02 \\ 9C & 9F & 5B & 6A \\ 7C & 35 & EA & 50 \\ F2 & 2B & 43 & 49 \end{bmatrix}$$

Així, el valor de la matriu d'estat a l'inici de la segona iteració valdrà:

$$S = \begin{bmatrix} A4 & 68 & 6B & 02 \\ 9C & 9F & 5B & 6A \\ 7F & 35 & EA & 50 \\ F2 & 2B & 43 & 49 \end{bmatrix}$$

Glossari

AES: *Advanced Encryption Standard*. Criptosistema Rijndael, que xifra blocs de 128 bits per mitjà d'una clau que pot variar la longitud entre 128, 192 o 256 bits.

CBC: *Cipher Bloc Chaining*. Mode de xifratge de bloc en què es crea un encadenament dels blocs, de manera que el xifratge d'un bloc depèn de l'anterior per mitjà d'un bloc inicial aleatori per al xifratge.

CFB: *Cipher Feedback*. Mode de xifratge de bloc en què la llargada dels blocs de text no ha de coincidir amb la dels blocs del criptosistema.

Confusió: tècnica que té per objectiu que la relació entre la clau i el text xifrat sigui tan complicada com es pugui.

DES: *Data Encryption Standard*. Criptosistema de xifratge de bloc que xifra blocs de dades de 64 bits de llargada per mitjà d'una clau de 56 bits i l'acció de caixes *S*.

Difusió: tècnica que té per objectiu la dissipació de les propietats estadístiques del text en clar mitjançant el text xifrat.

ECB: *Electronic Code Book*. Mode de xifratge de bloc en què el xifratge dels blocs és independent l'un de l'altre i es porta a terme amb una mateixa clau.

IDEA: *International Data Encryption Algorithm*. Criptosistema de xifratge de bloc que xifra blocs de text en clar de 64 bits de llargada per mitjà d'una clau de 128, per mitjà de vuit iteracions idèntiques i una transformació de sortida.

OFB: *Ouput Feedback*. Mode de xifratge de bloc en què el vector inicial es realimenta directament amb el resultat del xifratge de bloc.

Triple DES: protocol de xifratge triple que utilitza el DES com a base per a obtenir un xifrador amb un espai de claus superior.

Bibliografia

Biham, E.; Shamir, A. (1991). "Differential Cryptanalysis of DES-like Cryptosystems". *Journal of Cryptology* (vol. 4, núm. 1, pàg. 3-72).

Lai, X.; Massey, J. (1990). "A proposal for a new block encryption standard". *Advances in Cryptology - Eurocrypt'90* (pàg. 389-404).

Meier,W. (1993). "On the Security of the IDEA Block Cipher". *Advances in Cryptology - Eurocrypt'93* (pàg. 371-385).

Merkle, R.; Hellman, M. (1981). "On the Security of Multiple Encryption". *Communications of the ACM* (vol. 24, núm. 7, pàg. 465-467).

Meyer, C.H.; Matyas, S.M. (1982). "Cryptography: A New Dimension in Computer Data Security". Nova York: John Willey & Sons.

National Bureau of Standards (NBS) (1977). "Data Encription Standard". *Federal Information Processing Standards* (núm. 46). Washington.

National Bureau of Standards (NBS) (1981). "DES, Modes of Operation". *Federal Information Processing Standards* (núm. 81). Washington.

National Bureau of Standards (NBS) (2001). "Advanced Encryption Standard (AES)". *Federal Information Processing Standards* (núm. 197). Washington.

Xifres de clau pública

Josep Domingo Ferrer

P03/05024/02263

UOC
www.uoc.edu

Índex

Xifres de clau pública

Introducció

En aquest mòdul didàctic presentem el concepte de **criptografia de clau pública**, que ha suposat una revolució en la criptografia moderna. Per a estudiar-lo, tractarem dels aspectes següents:

a) Comencem amb unes nocions preliminars de teoria dels nombres i de teoria de la complexitat, que són necessàries per a entendre el funcionament dels criptosistemes de clau pública.

b) A continuació expliquem els conceptes de *clau pública*, *funcions unidireccionals* i *funcions unidireccionals amb trapa*.

c) Després introduïm l'intercanvi de claus de Diffie-Hellman.

d) Finalment, descrivim els dos criptosistemes de clau pública principals: l'RSA i l'ElGamal, que són els més usats actualment.

Objectius

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

1. Entendre els conceptes de clau pública, de funció unidireccional i de funció unidireccional amb trapa.

2. Adquirir o recuperar els coneixements d'aritmètica modular i de matemàtica discreta necessaris per a entendre el funcionament dels criptosistemes de clau pública.

3. Conèixer els dos criptosistemes de clau pública més importants.

1. Conceptes preliminars

En aquest apartat resumirem els conceptes bàsics de la teoria dels nombres, que ens són necessaris per a entendre la resta del mòdul. L'estudiant que ja tingui un cert nivell de coneixements d'aritmètica modular o de matemàtica discreta pot passar per alt aquest apartat.

1.1. Aritmètica modular

Donats els enters a, b i $n \neq 0$, es diu que a és congruent amb b mòdul n, que s'escriu:

$$a \equiv_n b,$$

si i només si $a - b = kn$ per a algun enter k.

$20 \equiv_7 6$, perquè $(20 - 6) = 2 \cdot 7$.

Una definició alternativa seria: $a \equiv_n b$ si i només si n divideix $(a - b)$, que s'escriu $n|(a - b)$.

Si $a \equiv_n b$, llavors b s'anomena **residu de *a* mòdul *n***. Recíprocament, a és un residu de b mòdul n. Escriurem ***a* mod *n*** per a representar el residu de a mòdul n al rang $[0,n - 1]$.

Un conjunt d'enters $r_1, \ldots, r_n$ és un **conjunt complet de residus mòdul *n*** si, per cada enter a, hi ha exactament un r_i en el conjunt tal que $a \equiv_n r_i$.

Per exemple, 13 mod 7 = 6.

Per a qualsevol mòdul, el conjunt d'enters $\{0, 1, \ldots, n - 1\}$ és un conjunt complet de residus mòdul n.

Com passa en els enters, els enters mod n amb les operacions de suma i de multiplicació formen un **anell commutatiu**. Això significa que la suma compleix les propietats associativa i commutativa i existeixen l'element neutre i l'element invers, i per al producte es compleix la propietat associativa respecte de la suma i també té element neutre.

A més, calcular amb aritmètica modular (és a dir, reduir cada resultat intermedi mòdul n) dóna el mateix resultat que calcular amb aritmètica entera ordinària i reduir el resultat final mòdul n. D'una manera més formal, es pot expressar aquest fet amb el teorema de l'aritmètica modular.

Teorema 1: principi de l'aritmètica modular. Siguin a_1 i a_2 enters, i sigui OP un dels operadors binaris $+$, $-$ o $\cdot$, llavors la reducció mòdul n és un homomorfisme dels enters als enters mòdul n, és a dir:

$$(a_1 \text{ OP } a_2) \bmod n = [(a_1 \bmod n) \text{ OP } (a_2 \bmod n)] \bmod n.$$

Equivalència de les operacions amb aritmètica entera i modular

Podem sumar 9.739 + 7.777 + 5.345 mòdul 7 fent-ho de dues maneres equivalents:

a) Primer calculem la suma de manera estàndard: 9.739 + 7.777 + 5.345 = 22.861, i després reduïm el resultat; 22.861 mod 7 = 6.

b) Primer reduïm cada sumand: 9.739 mod 7 = 2; 7.777 mod 7 = 0; 5.345 mod 7 = 4. Després sumem els resultats de les reduccions:
(2 + 0 + 4) mod 7 = 6.

Calcular amb aritmètica modular té l'avantatge que redueix el rang dels valors intermedis. Per a un mòdul n de k bits, el valor de qualsevol suma, resta o multiplicació cabrà en $2k$ bits, pel cap alt. Això, per exemple, ens permet fer exponenciacions modulars de l'estil $a^z \bmod n$ sense generar resultats intermedis monstruosos.

Noteu que per al càlcul informatitzat...

... és molt més adequat operar amb aritmètica modular, així ens assegurem que tots els resultats intermedis seran més petits que el mòdul, amb la qual cosa prevenim possibles sobreeiximents (en anglès, *overflows*) intermedis.

Com que alguns dels algorismes de xifratge descrits en els apartats següents d'aquest mòdul es basen en l'exponenciació mod n, esbossarem tot seguit un mètode d'exponenciació ràpida per a calcular $a^z \bmod n$:

exp_rapid(*a*,*z*,*n*)

begin "Retorna $x = a^z \bmod n$"
  $a_1 := a$; $z_1 := z$;
  $x := 1$;
  while $z_1 \neq 0$ **do** "$x(a_1^{z_1} \bmod n) = a^z \bmod n$"
  begin
    while $z_1 \bmod 2 = 0$ **do**
    begin "elevar al quadrat a_1 mentre z_1 parell"
      $z_1 := \lfloor z_1/2 \rfloor$;
      $a_1 := (a_1 * a_1) \bmod n$
    end;
    $z_1 := z_1 - 1$;
    $x := (x * a_1) \bmod n$ "Multiplicació"
  end;
  exp_rapid := x
end

Si $(z_{k-1}, ..., z_1, z_0)$ és la representació binària de l'exponent z, l'algorisme anterior en processa els bits en ordre $z_0, z_1, ..., z_{k-1}$ (de menys a més significatiu).

Si $z_i = 0$, calcula un quadrat; si $z_i = 1$, multiplica i calcula un quadrat. Per cada bit de z es fan, doncs, fins a dues multiplicacions (excepte per al més significatiu, que només dóna lloc a una multiplicació). Com que la llargada en bits de z és $\lceil \log_2 z \rceil$, tenim que el nombre de multiplicacions és, a tot estirar:

$$2 \cdot (\lceil \log_2 z \rceil - 1) + 1 = 2 \cdot \lfloor \log_2 z \rfloor + 1.$$

Cal tenir present que exponenciar pel mètode "innocent" de les multiplicacions successives requeriria z multiplicacions.

Càlcul de a^{37} amb l'algorisme exp_rapid(a,z,n)

Per a il·lustrar l'algorisme exp_rapid(a,z,n) prenem, per exemple, $z = 37$. La taula següent reflecteix els valors intermedis de les variables (només es presenten els canvis de valor):

a_1	z_1	x
a	37	1
–	36	a
a^2	18	–
a^4	9	–
–	8	a^5
a^8	4	–
a^{16}	2	–
a^{32}	1	–
–	0	a^{37}

En total hi ha vuit multiplicacions, que és inferior a la fita $2\lfloor \log_2 37 \rfloor + 1 = 11$. De fet, l'algorisme anterior es basa en la descomposició següent:

$$a^{37} = a^{32+4+1} = ((((a^2)^2)^2)^2)^2(a^2)^2a.$$

1.2. Càlcul d'inversos en aritmètica modular

Si treballem amb aritmètica entera ordinària, no hi ha inversos multiplicatius. És a dir, per a un enter $a \neq 1$ no hi ha cap altre enter x tal que es verifiqui $ax = 1$.

En canvi, si treballem amb aritmètica modular, per a un enter a en el rang de valors $[0,n-1]$ de vegades es pot trobar un únic enter en el mateix rang tal que es verifiqui $ax \bmod n = 1$.

Inversos multiplicatius

Els enters 11 i 3 són inversos multiplicatius mod 16 perquè:

$$11 \cdot 3 \bmod 16 = 33 \bmod 16 = 1.$$

En els algorismes de xifratge descrits en aquest apartat s'empra sovint l'invers multiplicatiu. El teorema següent dóna la condició que s'ha de complir perquè hi hagi un invers multiplicatiu.

Teorema 2. Donat $a \in [0,n-1]$, a té un únic invers multiplicatiu mòdul n si a és coprimer amb n, és a dir, $\text{mcd}(a,n) = 1$.

Ara bé, no serveix de gaire saber que hi ha l'invers si no disposem d'algorismes per a calcular-lo. Per avançar cal presentar alguns conceptes addicionals.

La **funció totient d'Euler** $\phi(n)$ és el nombre d'elements del conjunt $\{0, ..., n\}$ que són coprimers amb n.

Teorema 3. La funció totient d'Euler es pot calcular amb les fórmules següents:

1) Per a p primer, $\phi(p) = p - 1$ i $\phi(1) = 1$

2) Per a $n = pq$ amb p, q primers: $\phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$.

3) Per a n arbitrari tenim:

$$\phi(n) = \prod_{i=1}^{t} p_i^{e_i - 1}(p_i - 1),$$

en què $n = p_1^{e_1} p_2^{e_2} \ldots p_t^{e_t}$ és la descomposició de n en factors primers $p_1, ..., p_t$, on p_i té multiplicitat e_i.

Els teoremes següents relacionats amb la funció d'Euler són importants en la teoria dels nombres.

Teorema 4: teorema petit de Fermat. Sigui p primer, llavors per a tot a tal que $\text{mcd}(a,p) = 1$ es compleix:

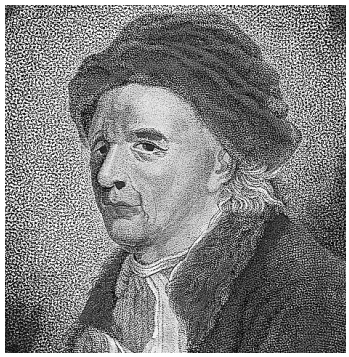
$$a^{p-1} \bmod p = 1.$$

Teorema 5: teorema d'Euler. Per a tot a i n tals que $\text{mcd}(a,n) = 1$ es compleix que:

$$a^{\phi(n)} \bmod n = 1.$$

El teorema d'Euler dóna una primera manera de calcular l'invers multiplicatiu de a mod n resolent l'equació:

$$ax \bmod n = 1.$$



Leonhard Euler

Matemàtic suís del segle XVIII l'obra del qual abasta tota la matemàtica de la seva època.

Elements de $\phi(10)$

$\phi(10) = 4$ perquè hi ha quatre elements a $\{0, ..., 10\}$ que són coprimers amb 10; concretament són l'1, el 3, el 7 i el 9.

Pierre de Fermat

Matemàtic francès del segle XVII que va tenir un paper precursor en càlcul diferencial, en geometria analítica, en probabilitat i també en teoria de nombres. El teorema petit de Fermat no és el cèlebre teorema indemostrat de Fermat. Aquest darrer no ha estat demostrat fins al 1995.

Quan mcd(a,n) = 1, la solució s'obté de l'expressió següent:

$$x = a^{\phi(n)-1} \bmod n.$$

Càlcul de l'invers multiplicatiu de 7 mod 15

Suposem $a = 7$ i $n = 15$; aleshores la funció totient d'Euler de 15 és:

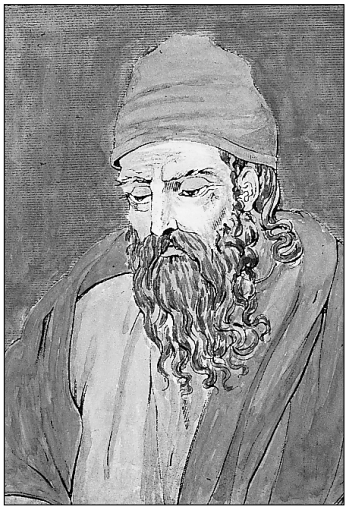
$$\phi(15) = \phi(3)\phi(5) = 2 \cdot 4 = 8.$$

Llavors, si apliquem el teorema d'Euler per a trobar l'invers multiplicatiu obtenim:

$$x = 7^{8-1} \bmod 15 = 13.$$

En efecte, 13 és l'invers de 7, ja que $7 \cdot 13 \bmod 15 = 1$.

Si es coneix $\phi(n)$, l'invers de a mòdul n es pot calcular amb l'equació anterior i l'algorisme exp_rapid(a,z,n). El problema és que si el mòdul n és molt gran (com en els criptosistemes que veurem), trobar $\phi(n)$ és difícil. L'algorisme d'Euclides estès permet calcular inversos sense conèixer $\phi(n)$.



Euclides

Matemàtic grec del segle III aC. La seva obra *Elements* és considerada el llibre de geometria per excel·lència i resumeix totes les aportacions gregues anteriors a Arquímedes.

L'**algorisme d'Euclides bàsic** serveix per a calcular el màxim comú divisor de dos nombres, a i n:

Algorisme mcd(a,n)

begin
  $g_0 := n$;
  $g_1 := a$;
  $i := 1$;
  while $g_i \neq 0$ **do**
  begin
    $g_{i+1} := g_{i-1} \bmod g_i$;
    $i := i + 1$;
  end;
  mcd := g_{i-1}
end

La **versió estesa de l'algorisme d'Euclides** permet trobar l'invers de a mòdul n quan mcd(a,n) = 1:

Algorisme inv(a,n)

begin "Torna x tal que $ax \bmod n = 1$, en què $0 < a < n$"
  $g_0 := n$; $g_1 := a$;
  $u_0 := 1$; $v_0 := 0$;

```
u1 := 0; v1 := 1;
i := 1;
while gi ≠ 0 do "gi = ui n + vi a"
begin
    y := ⌊gi−1/gi⌋;
    gi+1 := gi−1 − y * gi;
    u i+1 := u i−1 − y * ui;
    v i+1 := v i−1 − y* vi;
    i := i + 1;
end;
x := v i−1
if x ≥ 0 then inv := x else inv := x + n
end
```

Si el mòdul respecte al qual s'inverteix és n, llavors l'algorisme inv(a,n) requereix de l'ordre de ln(n) divisions; és força eficient, doncs.

Utilització de l'algorisme d'Euclides estès

Il·lustrarem el funcionament de l'algorisme d'Euclides estès trobant l'invers de 3 mòdul 7, és a dir, resolent l'equació $3x \bmod 7 = 1$.

La taula següent reflecteix els valors intermedis de les variables (només es presenten els canvis de valor):

i	g_i	u_i	v_i	y
0	7	1	0	–
1	3	0	1	2
2	1	1	−2	3
3	0	–	–	–

Com que $v_2 = -2$ és negatiu, la solució és $x = -2 + 7 = 5$.

Finalment, esmentarem un darrer resultat (ja mil·lenari) que és útil en alguns protocols criptogràfics. La idea és resoldre un determinat tipus de sistemes d'equacions modulars.

Teorema 6: teorema xinès del residu. Siguin $d_1, ..., d_t$ enters coprimers dos a dos, i sigui $n = d_1\, d_2\, ...\, d_t$, llavors el sistema d'equacions:

$$(x \bmod d_i) = x_i; \text{ per a } i = 1, ..., t$$

té una solució comuna x en el rang $[0,n-1]$.

Demostració: per a cada $i = 1, ..., t$ tenim $\text{mcd}(d_i, n/d_i) = 1$. Per tant, hi ha l'invers de n/d_i mòdul d_i, és a dir, y_i tal que $(n/d_i)y_i \bmod d_i = 1$. D'altra banda, $(n/d_i)y_i \bmod d_j = 0$ per a $j \neq i$, perquè d_j és un factor de n/d_i. Sigui:

$$x = \left[\sum_{i=1}^{t}\left(\frac{n}{d_i}\right)y_i x_i\right] \bmod n,$$

llavors x és la solució de l'equació següent:

$$(x \bmod d_i) = x_i; \text{ per a } i = 1, ..., t$$

atès que:

$$x \bmod d_i = \left(\frac{n}{d_i}\right)y_i x_i \bmod d_i = x_i.$$

Utilització del teorema xinès del residu

Farem servir el teorema xinès del residu per a resoldre l'equació $3x \bmod 10 = 1$ a base d'anar resolent equacions amb mòduls més petits. Observem que $10 = 2 \cdot 5$; per consegüent, $d_1 = 2$ i $d_2 = 5$. Llavors hem de trobar les solucions x_1 i x_2 de les equacions:

- $3x \bmod 2 = 1$,
- $3x \bmod 5 = 1$.

Aplicant l'algorisme inv(a,n) obtenim $x_1 = 1$ i $x_2 = 2$. Aleshores el teorema xinès dels residus permet trobar una solució comuna x de les equacions:

- $x \bmod 2 = x_1 = 1$,
- $x \bmod 5 = x_2 = 2$.

Trobem y_1 i y_2 tals que:

- $\left(\frac{10}{2}\right)y_1 \bmod 2 = 1$,
- $\left(\frac{10}{5}\right)y_2 \bmod 5 = 1$.

Obtenim $y_1 = 1$ i $y_2 = 3$, amb la qual cosa:

$$x = \left[\left(\frac{10}{2}\right)y_1 x_1 + \left(\frac{10}{5}\right)y_2 x_2\right] \bmod 10 = [5 \cdot 1 \cdot 1 + 2 \cdot 3 \cdot 2] \bmod 10 = 7.$$

Per tant, 7 és l'invers de 3 mòdul 10.

1.3. Terminologia bàsica de la teoria de la complexitat

La complexitat computacional permet fonamentar l'anàlisi dels requisits computacionals de les tècniques de criptoanàlisi, és a dir, la dificultat que comporta trencar una xifra.

1.3.1. Complexitat d'un algorisme

La fortalesa d'una xifra segura computacionalment* queda determinada per la complexitat de càlcul dels algorismes que són necessaris per a trencar-la.

*** Categoria a la qual pertanyen totes les xifres emprades en la pràctica.**

Aquesta complexitat de càlcul es mesura pel temps T i l'espai E que es tarda, on T i E s'expressen com a funcions de la mida n de l'entrada de l'algorisme. Més que fer servir complexitats exactes $f(n)$, se solen emprar ordres de magnitud $O(g(n))$, de tal manera que $f(n) = O(g(n))$ vol dir que hi ha constants c i n_0 tals que:

$$f(n) \leq c\,|g(n)|; \text{ per a } n \geq n_0.$$

El propòsit que hi ha darrere l'ús d'ordres de magnitud és que $g(n)$ sigui més simple que $f(n)$.

Càlcul de l'ordre de magnitud de la complexitat d'un algorisme

Si la complexitat exacta d'un algorisme és $f(n) = 24n + 16$, podem dir que $f(n) = O(n)$, ja que:

$$24n + 16 \leq 25n, \text{ per a } n \geq 16.$$

De la mateixa manera, si $f(n)$ és un polinomi de grau t en n, podem escriure $f(n) = O(n^t)$.

Un **algorisme** és **polinòmic** o, més exactament, **de temps polinòmic**, si el seu temps d'execució és $T = O(n^t)$ per a alguna constant t. Un algorisme polinòmic és: **constant**, si $t = 0$; **lineal**, si $t = 1$; **quadràtic**, si $t = 2$; etc.

Així mateix, un **algorisme** és **exponencial**, o **de temps exponencial**, si $T = O(t^{h(n)})$ per a una constant t i un polinomi $h(n)$.

Temps d'execució de diferents algorismes

Per a n gran, les diferents classes de complexitat temporal donen lloc a temps molt diferents. Per exemple, suposem que tenim una màquina capaç de processar 10^8 instruccions per segon. La taula següent dóna el temps d'execució per a les diferents classes d'algorismes esmentades més amunt.

Classe	Complexitat	Operacions per $n = 10^8$	Temps
Pol. constant	$O(1)$	1	10^{-8} segons
Pol. lineal	$O(n)$	10^8	1 segon
Pol. quadràtic	$O(n^2)$	10^{16}	1.000 dies
Pol. cúbic	$O(n^3)$	10^{24}	$2{,}7 \cdot 10^8$ anys
Exponencial	$O(2^n)$	$10^{9.030.900}$	$10^{9.030.886}$ anys

Les xifres segures computacionalment es poden trencar cercant exhaustivament l'espai de claus i provant cada clau possible per saber si desxifra bé (amb sentit) o no. Si la mida de l'espai de claus és $n = 2^{H(K)}$, llavors el temps d'execució d'aquesta estratègia és $T = O(n) = O(2^{H(K)})$. Per tant, el temps és lineal en el nombre de claus n, però és exponencial en la longitud de les claus $H(K)$.

Aquesta és la raó per la qual doblar la llargada de la clau del DES (*Data Encryption Standard*) de 56 a 112 bits pot tenir un impacte enorme en la dificultat de trencar-lo (tot i que la distància d'unicitat només es dobla).

Vegeu el DES al subapartat 2.1 del mòdul didàctic "Xifres de clau compartida: xifres de bloc" d'aquesta assignatura.

1.3.2. Complexitat d'un problema

La teoria de la complexitat classifica un problema segons l'espai i el temps mínims que es requereixen per a resoldre'n les instàncies* més difícils amb una màquina de Turing. Una màquina de Turing és una màquina d'estats finits amb una cinta infinita de lectura/escriptura. Els problemes que es poden resoldre polinòmicament en una màquina de Turing es poden resoldre polinòmicament en un sistema real, i a l'inrevés.

*** Una instància és el problema per a un valor concret de l'entrada.**

Els **problemes tractables** són aquells que es poden resoldre en temps polinòmic en una màquina de Turing. La resta són els problemes intractables o difícils. Hi ha problemes que són tan difícils que ni tan sols no es pot escriure cap algorisme capaç de resoldre'ls; són els **problemes indecidibles**.

El conjunt dels problemes resolubles en temps polinòmic s'anomena **classe P**. La classe polinòmica no determinista, **classe NP**, engloba tots els problemes que es poden resoldre en temps polinòmic en una màquina de Turing no determinista, és a dir, si la màquina troba la solució, pot comprovar-ne la correcció en temps polinòmic. Noteu que això no significa que el problema es pugui resoldre, perquè no hi ha cap garantia que la màquina encerti la resposta correcta.

És evident que la classe NP inclou la classe P. Ara bé, hi ha problemes a NP que sembla que demanin un temps exponencial. De tota manera, no s'ha pogut demostrar que $P \neq NP$; així, doncs, de moment, no es pot excloure que algun dia es trobin algorismes polinòmics per a resoldre tots els problemes de la classe NP.

Un **problema NP-difícil** és aquell que no es pot resoldre en temps polinòmic, si no és que $P = NP$.

Un **problema NP-complet** és aquell a què qualsevol altre problema de NP es pot reduir en un temps polinòmic.

Si es trobés un algorisme polinòmic per a un problema NP-complet, aleshores s'hauria demostrat que $P = NP$.

1.4. Problemes difícils: logaritme discret i factorització

La seguretat dels criptosistemes de clau pública que funcionen en la pràctica es basa en la dificultat computacional d'alguns problemes NP de la teoria dels nombres. N'esmentarem dos: el problema del logaritme discret, sobre el qual se sosté el criptosistema d'ElGamal, i el problema de la factorització, en el qual es basa el criptosistema RSA. Per a poder entendre la formulació d'ambdós problemes, ens cal tenir prèviament unes nocions bàsiques de teoria dels grups.

1.4.1. Grups i elements primitius

Un **grup** és un conjunt G juntament amb una operació, que representarem per $*$. L'operació fa correspondre a cada parella d'elements a i b un tercer element, en concret el resultat $a * b$ d'aplicar-los l'operació. Les quatre propietats d'un grup són les següents:

- L'operació és interna: si $a, b \in G$, llavors $a * b \in G$.
- L'operació és associativa: $(a * b) * c = a * (b * c)$.
- Hi ha un element d'identitat I tal que $I * a = a * I = a$ per a tot $a \in G$.
- Tot $a \in G$ té un invers, representat per a^{-1}, tal que $a * a^{-1} = a^{-1} * a = I$.

Exemples de grups

Hi ha molts exemples de grups, com ara els següents:

- Els enters $\mathbb{Z}$ amb la suma. Tenim que $I = 0$ i que $a^{-1} = -a$.
- Els reals $\mathbb{R}$ amb la multiplicació. Tenim que $I = 1$ i que $a^{-1} = 1/a$.
- Els enters mòdul n, representats per $\mathbb{Z}_n = \{0, ..., n - 1\}$, amb la suma mòdul n. Tenim que $I = 0$ i que $a^{-1} = n - a$.
- Els enters $\mathbb{Z}_n^* = \{m : 1 \leq m \leq n$ i $\text{mcd}(m,n) = 1\}$ amb la multiplicació. Tenim que $I = 1$ i, d'altra banda, hem vist que tot enter coprimer amb n té un invers multiplicatiu mòdul n.

En un grup finit G, l'**ordre del grup** és el seu cardinal $|G|$.

Si G és un grup i $a \in G$, en col·leccionar les potències de a, és a dir, a^0, a^1, a^2, ..., podem formar el conjunt:

$$< a > = \{a^i : i \geq 0\}.$$

Si $m = |G|$ és l'ordre de G, es compleix que $a^m = I$. Per consegüent, la seqüència es repeteix al cap de m passos, és a dir, $a^{m+1} = a$, però també es podria repetir abans.

Si G és un grup i $a \in G$, el subgrup $< a > = \{a^i : i \geq 0\}$ s'anomena **subgrup generat** per a i té un ordre $t = | < a > |$ que divideix l'ordre $m = |G|$.

Subgrups generats per elements del grup $\mathbb{Z}_9^*$

Si $\mathbb{Z}_9^* = \{1,2,4,5,7,8\}$, l'ordre del grup (mida del conjunt) és el nombre d'enters menors que 9 que hi són coprimers, és a dir, $\phi(9) = 6$. En aquest cas, alguns dels subgrups generats pels seus elements són els següents:

- $< 1 > = \{1\}$.
- $< 2 > = \{1,2,4,8,7,5\}$.
- $< 4 > = \{1,4,7\}$.
- $< 5 > = \{1,5,7,8,4,2\}$.

Així, doncs, segons quin element prenguem, podem fer la volta sense que arribin a sortir tots els elements de $\mathbb{Z}_9^*$.

Un element $g \in G$ s'anomena **primitiu** o **generador de G** si les potències de g generen G. És a dir, $< g > = G$ és tot el grup G. Igualment, un grup s'anomena **cíclic** si té un element primitiu.

El grup $\mathbb{Z}_p^*$

Se sap que si p és primer, el grup següent:

$$\mathbb{Z}_p^* = \{1, 2, ..., p - 1\},$$

és cíclic. Noteu que l'ordre del grup és el nombre d'enters menors que p que hi són coprimers, és a dir, $\phi(p) = p - 1$.

1.4.2. El problema del logaritme discret

En un grup cíclic d'ordre m i generador g, per a qualsevol $y \in G$ hi ha un únic $i \in \{0, ..., m - 1\}$ tal que $g^i = y$. Aquest únic i es representa per $\log_g y$ i s'anomena **logaritme discret de y a la base g**.

És especialment interessant en criptografia el cas en què $G = \mathbb{Z}_p^*$ per a p primer i g és un generador de $\mathbb{Z}_p^*$. Aleshores es poden calcular potències mòdul p de manera ràpida. Per exemple, amb l'algorisme *exp_rapid*(a,z,n).

Donat $y = g^i \bmod p$, trobar-ne el logaritme discret i és difícil (problema del logaritme discret). El millor algorisme actual tarda un temps $O(\exp[\sqrt{\ln p \ln\ln p}])$, és a dir, un temps exponencial. Hi ha una variant recent d'aquest algorisme que sembla requerir només un temps $O(\exp[\ln p \ln\ln p]^{1/3})$.

1.4.3. El problema de la factorització

Factoritzar un enter *n* significa trobar-ne la descomposició en factors primers. Aquesta descomposició existeix sempre i és única.

En criptografia és especialment interessant la factorització d'enters de la forma $n = pq$, en què p i q són primers. En aquest cas algú coneix n, però no coneix ni p ni q i voldria trobar-los (problema de la factorització).

El problema de la factorització també és difícil. Els millors algorismes, que tarden un temps $O(\exp[\sqrt{\ln n \ln\ln n}])$, són els de Dixon i del garbell quadràtic*. Si p és el divisor primer més petit de n, hi ha un algorisme anomenat *de corba el·líptica* que tarda un temps $O(\exp[\sqrt{2\ln p \ln\ln p}])$. Aquest darrer algorisme, doncs, només s'aconsella quan se sospita que un dels dos factors de n és molt més petit que no l'altre. Hi ha una proposta recent d'algorisme de tipus garbell que sembla tardar només un temps $O(\exp[\ln n \ln\ln n]^{1/3})$.

* En anglès, *quadratic sieve*.

La complexitat del càlcul de la factorització de $n = pq$, doncs, és exponencial amb els millors algorismes que es coneixen actualment. De fet, és sorprenent veure la semblança que hi ha entre els millors algorismes coneguts per a calcular logaritmes discrets i per a factoritzar, tenint en compte que els algorismes emprats per a resoldre un problema no tenen gaire en comú amb els usats per a l'altre problema.

2. Fonaments dels criptosistemes de clau pública

La criptografia de clau compartida presenta tres inconvenients:

Vegeu els criptosistemes de clau compartida al mòdul didàctic "Xifres de clau compartida: xifres de bloc" d'aquesta assignatura.

1) La distribució de claus: dos usuaris han d'elegir una clau secreta abans de començar-se a comunicar. En aquest cas, o bé s'han de trobar personalment o bé han de confiar en un canal segur per a distribuir-se les claus. També és perfectament possible que no es puguin trobar personalment i que no disposin de cap canal segur.

2) La gestió de claus: en una xarxa de n usuaris, cada parella d'usuaris ha de tenir la seva clau compartida particular, la qual cosa implica un total de $n(n-1)/2$* claus per a tota la xarxa.

*** Combinacions de n elements presos de dos en dos.**

3) No hi ha signatura digital: la signatura digital és l'equivalent de les signatures manuals en el cas de la informació electrònica; amb un criptosistema de clau compartida, generalment no hi ha la possibilitat de signar els missatges, pel mateix fet que totes les claus són compartides almenys per dos usuaris.

El concepte de *criptosistema de clau pública*, que permet superar els inconvenients anteriors, va ser proposat per W. Diffie i M.E. Hellman en l'article "New directions in cryptography" aparegut l'any 1976. La idea, que es pot titllar de revolucionària, era permetre un intercanvi segur de missatges entre emissor i receptor sense ni tan sols haver-se de trobar prèviament per acordar una clau secreta comuna.

2.1. Concepte de clau pública

Un criptosistema de clau pública s'adreça més sovint a una xarxa d'usuaris que no a una sola parella. En aquest criptosistema cada usuari, u, té associada una parella de claus $< P_u, S_u >$: la **clau pública**, P_u, que es publica amb el nom de l'usuari en un directori públic que tothom pot llegir, mentre que la *clau privada*, S_u, només la coneix u. Els parells de claus es generen mitjançant un algorisme de generació de claus.

Per a enviar un missatge secret m a u, tothom de la xarxa fa servir el mateix mètode:

1) Cercar P_u.

2) Calcular $c = E(P_u, m)$, en què E és un **algorisme públic de xifratge**.

3) Enviar c a l'usuari u.

En rebre el text xifrat c, l'usuari u el pot desxifrar de la manera següent:

1) Cercar la seva clau privada S_u.

2) Calcular $D(S_u,c)$, en què D és un **algorisme públic de desxifratge**.

Perquè aquest procediment funcioni cal que $D(S_u,E(P_u,m)) = m$. D'aquesta manera la gestió de claus queda simplificada, perquè ara només hi ha n parelles de claus per a n usuaris, en comptes de les $n(n - 1) / 2$ claus que calien amb la criptografia de clau compartida.

2.2. Funcions unidireccional i unidireccional amb trapa

Els criptosistemes de clau pública es basen en l'existència d'alguns tipus de funcions que són difícils d'invertir.

Una **funció unidireccional*** $f: M \rightarrow C$ és una funció invertible tal que és *fàcil* de calcular $f(m) = c$, mentre que és *difícil* de calcular $f^{-1}(c) = m$.

Una **funció unidireccional amb trapa**** és una funció unidireccional que pot ser invertida fàcilment quan es coneix certa informació addicional. Aquesta informació s'anomena *trapa* (amb el sentit figurat de 'porta falsa').

*** En anglès, *one-way*.**
**** En anglès, *trapdoor one-way*.**

A les definicions anteriors, *fàcil* vol dir en temps polinòmic i *difícil* en temps exponencial. Curiosament, no se sap del cert si hi ha funcions unidireccionals ni, encara menys, funcions unidireccionals amb trapa. De tota manera, hi ha funcions susceptibles de ser considerades així.

Possibles funcions unidireccionals

Basant-nos en allò que hem dit en explicar els problemes difícils, podrien ser unidireccionals les funcions següents:

- Exponencial discreta. Amb l'algorisme exp_rapid(a,z,n) o un de semblant, és fàcil d'exponenciar. En canvi, tornar enrere calculant logaritmes discrets és difícil.
- Producte de primers. És fàcil multiplicar nombres primers. En canvi, és difícil de tornar enrere factoritzant el producte obtingut.

3. Intercanvi de claus de Diffie-Hellman

En un article de 1976, W. Diffie i M.E. Hellman van descriure un protocol d'intercanvi de claus que evita l'inconvenient de la distribució de claus. El fet innovador és que dos usuaris poden pactar una clau secreta compartida sobre un canal insegur.

Recordeu que hem estudiat el problema de la distribució de claus a l'apartat 2 d'aquest mòdul didàctic.

Protocol 1: Intercanvi Diffie-Hellman

Per a dur a terme correctament el protocol d'intercanvi de Diffie-Hellman, s'han de seguir els passos següents:

1) Els dos usuaris, A i B, trien públicament un grup multiplicatiu finit G d'ordre n i un generador $\alpha \in G$.

2) A genera un nombre aleatori a, calcula α^a dins de G i transmet el resultat a B.

3) B genera un nombre aleatori b, calcula α^b dins de G i transmet el resultat a A.

4) A rep α^b i calcula $(\alpha^b)^a$ dins de G.

5) B rep α^a i calcula $(\alpha^a)^b$ dins de G.

Al final del protocol 1, A i B han pactat un element α^{ab} del grup G que és comú entre ells dos i secret per a la resta d'usuaris. La seguretat de l'intercanvi de Diffie-Hellman es basa en la dificultat del problema de Diffie-Hellman generalitzat.

La idea de l'intercanvi...

... la va tenir Hellman, segons explica W. Diffie:

"Finalment, Marty [Hellman] va fer el gran pas un matí de maig del 1976. [...] Marty em va trucar i m'explicà l'intercanvi de claus exponencial: era d'una simplicitat desconcertant. Mentre l'escoltava em vaig adonar que feia temps que ja ho sabia, però no n'era conscient."

Problema de Diffie-Hellman generalitzat (PDHG): el criptoanalista pot conèixer tota l'execució del protocol 1, amb la qual cosa obté G, n, α , α^a i α^b. Amb aquesta informació vol calcular α^{ab}.

Si $G = \mathbb{Z}_p^*$, amb p primer, el PDHG s'anomena simplement *problema de Diffie-Hellman* (PDH). Remarquem la semblança amb el problema del logaritme discret generalitzat, que enunciem tot seguit.

Problema del logaritme discret generalitzat (PLDG): un criptoanalista obté G, n, α i α^a, i vol calcular a.

En realitat, no s'ha demostrat que PDHG i PLDG siguin equivalents. De tota manera, no s'ha trobat com resoldre PDHG sense resoldre PLDG. El que és clar és que si el criptoanalista sabés resoldre PLDG, aleshores podria trobar a (respectivament, b) a partir de α^a (respectivament, α^b) i, per consegüent, α^{ab}.

Funcionament del protocol de Diffie-Hellman

Il·lustrem el funcionament del protocol de Diffie-Hellman amb l'exemple següent:

1) A i B elegeixen públicament $G = \mathbb{Z}_{53}^*$ i el generador $\alpha = 2$ (podeu comprovar que les potències de 2 generen G).

2) A tria $a = 29$, calcula $\alpha^a = 2^{29} \bmod 53 = 45$ i envia 45 a B.

3) B tria $b = 19$, calcula $\alpha^b = 2^{19} \bmod 53 = 12$ i envia 12 a A.

4) A rep 12 i calcula $12^{29} \bmod 53 = 21$.

5) B rep 45 i calcula $45^{19} \bmod 53 = 21$.

Només A i B saben que el nombre compartit és 21. Un criptoanalista enemic veu $\mathbb{Z}_{53}^*$, 2, 45 i 12.

4. Criptosistemes de clau pública

En aquest apartat tractem dels dos criptosistemes de clau pública més utilitzats actualment: l'RSA i l'ElGamal. El primer es basa en el problema de la factorització, mentre que el segon ho fa en el problema del logaritme discret.

Lectura complementària

L'estudiant que estigui interessat a trobar més informació sobre altres criptosistemes de clau pública pot consultar el llibre següent:

D. Stinson (1995). *Cryptography: Theory and Practice*. Boca Ratón: CRC Press.

Val a dir que hi ha altres criptosistemes de clau pública, com ara el de McEliece, el de Rabin, els de motxilla, etc. De tota manera, per raons de poca eficiència (McEliece i Rabin) o bé de poca seguretat (els criptosistemes de motxilla o *knapsack*) no n'hi ha cap que sigui una alternativa pràctica a l'RSA i a l'ElGamal.

Per simplicitat, els exemples que posarem tant d'RSA com d'ElGamal són sobre grups multiplicatius enters del tipus $\mathbb{Z}_n^*$. No obstant això, es poden emprar les anomenades *corbes el·líptiques* com a alternativa als grups multiplicatius enters. L'avantatge de fer servir aquestes corbes és que s'obtenen implementacions més ràpides dels criptosistemes.

Lectures complementàries

L'estudiant interessat en les corbes el·líptiques pot consultar els llibres següents:

A. Fuster, D. De la Guía, L. Hernández, F. Montoya, J. Muñoz (1997). T*écnicas criptográficas de protección de datos*. Madrid: Ra-ma.

D. Stinson (1995). *Cryptography: Theory and Practice*. Boca Ratón: CRC Press.

4.1. El criptosistema RSA

En l'article inicial de 1976, Diffie i Hellman van formular la idea d'un criptosistema de clau pública, però no en van donar cap exemple pràctic. La primera aplicació de la proposta va ser el criptosistema RSA, publicat per Rivest, Shamir i Adleman el 1978 en el cèlebre article "A method for obtaining digital signatures and public-key cryptosystems". L'algorisme de generació de claus de l'RSA l'exposem tot seguit.

Algorisme de generació de claus de l'RSA

Per a generar les claus que es fan servir en l'RSA cal seguir el passos següents:

1) Cada usuari, u, tria dos nombres primers, p i q*. Llavors calcula $n = pq$, amb la qual cosa el grup multiplicatiu que farà servir u és $\mathbb{Z}_n^*$. L'ordre d'aquest grup és $\phi(n) = \phi(pq) = (p - 1)(q - 1)$. Per a u és fàcil de calcular aquest ordre, ja que coneix p i q.

*** Actualment es recomana que cadascun d'aquests primers tingui més de 200 dígits.**

2) L'usuari u tria un enter positiu, e, que compleixi les relacions $1 \leq e < \phi(n)$ i $\text{mcd}(e,\phi(n)) = 1$.

3) Amb l'algorisme d'Euclides estès, u calcula l'invers d de e a $\mathbb{Z}_{\phi(n)}$*. Tenim, doncs, $ed \bmod \phi(n) = 1$.

*** L'invers existeix per la manera com s'ha elegit *e*.**

4) La clau pública de u és el parell (n,e), mentre que la seva clau privada és d. Per descomptat, p, q i $\phi(n)$ també romanen secrets.

Si un usuari A vol enviar un missatge $m \in \mathbb{Z}^*_{n_b}$ a un altre usuari B, fa servir la clau pública de B, (n_b,e_b), per a calcular el valor $m^{e_b} \bmod n_b = c$, el qual és enviat a B.

Per a recuperar el missatge original, B calcula:

$$c^{d_b} \bmod n_b = (m^{e_b})^{d_b} \bmod n_b = m^{e_b d_b} \bmod n_b =$$
$$= m^{1+t\phi(n_b)} \bmod n_b = (m \bmod n_b)(m^{\phi(n_b)} \bmod n_b)^t \bmod n_b = m.$$

En l'última igualtat anterior s'ha fet servir el teorema d'Euler, que garanteix que $m^{\phi(n_b)} \bmod n_b = 1$ si $\text{mcd}(m,n_b) = 1$ (i això darrer és gairebé segur que passarà, perquè n_b és molt gran i només té dos divisors, p_b i q_b).

Codificació i descodificació d'un missatge utilitzant una clau pública

Considerem una codificació de l'alfabet que transforma les lletres A-Z en els nombres del 0 al 25. Suposem que:

a) L'usuari B ha elegit els primers $p_b = 281$ i $q_b = 167$, amb la qual cosa ha obtingut $n_b = 281 \cdot 167 = 46.297$ i fa servir el grup $\mathbb{Z}^*_{46.927}$.

b) L'ordre d'aquest grup és $\phi(46.927) = 280 \cdot 166 = 46.480$. B elegeix $e_b = 39.423$ i comprova que $\text{mcd}(39.423,46.480) = 1$. Llavors determina l'invers de 39.423 mòdul 46.480 i obté $d_b = 26.767$.

c) La clau pública de B és $(n_b,e_b) = (46.927, 39.423)$ i la resta de valors es mantenen secrets.

Per a enviar un missatge a B, hem de mirar primer quina llargada té: aquest ha de pertànyer al grup on treballem i, per tant, no pot superar $n_b = 46.927$. D'altra banda, codifiquem cada lletra en base 26 i com que:

$$26^3 = 17.576 < n_b < 456.976 = 26^4,$$

el missatge pot tenir fins a tres lletres. Si volguéssim enviar un missatge més llarg, l'hauríem de partir en blocs de tres lletres. Cal tenir en compte que les n_b emprades són molt més grans en la pràctica (tenen almenys 400 dígits decimals).

Imaginem que enviem a B el missatge YES. Codificat en base 26, té la forma següent:

$$Y \cdot 26^2 + E \cdot 26 + S = 24 \cdot 26^2 + 4 \cdot 26 + 18 = 16.346 = m.$$

Aleshores xifrem m amb la clau pública de B, amb la qual cosa obtenim:

$$c = m^{e_b} \bmod n_b = 16.346^{39.423} \bmod 46.927 = 21.166.$$

Si volem expressar el missatge xifrat com a lletres, tenim que 21.166 és la codificació en base 26 de les lletres BFIC.

Suposem que B ha rebut BFIC. Llavors el codifica en base 26 i obté 21.166. Tot seguit recupera:

$$m = c^{d_b} \bmod n_b = 21.166^{26.767} \bmod 46.927 = 16.346.$$

Finalment, B descodifica m i obté el text original, YES.

Una qüestió tècnica és que l'algorisme de generació de claus RSA requereix trobar primers molt grans (200 dígits decimals). Hi ha diversos mètodes eficients per a confirmar que un nombre gran és primer; per exemple, els tests

de primeritat de Solovay-Strassen, Miller-Rabin i Goldwasser-Kilian. Detallem aquí el primer d'aquests tests i remetem el lector als llibres indicats a la bibliografia, si desitja conèixer altres mètodes.

Obtenció de nombres primers amb el test de primeritat de Solovay-Strassen

Per a trobar un nombre primer que sigui elevat, els autors de l'RSA recomanen generar nombres aleatoris fins que se'n trobi un que sigui probablement primer. Per testejar la primeritat d'un nombre, b, se'n poden generar cent d'aleatoris $a \in [1,b-1]$ i verificar si tots els nombres a passen el test de Solovay-Strassen:

$$\text{mcd}(a,b) = 1 \text{ i } J(a,b) \bmod b = a^{(b-1)/2} \bmod b,$$

on $J(a,b)$ és el **símbol de Jacobi**.

Si b és primer, l'equació anterior es compleix per a tot $a \in [1,b-1]$. Si b no és primer, l'equació es compleix amb una probabilitat màxima d'1/2 per a cada a, de tal manera que la probabilitat que cent nombres a diferents passin el test si b no és primer és només d'$1/2^{100}$.

Quan $\text{mcd}(a,b) = 1$, el símbol de Jacobi $J(a,b)$ es pot calcular d'una manera eficient amb l'algorisme recursiu següent:

Algorisme *J*(*a*,*b*)

```
begin
   if a = 1 then J := 1
   else if a mod 2 = 0 then
   begin
      if (b * b - 1) / 8 mod 2 = 0 then
      J := J(a/2,b) else J := -J(a/2,b)
   end
   else if (a - 1) * (b - 1) / 4 mod 2 = 0 then
   J := J(b mod a,a) else J := - J(b mod a,a)
end
```

4.1.1. Velocitat de l'RSA

En la pràctica, perquè el procés de xifratge amb l'RSA sigui més ràpid se sol escollir un exponent públic e petit; sovint es pren $e = 3$. Aquesta tria no compromet la seguretat del sistema, que descansa en el secret de la clau privada d.

Si ens fixem en la velocitat que s'aconsegueix en les implementacions RSA, cal reconèixer dos fets:

a) En programari, el criptosistema de clau compartida DES arriba a funcionar cent vegades més de pressa que les millors implementacions d'RSA.

b) En maquinari, el DES arriba a ser entre 1.000 i 10.000 vegades més ràpid que no l'RSA.

Malgrat la lentitud relativa, l'RSA i els altres criptosistemes de clau pública es fan servir perquè permeten també signar digitalment els missatges enviats. Normalment es fa per mitjà d'una combinació de clau pública i de clau secreta coneguda com a **sobre digital**, que ofereix la velocitat de la clau compartida i la flexibilitat de la clau pública. Per a treballar en aquesta modalitat cal seguir els passos següents:

Vegeu la signatura digital a l'apartat 1 del mòdul didàctic "Signatures digitals" d'aquesta assignatura.

1) L'usuari, *A*, xifra el missatge, *m*, amb un criptosistema de clau compartida (per exemple, el DES) i una clau aleatòria.

2) *A* envia a *B* el missatge xifrat, *c*. Tot seguit, *A* envia a *B* la clau aleatòria xifrada amb l'RSA sota la clau pública de *B*.

3) Gràcies a la seva clau privada RSA, *B* recupera la clau aleatòria DES i la fa servir per a desxifrar el missatge *c* i recuperar *m*.

4.1.2. Seguretat de l'RSA

Per a trencar el criptosistema RSA, un criptoanalista ha de trobar el valor *d*, que és la trapa de la funció unidireccional $f(m) = m^e \bmod n$, però això requereix conèixer $\phi(n)$. Amb tot, no és fàcil trobar $\phi(n)$ a partir de *n*, si no és que se sap la factorització $n = pq$.

D'acord amb el que acabem d'explicar, podem dir que la seguretat del criptosistema RSA es basa en el problema de la factorització.

No obstant això, no s'ha pogut demostrar que l'única manera de trobar *d* a partir de *e* passi per factoritzar *n*.

Com que la seguretat de l'RSA està relacionada amb el problema de la factorització, convé elegir els dos primers *p* i *q* que componen *n* de manera que la factorització sigui tan difícil com es pugui. El que se sol fer és triar *p* i *q* aleatòriament, entre els primers que compleixen les condicions següents:

1) p i q han de tenir una magnitud semblant, però no han de ser massa propers.

2) $(p - 1)$ i $(q - 1)$ han de tenir factors primers grans.

3) mcd$(p - 1, q - 1)$ ha de ser petit.

Justifiquem a continuació cadascuna d'aquestes condicions:

1) Si p i q fossin massa propers, amb $p > q$, llavors $(p - q) / 2$ seria petit i $(p + q) / 2$ només seria una mica més gran que $\sqrt{n}$.

D'altra banda,

$$\frac{(p + q)^2}{4} - n = \frac{(p - q)^2}{4}.$$

Així, doncs, el membre de l'esquerra de l'equació anterior és un quadrat perfecte. Per a factoritzar n n'hi ha prou de provar els enters $x > \sqrt{n}$ fins que n'hi hagi un que faci que $x^2 - n$ sigui un quadrat perfecte y^2. Aleshores, $p = x + y$ i $q = x - y$. L'atac no tarda gaire a triomfar, perquè l'x que cerquem és $(p + q) / 2$ i sabem que no és gaire més gran que $\sqrt{n}$.

2) La segona condició sobre p i q demana que $(p - 1)$ i $(q - 1)$ tinguin factors primers grans. Si això no passés, llavors tots els factors de $\phi(n) = (p - 1)(q - 1)$ serien petits. Si k és una fita superior dels factors primers de $\phi(n)$, és senzill provar tots els valors v candidats a ser $\phi(n)$: cal provar combinacions de factors primers menors que k en què cada factor r va elevat a un exponent entre 1 i $\lfloor \log_v n \rfloor$. Per comprovar si ho hem fet bé, podem elevar un criptograma m^e a $(v + 1) / e$ si aquest exponent és enter. Si $v = \phi(n)$, això vol dir elevar m^e a $(\phi(n) + 1) / e$, amb la qual cosa recuperarem el missatge en clar m (teorema d'Euler) i haurem confirmat l'encert.

3) La tercera condició (sobre el mcd(p,q)) es justifica perquè, com que $p - 1$ i $q - 1$ són parells, $\phi(n)$ és divisible per 4. Si mcd$(p - 1, q - 1)$ fos gran, llavors mcm$(p - 1, q - 1) = mcm$ seria petit en comparació amb $\phi(n) = (p - 1)(q - 1)$. Convé que ens adonem que qualsevol invers de e mòdul mcm serveix per a desxifrar, i que trobar mcm és fàcil si se sap que és petit.

Una manera d'elegir p i q perquè compleixin les tres condicions anteriors és triar un primer r gran tal que $p = 2r + 1$ i $q = 2p + 1$ siguin primers. Un nombre p triat així s'anomena **primer segur**.

La primera condició es compleix: p i q són de magnitud semblant, però q és el doble de p. La segona condició també es compleix, ja que $p - 1 = 2r$ i $q - 1 = 2p$ tenen factors primers grans, r i p respectivament. Finalment, veiem que mcd$(p - 1, q - 1) = 2$ és petit, cosa que satisfà la tercera condició.

4.2. El criptosistema d'ElGamal

El 1985, T. ElGamal va publicar un criptosistema de clau pública basat en l'exponenciació discreta sobre un grup multiplicatiu $\mathbb{Z}_p^*$, en què p és un primer gran (400 dígits decimals almenys). És fàcil de generalitzar la proposta d'ElGamal a un grup finit G.

Protocol de generació de claus de l'ElGamal

Per a generar les claus que es fan servir en l'ElGamal cal seguir els passos següents:

1) Es tria un grup finit, G, i un element $\alpha \in G$ que no ha de ser necessàriament un generador.

2) Cada usuari A tria un nombre aleatori, a, que serà la seva clau privada, i calcula α^a dins de G, que serà la seva clau pública.

Si un usuari A vol enviar un missatge, $m \in G$, a un usuari B, llavors fa el següent:

1) A genera un nombre aleatori v i calcula α^v dins de G.

2) A mira la clau pública de B, α^b, i calcula $(\alpha^b)^v$ i $m \cdot \alpha^{bv}$ dins de G.

3) A envia a B el parell $(\alpha^v, m \cdot \alpha^{bv})$.

Per a recuperar el missatge original, B ha de fer el següent:

1) B calcula $(\alpha^v)^b$ dins de G.

2) B obté m només dividint la segona part del criptograma:

$$\frac{m \cdot \alpha^{vb}}{\alpha^{vb}}.$$

Codificació i descodificació d'un missatge amb el criptosistema d'ElGamal

Per a il·lustrar el funcionament de l'ElGamal, suposem que s'ha elegit el grup $G = \mathbb{Z}_{15.485.863}^*$ i $\alpha = 7$ (de fet, les potències de 7 generen tot G).

L'usuari B ha triat $b = 21.702$ i ha calculat la seva clau pública de la manera corresponent:

$$\alpha^b = 7^{21702} \bmod 15.485.863 = 8.890.431.$$

Suposem que A vol enviar a B el missatge AMIC. Codificat en base 26, el missatge té la forma següent:

$$\text{AMIC} = 0 \cdot 26^3 + 12 \cdot 26^2 + 8 \cdot 26 + 2 = 8.322 = m.$$

Després A tria el nombre aleatori $v = 480$ i calcula:

$$\alpha^v = 7^{480} \bmod 15.485.863 = 12.001.315.$$

Tot seguit, A calcula:

$$(\alpha^b)^v = 8.890.431^{480} \bmod 15.485.863 = 9.846.598$$

i codifica el missatge:

$$m \cdot (\alpha^b)^v = 8.322 \cdot 9.846.598 \bmod 15.485.863 = 7.687.423.$$

Llavors A envia a B els nombres 12.001.315 i 7.687.423 o les lletres de l'alfabet resultants de descodificar aquests nombres en base 26.

B rep el missatge (si és en format lletra li caldrà codificar-lo en base 26). Tot seguit, B calcula:

$$(\alpha^v)^b = 12.001.315^{21.702} \bmod 15.485.863 = 9.846.598.$$

Després B ha de calcular $(m \cdot \alpha^{vb}) / \alpha^{vb}$. Per a fer-ho, ha de trobar l'invers de α^{vb} mòdul 15.485.843. Fent servir l'algorisme d'Euclides estès, tenim que $\alpha^{-vb} = 14.823.281$.

Finalment, descodifica el missatge:

$$m = \frac{m \cdot \alpha^{vb}}{\alpha^{vb}} = (7.687.423 \cdot 14.823.281) \bmod 15.485.863 = 8.322.$$

4.2.1. Velocitat de l'ElGamal

Igual que l'RSA, el criptosistema d'ElGamal ha de fer exponenciacions. La diferència és que les exponenciacions de xifratge no depenen del missatge m, sinó d'un exponent aleatori v elegit per l'emissor. Per tant, es poden calcular abans d'enviar el missatge. De fet, l'única operació que cal fer forçosament en el mateix moment que es xifra m és un producte (molt més ràpid de fer que no una exponenciació). En conseqüència, el xifratge de l'ElGamal pot ser més ràpid que no el xifratge de l'RSA.

El desxifratge de l'ElGamal requereix una exponenciació i un producte, és a dir, pràcticament el mateix temps que un desxifratge de l'RSA, que requeria una exponenciació. Un petit inconvenient de l'ElGamal és que el criptosistema és una parella d'enters, que solen ocupar el doble de bits que el missatge en clar (expansió del text xifrat)

4.2.2. Seguretat de l'ElGamal

El criptosistema d'ElGamal fa servir l'exponenciació discreta com a funció unidireccional; més concretament, per a enviar a l'usuari B la funció unidireccional és $f(m) = (\alpha^v, m \cdot \alpha^{bv})$ i la trapa és la clau privada b. La seguretat del criptosistema d'ElGamal es basa en la dificultat del problema que presentem a continuació.

Problema d'ElGamal (PEG): el criptoanalista que escolta la transmissió xifrada de A cap a B sap G, n, α, α^a, α^b, α^v i $m \cdot \alpha^{vb}$. Ell vol calcular m.

És fàcil adonar-se que el PEG equival al problema de Diffie-Hellman. Per tant, com en el cas del PDHG, no s'ha demostrat que el PEG sigui equivalent al problema del logaritme discret (PLDG), però tampoc no s'ha trobat fins ara cap atac general contra l'ElGamal que no passi per calcular logaritmes discrets. És evident que si el criptoanalista sabés resoldre logaritmes discrets, llavors podria trobar la clau privada b del destinatari B a partir de la clau pública α^b, i amb la clau b podria desxifrar el criptograma i recuperar m.

Vegeu el problema de Diffie-Hellman a l'apartat 3 d'aquest mòdul didàctic.

Cal remarcar aquí que la proposta original d'ElGamal feia servir $G = \mathbb{Z}_p^*$ per a p molt gran i α un generador de G. Si ho generalitzem a qualsevol grup G i qualsevol element $\alpha \in G$, cal que ens assegurem que el subgrup cíclic $< \alpha >$ generat per les potències de α sigui prou gran perquè continuï essent difícil calcular-hi logaritmes.

Resum

L'**aritmètica modular** i, més generalment, la **teoria dels nombres** són la base sobre la qual s'implementen els criptosistemes de clau pública.

L'**intercanvi de claus de Diffie-Hellman** permet a dos usuaris pactar una clau secreta compartida sobre un canal insegur.

Si emprem criptografia de clau pública en una xarxa de n usuaris, n'hi ha prou que cada usuari tingui dues claus (una de pública i una de privada) per a permetre la comunicació confidencial entre qualsevol parella d'usuaris. Recordem que, amb clau compartida, caldrien $n(n - 1) / 2$ claus compartides. En efecte, per a enviar un missatge confidencial a un usuari B, qualsevol altre usuari A ha de xifrar el missatge sota la clau pública de B. Només B podrà recuperar el missatge original desxifrant el criptograma rebut amb la seva clau privada.

Els criptosistemes de clau pública pràctics ofereixen seguretat computacional. La fortalesa d'una xifra segura computacionalment queda determinada per la **complexitat de càlcul dels algorismes** que són necessaris per a trencar-la.

Els dos criptosistemes més emprats són l'**RSA** i l'**ElGamal**, que fonamenten la seguretat en els problemes de la factorització i del logaritme discret, respectivament.

Tot i que l'RSA i l'ElGamal es poden fer una mica més ràpids si s'implementen amb corbes el·líptiques, el fet és que són considerablement més lents que no els criptosistemes de clau compartida de tipus DES. Tot i així, la simplicitat en la gestió de claus i la possibilitat d'usar-los per a fer signatures digitals els fan atractius. El **sobre digital** és una tècnica que combina els avantatges de la clau pública i de la clau compartida: el missatge s'envia xifrat amb un criptosistema de clau compartida, sota una clau aleatòria, i tot seguit s'envia la clau aleatòria xifrada amb la clau pública del destinatari.

Activitats

1. Cerqueu a Internet implementacions d'RSA i d'ElGamal.

2. Programeu l'algorisme exp_rapid(n,z,a).

3. Programeu l'algorisme d'Euclides estès.

4. Programeu el test de primeritat de Solovay-Strassen.

5. Escriviu un programa que simuli l'intercanvi de Diffie-Hellman amb nombres petits (per exemple, p de 8 dígits decimals). Simuleu l'execució del protocol per dos usuaris.

6. Escriviu programes que implementin el xifratge/desxifratge de l'RSA amb nombres petits (per exemple, un mòdul n de 8 dígits decimals). Us caldrà fer servir l'algorisme d'exponenciació ràpida, l'algorisme d'Euclides estès i el test de primeritat que heu programat en les activitats 3, 4 i 5, respectivament.

7. Escriviu programes que implementin el xifratge/desxifratge de l'ElGamal amb nombres petits (per exemple, un p de 8 dígits decimals). Us caldrà fer servir l'algorisme d'exponenciació ràpida i l'algorisme d'Euclides estès que heu programat en les activitats 2 i 3, respectivament.

8. Cerqueu a Internet informació sobre llibreries multiprecisió. Podeu cercar *multiprecision* o bé llibreries concretes, com *gpm* i *LiDIA*. Es tracta de llibreries que tenen rutines aritmètiques per a sumar, restar, exponenciar, etc. nombres de llargada arbitràriament gran (centenars de dígits decimals). Aquestes llibreries són necessàries per a produir programaris seriosos dels criptosistemes RSA i ElGamal.

Exercicis d'autoavaluació

1. Demostreu el principi de l'aritmètica modular (teorema 1).

2. Amb l'algorisme d'Euclides bàsic (algorisme mcd(a,n)), calculeu pas per pas el màxim comú divisor de 2.678 i 346.

3. Trobeu l'invers de 24 a $\mathbb{Z}^*_{37}$.

4. Comproveu que $80 = \sum_{d|80} \phi(d)$.

5. Per a cada equació de la forma $ax \bmod n = b$ indicada a continuació, trobeu el valor x al rang $[0,n-1]$.
- $5x \bmod 17 = 1$.
- $19x \bmod 26 = 1$.
- $17x \bmod 100 = 1$.
- $17x \bmod 100 = 10$.

6. A partir de les congruències $x \bmod 3 = 2$ i $x \bmod 5 = 4$, trobeu x fent servir el teorema xinès del residu. En quin rang es troba x?

7. Al criptosistema d'ElGamal, preneu $p = 23$, $\alpha = 5$ (α és primitiu a $\mathbb{Z}^*_{23}$). Detalleu com un usuari B generaria el seu parell de claus, pública i privada. Detalleu pas per pas com vosaltres (usuari A) xifraríeu la vostra edat en anys mòdul 23 per a enviar-la a l'usuari B.

Solucionari

Exercicis d'autoavaluació

1. Per començar, escrivim:
- $a_1 = k_1\, n + r_1$,
- $a_2 = k_2\, n + r_2$,

en què $r_1, r_2 \in [0, n-1]$.

a) Per a la suma tenim:

$$(a_1 + a_2) \bmod n = [(k_1\, n + r_1) + (k_2\, n + r_2)] \bmod n =$$
$$= [(k_1 + k_2)n + (r_1 + r_2)] \bmod n = [r_1 + r_2] \bmod n =$$
$$= [(a_1 \bmod n) + (a_2 \bmod n)] \bmod n.$$

b) La subtracció es demostra igual que la suma.
c) Finalment, per a la multiplicació tenim:

$$(a_1 \cdot a_2) \bmod n = [(k_1\, n + r_1) \cdot (k_2\, n + r_2)] \bmod n =$$
$$= [(k_1\, k_2\, n + r_1\, k_2 + r_2\, k_1)\, n + r_1\, r_2] \bmod n = [r_1 \cdot r_2] \bmod n =$$
$$= [(a_1 \bmod n) \cdot (a_2 \bmod n)] \bmod n.$$

2. Cal seguir l'algorisme mcd(a,n) fent una taula amb l'estat de les variables en cada pas, i al final obtenim que el màxim comú divisor és 2.

3. Cal seguir l'algorisme inv(a,n) fent una taula amb l'estat de les variables en cada pas, i al final obtenim que l'invers de 24 mòdul 37 és 17.

4. Els divisors de $80 = 2^4 \cdot 5$ són 1, 2, 4, 5, 8, 10, 16, 20, 40 i 80.
D'altra banda, $\phi(1) = 1$, $\phi(2) = 2 - 1 = 1$, $\phi(4) = 2(2-1) = 2$, $\phi(5) = 5 - 1 = 4$, $\phi(8) = 2^2(2-1) = = 4$, $\phi(10) = (2-1)(5-1) = 4$, $\phi(16) = 2^3(2-1) = 8$, $\phi(20) = 2(2-1)(5-1) = = 8$, $\phi(40) = 2^2(2-1)(5-1) = 16$ i $\phi(80) = 2^3(2-1)(5-1) = 32$.

Si sumem tots els valors de ϕ tenim:

$$\sum_{d|80} \phi(d) = 1 + 1 + 2 + 4 + 4 + 4 + 8 + 8 + 16 + 32 = 80.$$

5. Per a resoldre $5x \bmod 17 = 1$ cal trobar l'invers de 5 mòdul 17. Aquest invers existeix, ja que mcd(5,17) = 1, i amb l'algorisme d'Euclides estès tenim $x = 7$.
La segona equació i la tercera es resolen igual si ens adonem que els inversos respectius existeixen, perquè mcd(19,26) = 1 i mcd(17,100) = 1.
Per a resoldre l'equació $17x \bmod 100 = 10$, primer dividim el membre de la dreta per 10 i el deixem a 1.
Llavors resolem $17y \bmod 100 = 1$, cosa que implica trobar l'invers y de 17 mòdul 100. Amb l'algorisme d'Euclides estès obtenim $y = 53$. Aleshores trobem la solució x de l'equació original multiplicant y per 10 mòdul 100, és a dir:

$$x = 10y \bmod 100 = 10 \cdot 53 \bmod 100 = 30.$$

6. Els mòduls de les equacions són coprimers i, per tant, podem fer servir el teorema xinès del residu. Trobem y_1 com l'invers de $(3 \cdot 5) / 3 = 5$ mòdul 3; tenim $y_1 = 2$. Trobem y_2 com l'invers de $(3 \cdot 5) / 5 = 3$ mòdul 5; tenim $y_2 = 2$. Llavors:

$$x = \left[\left(\frac{n}{d_1}\right) y_1 x_1 + \left(\frac{n}{d_2}\right) y_2 x_2\right] \bmod n = \left[\frac{15}{3} \cdot 2 \cdot 2 + \frac{15}{5} \cdot 2 \cdot 4\right] \bmod 15 = 14,$$

x s'havia de trobar al rang $[1, 15-1]$, és a dir, no podia ser superior a 14.

7. Tenim $p = 23$ i $\alpha = 5$. Suposem que l'usuari B tria la seva clau privada $b = 3$. Llavors calcula la seva clau pública com $\alpha^b = 5^3 \bmod 23 = 10$. Suposem que la nostra edat en anys mòdul 23 és $m = 11$. Llavors, per xifrar m triem aleatòriament v, per exemple $v = 6$, i calculem $\alpha^v = = 5^6 \bmod 23 = 8$. Prenem la clau pública de B i calculem $(\alpha^b)^v = 10^6 \bmod 23 = 6$. Després calculem $m\, \alpha^{bv} = (11 \cdot 6) \bmod 23 = 20$. El criptograma que s'ha d'enviar a B és (8,20).

Glossari

Algorisme de temps exponencial: algorisme el temps d'execució del qual és $T = O(t^{h(n)})$ per a una constant t i un polinomi $h(n)$.

Algorisme de temps polinòmic: algorisme el temps d'execució del qual és $T = O(n^t)$ per a una constant t.

Cíclic: grup que conté un element primitiu o generador.

Complexitat d'un algorisme: temps de càlcul i espai d'emmagatzematge que requereix un algorisme determinat.

Complexitat d'un problema: espai i temps mínims que són necessaris per a resoldre les instàncies més difícils del problema amb una màquina de Turing.

Criptosistema de clau pública: criptosistema en què cada usuari té una clau pública i una de privada; amb aquest criptosistema una parella d'usuaris es pot comunicar confidencialment sense compartir una clau secreta.

Element primitiu: element generador.

ElGamal: criptosistema de clau pública i basat en el problema del logaritme discret.

Factorització: problema que consisteix a trobar els factors primers d'un enter i que actualment es considera intractable.

Funció unidireccional: funció unidireccional invertible que és fàcil de calcular en sentit directe i difícil de calcular en sentit invers.

Funció unidireccional amb trapa: funció que és fàcil de calcular en sentit invers només si es coneix una informació addicional anomenada *trapa* o *porta falsa*.

Generador: element d'un grup les potències successives del qual generen tot el grup.

Grup: conjunt tancat sota una certa operació associativa, dins el qual tots els elements tenen invers.

Ordre d'un grup: nombre d'elements, finit o infinit, que conté un grup.

Problema del logaritme discret: problema que consisteix a invertir l'exponenciació modular i que actualment es considera intractable.

Primer segur: primer que compleix els requisits de seguretat establerts per l'RSA.

Problema difícil: manera d'anomenar un problema intractable.

Problema indecidible: problema per al qual no és possible trobar un algorisme que el resolgui.

Problema intractable: problema que no es pot resoldre en temps polinòmic amb una màquina de Turing.

Problema tractable: problema que es pot resoldre en temps polinòmic amb una màquina de Turing.

RSA: criptosistema de clau pública molt utilitzat, publicat per Rivest, Shamir i Adleman l'any 1978; es basa en el problema de la factorització.

Sobre digital: tècnica que combina la criptografia de clau compartida i la de clau pública per tal d'aprofitar la velocitat de la primera i la flexibilitat de la segona.

Subgrup generat: conjunt de potències diferents de l'element, calculades amb l'operació del grup.

Test de primeritat: procediment per a determinar, sovint de manera probabilística, si un enter és primer.

Bibliografia

Denning, D.E. (1982). *Cryptography and Data Security*. Reading (Massachusetts): Addison-Wesley.

Fuster, A.; De la Guía, D.; Hernández, L.; Montoya, F.; Muñoz, J. (1997). *Técnicas criptográficas de protección de datos*. Madrid: Ra-ma.

Stinson, D. (1995). *Cryptography: Theory and Practice*. Boca Ratón: CRC Press.

Signatures digitals

Josep Domingo Ferrer

P03/05024/02264

UOC
www.uoc.edu

Índex

Signatures digitals

Introducció

En aquest mòdul didàctic presentem el concepte de *signatura digital*, que és una analogia per a documents electrònics de la signatura manual per a documents en paper. Veurem, però, que la seguretat que ofereixen les signatures digitals és molt superior.

Comencem el primer apartat definint el concepte de **signatura digital**, relacionant-lo amb el de *criptosistema de clau pública*.

Tot seguit, en el segon apartat, veiem les signatures digitals més emprades actualment: la **signatura RSA**, la **signatura d'ElGamal** i el ***Digital Signature Standard*** (DSS).

En el darrer apartat definim el concepte de **funció resum** o ***hash***. Les funcions *hash* proporcionen un resum de longitud fixa d'un missatge arbitràriament llarg. Normalment, per a signar un document, se'n calcula la imatge per una funció *hash* i se signa aquesta imatge en comptes del document original.

Objectius

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

1. Entendre el concepte i la utilitat de les signatures digitals.

2. Conèixer els principals algorismes de signatura emprats avui dia.

3. Entendre el paper de les funcions *hash* en la signatura digital.

1. Signatura digital

La noció de **signatura digital** és probablement una de les troballes fonamentals i més útils de la criptografia moderna. Amb un esquema de signatura digital cada usuari pot signar missatges de tal manera que les signatures poden ser verificades més tard per qualsevol persona.

1.1. Idea bàsica

Les signatures mantenen una estreta relació amb la criptografia de clau pública. Més concretament, cada usuari crea un parell de claus: una de pública i una de privada. Per a signar un missatge, l'usuari fa servir la seva clau privada; per a verificar una signatura, qualsevol pot fer servir la clau pública del signatari.

Vegeu el concepte de *clau pública* al subapartat 2.1 del mòdul didàctic "Xifres de clau pública" d'aquesta assignatura.

El verificador queda convençut que el missatge no ha estat alterat perquè està signat. A més, amb aquest procediment, el signatari no pot repudiar més tard el fet d'haver signat el missatge, perquè ningú llevat del signatari no té la clau privada necessària per a produir la signatura.

1.2. Formalització

En aquest subapartat, formalitzem les idees donades als subapartats anteriors. Suposem un usuari A que vol signar un missatge m. Sigui $f_A(\ )$ el xifratge sota la clau pública de A i $f_A^{-1}(\ )$ el xifratge sota la clau privada de A. Podem formalitzar un protocol de signatura i el protocol de verificació associat.

Protocol de signatura

Per a signar m, A calcula $s = f_A^{-1}(m)$, és a dir, xifra el missatge amb la seva clau privada. s és la signatura del missatge m.

Protocol de verificació

Per a verificar la signatura del missatge m, qualsevol usuari B pot calcular $f_A(s)$ i veure si es compleix $f_A(s) \stackrel{?}{=} m$. En cas afirmatiu, s'accepta la signatura s com a vàlida; en cas negatiu, la signatura no és vàlida.

Més endavant veurem que normalment no se signa directament el missatge m, sinó que se'n signa un resum de longitud fixa.

Vegeu la signatura de resums a l'apartat 3 d'aquest mòdul didàctic.

2. Esquemes de signatura digital

En aquest apartat veurem tres dels esquemes de signatura digital més emprats:

1) El primer esquema es basa en el criptosistema RSA i, per tant, la seva infalsificabilitat està relacionada amb la dificultat que suposa el problema de la factorització.

2) El segon esquema es basa en el criptosistema d'ElGamal i, per tant, la seva infalsificabilitat està relacionada amb la dificultat que suposa el problema del logaritme discret.

3) El tercer esquema és l'algorisme estàndard de signatura homologat pel govern dels EUA, que, de fet, és molt semblant a la signatura d'ElGamal (amb algunes millores).

2.1. Signatura RSA

Suposem que un usuari A té una clau pública RSA (n_A,e_A) i una clau privada d_A. Per a signar digitalment el missatge m amb l'RSA, l'usuari A calcula la signatura $s = m^{d_A} \bmod n_A$. Tot seguit A publica el missatge signat difonent la parella (m,s).

Vegeu el criptosistema RSA al subapartat 4.T del mòdul didàctic "Xifres de clau pública" d'aquesta assignatura.

Per a verificar la signatura s, qualsevol usuari B pot mirar si es compleix $s^{e_A} \bmod n_A \stackrel{?}{=} m$. En cas afirmatiu, la signatura és vàlida; en cas negatiu, la signatura no ho és.

Remarquem que el protocol de verificació suposa implícitament que B disposa de l'autèntica clau pública de l'usuari A. Tal com s'ha comentat en el subapartat anterior, una manera d'assegurar-se que la clau pública de A és correcta és recuperar-la a partir d'un certificat emès per un gestor de directori de claus públiques o, generalment, per una autoritat de certificació.

Utilització de la signatura RSA

Suposem que la clau pública de l'usuari A és $(n_A,e_A) = (34.121,15.775)$ i que la seva clau privada és $d_A = 26.623$. Si YES és el missatge que ha de signar A, el missatge codificat en base 26 és $m = 16.346$. Llavors A calcula la signatura:

$$s = m^{d_A} \bmod n_A = 16.346^{26.623} \bmod 34.121 = 20.904.$$

Descodificant el valor de s en base 26, obtenim que la signatura és BEYA. Per tant, allò que s'envia és la parella (YES,BEYA).

Qualsevol usuari B que vegi aquesta parella, pot verificar la signatura del missatge. Per fer-ho, codifica en base 26 i obté $m = 16.346$, $s = 20.904$. Llavors, B recupera la clau pública de A d'un directori de claus públiques. Finalment verifica que es compleixi l'expressió següent:

$$s^{e_A} \bmod n_A = 20.904^{15.775} \bmod 34.121 = m.$$

2.2. Signatura d'ElGamal

Vegeu el criptosistema d'ElGamal al subapartat 4.2 del mòdul didàctic "Xifres de clau pública" d'aquesta assignatura.

Suposem que el signatari A ha fet una generació de claus d'ElGamal dins un grup G usant un element $\alpha \in G$. Suposem que a és la clau privada de A, que $\alpha^a \in G$ és la seva clau pública i n és l'ordre del grup G. Si A vol signar un missatge $m \in G$, llavors fa el procés següent:

1) A genera un nombre aleatori h tal que $\text{mcd}(h,\phi(n)) = 1$.

2) A calcula $r = \alpha^h$ dins de G.

3) A troba el valor de s resolent la congruència:

$$m \equiv ar + hs \bmod \phi(n). \qquad (2.1)$$

La signatura digital per al missatge m és la parella (r,s).

Vegeu el càlcul d'inversos al subapartat 1.2 del mòdul didàctic "Xifres de clau pública" d'aquesta assignatura.

Noteu que la solució de l'equació 2.1 és $s \equiv (m - ar)h^{-1} \bmod \phi(n)$ i existeix perquè la condició $\text{mcd}(h,\phi(n)) = 1$ assegura que h té invers multiplicatiu mòdul $\phi(n)$.

Per a verificar la signatura del missatge m, qualsevol usuari B pot fer el procés següent:

1) B calcula $r^s = (\alpha^h)^s$ dins de G. També calcula $(\alpha^a)^r$ dins de G.

2) B verifica si es compleix: $(\alpha^a)^r\,(\alpha^h)^s \stackrel{?}{=} \alpha^m$, on totes les operacions són dins de G.

Utilització de la signatura d'ElGamal

Suposem que hem triat el grup $G = \mathbb{Z}^*_{15.485.863}$ i $\alpha = 7$ (de fet, les potències de 7 generen tot G). Com que $p = 15.485.863$ és primer, l'ordre de G és $\phi(p) = p - 1 = 15.485.862$.

L'usuari A té com a clau privada $a = 28.236$ i ha calculat la seva clau pública:

$$\alpha^a = 7^{28.236} \bmod 15.485.863 = 12.506.884.$$

Suposem que A vol signar el missatge $m = 128.688$. Per a fer-ho seguirà els passos següents:

1) A tria el número aleatori $h = 90.725$, que és coprimer amb l'ordre del grup, és a dir, $\text{mcd}(90.725,15.485.862) = 1$.

2) Llavors A calcula:

$$r = \alpha^h = 7^{90.725} \bmod 15.485.863 = 7.635.256.$$

3) Tot seguit resol la congruència $m \equiv ar + hs \bmod (p - 1)$, la solució de la qual és:

$$s = (m - ar)h^{-1} \bmod (p - 1) =$$
$$= (128.688 - 28.236 \cdot 7.635.256) \cdot 90.725^{-1} \bmod 15.485.862 =$$
$$= 11.047.464,$$

on l'invers de h s'ha trobat amb l'algorisme d'Euclides estès. La signatura de A per al missatge m és:

$$(r,s) = (7.635.256,11.047.464).$$

Per a comprovar la signatura de A, qualsevol usuari B pot calcular:

$$(\alpha^h)^s = r^s = 7.635.256^{11.047.464} \bmod 15.485.863 = 8.799.713.$$

Després B calcula:

$$(\alpha^a)^r = 12.506.884^{7.635.256} \bmod 15.485.863 = 1.260.686,$$

i també:

$$\alpha^m = 7^{128.688} \bmod 15.485.863 = 5.362.356.$$

Finalment, B verifica que:

$$r^s \cdot (\alpha^a)^r = 8.799.713 \cdot 1.260.686 \bmod 15.485.863 = 5.362.356 = \alpha^m.$$

2.2.1. Velocitat d'ElGamal

Signar un missatge amb l'ElGamal requereix fer una exponenciació, igual que amb l'RSA*. L'avantatge sobre l'RSA és que l'exponent h no depèn del missatge que s'ha de signar i per tant l'exponenciació pot ser precalculada. De fet, el signatari pot guardar en un lloc segur unes quantes exponenciacions precalculades, que farà servir a mesura que hagi de signar missatges.

*** El temps de resoldre la congruència és negligible.**

En canvi, verificar una signatura d'ElGamal és mes lent que verificar una signatura RSA. En efecte, la verificació implica dur a terme tres exponenciacions, enfront d'una sola exponenciació per a l'RSA. Això és un inconvenient greu, ja que se signa una sola vegada, mentre que una signatura és normalment verificada moltes vegades.

Finalment, les signatures d'ElGamal tenen l'inconvenient de ser més llargues que les signatures RSA, perquè consisteixen en parells d'enters grans (r,s), mentre que la signatura RSA és un sol enter gran s.

2.2.2. Seguretat d'ElGamal

Per a falsificar la signatura de A sobre el missatge m, un criptoanalista enemic hauria de resoldre l'equació $\alpha^m = (\alpha^a)^r r^s$ amb les incògnites r i s: per a fer-ho pot plantejar el seu atac de dues maneres diferents:

a) Si fixa *r* i mira de trobar *s*, el criptoanalista ha de resoldre el problema del logaritme discret (PLG).

b) Si fixa *s* i mira de trobar *r*, el criptoanalista s'enfronta a una congruència exponencial mixta, per a la qual no hi ha algorisme polinòmic conegut, és a dir, es troba amb el problema de la signatura d'ElGamal (PSE).

Vegeu el problema del logaritme discret al subapartat 1.4.2 del mòdul "Xifres de clau pública" d'aquesta assignatura.

Per tant, falsificar una signatura d'ElGamal no és equivalent a resoldre el problema del logaritme discret. Ara bé, si se sabés resoldre el PLG, es podria falsificar la signatura d'ElGamal.

2.3. L'estàndard DSS

El 1991, el National Institute of Standards and Technology (NIST) dels EUA va proposar un **estàndard de signatura digital** i va sol·licitar comentaris públics per a l'adopció de l'estàndard proposat. L'objectiu era que les oficines governamentals nord-americanes tinguessin una manera estàndard de signar les comunicacions en cas necessari.

L'algorisme proposat és una variant de la signatura d'ElGamal i corre el rumor que una de les raons de no haver adoptat exactament la signatura d'ElGamal és una qüestió de patents.

Protocol de generació de claus DSS

Per a generar una clau DSS cal que cada usuari triï els elements següents:

- p, nombre primer que compleixi que $2^{511} < p < 2^{512}$.

- q, nombre primer divisor de $p - 1$, que compleixi que $2^{159} < q < 2^{160}$.
- g, generador de l'únic subgrup cíclic de $\mathbb{Z}_p^*$ d'ordre q.
- x, clau privada de l'usuari, amb $0 < x < q$.
- $y = g^x \bmod p$, clau pública de l'usuari.

Per a triar p, q i g es procedeix de la manera següent:

a) Es genera un nombre aleatori i senar q tal que $2^{159} < q < 2^{160}$ i se'n comprova la primeritat. El procés es repeteix fins a obtenir un q primer.

b) Tot seguit es genera el primer p repetint la tria aleatòria d'un enter n tal que verifiqui:

$$\frac{2^{511} - 1}{2q} < n < \frac{2^{512} - 1}{2q},$$

fins que $p = 2nq + 1$ sigui primer.

c) Finalment, es genera un element g d'ordre q del grup $\mathbb{Z}_p^*$ repetint la tria aleatòria d'un enter h amb $1 < h < p - 1$ i calculant $g = h^{(p-1)/q}$ fins que $g \neq 1$.

Protocol de signatura DSS

Si m és el missatge que s'ha de signar i $H : \mathbb{N} \to \mathbb{Z}$ és una funció *hash* unidireccional, aleshores el signatari haurà de fer els passos que exposem tot seguit:

Vegeu les funcions *hash* a l'apartat 3 d'aquest mòdul didàctic.

1) Triar un enter aleatori k específic per a m, amb $0 < k < q$.

2) Calcular $r = (g^k \bmod p) \bmod q$.

3) Trobar s resolent la congruència $H(m) \equiv -xr + ks \bmod q$.

4) Indicar que la signatura de m és el parell (r,s).

Protocol de verificació DSS

Si (r,s) és la signatura DSS que es vol verificar, el receptor haurà de fer els passos següents:

1) Calcular $w = s^{-1} \bmod q$.

2) Calcular $u_1 = H(m)\, w \bmod q$ i $u_2 = rw \bmod q$.

3) Verificar si es compleix la relació:

$$r = (g^{u_1} y^{u_2} \bmod p) \bmod q.$$

2.3.1. Velocitat del DSS

Com que el DSS treballa amb un subgrup de $\mathbb{Z}_p^*$ d'ordre q, les signatures digitals són més curtes que amb la signatura d'ElGamal (són dos enters de la mida de q, en comptes de dos enters de la mida de p). A part de representar una reducció d'emmagatzematge, disposar de signatures més curtes suposa que la verificació és més ràpida que amb l'ElGamal. En efecte, les exponenciacions que es calculen per a verificar una signatura tenen exponents de la mida de q (160 bits).

Tot i els avantatges anteriors, la signatura DSS continua requerint més càlcul que la signatura RSA, particularment pel que fa a la verificació. També s'assenyala com a inconvenient la necessitat de generar un enter aleatori k de 160 bits per a signar cada missatge.

A més, a diferència d'RSA i d'ElGamal, la signatura DSS no té la doble utilitat de ser alhora un criptosistema de clau pública per a encriptar missatges.

2.3.2. Seguretat del DSS

La seguretat del DSS és semblant a la de la signatura d'ElGamal, amb l'única diferència que els logaritmes discrets són sobre el subgrup cíclic d'ordre q de $\mathbb{Z}_p^*$ generat per g.

Fins ara, el millor algorisme per a trobar logaritmes discrets al subgrup cíclic generat per g requereix calcular logaritmes a $\mathbb{Z}_p^*$. Per tant, això fa que la seguretat del DSS sigui de moment la mateixa de la signatura d'ElGamal. Ara bé, no es pot excloure la possibilitat que en el futur es trobin algorismes per a calcular logaritmes en un subgrup cíclic d'un grup finit donat. Una de les raons de la no-disponibilitat de tals algorismes és probablement que abans de la introducció del DSS el problema no tenia gaire interès.

S'ha criticat també el fet que la longitud de p i de q quedi tan fixada a l'estàndard. Hi ha hagut veus en el sentit que les longituds són insuficients per a proporcionar una seguretat adequada. Possiblement, futures versions de l'estàndard permetran més flexibilitat en la tria de p i de q.

3. Funcions *hash*

Les signatures digitals en ús actualment són lentes (amb relació a criptosistemes de clau compartida, com el DES). Per tant, és desitjable signar només un resum del missatge en comptes del missatge sencer. Les funcions *hash* serveixen per a crear resums.

Una **funció *hash*** és una funció que fa correspondre a un missatge m de mida variable una representació $H(m)$ de mida fixa. Típicament, $H(m)$ té de 64 a 160 bits i s'anomena el *valor hash del missatge*.

La projecció...

... dels enters $\mathbb{Z}$ en $\mathbb{Z}_p$ per p fixat és una funció *hash*. En efecte, a un enter de longitud arbitrària se li fa correspondre un enter de longitud com a màxim $|p|$ bits, on $|p|$ és la longitud de p en bits.

Si una funció *hash* ha de ser emprada per a aplicacions criptogràfiques, no n'hi ha prou que resumeixi la seva entrada de manera aparentment aleatòria. Cal que sigui també unidireccional.

Una **funció *hash* unidireccional** és una funció *hash* H tal que, per a qualsevol missatge m' del recorregut de H, és difícil de trobar m tal que $m' = H(m)$.

Així, doncs, una funció *hash* unidireccional és una funció *hash* que és també una funció unidireccional.

La combinació de funcions *hash* unidireccionals amb signatures digitals dóna peu al protocol de signatura amb *hash* i al corresponent protocol de verificació.

Protocol de signatura amb *hash*

Si A vol signar el missatge m amb una funció *hash* fa dues accions:

1) Calcula $H(m)$, on H és una funció *hash* unidireccional públicament coneguda.

2) Signa $H(m)$ amb la seva clau privada i obté la signatura s. s és considerada la signatura de m.

Protocol de verificació amb *hash*

Per a verificar la signatura s del missatge m, qualsevol usuari B pot fer:

1) Primer calcula $H(m)$.

2) Tot seguit, verifica si s és una signatura vàlida per a $H(m)$.

Noteu que és fonamental que H sigui unidireccional, ja que es considera equivalent signar $H(m)$ i signar m. Mal aniríem si, donat el valor *hash* $H(m_1)$ d'un missatge m_1 signat per A, fos fàcil de trobar un altre missatge $m_2 \neq m_1$ tal que $H(m_1) = H(m_2)$. En aquest cas, qualsevol podria pretendre que A ha signat m_2.

Lectura complementària

L'estudiant interessat pot trobar descripcions de les funcions *hash* de la família MD i també de la funció SHA-1 en el llibre següent:

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

Les funcions *hash* més utilitzades són els *message digest* de Rivest, coneguts com MD2, MD4 i MD5. Aquestes funcions produeixen resums de 128 bits i l'únic atac que es coneix contra aquestes és el de la cerca exhaustiva. Recentment, el NIST ha proposat una funció *hash* estàndard, coneguda com a *Secure Hash Algorithm* (SHA-1).

Els algorismes que implementen aquestes funcions tenen una aparença semblant als dels criptosistemes de tipus DES* amb la diferència que no depenen de cap clau. Llur velocitat en programari és superior a la dels criptosistemes de tipus DES. Finalment, remarquem que les restriccions dels EUA a l'exportació de criptografia no afecten les funcions *hash*, per la qual cosa és extremament fàcil de trobar-ne implementacions a Internet.

*** Iteracions, operacions a nivell de bit fàcils d'implementar en maquinari, etc.**

Resum

Amb un **esquema de signatura digital** cada usuari pot signar missatges en format electrònic de manera que les signatures puguin ser verificades més tard per qualsevol persona.

Els dos algorismes de signatura digital més emprats avui dia s'obtenen dels dos criptosistemes de clau pública més utilitzats: l'RSA i l'ElGamal. La infalsificabilitat de la signatura RSA està relacionada amb el problema de la factorització i la infalsificabilitat de la signatura d'ElGamal està relacionada amb el problema del logaritme discret.

Hi ha un **estàndard de signatura digital**, el DSS, proposat pel govern dels EUA, que en realitat és una variant de la signatura d'ElGamal.

Com els criptosistemes de clau pública, els algorismes de signatura digital són lents comparats amb els criptosistemes de clau compartida. Per aquesta raó, es tendeix a signar resums dels missatges en comptes dels missatges sencers. Les **funcions *hash*** permeten obtenir un resum de longitud fixa d'un missatge de longitud arbitrària, de tal manera que sigui difícil per a un criptoanalista enemic de trobar un missatge diferent amb el mateix resum.

Activitats

1. Cerqueu a Internet implementacions de les funcions *hash* de la família MD i també de la funció estàndard SHS.

2. Escriviu programes que implementin les signatures RSA i d'ElGamal amb nombres petits. Podeu aprofitar implementacions dels criptosistemes RSA i ElGamal.

Exercicis d'autoavaluació

1. Demostreu que la signatura RSA és existencialment falsificable, és a dir, que donades signatures legítimes per a dos missatges m_1 i m_2, qualsevol criptoanalista pot calcular la signatura per al missatge producte $m_1 \cdot m_2$.

2. A la signatura d'ElGamal, preneu $p = 23$, $\alpha = 5$ (α és primitiu a $\mathbb{Z}_{23}^*$). Detalleu com un usuari A generaria el seu parell de claus pública i privada. Detalleu pas per pas com signaria l'usuari A la vostra edat en anys mòdul 23.

3. Seguiu els passos indicats per a generar els paràmetres del DSS suposant que $p = 23$:
a) Trobeu un factor primer q de $p - 1$ tal que $q > 10$.
b) Comproveu que $x^{q-1} = 1 \bmod q$ per a tot $1 \leq x \leq q - 1$.
c) Trobeu un nombre $h < p - 1$ tal que $g = h^{(p-1)/q} \bmod p > 1$.
d) Detalleu els elements del subgrup multiplicatiu generat per g.

4. Suposeu que A fa servir un esquema de signatura d'ElGamal i que signa dos missatges m_1 i m_2 amb signatures (r,s_1) i (r,s_2), fent servir el mateix r per a totes dues signatures. Suposeu també que $\text{mcd}(s_1 - s_2, p - 1) = 1$. Demostreu com es pot calcular eficientment la clau secreta a del signatari i com a conseqüència es pot trencar l'esquema de signatura.

Solucionari

Exercicis d'autoavaluació

Si heu fet les activitats del mòdul "Xifres de clau pública" podeu aprofitar les implementacions que hàgiu fet per a dur a terme l'activitat 2.

1. Si el signatari té clau pública (e,n) i clau privada d, la signatura RSA de m_1 és $s(m_1) = m_1^d \bmod n$ i la signatura RSA de m_2 és $s(m_2) = m_2^d \bmod n$. La signatura del missatge producte pot ser calculada a partir de les signatures anteriors de la manera següent:

$$s(m_1m_2) = (m_1m_2)^d \bmod n = ((m_1)^d \bmod n)((m_2)^d \bmod n) \bmod n =$$
$$= (s(m_1)s(m_2)) \bmod n.$$

2. Tenim $p = 23$ i $\alpha = 5$. Suposem que l'usuari A tria la seva clau privada $a = 3$. Llavors calcula la seva clau pública com $\alpha^a = 5^3 \bmod 23 = 10$. Suposem que la nostra edat en anys mòdul 23 és $m = 11$. Llavors, per a signar m, A tria aleatòriament h, tal que $\text{mcd}(h,\phi(23)) = \text{mcd}(h,22) = 1$. Per exemple, $h = 5$. Llavors, A calcula $r = \alpha^h = 5^5 \bmod 23 = 20$. Finalment, A troba el valor de s com:

$$s = (m - ar)h^{-1} \bmod 22 = (11 - 3 \cdot 20)\ 9 \bmod 22 = 21.$$

La signatura del missatge $m = 11$ és $(r,s) = (20,21)$.
Per a verificar la signatura, es calcula:
- $r^s = 20^{21} \bmod 23 = 15$,
- $(\alpha^a)^r = 10^{20} \bmod 23 = 3$,
- $\alpha^m = 5^{11} \bmod 23 = 22$,

i es comprova que:

$$r^s\ (\alpha^a)^r = (15 \cdot 3) \bmod 23 = 22 = \alpha^m.$$

3.
a) Podem prendre $q = 11$.
b) No cal comprovar que $x^{q-1} = 1 \bmod q$ per a tot $1 \leq x \leq q - 1$ perquè això se segueix del teorema petit de Fermat.
c) Prenem $h = 3 < 22$ i comprovem que:

$$g = 3^{22/11} \bmod 23 = 9 \neq 1.$$

d) Els elements del subgrup multiplicatiu generat per g són: {9, 12, 16, 6, 8, 3, 4, 13, 2, 18, 1}. El subgrup en qüestió té ordre 11, que és divisor de l'ordre del grup multiplicatiu sencer, que és 22.

4. Si r és igual a totes dues signatures, llavors $h = \log_\alpha r$ també ho és. Llavors, es compleixen les dues relacions següents:
- $m_1 \equiv ar + hs_1 \bmod (p - 1)$,
- $m_2 \equiv ar + hs_2 \bmod (p - 1)$.

Restant les dues equacions anteriors obtenim:

$$m_1 - m_2 \equiv h(s_1 - s_2) \bmod (p - 1).$$

Com que se suposa que $\text{mcd}(s_1 - s_2, p - 1)$, en podem trobar l'invers mòdul $p - 1$ i així obtenir:

$$h = (m_1 - m_2)(s_1 - s_2)^{-1} \bmod (p - 1).$$

Un cop trobat h, podem aïllar a de qualsevol de les dues equacions.

Hem trobat la clau privada del signatari i, per tant, hem trencat l'esquema. Per això és molt important que el valor h (i per tant r) sigui aleatori i diferent en cada signatura d'ElGamal.

Glossari

Autoritat de certificació: tercera part fiable que emet certificats, és a dir, documents electrònics que acrediten l'autenticitat de les claus públiques dels usuaris pertanyents al domini de l'autoritat.

Certificat: document electrònic consistent en una còpia de la clau pública d'un usuari signada pel gestor del directori de claus públiques o per una tercera part fiable.

Falsificació: atac criptoanalític que pretén obtenir la signatura d'un cert missatge sense la intervenció del signatari.

Funció *hash*: funció que dóna com a sortida un resum de longitud fixa a partir d'una entrada consistent en un missatge arbitràriament llarg.

Funció *hash* unidireccional: funció *hash* que a més és unidireccional; és a dir, la sortida és fàcil de calcular a partir de l'entrada, però l'entrada és difícil de calcular a partir de la sortida.

RSA: criptosistema de clau pública, basat en el problema de la factorització, publicat per Rivest, Shamir i Adleman l'any 1978.

Signatura digital: procediment per a signar documents en format electrònic que consisteix en un algorisme de signatura privat del signatari i un algorisme públic per a la verificació de la signatura.

Signatura DSS: signatura digital estàndard del govern dels EUA, molt semblant a la signatura d'ElGamal.

Signatura d'ElGamal: signatura digital basada en el criptosistema de clau pública d'ElGamal.

Signatura RSA: signatura digital basada en el criptosistema de clau pública RSA.

Verificació: comprovació que una signatura és vàlida, és a dir, que ha estat efectuada pel pretès signatari. Ha de ser possible per a tothom, és a dir, no ha de requerir coneixement de paràmetres secrets.

Bibliografia

Fuster, A.; De la Guía, D.; Hernández, L.; Montoya, F.; Muñoz, J. (1997). *Técnicas criptográficas de protección de datos*. Madrid: Ra-ma.

Goldwasser, S.; Bellare, M. (1996). *Lecture Notes on Cryptography* (Manuscrit).

Rifà, J. (1995). *Seguretat Computacional*. Bellaterra: 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.

Infraestructura de clau pública

PKI

Helena Rifà Pous

P03/05024/02265

UOC
www.uoc.edu

Índex

Introducció

En aquest mòdul estudiarem de quina manera és possible implantar sistemes criptogràfics basats en clau pública que permetin l'ús generalitzat de serveis d'integritat, confidencialitat, autenticitat i no-repudi. En primer lloc, veurem quina és la problemàtica associada a la implementació de la criptografia asimètrica, i presentarem la solució de la infraestructura de clau pública (o PKI, *public key infrastructure*).

La tecnologia d'infraestructura de clau pública està en fase de desenvolupament. Hi participen moltes institucions i grups d'usuaris, i el procés d'estandardització no és fàcil ni ràpid. En aquest mòdul no pretenem veure totes les possibilitats d'una infraestructura de clau pública, sinó només donar una idea general de la tecnologia. N'estudiarem els components essencials, en especial el certificat digital X509, i veurem com s'interrelacionen per a crear els diferents models de PKI. D'aquesta manera, obtindrem una visió global dels passos que calen per a poder oferir serveis de criptografia de clau pública a un conjunt d'usuaris.

Un cop coneguts els principis d'una infraestructura de clau pública, veurem de quina manera és possible generar missatges segurs entre diferents aplicacions d'acord amb els estàndards actuals.

Finalment, veurem alguns dels protocols i serveis de seguretat actuals més coneguts, que es fonamenten en infraestructures de clau pública per al seu correcte funcionament.

Objectius

Els materials didàctics d'aquest mòdul us permetran assolir els objectius següents:

1. Entendre els principis bàsics d'una infraestructura de clau pública: els problemes que resol, els seus components i les seves funcions.

2. Assimilar les característiques de les estructures bàsiques que s'utilitzen dins un entorn d'infraestructura de clau pública, quina mena d'informació contenen i per a què serveixen.

3. Comprendre el format de les estructures de dades PKCS més importants emprades en criptografia de clau pública.

4. Conèixer el funcionament de diversos protocols de seguretat que es basen en la tecnologia d'infraestructura de clau pública.

1. Conceptes bàsics

En aquest apartat veurem quines són les motivacions del naixement de les infraestructures de clau pública.

1.1. Problemàtica de la distribució de claus

La criptografia de clau pública permet l'intercanvi de missatges confidencials i íntegres de manera àgil, sempre que disposem de la clau pública de l'interlocutor amb qui ens comuniquem. El problema és com obtenir aquesta clau pública i poder estar segurs que pertany a qui ens pensem.

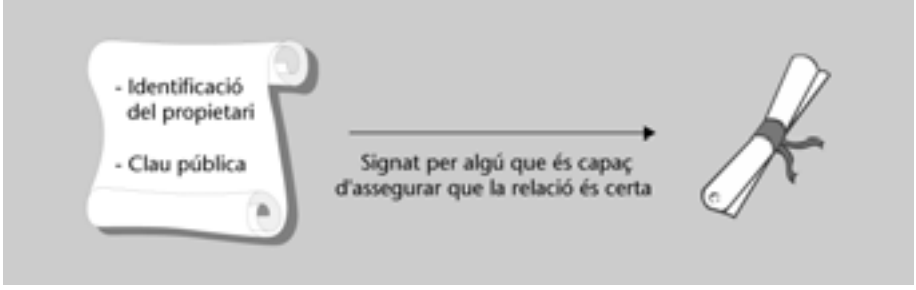
Les claus públiques són justament això: públiques; i un intrús prou hàbil podria tenir accés al directori on estan ubicades i substituir la clau pública d'algun usuari per la seva pròpia. D'aquesta manera, els interlocutors que utilitzessin la clau pensarien que s'estan comunicant amb aquell usuari quan, en realitat, ho farien amb l'intrús. És el que es coneix com *atac de l'home a mig camí*.

Vegeu els criptosistemes de clau pública en el mòdul "Xifres de clau pública" d'aquesta assignatura.

Diffie i Hellman (1976) van pensar que el maldecap de la distribució de claus es resoldria amb un directori segur en línia en el qual s'establís de manera unívoca el lligam entre un nom distintiu d'usuari i una clau pública. El directori públic hauria de signar totes les seves transaccions de manera que ningú no en pogués suplantar la identitat. El problema raïa en el baix rendiment que el sistema oferia en poblacions d'usuaris mitjanes o grans. L. Kohnfelder (1978) es va basar en la idea d'una autoritat central de confiança introduïda per Diffie i Hellman i va proposar crear uns registres de dades signades –els certificats– que permetrien que la distribució de claus es fes des de directoris públics que no requerissin confiança.

Un **certificat digital** és una estructura de dades que conté informació del propietari de les claus criptogràfiques, la clau pública en si i una signatura digital dels dos camps anteriors que hi dóna validesa.

La signatura, realitzada per un usuari o entitat externa lleial, assegura la integritat contra una possible modificació no desitjada de les dades. La confiança en el signatari s'estén al subjecte del certificat.



D'aleshores ençà, els certificats han esdevingut el principal mitjà de distribució de les claus públiques dels sistemes de clau asimètrica. De tota manera, l'ús de certificats no resol totalment el problema de la distribució de claus, sinó que la trasllada a un nivell superior. Qui ha de signar el certificat? Quins mecanismes són necessaris perquè dos usuaris que no es coneixen puguin assegurar la seva identitat en una comunicació virtual? En definitiva, quin és el model de confiança?

En l'apartat 3 d'aquest mòdul veurem els models de confiança més comuns i què ofereix cadascun.

1.2. Propòsits d'una infraestructura de clau pública (PKI)

L'objectiu d'una infraestructura de clau pública és la gestió eficient i confiable de les claus criptogràfiques i els certificats perquè es puguin utilitzar per a funcions d'autenticació, integritat, no-repudi i confidencialitat. La infraestructura de clau pública crea un marc segur d'intercanvi de dades en un entorn típicament insegur com Internet.

Una **infraestructura de clau pública*** (**PKI**) és el conjunt de maquinari, programari, persones, polítiques i procediments necessaris per a crear i gestionar certificats digitals basats en criptografia de clau pública.

*** En anglès,** ***public key infrastructure.***

El certificat és l'element central de la infraestructura de clau pública al voltant del qual es crea aquesta infraestructura de suport que abraça serveis com el registre d'usuaris, l'emissió de certificats, la seva distribució des de directoris públics, la seva renovació i revocació, la recuperació de claus, etc.

1.3. Estàndards d'infraestructura de clau pública

Per a poder tenir aplicacions interoperables, cal establir uns estàndards en matèria d'infraestructura de clau pública. El fet que diversos fabricants es posin d'acord en la sintaxi i l'estructura de les dades atreu tant desenvolupadors com usuaris finals, i fa que el mercat creixi i millori.

El mercat de la seguretat es troba a mig camí de la transició entre solucions propietàries i estàndards oberts. Alguns fabricants no volen donar a conèixer les seves solucions perquè basen la seguretat en l'obscuritat i el secret; aquesta no és una bona política, ja que no compleix la suposició de Kerckhoff.

Vegeu la suposició de Kerckhoff en el subapartat 4.1 del mòdul "Fonaments de criptografia" d'aquesta assignatura.

Els principals organismes estandarditzadors en matèria d'infraestructura de clau pública a Internet són els següents:

a) Internet Engineering Task Force (**IETF**). La IETF, com a fòrum global per al desenvolupament d'estàndards de xarxes, és el lloc preferent per al desenvolupament d'especificacions i estàndards que ofereixin suport al creixement

de les aplicacions de serveis de confiança. El treball fet en el camp de les infraestructures de clau pública dins del grup anomenat PKIX és un exemple d'aquesta activitat, ja que és el model més àmpliament acceptat i en el qual ens basarem en aquest mòdul. El mitjà de publicació dels estàndards i la informació relacionada és la sèrie de documents anomenats ***requests for comments*** (**RFC**), dels quals n'hi ha dues categories principals:

- Els ***standards track*** són aquells que segueixen el progrés de l'RFC des de la proposta o esborrany fins a l'estàndard.

- Els ***non-standards track*** no tenen l'aprovació i el consens de tota la comunitat Internet i es classifiquen en experimentals, informatius i històrics.

b) International Standards Organisation (**ISO**). L'ISO es focalitza en la ratificació d'estàndards generats tant per la IETF com per altres organismes nacionals per tal d'apropar més determinades normes a la indústria.

c) European Telecommunications Standards Institute (**ETSI**). L'ETSI és un organisme europeu format per empreses actives en la indústria de telecomunicacions, principalment com a proveïdors de serveis i venedors.

d) European Committee for Standardisation (**CEN**). El CEN comprèn els organismes d'estandardització nacionals d'Europa.

Per altra banda, els **RSA Laboratories** han fet importants aportacions en els formats de dades basades en infraestructura de clau pública i han aconseguit que moltes de les seves especificacions s'hagin convertit en estàndards *de facto*, com els PKCS.

Veurem alguns exemples de PKCS en l'apartat 6 d'aquest mòdul.

2. Components d'una infraestructura de clau pública

En aquest apartat veurem els components d'una infraestructura de clau pública típica. Malgrat que els components essencials són l'autoritat de certificació, els subscriptors i els repositoris, n'hem afegit d'altres com l'autoritat de registre, l'autoritat de validació o l'autoritat de segellat de temps, que donen una visió completa de les funcions que pot arribar a proporcionar una infraestructura de clau pública. Altres components opcionals que no detallarem són el repositori de claus (que permet donar serveis de recuperació de claus criptogràfiques) o l'autoritat documental.

2.1. Autoritat de certificació

L'**autoritat de certificació*** (CA) és la responsable d'emetre i revocar certificats. És l'entitat de confiança que dóna legitimitat a la relació d'una clau pública amb la identitat d'un usuari o servei.

* En anglès, *certification authority*.

Polítiques de certificat i pràctiques de certificació

La relació legal i tècnica entre una autoritat de certificació i els seus subscriptors i usuaris està regida per les polítiques de certificats (*certificate policy*, CP) i la declaració de pràctiques de certificació (*certification practice statement*, CPS). Una **política de certificat** exposa la garantia de seguretat que es pot donar al certificat i els usos per als quals és adequat. És un concepte general, no particular d'una organització. Serveix com a vehicle per a establir les bases de la interoperabilitat. Les **pràctiques de certificació** estipulen la manera com una autoritat de certificació en particular estableix aquesta garantia de seguretat. Normalment, la CPS se sol presentar en forma de document a l'abast del públic.

Una infraestructura de clau pública pot tenir una o més autoritats de certificació. La creació d'una autoritat de certificació comença amb la generació del parell de claus (pública i privada) que s'utilitzaran per a signar i validar els certificats digitals que emeti l'autoritat de certificació. Les claus han de ser prou fortes perquè la probabilitat que un atacant les trenqui sigui extremadament durant el temps de vida dels certificats que s'hi signaran. Això dependrà de la combinació de la longitud de la clau i la qualitat de l'algorisme de generació de claus.

La relació entre les diferents autoritats de certificació es descriu en l'apartat 3 d'aquest mòdul.

Un cop generat el parell de claus, la clau pública s'ha de distribuir d'una manera segura a totes les entitats de confiança potencials. Aquesta distribució es pot fer tant per mitjà d'un certificat digital emès per una autoritat de certificació en la qual els usuaris ja confien, o via un certificat generat per la pròpia autoritat de certificació que s'acabi de crear (aquest tipus de certificats reben el nom de *certificats autosignats*). Per a garantir la seguretat de la transmissió,

aquests certificats digitals de les autoritats de certificació s'hauran de distribuir des d'un canal fora de banda (caldrà passar el certificat físicament, per correu, incorporat en el programari, etc.).

L'autoritat de certificació també ha de protegir la seva clau privada d'un ús no autoritzat. La seguretat d'una infraestructura de clau pública depèn de les bones pràctiques que s'utilitzin per a crear i gestionar tota la infraestructura de confiança. És important que el control d'accés a les claus de l'autoritat de certificació sigui molt estricte; per això, generalment les claus es guarden en una targeta o maquinari criptogràfic i es fa ús de la protecció física que representa tenir l'autoritat de certificació aïllada i sense cap connexió de xarxa.

Els dispositius més emprat

Els dispositius criptogràfics més utilitzats són les targetes intel·ligents (*smartcards*), HSM (*hardware security modules*) i *tokens* USB.

El nucli bàsic d'una autoritat de certificació és el parell de claus criptogràfiques. Però a més de les claus, les autoritats de certificació acostumen a estar formades per diversos subcomponents o serveis, com el repositori de certificats, el repositori de claus, serveis segurs de *logs*, etc.

2.2. Autoritat de registre

L'**autoritat de registre*** (**RA**) és l'encarregada de verificar el lligam entre les claus públiques i la identitat dels seus titulars.

*** En anglès, *registration authority*.**

A l'Estat espanyol

A l'Estat espanyol, les autoritats de registre de la Fàbrica Nacional de Moneda i Timbre són les delegacions i administracions que l'Agència Tributària té en cada localitat, i les oficines consulars.

Les autoritats de registre són un component opcional d'una infraestructura de clau pública que s'utilitza per a descarregar l'autoritat de certificació de moltes de les funcions administratives. En particular, són especialment útils en organitzacions grans i geogràficament disperses.

Poca seguretat

Quan el model de d'infraestructura de clau pública no requereix un nivell de seguretat gaire elevat, l'autoritat de registre es pot automatitzar de manera que la validació sigui més laxa i s'efectuï més mecànicament.

2.3. Subscriptors i entitats finals

Els **subscriptors** i les **entitats finals** són aquells que posseeixen un parell de claus (pública i privada) i un certificat associat a la clau pública.

Amb aquest parell de claus podran efectuar signatures digitals i xifrar i desxifrar documents. Una entitat final representa un organisme, mentre que un subscriptor és una persona.

2.4. Usuaris

Els **usuaris** són els agents que validen signatures digitals i la seva ruta de certificació a partir de claus públiques emeses per autoritats de certificació de confiança. També poden xifrar documents per a subscriptors i entitats finals.

Observeu que, a diferència dels subscriptors i les entitats finals d'una infraestructura de clau pública, els usuaris no tenen per què tenir cap parell de claus ni cap certificat. Evidentment, subscriptors i entitats finals són, en particular, usuaris.

2.5. Repositoris

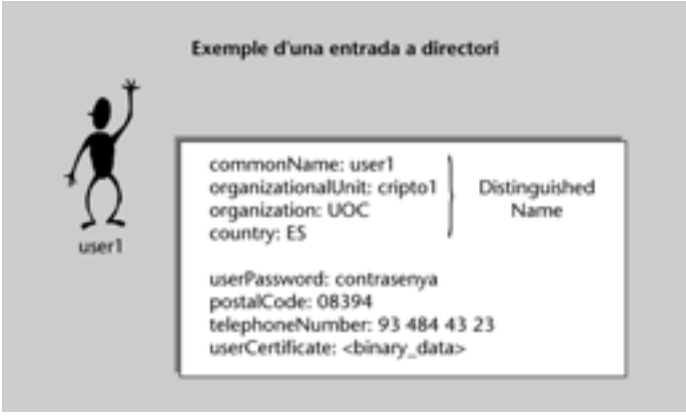
Els **repositoris** són les estructures encarregades d'emmagatzemar la informació relativa a la infraestructura de clau pública. Els dos repositoris més importants en una infraestructura de clau pública són el repositori de certificats i el repositori de llistes de revocació de certificats.

Una **llista de revocació de certificats*** (**CRL**) inclou tots aquells certificats que per diversos motius són invàlids abans de la data de caducitat establerta en el mateix certificat.

*** En anglès,** ***certification revocation list.***

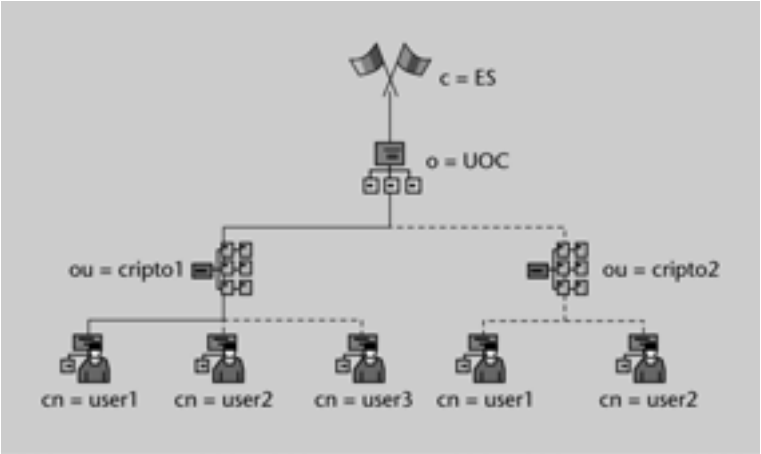
El tipus de repositoris més utilitzats en infraestructura de clau pública són els directoris. Un **directori** és una base de dades especialitzada en la qual s'emmagatzema informació tipificada i organitzada sobre objectes. Està optimitzat per als accessos de lectura, les navegacions i les grans cerques, i el seu objectiu és donar respostes ràpides a un alt volum de peticions.

L'X.500 és un directori basat en l'arquitectura OSI i dissenyat sota els auspicis de l'ISI i altres organitzacions d'estandardització internacionals. En el model X.500, la informació està organitzada de manera jeràrquica (estructura típica d'arbre) i cada node conté objectes d'una classe determinada amb diferents atributs que són els que realment contenen la informació.



Les entrades al directori estan disposades en un arbre i utilitzen el subconjunt d'atributs anomenat **nom distingit*** (**DN**). La situació de l'entrada anterior en l'arbre X.500 seria:

*** En anglès,** ***distinguished name.***



Per a accedir als directoris X.500 es definí el **protocol d'accés al directori*** (**DAP**). Però el DAP era un protocol del llegat ISO i tenia el defecte que no era compatible amb el protocol d'Internet TCP/IP. La IETF (1995) va afegir a la seva línia de treball el desenvolupament d'un protocol que utilitzés un model de dades X.500, però que funcionés sobre TCP/IP i que fos més fàcil d'implementar i de posar en marxa. Es va anomenar LDAP (*lightweight directory access protocol*) i el desembre de 1997 en van treure la versió 2 com a proposta d'estàndard Internet.

*** En anglès,** ***directory access protocol.***

LDAP

L'LDAP va ser originàriament desenvolupat a la Universitat de Michigan.

2.6. Autoritat de validació

L'**autoritat de validació*** (**VA**) és l'encarregada de comprovar la validesa dels certificats digitals.

*** En anglès,** ***validation authority.***

Aquesta autoritat pot ser la pròpia autoritat de certificació o una entitat externa.

Els protocols de validació de certificats són mecanismes que proporcionen informació sobre l'estat actual del certificat (revocat, suspès, vàlid o estat desconegut) o relativa a la cadena de certificació necessària per a validar l'autenticitat del certificat.

Una de les principals deficiències de les llistes de revocació de certificats és que la periodicitat amb què es publiquen les renovacions no està sota el control de les aplicacions que han de validar els certificats. Idealment, el coneixement sobre si un certificat està revocat o no, hauria d'obtenir-se en el mateix moment en què es vol utilitzar. Hi ha diversos protocols de verificació en línia per a obtenir respostes instantànies sobre l'estat dels certificats implicats.

A continuació analitzarem dos protocols per a validar certificats: l'OCSP (*online certificate status protocol*) i l'SCVP (*simple certificate validation protocol*).

2.6.1. *Online certificate status protocol* (OCSP)

L'**online certificate status protocol** (**OCSP**) és un protocol client/servidor que valida l'estat dels certificats sol·licitats i retorna una resposta signada digitalment.

El servidor, implementat en l'autoritat de validació, connecta amb les autoritats de certificació apropiades o per algun altre mecanisme de confiança, i obté les dades de l'estat dels certificats a partir de llistes de revocació de certificats. L'autoritat de validació es fa responsable de les respostes que envia als seus clients (per això les signa), i els clients confien en l'autoritat de validació i n'accepten la sentència.

L'OCSP ofereix dues millores respecte a la validació de certificats a partir de llistes de revocació de certificats:

1) Gestió eficient dels riscos: l'autoritat de validació pot respondre a l'usuari amb informació en temps real.

2) Reducció del trànsit generat: els usuaris no reben grans llistes de certificats de les quals només els interessa una entrada, sinó només la informació que necessiten. Per a assegurar la màxima compatibilitat entre xarxes, es fa servir l'HTTP com a protocol de transport de les peticions OCSP.

2.6.2. *Simple certificate validation protocol* (SCVP)

SCVP és una línia actual de treball de PKIX per a desenvolupar un protocol de validació de certificats que s'adapti a les necessitats de dispositius clients que no tenen gaire capacitat de procés ni d'emmagatzematge. A més de les funcions aportades pel protocol OCSP, l'SCVP ofereix serveis relacionats amb la recerca i el processament de cadenes de certificació. Així, quan un client vol verificar una signatura digital, no li cal construir-se tota la ruta de certificació, des del certificat que ha estat utilitzat per a generar la signatura fins a l'autoritat de certificació arrel, i validar l'estat de tots aquests certificats. Simplement pot demanar per l'estat dels certificats associats al certificat de la signatura que vol verificar i el servidor s'encarregarà de crear tota la cadena de certificació i validar-ne cada un dels components.

2.7. Autoritat de segellat de temps

L'**autoritat de segellat de temps*** (**TSA**) és l'encarregada de signar un missatge amb la finalitat de provar que existia abans d'un determinat instant de temps.

*** En anglès,** ***time stamping authority.***

En el document de definició del servei, s'especifica el protocol basat en missatges de petició i resposta que permeten associar marques temporals de confiança als documents.

La necessitat d'una autoritat de segellat de temps és important per a la propietat de no-repudi. Els serveis de no-repudi han de poder establir l'existència d'unes dades abans de determinats moments. El paper de la TSA consisteix a segellar aquestes dades per tal d'establir-ne l'evidència. Alguns exemples en què la TSA té un paper important són:

a) Verificació de la signatura digital d'un document: si el certificat corresponent ha estat revocat, el segell de temps ens permetrà establir si en el moment en què es va efectuar la signatura del document, el certificat encara era vàlid o no.

b) Lliurament de documents abans d'una data límit: el segell de temps ens permet indicar el moment en què es va fer el lliurament de determinada informació.

c) Auditories: el segell de temps permet tenir datades totes les entrades dels històrics.

Perquè el segellat de temps sigui vàlid, l'estructura de dades que el conté ha d'estar protegida criptogràficament i, a més, la marca de temps que porta s'ha d'haver obtingut d'una font de confiança. Això es pot assegurar amb la utilització de proveïdors oficials de valors temporals basats en el temps coordinat universal (UTC, *Universal Time Coordinated*), que garanteixen una alta precisió en les dades temporals que subministren.

Els passos per a posar un segell de temps en un missatge són els següents:

1) Es calcula un resum digital de les dades que es volen segellar.

2) S'envia el resum a una TSA, que hi afegeix la data i l'hora, signa tot el bloc amb la seva clau privada i torna el resultat al client.

3) El client verifica la signatura i comprova que la TSA ha signat el *hash* de les seves dades i hi ha afegit un segell de temps actual.

3. Models de confiança

Hem vist que per a treballar amb criptografia de clau pública és necessari l'ús d'uns certificats que ens assegurin la correspondència entre la clau pública i algun atribut, típicament la identitat de l'entitat o la persona propietària del certificat. Però els mètodes de certificació absoluts són impossibles, ja que un certificat no es pot certificar ell mateix. Per aquesta raó s'han proposat diferents mètodes que pretenen tractar aquesta situació: el model distribuït de xarxa de confiança, el model pla, el model jeràrquic, el model de navegació per llista de confiança, el model de certificats creuats i el model Bridge-CA.

3.1. El model distribuït

El sistema distribuït de xarxa de confiança és el model més senzill d'utilitzar, adequat per a un grup petit d'usuaris que ja tenien tractes abans de la implantació de la infraestructura de clau pública.

En el **model distribuït** cada usuari crea i signa certificats per a la gent que coneix. No s'ha de desenvolupar cap infraestructura central amb una tercera entitat de confiança que doni fe de la identitat dels usuaris.

Un exemple d'aquest sistema és el programari PGP, que correspon a les sigles de *Pretty Good Privacy**. Es tracta d'un projecte iniciat a principi dels anys 90 per Phill Zimmerman. Amb el pas del temps, el PGP s'ha convertit en un dels mecanismes més populars i fiables per a proporcionar serveis criptogràfics en les comunicacions, sobretot per correu electrònic.

*** En anglès, *privadesa prou bona*.**

El programari PGP assumeix que només els usuaris individuals tenen competències per a decidir en qui confien i en quin grau; els mateixos usuaris signen o donen autenticitat als certificats. Si un usuari A es vol comunicar amb un usuari B, s'hauran d'intercanviar les claus públiques per un canal insegur. Aquestes claus estaran empaquetades en un format de certificat autosignat que, a més, podrà ser signat per altres persones. Per tal d'assegurar que les claus públiques pertanyen efectivament a cada usuari, tant A com B calculen el valor del *hash* de les seves claus i se l'intercanvien per un canal insegur diferent al que han fet servir per a intercanviar-se la clau (per exemple, per telèfon o correu regular). Els usuaris calculen el *hash* de la clau del certificat que han rebut i el comparen amb el que els han enviat; si coincideixen, assumeixen que la clau és autèntica.

En aquest model, hem de distingir dos tipus de confiança:

- La que ens permet creure en la validesa d'una clau, és a dir, la que ens permet refiar-nos que aquella clau pertany a aquella persona.

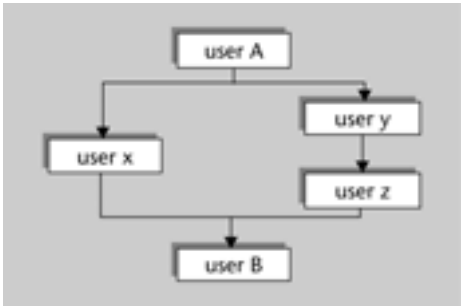
- La que ens permet confiar en una persona com a certificadora de claus (aquesta confiança es basa en una decisió personal, i no en algorismes matemàtics).

No hem d'oblidar que el fet que un certificat sigui autèntic no ens diu res del propietari. Per exemple, hom pot tenir la seguretat que una clau pertany a una persona, però aquesta persona es podria dedicar a signar tots els certificats que li arribessin sense assegurar-se de la seva autenticitat, de manera que en cap cas es mereixeria la nostra confiança.

En el cas del PGP, cada usuari guarda les claus públiques en unes estructures denominades **anells de claus***. Una clau és vàlida si està signada (certificada) pel mateix usuari propietari de l'anell o per prou persones de fiar. Cada clau té associat un nivell de confiança que estableix fins a quin punt es confia en els certificats emesos per aquesta clau. Com més baix és el nivell de confiança, més certificats cal per a validar la clau.

*** En anglès, *key rings*.**

Quan volem validar el certificat d'un usuari qualsevol, fem servir aplicacions de cerca de rutes de certificació per a crear una cadena de confiança adaptada a les nostres necessitats. Donem a l'aplicació l'identificador de clau del nostre certificat i l'identificador del certificat per validar. El servidor de rutes retorna un resultat com el següent:



L'usuari *user x* confirma amb l'emissió d'una signatura que la clau B pertany realment al *user B*. Per altra banda, com que nosaltres (*user A*) hem signat la clau del *user x* podem confirmar que la clau B pertany a l'usuari esperat. A més, hi ha un altre camí de validació que també va des de la nostra clau fins a la del *user B*.

És important que tothom signi claus perquè altres usuaris es puguin beneficiar de les signatures. Totes aquestes signatures formen una espècie de xarxa, i per això es diu que el PGP és un model de xarxa de confiança.

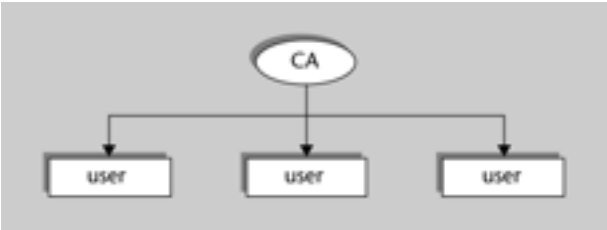
El PGP fa servir dues formes diferents de clau pública: RSA (amb funció de *hash* MD5) i Diffie-Hellman (amb funció de *hash* SHA-1), amb longitud de clau de fins a 2.048 (RSA) o 2.096 (DH). Els algorismes de clau simètrica que pot utilitzar són CAST (predeterminat), IDEA i TripleDES.

3.2. El model pla

El model pla és el sistema més senzill d'infraestructura de clau pública que inclou una única autoritat de certificació com a tercera part de confiança encarregada de l'emissió i la gestió dels certificats dels subscriptors. Els usuaris poden validar la identitat dels subscriptors a partir del certificat de l'autoritat de certificació. L'autoritat de certificació posseeix un certificat que ella mateixa ha generat i en el qual els usuaris hi dipositen la confiança.

En un **certificat autosignat**, la clau pública que se certifica correspon a la clau privada que s'utilitza per a signar el certificat. El nom de l'emissor i el titular del certificat són el mateix.

El model pla s'acostuma a fer servir en l'àmbit de les intranets. Per a entorns més extensos i oberts, aquest model s'ha ampliat al model jeràrquic.



3.3. El model jeràrquic

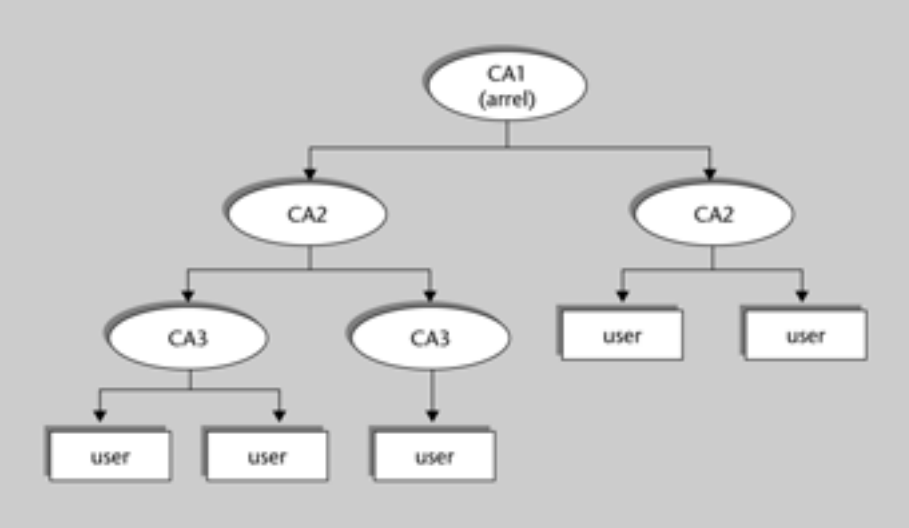
El model jeràrquic representa la implementació més típica d'una infraestructura de clau pública.

En el **model jeràrquic**, els certificats dels subscriptors i les entitats finals estan signats per una entitat externa que també s'identifica amb certificats que emetrà una autoritat de certificació de jerarquia superior.

Els certificats de l'autoritat de certificació de jerarquia superior poden estar a la vegada certificats per altres autoritats de certificació, i així successivament, fins a arribar a una autoritat de certificació que té un certificat autosignat. Aquesta autoritat de certificació s'anomena **autoritat de certificació arrel***, i les que en depenen són les **autoritats de certificació subordinades**. Si els usuaris

* En anglès, *root CA*.

de la infraestructura de clau pública confien en l'autoritat de certificació arrel, aquest vot de confiança s'estendrà per tota la jerarquia.



Els certificats s'estructuren de manera jeràrquica; així, per a verificar l'autenticitat d'un certificat un usuari pot comprovar la signatura de l'autoritat que el va emetre, que a la vegada tindrà un altre certificat expedit per una altra autoritat de nivell superior. D'aquesta manera, anem pujant per la jerarquia fins a arribar al nivell més alt, que estarà ocupat per un certificat que gaudeixi de la plena confiança de tota la comunitat. Les claus públiques dels certificats de més alt nivell fins i tot es publiquen en suport paper perquè qualsevol les pugui verificar.

El PKIX

El PKIX és la plataforma d'infraestructura de clau pública més important que utilitza el model jeràrquic de confiança.

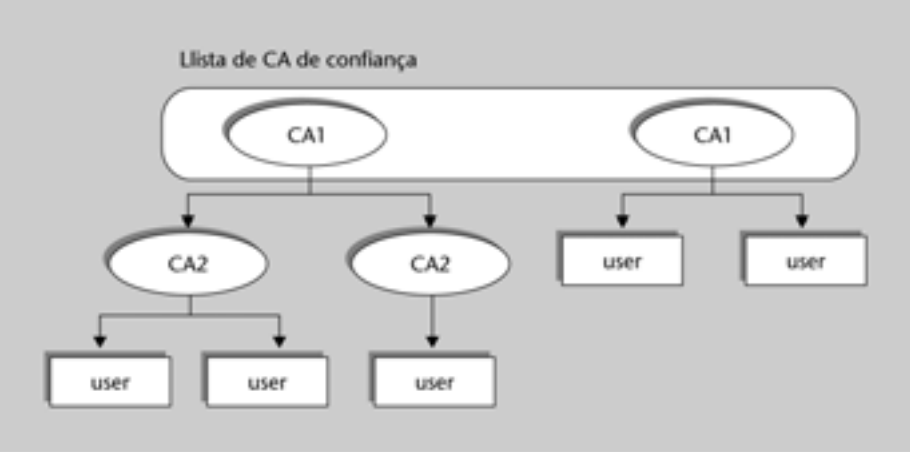
3.4. El model de navegació per llista de confiança

Hem vist que el model jeràrquic requereix que s'estableixi una única autoritat de certificació arrel en la qual tots els usuaris dipositin confiança. Idealment aquesta autoritat de certificació hauria de tenir abast mundial i l'organització que la gestionés hauria de ser prou reconeguda i prestigiosa com perquè qualsevol usuari se'n refiés.

En la pràctica això no és així. Hi ha moltes i molt diferents autoritats de certificació arrels, i cadascuna ofereix servei a petites comunitats d'usuaris. Per a interconnectar aquestes illes d'infraestructures de clau pública han sorgit formes híbrides del model jeràrquic que permeten la interfuncionalitat entre diferents grups d'usuaris controlats per una autoritat de certificació arrel.

La solució més estesa per a interconnectar infraestructures de clau pública jeràrquiques és el **model de navegació per llista de confiança**, també conegut com **model centrat en l'usuari**, en què cada aplicació final té una llista de les claus públiques de totes les autoritats de certificació en què confia.

Aquest model, implementat en la majoria de navegadors web (Netscape, Mozilla, Explorer), permet a l'usuari un gran nivell de flexibilitat per a afegir i esborrar autoritats de certificació de la seva llista de confiança. L'accés a aquesta llista de confiança hauria d'estar totalment protegit contra modificacions externes, ja que determina en quines autoritats de certificació confiem i, per extensió, en quins altres usuaris (certificats per aquestes autoritats de certificació).



El principal problema d'aquest sistema és que no hi ha cap diferència entre una infraestructura de clau pública forta i una de dèbil. Per exemple, la majoria d'usuaris de certificats no miren quina política té un certificat específic abans de confiar-hi. Així, la majoria d'infraestructures de clau pública comercials incloses en la llista dels navegadors no inclouen molts dels controls que els diferents governs exigeixen a les autoritats de certificació.

Un avantatge d'aquest model és que cada aplicació pot triar confiar en un conjunt diferent d'autoritats de certificació.

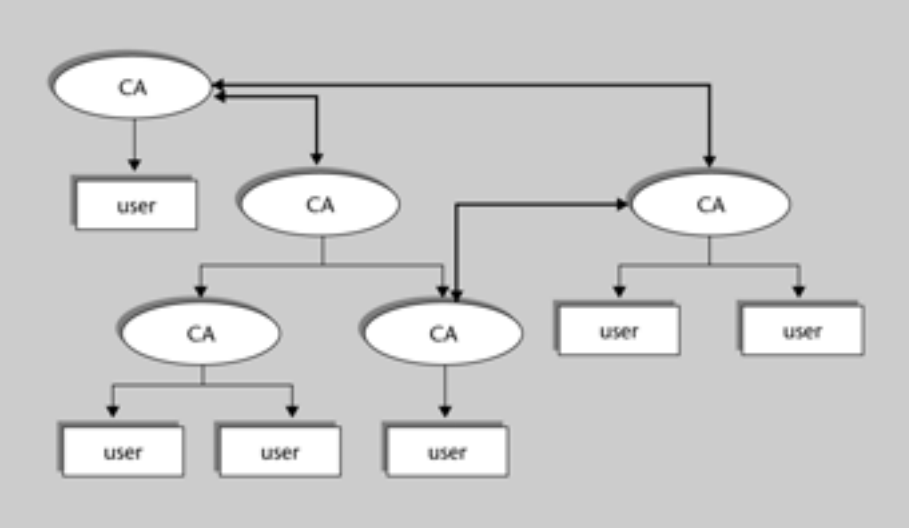
3.5. El model de certificats creuats

En el model de certificats creuats, les autoritats de certificació arrel de cada comunitat d'usuaris emeten certificats per a altres autoritats de certificació que tenen funcions i polítiques equivalents a les seves. Com en el model jeràrquic, cada usuari només confia en una autoritat de certificació arrel, però en aquest model, l'autoritat de certificació arrel és d'àmbit local, i no pas central, com en aquell. La interconnexió entre diferents illes d'infraestructures de clau pública es fa pel certificat creuat d'una autoritat de certificació a una altra.

Vegeu el model jeràrquic en el subapartat 3.3 d'aquest mòdul.

En un **certificat creuat**, el titular i l'emissor són autoritats de certificació diferents. Aquest tipus de certificats s'utilitzen perquè una autoritat de certificació pugui certificar la identitat d'una altra autoritat de certificació.

Els usuaris finals poden validar els certificats d'usuaris d'altres comunitats amb el certificat que ha emès l'autoritat de certificació externa en què confien. L'estructura que l'usuari final veu és la d'una infraestructura de clau pública jeràrquica, però cada usuari tindrà una vista diferent.



El problema bàsic d'aquest model és la dificultat d'implementar aplicacions client capaces de dur-lo a la pràctica. Saber si hi ha una cadena de certificació que enllaci un usuari X amb un usuari Y requereix un esforç de cerca important.

Un altre inconvenient és el gran volum de certificats amb què es treballa. Cada autoritat de certificació arrel ha d'emetre un certificat a totes les altres autoritats de certificació que vol reconèixer. Per exemple, per a establir un domini creuat bidireccional entre diferents infraestructures de clau pública caldria:

a) Per a la interconnexió de tres infraestructures de clau pública:

$$C_3^2 = \binom{3}{2} = 6 \text{ certificats creats}$$

b) Per a la interconnexió de cent infraestructures de clau pública:

$$C_{100}^2 = \binom{100}{2} = \frac{100!}{98!} = 9.900 \text{ certificats creuats}$$

El model de certificats creuats permet organitzar les autoritats de certificació en estructures horitzontals en comptes d'estructures jeràrquiques, semblant a la idea en què es fonamenta el model distribuït.

3.6. El model Bridge-CA

La interconnexió de diverses infraestructures de clau pública per certificats creuats és molt pràctica perquè no requereix l'establiment de cap autoritat de

certificació arrel comuna, però té el problema que és un model poc reescalable. Per a resoldre aquest inconvenient va sorgir el model Bridge-CA, que incorpora una autoritat externa que actuarà de pont entre les infraestructures de clau pública que es volen unir. Totes les autoritats de certificació estableixen una certificació creuada amb el punt central de referència, el Bridge-CA, i d'aquesta manera es crea una comunitat de confiança més extensa.

El US Federal Bridge-CA

El model d'autoritat de certificació US Federal Bridge és un dels projectes pioners en aquest camp. Permetrà interconnectar les autoritats de certificació arrel dels diferents estats i universitats dels Estats Units.

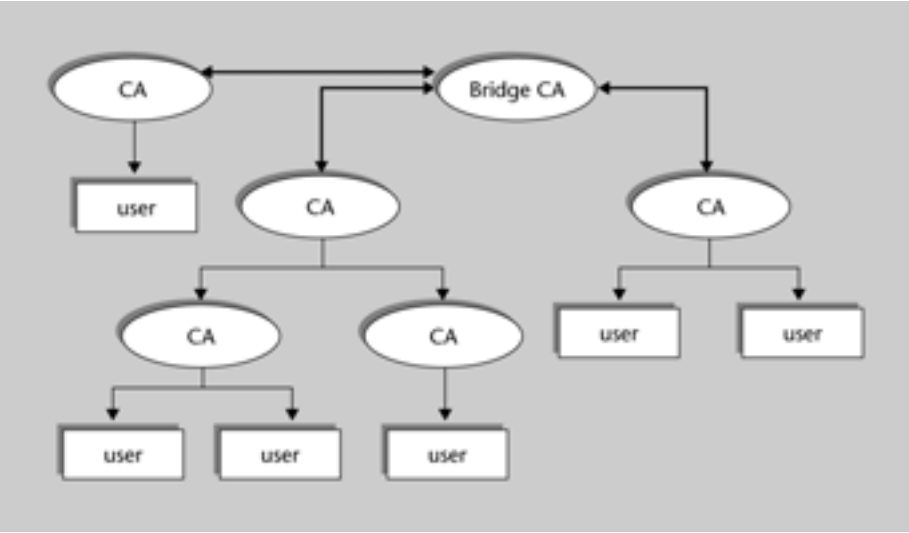
Per a establir un domini creuat bidireccional entre diferents infraestructures de clau pública calen:

a) Per a la interconnexió de tres infraestructures de clau pública:

$$2 \cdot 3 = 6 \text{ certificats creuats}$$

b) Per a la interconnexió de cent infraestructures de clau pública:

$$2 \cdot 100 = 200 \text{ certificats creuats}$$



L'estructura del model Bridge-CA és similar a la del model jeràrquic. La diferència bàsica rau en el fet que el Bridge-CA no és un punt central de confiança, sinó una entitat que facilita la certificació creuada entre totes les autoritats de certificació participants. Un usuari que confiï en l'autoritat de certificació de la seva jerarquia podrà validar certificats d'altres jerarquies que hi estiguin unides pel Bridge-CA, però no li caldrà confiar-hi explícitament.

Vegeu el model jeràrquic en el subapartat 3.3 d'aquest mòdul.

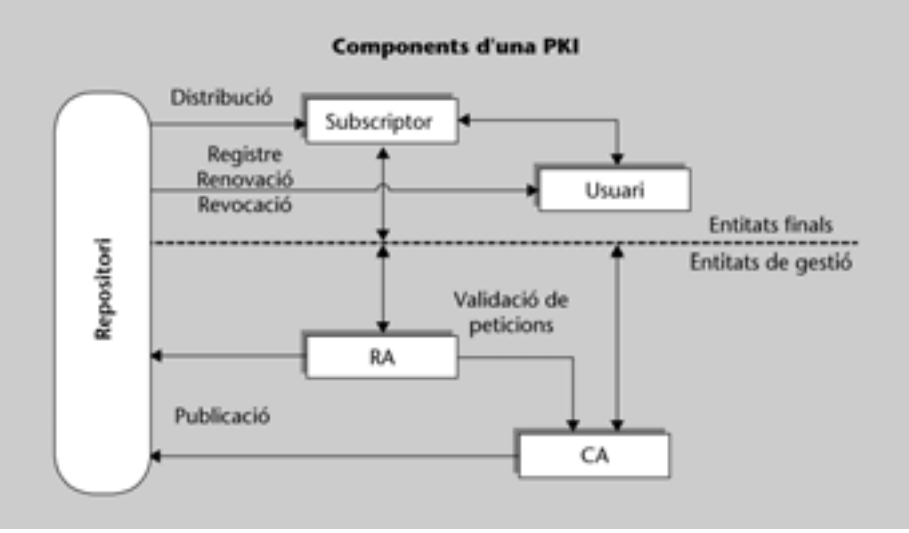
4. Cicle de vida de claus i certificats digitals

En aquest apartat veurem el cicle de vida de les claus i dels certificats digitals i, d'aquesta manera, il·lustrarem els processos bàsics que són comuns a totes les infraestructures de clau pública. Partirem del fet que ja existeix el parell de claus d'una autoritat de certificació i ens basarem en un model de confiança pla.

La figura de més avall il·lustra la relació entre els components bàsics d'una infraestructura de clau pública.

Els subscriptors generen un parell de claus criptogràfiques i es registren en una autoritat de registre per tal de poder demanar un certificat digital. Si l'autoritat de registre aprova la petició de certificació, envia les dades a l'autoritat de certificació perquè emeti el certificat. El certificat es publica en un repositori al qual tots els usuaris tenen accés i, d'aquesta manera, el subscriptor i l'usuari poden utilitzar serveis de seguretat per a comunicar-se.

En els subapartats següents descriurem amb més detall els passos que se segueixen.



4.1. Generació del parell de claus

La primera part del cicle de vida d'un parell de claus i del certificat digital relacionat és la generació de les claus. Segons la **política de certificació*** (**CPS**) de l'autoritat de certificació, tant el subscriptor en el seu entorn local com les entitats externes poden generar el parell de claus criptogràfiques. En aquest úl-

*** En anglès,** ***certification practice statement.***

tim cas, les claus s'han de distribuir als usuaris en fitxers xifrats o targetes criptogràfiques.

Els navegador

Els navegadors disposen d'eines per a generar les parelles de claus criptogràfiques.

Les claus criptogràfiques s'han de generar d'acord amb les necessitats de les aplicacions i els serveis que les utilitzaran. Per exemple, la vida útil de les dades que es volen protegir determina la longitud òptima de les claus. D'altra banda, el període de temps considerat de validesa d'una clau o certificat dependrà del tipus d'operacions que es vulgui efectuar.

Per tal de proporcionar un mitjà coherent de discernir quines claus són apropiades per a certes aplicacions i quines no, els certificats contenen un camp amb la informació del tipus de certificat que representen. Bàsicament n'hi ha de dos tipus:

a) En els **certificats de xifratge** pot ser necessari tenir una còpia de seguretat de la clau privada per tal de poder recuperar la informació en clar, si la clau de desxifratge es perd o es destrueix. Aquesta pràctica és comuna en l'entorn empresarial.

b) Els **certificats de signatura** es fan servir per a donar serveis d'autenticació i de no-repudi, i és important que no hi hagi cap còpia de la clau privada corresponent. Qualsevol sistema de recuperació de claus per a aquest tipus de certificats n'invalidaria les propietats.

4.2. Registre

El registre és el procés pel qual un subscriptor o entitat final es dóna a conèixer per primera vegada a una autoritat de certificació (directament o des d'una autoritat de registre) per tal de demanar-hi un certificat.

El subscriptor ha de proporcionar informació referent a la seva identitat, com el seu nom, un identificador únic, alguna prova d'identitat i altra informació addicional que estigui especificada en la CPS de l'autoritat de certificació. Si ell mateix ha generat el parell de claus criptogràfiques, també ha de donar la clau pública que vol certificar. Tota aquesta informació s'estructura en una petició de certificat amb format PKCS#10. La verificació de les dades aportades en el procés de registre es fa seguint les directrius establertes en la CPS.

Veurem el format PKCS#10 en el subapartat 6.2 d'aquest mòdul.

4.3. Certificació

La certificació és el procés pel qual l'autoritat de certificació emet un certificat, és a dir, signa digitalment la clau pública i les dades d'un subjecte i, per tant, dóna fe d'aquesta correspondència.

El subscriptor proporciona les dades que el certificat ha de contenir per mitjà d'una petició de certificat PKCS#10. Aquestes dades han estat prèviament validades, i s'ha comprovat que el subscriptor té drets per a tenir un certificat d'aquestes característiques. En el procés de certificació també es defineix el període de validesa del certificat i l'ús que se'n podrà fer.

Un cop generat el certificat, l'autoritat de registre (o, si no n'hi ha, l'autoritat de certificació) s'encarregarà de fer-lo arribar al subscriptor i penjar-lo en el directori públic adequat.

4.4. Recuperació de claus

En algunes situacions és important poder recuperar una clau privada per a accedir a informació xifrada prèviament. Per exemple, una empresa pot necessitar accés a documents prèviament xifrats per un empleat que s'hagi donat de baixa de l'empresa.

L'autoritat de certificació pot oferir sistemes de recuperació de claus mitjançant un sistema de còpies de seguretat al qual només puguin accedir els usuaris autoritzats per obtenir les dades. El sistema ha de garantir que les claus no es poden comprometre.

4.5. Revocació de certificats

La revocació de certificats és el procés pel qual una autoritat de certificació invalida un certificat abans del seu període d'expiració.

Hi ha diverses raons per les quals un certificat pot ser revocat:

a) Compromís de la clau privada associada al certificat.

b) Canvi del subscriptor per un altre proveïdor de serveis d'infraestructura de clau pública.

c) Canvi en la informació que porta el certificat (nom del subscriptor, companyia a què pertany, etc.).

d) Substitució prematura de la parella de claus de l'autoritat de certificació.

En qualsevol d'aquestes situacions, el certificat deixa de tenir utilitat i el subscriptor ho ha de notificar a l'autoritat de certificació/registre. La notificació es

pot fer en persona, per correu electrònic signat o per qualsevol altre canal autenticat. La clau privada corresponent al certificat revocat es pot utilitzar per a autenticar una petició de revocació, però per raons òbvies, si la petició és la causa del compromís de la clau, aquesta no valdrà per a l'autenticació en la generació d'un nou certificat.

El mètode predominant de distribució de la informació dels certificats que han estat revocats són les llistes de revocació de certificats (CRL).

Una **llista de revocació de certificats** (**CRL**) és una llista dels certificats que han estat revocats, signada i datada per l'autoritat de certificació corresponent.

L'autoritat de certificació publica les llistes de revocació de certificats en un servei de directori a intervals regulars, i és responsabilitat del client obtenir i comprovar la llista per determinar la validesa dels certificats.

El cicle de publicació de CRL

El cicle de publicació de les CRL és crític en una aplicació d'autorització d'usuaris per a accedir a un registre amb dades financeres.

El temps transcorregut entre la revocació d'un certificat i la publicació de la llista de revocació de certificats que en porta la informació és crític, ja que hi ha una escletxa de confiança en el certificat en un moment en què la clau privada pot estar compromesa i durant el qual es poden prendre decisions crítiques.

4.6. Renovació de certificats

La renovació d'un certificat digital pot venir donada bàsicament per dos motius:

Els períodes de renovació

El període apropiat per a una clau RSA de 1.024 bits que s'utilitzi per a autenticació és d'uns dos anys.

- Pèrdua de validesa del certificat (caducitat del certificat, revocació per canvi d'informació del certificat, etc.).

- Pèrdua de validesa de les claus que certifica (revocació del certificat per compromís de la clau privada associada).

El certificat digital caduca quan n'expira el període de validesa. El **període de validesa** d'un certificat s'estableix d'acord amb les restriccions d'una sèrie de factors: l'algorisme de clau pública i la llargada de la clau, la utilització del certificat, els acords financers i de manteniment amb l'autoritat de certificació, els requisits del cicle de vida dels serveis i el programari que el certificat necessita.

La renovació del certificat digital per pèrdua de validesa és simple: la petició de renovació se signa amb la clau privada associada al certificat que es vol renovar, que encara té validesa. D'aquesta manera, el subscriptor es pot autenticar davant l'autoritat de certificació sense haver de tornar a passar tota la fase de registre. En aquest cas es pot certificar el mateix parell de claus o generar-

ne un altre parell. Normalment, en cas de caducitat del certificat es genera un nou parell de claus, ja que s'utilitza justament la data de caducitat dels certificats per a limitar el temps de vida de les claus i reduir els possibles atacs al criptosistema per força bruta.

En el cas de revocació del certificat per compromís de la clau privada, la renovació del certificat no és tan transparent. La infraestructura de clau pública ha d'oferir mitjans en els quals no intervingui el parell de claus per a anunciar que el certificat vell deixa de ser vàlid en favor d'un nou certificat. Sovint s'associa una contrasenya al procés de generació del certificat per tal de poder-lo revocar. En aquest cas, la renovació del certificat passa per la generació d'un nou parell de claus i l'emissió d'un nou certificat per a les claus noves.

El pitjor cas és quan la clau compromesa és la de la pròpia autoritat de certificació. Aleshores s'ha d'invalidar tant el certificat de l'autoritat de certificació com el de tots els subscriptors que en depenien.

El fet que l'autoritat de certificació tingui les claus compromeses fa dubtar de la confiança en qualsevol missatge electrònic de l'autoritat de certificació que notifiqui aquest fet (qui signa el missatge, si les claus no són vàlides?). Una manera de pal·liar els efectes de la caiguda d'una autoritat de certificació és que es pregenerin i distribueixin unes claus substitutòries que s'utilitzaran en cas de compromís de les nominals. A més, caldrà reemetre certificats a tots els subscriptors, de manera que estiguin signats per una clau no revocada.

5. Estructures de dades bàsiques de PKIX

En aquest apartat ens centrarem en les estructures de dades definides per la IETF en els seus estàndards del grup PKIX.

5.1. Certificats X.509

El format de certificats més àmpliament acceptat en infraestructura de clau pública està definit per l'ISO/IEC JTC1 SC21 i es coneix com X.509v3.

L'X.509

El terme X.509 ve de l'especificació X.500 de serveis de directori.

L'**estàndard internacional X.509v3**, que també es publica amb el nom d'**ITU-T Recommendation X.509**, és important per dues raons bàsiques: defineix el marc per a la provisió de serveis d'autenticació i un format de certificat per a les claus públiques.

El format dels certificats X.509 ha evolucionat a partir de tres versions en diferents edicions de l'estàndard. La primera versió va aparèixer el 1988. El 1993 es va revisar i se n'obtingué la versió 2, en què s'hi van afegir dos camps opcionals d'identificació única. L'ús de l'X.509 junt amb l'*Internet privacy enhanced mail* (PEM) va fer sorgir la necessitat de dotar el certificat de més flexibilitat. El juny de 1996 va aparèixer la versió 3 del certificat, que inclou la possibilitat de tenir camps d'ampliació.

L'X.509 es va dissenyar com un marc general d'autenticació per a donar suport als serveis de directori X.500. Aquest caràcter obert origina múltiples possibilitats per a alguns dels camps del certificat (principalment en les ampliacions). Per a regular l'ús de les ampliacions i el seu significat, apareixen els perfils de certificats, que concreten i limiten les combinacions d'atributs que es poden definir per a cada tipus de certificat. El perfil de certificat més estès és el definit pel grup de treball PKIX de la IETF. Aquest grup s'encarrega tant de definir el perfil de certificats i llistes de revocació, com de desenvolupar especificacions i protocols relacionats amb l'entorn d'infraestructura de clau pública.

En els subapartats següents veurem la manera de designar les entitats participants en l'estructura de clau pública, i després comentarem el format general d'un certificat i n'analitzarem cadascun dels camps.

5.1.1. Nom dels components d'un certificat

Els certificats X.509, com la resta de components de la infraestructura de clau pública, es designen per un nom distingit.

Un **nom distingit** (**DN**) és un conjunt d'atributs amb valors associats. Aquest tipus d'estructura permet que cada organització decideixi quins atributs són interessants per a identificar les entitats dins de la seva empresa.

Els atributs més habituals es mostren a la taula del marge. Per a alguns usos del certificat, el sistema de noms per noms distingits pot resultar insuficient. En aquests casos es fan servir uns camps especials del certificat anomenats *noms alternatius* que permeten especificar l'adreça de correu electrònic, l'adreça IP d'una web, el DNS, el nom EDI, etc.

Atributs més usuals dels noms distingits

CN:	Nom comú
OU:	Unitat organitzativa
O:	Organització
C:	País
L:	Localitat
S:	Estat
STREET:	Carrer
T:	Títol de la persona

Camps especials

Les adreces de correu electrònic s'especifiquen en el camp *Nom alternatiu del titular* (*subjectAltName*) seguint el format rfc822Name definit en [RFC822].

5.1.2. Estructura d'un certificat

En un certificat hi ha tres tipus principals de camps:

a) Bàsics: camps que aporten informació sobre l'autoritat de certificació que ha emès el certificat, l'entitat/subscriptor a què pertany la clau pública, la mateixa clau pública, i el període de validesa i la identificació del certificat.

b) Necessaris per a la signatura: camps que utilitzarà qui rebi el certificat per a comprovar que el document està signat correctament.

c) Ampliacions: camps que han aparegut per a cobrir les noves necessitats d'atributs en un certificat. Alguns dels possibles usos dels camps d'ampliació són la millora de la gestió de l'herència de certificació, les polítiques de seguretat o les dades sobre l'usuari i la seva clau. Les ampliacions poden estar definides en estàndards o per una organització particular que faci ús de certificats.

Llenguatge dels certificats

Els certificats X.509, com les altres estructures definides per PKIX, estan especificades en llenguatge ASN.1 (*Abstract Syntax Notation One*).

Els camps bàsics d'un certificat X.509v3 són els següents:

Camp	Descripció
version	Identifica el número de versió del certificat: v1, v2, v3.
serialnumber	Enter assignat per l'autoritat de certificació. Identifica unívocament el certificat d'una autoritat de certificació.
signature	Ha de coincidir amb l'algorisme de signatura dels camps necessaris per a la signatura.
issuer	Nom distingit de l'autoritat de certificació que ha signat i emès el certificat.
validity	Dos camps que identifiquen el període en què el certificat és vàlid.
subject	Nom distingit que identifica el propietari del certificat.
subjectPublicKeyInfo	Camp amb informació sobre la clau pública per la qual s'emet el certificat.

Camp	Descripció
issuerUniqueID	Camp opcional i poc utilitzat que permet la reutilització de noms d'autoritat de certificació.
subjectUniqueID	Camp opcional que permet la reutilització de noms de titulars.
extensions	Camp opcional que conté els camps d'ampliacions del certificat.

Els camps necessaris per a la signatura són:

Camp	Descripció
signatureAlgorithm	Conté l'identificador de l'algorisme de *hash* (MD2, MD5, SHA-1) i l'algorisme de clau pública (RSA, DSA) que s'ha utilitzat en la signatura del certificat digital.
signature	Conté el valor de la signatura dels camps bàsics i les ampliacions del certificat amb l'algorisme de signatura descrit en el camp anterior.

Identificació dels certificats

El número de sèrie del certificat, junt amb el nom de l'autoritat de certificació, ha de servir per a identificar unívocament un certificat en qualsevol context.

Totes les ampliacions apareixen dins del camp *extensions* del certificat, i cadascuna està formada per tres subcamps:

Camp d'una ampliació	Descripció
extnID	Identificador unívoc de l'ampliació que representa.
critical	Valor lògic que marca si l'ampliació és crítica o no. Les aplicacions poden ignorar les ampliacions no crítiques en cas que no les reconeguin. La presència d'una ampliació crítica no reconeguda provoca el rebuig del certificat.
extnValue	Conté el valor de l'ampliació.

Cada ampliació només pot aparèixer una vegada en cada certificat.

A continuació, comentem algunes de les ampliacions més destacades dels certificats i n'indiquem el contingut del subcamp *extnValue*:

Tipus d'ampliació	Descripció
subjectAltName	Nom alternatiu del titular (adreça de correu electrònic, etc.).
issuerAltName	Nom alternatiu de l'autoritat de certificació emissora.
keyUsage	Defineix i limita els possibles usos de la clau certificada.
extKeyUsageSyntax	Complement del *keyUsage* i el *basicConstraints* per a delimitar els possibles usos de la clau dins les necessitats particulars d'una organització.
basicConstraints	Distingeix si el certificat és d'una entitat final (fals) o d'una autoritat de certificació (veritable). Permet limitar la longitud de cadena de certificació.
nameConstraints	Limita l'espai de noms que poden utilitzar els certificats següents de la cadena.

Tipus d'ampliació	Descripció
certificatePolicies	Informació sobre la CPS de l'autoritat de certificació.
policyMappings	Identifica les polítiques d'una autoritat de certificació amb les d'una altra.
policyConstraints	Limita la verificació del camí de certificació segons unes polítiques.
inhibitAnyPolicy	Limita l'ús de la política *anyPolicy* a un nombre de certificats de la cadena.
privateKeyUsagePeriod	Especifica períodes de validesa per a la clau privada que són diferents dels del certificat.
authorityKeyIdentifier	Identificador únic de la clau pública necessari per a validar el certificat. Útil quan una autoritat de certificació té diverses claus de signatura.
subjectKeyIdentifier	Identificador únic de certificats que fa el *hash* sobre la clau pública o genera un nombre únic.
crlDistributionPoints	Indica la localització de les llistes de revocació de certificats relacionades amb el certificat a partir d'uns punts de distribució.
fraeshestCRL	Indica on trobar informació de les llistes de revocació de certificats més recents.
authorityInformationAccess	Ampliació privada definida per PKIX que indica com accedir a informació i serveis que l'autoritat de certificació emissora proporciona.

L'ampliació *keyUsage*, que defineix els propòsits del certificat, és una de les més importants. El PKIX recomana que en cas de fer-la servir, aquesta ampliació es marqui com a crítica. Els valors que pot prendre són una combinació de les opcions següents:

Valors de *keyUsage*	Descripció
digitalSignature	El certificat pot verificar signatures de documents.
nonRepudiation	Es poden verificar signatures amb la propietat de no-repudi.
keyEncipherment	La clau pot ser utilitzada per a transportar claus.
dataEncipherment	La clau es pot fer servir per a xifrar dades d'usuari que no són claus.
keyAgreement	S'utilitza la clau per a algorismes d'establiment d'una clau de sessió.
keyCertSign	La clau es fa servir per a validar la signatura d'altres certificats (emprat en autoritats de certificació).
encipherOnly	S'utilitza conjuntament amb *keyAgreement* per a limitar l'ús de la clau a xifrar dades durant el procés d'establiment de la clau.
decipherOnly	Es fa servir conjuntament amb *keyAgreement* per a limitar l'ús de la clau a desxifrar dades durant el procés d'establiment de la clau.
subjectUniqueID	Camp opcional que permet la reutilització de noms de titulars.

L'estructura de dades del certificat indicat és la següent:

```
Certificate: SEQUENCE (1006)
   toBeSigned SEQUENCE (726)
     version [0] INTEGER (1) 0x02 (2)
     serialNumber INTEGER (1) 0x24 (36)
     signature SEQUENCE (13)
       algorithm OBJECT IDENTIFIER (9) pkcs1-sha1WithRsaSignature
       parameters TYPE (2) NULL (0)
     issuer SEQUENCE OF (71) OU=Coyote Subordinate CA (MIR), O=UOC, C=ES
     validity SEQUENCE (30)
       notBefore UTCTime (13) "030929144413Z"
       notAfter UTCTime (13) "040929144251Z"
     subject SEQUENCE OF (65) CN=user, OU=Criptografia, O=UOC, C=ES
     subjectPublicKeyInfo SEQUENCE (159)
      algorithm SEQUENCE (13)
        algorithm OBJECT IDENTIFIER (9) pkcs1-rsaEncryption
        parameters TYPE (2) NULL (0)
      subjectPublicKey BIT STRING (141) Encapsulates TYPE (140) with
          rSAPublicKey SEQUENCE (137)
            smodulus INTEGER (129) 0x00D0A60F7EF71D4FCB342982E2333C575392AFF9B..
       exponent INTEGER (3) 0x010001 (65537)
     issuerUId [1] IMPLICIT BIT STRING OPTIONAL NOT PRESENT
     subjectUId [2] IMPLICIT BIT STRING OPTIONAL NOT PRESENT
     extensions [3] SEQUENCE OF (361)
      extension SEQUENCE (14)
        extnId OBJECT IDENTIFIER (3) id-ce-keyUsage
        critical BOOLEAN (1) TRUE
        extnValue OCTET STRING (4) Encapsulates TYPE (4) with
           BIT STRING (2) 03 F8
     extension SEQUENCE (29)
         extnId OBJECT IDENTIFIER (3) id-ce-extKeyUsage
         critical BOOLEAN (1) FALSE DEFAULT
         extnValue OCTET STRING (22) Encapsulates TYPE (22) with
            SEQUENCE OF (20)
               OBJECT IDENTIFIER (8) id-kp-clientAuth
               OBJECT IDENTIFIER (8) id-kp-emailProtection
     extension SEQUENCE (17)
                              ...
signatureAlgorithm SEQUENCE (13)
   algorithm OBJECT IDENTIFIER (9) pkcs1-sha1WithRsaSignature
   parameters TYPE (2) with NULL (0)
signature BIT STRING (257) 00 73E9098140B1DD4BBE66F0487AD1EDC85E4B...
```

5.2. Llistes de revocació (CRL)

Les llistes de revocació de certificats serveixen perquè una autoritat de certificació pugui distribuir informació sobre els certificats que ha emès i han estat revocats.

Una **llista de revocació de certificats** (**CRL**) és una estructura de dades signada per l'autoritat de certificació emissora que conté les dades necessàries per a determinar l'estat d'un certificat.

En els certificats X.509 es manté informació sobre on es pot obtenir la llista de revocació de certificats de l'autoritat de certificació emissora, en concret a l'ampliació *crlDistributionPoint*.

En el mateix estàndard X.509 s'especifica un format per a les llistes de revocació de certificats. La publicació de la primera versió de l'estàndard va ser l'any 1988. De la mateixa manera que va passar amb els certificats (vegeu el subapartat 5.1.2), més tard, a la versió 2, l'estàndard es va modificar amb l'objectiu d'introduir-hi els camps d'ampliació.

En els subapartats següents veurem els tipus bàsics de llistes de revocació de certificats, com es distribueixen i quin és el format de la seva estructura de dades.

A la figura veiem el tipus d'informació que mostra MS Internet Explorer d'un certificat.

5.2.1. Tipus de llistes de revocació de certificats

Hi ha tres tipus bàsics de llistes de revocació de certificats: CRL simple, delta CRL i CRL indirecta.

Una **llista de revocació de certificats simple** és un fitxer seqüencial que conté una llista dels certificats revocats.

Taxes de revocació

La taxa de revocació de certificats se situa en un 10% en un any.

Aquest fitxer creix a mesura que el temps passa i esdevé intractable i ineficient. Per aquest motiu, han sorgit altres formes de llistes de revocació de certificats que proporcionen a l'aplicació final només el conjunt d'informació de revocació que hi pot ser rellevant.

Una **delta CRL** és una llista de revocació de certificats que complementa la informació continguda en una altra llista de revocació de certificats base.

La delta CRL està dissenyada per a reduir les dimensions de la llista que es transmet a l'entitat final. Inicialment, i després en intervals regulars planificats, es publica una llista de revocació de certificats completa –la CRL base– que conté informació de totes les revocacions conegudes. A partir d'aquest

punt s'emeten una sèrie de llistes d'actualització que contenen els canvis que s'han produït des de l'última publicació d'una delta CRL o CRL base. Cada actualització es distribueix com una llista de revocació de certificats, i l'autoritat de certificació l'emet i la signa de la mateixa manera que una llista de revocació de certificats.

Una **CRL indirecta** és una llista de revocació de certificats que conté informació de revocació de diverses autoritats de certificació.

Aquests tipus de llistes de revocació de certificats s'utilitzen per a reduir el nombre de llistes de revocació de certificats que és necessari obtenir per a poder comprovar la validesa d'un certificat. Per tal que una entitat final pugui utilitzar una CRL indirecta, s'hi ha de confiar com es confia en l'autoritat de certificació que emet el certificat.

5.2.2. Punts de distribució de llistes de revocació de certificats

Els punts de distribució de llistes de revocació de certificats serveixen per a fragmentar la llista de revocació de certificats global d'una autoritat de certificació en unes quantes de més petites; cada subconjunt engloba un nombre determinat de certificats i/o un subconjunt de raons de revocació. Els certificats X.509 contenen un apuntador al punt de distribució corresponent.

Amb l'ús de punts de distribució de llistes de revocació de certificats disminueix el consum de recursos de xarxa (ample de banda i temps de transmissió) per la menor dimensió de cada llista de revocació de certificats d'un punt de distribució respecte d'una única llista que inclogués les revocacions de tots els certificats.

5.2.3. Format de les llistes de revocació de certificats

En una llista de revocació de certificats podem trobar quatre tipus principals de camps:

a) Bàsics: camps que aporten informació sobre l'autoritat de certificació que ha emès la llista de revocació de certificats, la data en què s'ha actualitzat, la data de la següent actualització i els certificats que han estat revocats.

b) Necessaris per a la signatura: camps que utilitzarà qui rebi la llista de revocació de certificats per a comprovar que el document està signat correctament.

c) Ampliacions de la CRL: ampliacions que proporcionen mètodes per a associar atributs addicionals a la llista de revocació de certificats. Poden ser definides en estàndards o per organitzacions particulars.

d) Ampliacions de l'entrada de la CRL: ampliacions que proporcionen informació addicional sobre les entrades de la llista de revocació de certificats.

Els camps bàsics d'una llista de revocació de certificats són els següents:

Camp	Descripció
version	Identifica el número de versió de la llista de revocació de certificats: v1, v2.
signature	Ha de coincidir amb l'algorisme de signatura dels camps necessaris per a la signatura.
issuer	Nom distingit de l'autoritat de certificació que ha signat i emès la llista de revocació de certificats.
thisUpdate	Data d'emissió de la llista de revocació de certificats.
nextUpdate	Data posterior en què s'emetrà la següent llista de revocació de certificats.
revokedCertificates	Llista amb els certificats revocats.
crlExtensions	Camp opcional que conté els camps d'ampliacions de la llista amb els certificats revocats.

Els camps necessaris per a la signatura són:

Camp	Descripció
signatureAlgorithm	Conté l'identificador de l'algorisme utilitzat per a signar la llista de revocació de certificats. És un camp anàleg al que trobem en un certificat.
signature	Conté la signatura de la llista de revocació de certificats segons l'algorisme establert.

Les ampliacions, que tenen el mateix format que les definides per a l'estàndard X.509, són dels tipus següents:

Vegeu l'estàndard X.509 en el subapartat 5.1 d'aquest mòdul.

Tipus d'ampliació	Descripció
authorityKeyIdentifier	Identifica la clau de l'autoritat de certificació amb la qual es pot validar la llista de revocació de certificats.
issuerAltName	Nom alternatiu de l'autoritat de certificació emissora.
crlNumber	Permet determinar quan una llista de revocació de certificats n'actualitza una altra.
deltaCrlIndicator	Indica que es treballa amb llista delta de revocació de certificats i identifica la llista de revocació de certificats base.
issuingDistributionPoint	Indica on es troba la llista de revocació de certificats i el tipus de certificats que hi haurà a la llista (d'autoritats de certificació, de subscriptors, revocats per un determinat motiu, etc.).

Les ampliacions d'entrada a la llista de revocació de certificats apareixen en el subcamp *crlEntryExtensions* de cada entrada de la llista de certificats revocats continguts en el camp *revokedCertificates*. Poden tenir els valors següents:

Ampliació d'entrada	Descripció
reasonCode	Identifica la causa per la qual s'ha revocat un certificat.
holdInstructionCode	Indica l'acció que cal dur a terme quan es troba un certificat que ha estat suspès.
invalidityDate	Conté la data en què se sap o se sospita que el certificat es va invalidar o la clau va ser compromesa. Pot ser anterior a la data de revocació.
certificateIssuer	Identifica l'emissor del certificat en una entrada d'una llista de revocació de certificats indirecta.
issuingDistributionPoint	Indica on es troba la llista de revocació de certificats i el tipus de certificats que hi haurà a la llista (d'autoritats de certificació, de subscriptors, revocats per un determinat motiu, etc.).

6. Format i procediments per a generar missatges: PKCS

Les **normes PKCS**, o ***public-key cryptography standards***, són un conjunt d'especificacions desenvolupades en els laboratoris RSA amb l'objectiu d'establir una norma comuna a la indústria sobre els formats de les dades utilitzades en criptografia de clau pública. En realitat no són estàndards, sinó una aproximació creada *ad hoc* per una única empresa amb el suport i la col·laboració d'altres fabricants (Apple, Microsoft, DEC, Lotus, Sun, MIT, etc.).

La primera publicació de PKCS es va fer el 1991 i, des d'aleshores, desenvolupadors de tot el món els han referenciat i implementat àmpliament. Algunes contribucions de les sèries PKCS han estat adoptades com a estàndards formals i *de facto*, com PKIX, SET; S/MIME i SSL. Actualment hi ha deu normes PKCS: PKCS #1, #3, #5, #7, #8, #9, #10, #11, #12 i #15. Els PKCS #13 i #14 encara no han estat publicats, els PKCS#2 i #4 han estat incorporats en el PKCS#1, i el PKCS#6 s'ha retirat en favor de la versió 3 de l'estàndard X.509.

Els PKCS inclouen normes que defineixen tant la sintaxi d'una estructura de dades independentment de l'algorisme, com formats específics per a un algorisme concret.

PKCS	Descripció
1	Defineix els mecanismes per a xifrar i signar dades amb el criptosistema de clau pública RSA. També defineix una sintaxi, idèntica a X.509, per a les claus públiques i privades.
3	Defineix un protocol Diffie-Hellman per a l'intercanvi de claus.
5	Descriu un mètode per a xifrar una cadena de text amb una clau secreta derivada d'una frase de pas. El seu objectiu primari és permetre la transmissió xifrada de claus privades entre ordinadors, com es descriu en el PKCS#8, encara que pot ser utilitzada per a xifrar missatges. Fa servir MD2 o MD5 per a produir una clau a partir d'una frase de pas. Aquesta clau s'utilitza per a xifrar amb DES (en mode CBC) el missatge en qüestió.
7	Defineix una sintaxi general per als missatges que inclouen millores criptogràfiques, com signatures digitals o xifratge.
8	Descriu el format de la informació de la clau privada. Aquesta informació inclou una clau privada per a algun algorisme de clau pública i, opcionalment, un conjunt d'atributs.
9	Defineix els tipus d'atributs seleccionats per utilitzar en altres estàndards PKCS.
10	Descriu una sintaxi per peticions de certificació.
11	Defineix una interfície de programació independent de la tecnologia, anomenada Cryptoki, per a dispositius criptogràfics com targetes intel·ligents i targetes PCMCIA.
12	Especifica un format portable per a emmagatzemar i transportar claus privades d'usuari, certificats, secrets diversos, etc.
13	Definirà els mecanismes per a xifrar i signar dades utilitzant criptografia basada en corbes el·líptiques.
14	Cobreix la generació de nombres pseudoaleatoris.
15	Complement del PKCS#11 que defineix el format de les credencials criptogràfiques emmagatzemades en dispositius criptogràfics.

6.1. El PKCS#7

El PKCS#7 descriu una sintaxi general per a dades que poden portar incorporada criptografia, com signatures digitals o sobres digitals. La sintaxi admet recursivitat, de manera que, per exemple, un sobre digital pot ser dins d'un altre i una entitat pot signar dades que abans han estat posades dins d'un sobre digital.

Els diferents tipus d'objectes PKCS#7 que ens hi trobem són: *data*, *signedData*, *envelopedData*, *signedAndEnvelopedData*, *digestedData* i *encryptedData*. A continuació els veurem amb més detall.

6.1.1. Els *data*

Es tracta d'informació en clar que no ha sofert cap tipus de manipulació criptogràfica.

6.1.2. Els *signedData*

L'objecte *signedData* engloba informació a la qual s'ha aplicat una signatura digital. La informació que porta aquesta estructura de dades és la següent:

signedData	**Descripció**
version	Versió de l'estàndard que es fa servir.
digestAlgorithms	Conjunt d'identificadors dels algorismes de *hash* que s'han utilitzat per a crear les signatures digitals.
contentInfo	Objecte que volem signar.
certificates	Cadena de certificació des dels certificats utilitzats per la signatura fins als certificats de les autoritats de certificació de nivell més alt.
crls	Conjunt de llistes de certificats revocats.
signerInfo	Conjunt d'informació sobre els signants.
version	Versió de l'estàndard que es fa servir.
issuerAndSerialNumber	Identifica el certificat del signant a partir del nom de l'autoritat de certificació que l'ha emès i el número de sèrie del certificat.
digestAlgorithm	Algorisme utilitzat per a resumir el contingut del missatge.
authenticateAttributes	Camp opcional que conté un conjunt d'atributs que han estat signats. Útil per a incloure la data en què es genera la signatura.
UnauthenticatedAttributes	Conjunt d'atributs que no han estat signats.

El contingut *signedData* de PKCS#7

El tipus de contingut *signedData* de PKCS#7 proporciona un cas d'ús degenerat en el qual no hi ha cap signatari del contingut. Aquest format s'utilitza per a distribuir cadenes de certificats i llistes de revocació.

L'objecte *signedData* pot portar signatures generades per diversos usuaris. Per a cada signant s'haurà de generar el resum de les dades per signar amb l'algorisme que aquest especifiqui, i després s'hauran de signar amb la clau privada de l'usuari que emet la signatura. Com es veu en la taula, el camp *signerInfo* inclou altres camps que donen informació més detallada sobre com es genera la signatura per a cada usuari.

Per a crear una estructura de signatura digital s'han de seguir els passos següents:

1) Per a cada signant, es calcula el resum de les dades amb l'algorisme indicat per l'usuari que signa. Si el signant defineix atributs autenticats, el resum serà el d'aquests atributs. Si no, només es tindrà en compte el missatge contingut en *contentInfo*.

2) Per a cada signant, se signa el resum calculat en el pas anterior utilitzant la clau privada.

3) Per a cada signant, la signatura digital i altra informació específica del remitent s'agrupen en el camp anomenat *signerInfo*.

4) Els algorismes que s'han utilitzat per a resumir el missatge de tots els remitents, els diferents camps *signerInfo* i el missatge original s'agrupen en un camp *signedData*.

Els passos que haurà de seguir el destinatari són:

a) Verificar els resums de cada signant amb la clau pública d'aquest.

b) Comparar que els resums obtinguts en el pas anterior coincideixen amb els que s'obtenen en fer els càlculs amb les dades que s'autentiquen.

Una signatura digital pot representar-se amb un PKCS#7 que contingui l'objecte *signedData*.

Exemple de l'estructura d'un PKCS#7 *signedData*

```
PKCS#7: SEQUENCE (2417)
   contentType OBJECT IDENTIFIER (9) pkcs7-signedData
   content [0] TYPE (2402) with SEQUENCE (2398)
     version INTEGER (1) 0x01 (1)
     digestAlgorithms SET OF (11) SEQUENCE (9)
       algorithm OBJECT IDENTIFIER (5) secsig-algorithm-sha1
       parameters TYPE (2) with NULL (0)
     contentInfo SEQUENCE (26)
       contentType OBJECT IDENTIFIER (9) pkcs7-data
       content [0] TYPE (13) with OCTET STRING (11) "Hello World"
     certificates [0] IMPLICIT SET OF (2074) SEQUENCE (997)
       toBeSigned SEQUENCE (717)
         version [0] INTEGER (1) 0x02 (2)
         serialNumber INTEGER (1) 0x13 (19)
```

```
crls [1] IMPLICIT SET OF OPTIONAL NOT PRESENT
signerInfos SET OF (272) SEQUENCE (268)
  version INTEGER (1) 0x01 (1)
  issuerAndSerialNumber SEQUENCE (76)
    issuer SEQUENCE OF (71) OU=Coyote Subordinate CA (MIR), O=UOC, C=ES
    serialNumber INTEGER (1) 0x13 (19)
  digestAlgorithm SEQUENCE (9)
    algorithm OBJECT IDENTIFIER (5) secsig-algorithm-sha1
    parameters TYPE (2) with NULL (0)
  authenticatedAttributes [0] IMPLICIT SET OF (93)
    SEQUENCE (24)
      attrType OBJECT IDENTIFIER (9) pkcs9-contentType
      attrValue TYPE (13) with SET OF (11) OBJECT IDENTIFIER (9) pkcs7-data
    SEQUENCE (28)
      attrType OBJECT IDENTIFIER (9) pkcs9-signingTime
      attrValue TYPE (17) with SET OF (15) UTCTime (13) "030929150419Z"
    SEQUENCE (35)
      attrType OBJECT IDENTIFIER (9) pkcs9-messageDigest
      attrValue TYPE (24) with SET OF (22) OCTET STRING (20) 0A4D55A8D...
  digestEncryptionAlgorithm SEQUENCE (13)
    algorithm OBJECT IDENTIFIER (9) pkcs1-rsaEncryption
    parameters TYPE (2) with NULL (0)
  encryptedDigest OCTET STRING (64) 63103EC00F6001BC760C7996...
  unauthenticatedAttributes [1] IMPLICIT SET OF OPTIONAL NOT PRESENT
```

6.1.3. Els *envelopedData*

L'objecte *envelopedData* consisteix en un sobre digital. Les dades estan xifrades amb una clau simètrica de sessió i aquesta clau està, al seu torn, xifrada amb les claus públiques dels destinataris (el sobre digital pot anar dirigit a un o més usuaris). Les dades que es xifren poden ser de qualsevol tipus; normalment són objectes *data*, *digestedData* o *signedData*.

La informació que porta una estructura de dades *envelopedData* és la següent:

envelopedData	Descripció
version	Versió de l'estàndard que es fa servir.
recipientInfo	Conjunt de la informació sobre els destinataris. N'hi ha d'haver com a mínim un.
Version	Versió de la sintaxi que es fa servir.
issuerAndSerialNumber	Identifica el certificat del destinatari amb el nom de l'autoritat de certificació que l'ha emès i el número de sèrie del certificat.
keyEncryptionAlgorithm	Algorisme de clau pública utilitzat per a xifrar la clau de sessió.
encryptedKey	Clau de sessió xifrada.
encryptedContentInfo	Contingut del missatge xifrat i informació sobre el xifratge: tipus de contingut del missatge original i algorisme de xifratge utilitzat.

Com es veu, la informació per a cada destinatari s'agrupa en una estructura de tipus *recipientInfo*.

Per a crear un sobre digital s'han de seguir els passos següents:

1) Es genera de forma aleatòria una clau de sessió per a un algorisme de xifratge simètric concret.

2) El contingut es xifra amb aquesta clau de sessió.

3) Per a cada destinatari, la clau de sessió es xifra amb la clau pública del destinatari.

4) Per a cada destinatari, la clau de sessió xifrada i alguna informació pròpia del destinatari s'agrupen en un valor del tipus *recipientInfo*.

5) Els valors del tipus *recipientInfo* –un per a cada destinatari– s'ajunten amb el contingut del missatge original per a crear el sobre digital.

El procés que el receptor ha de seguir és:

a) Desxifrar la clau de sessió que va dirigida al sobre digital i obrir-lo.

b) Desxifrar el contingut del missatge amb la clau de sessió obtinguda en el pas anterior.

El xifratge d'una informació amb la tècnica del sobre digital es pot representar amb un PKCS#7 que contingui l'objecte *envelopedData*.

Exemple de l'estructura d'un PKCS#7 *envelopedData*

```
PKCS#7: SEQUENCE (308)
  contentType OBJECT IDENTIFIER (9) pkcs7-envelopedData
  content [0] TYPE (293) with SEQUENCE (289)
     version INTEGER (1) 0x00 (0)
     recipientInfos SET OF (230) SEQUENCE (227)
       version INTEGER (1) 0x00 (0)
       issuerAndSerialNumber SEQUENCE (76)
         issuer SEQUENCE OF (71) OU=Coyote Subordinate CA (MIR), O=UOC, C=ES
         serialNumber INTEGER (1) 0x24 (36)
       keyEncryptionAlgorithm SEQUENCE (13)
         algorithm OBJECT IDENTIFIER (9) pkcs1-rsaEncryption
         parameters TYPE (2) with NULL (0)
       encryptedKey OCTET STRING (128) 1892CA739A7DE62DCC...
     encryptedContentInfo SEQUENCE (51)
       contentType OBJECT IDENTIFIER (9) pkcs7-data
       contentEncryptionAlgorithm SEQUENCE (20)
         algorithm OBJECT IDENTIFIER (8) rsadsi-encryptionAlgorithm-des-ede3-cbc
         parameters TYPE (10) with OCTET STRING (8) A76D20B015138DF0
       encryptedContent [0] IMPLICIT OCTET STRING (16) 7269AFD6A5DD...
```

6.1.4. Els *signedAndEnvelopedData*

L'objecte *signedAndEnvelopedData* consisteix en un sobre digital autenticat, és a dir, la informació està xifrada amb la tècnica del sobre digital i, a més, està signada. Les dades poden estar signades per un nombre qualsevol de signants en paral·lel i, alhora, tot el contingut pot estar xifrat per un o més destinataris.

La informació que porta una estructura de dades *signedAndEnvelopedData* és la següent:

signedAndEnvelopedData	**Descripció**
version	Versió de la sintaxi que es fa servir.
recipientInfos	Conjunt de la informació sobre els destinataris. N'hi ha d'haver com a mínim un.
digestAlgorithms	Conjunt d'identificadors dels algorismes de *hash* que s'han fet servir per a crear les signatures digitals.
encryptedContentInfo	Contingut del missatge xifrat i informació sobre el xifratge: tipus de contingut del missatge original i algorisme de xifratge utilitzat.
certificates	Cadena de certificació des dels certificats utilitzats per la signatura fins als certificats de les autoritats de certificació de nivell més alt.
crls	Conjunt de llistes de certificats revocats.
signerinfo	Conjunt d'informació sobre els signants.

Per a crear un sobre digital autenticat s'han de seguir els passos següents:

1) Es genera una clau de sessió de manera aleatòria amb algun algorisme de xifratge simètric.

2) El contingut del missatge es xifra amb aquesta clau de sessió.

3) Per a cada destinatari, la clau de sessió es xifra amb la clau pública del destinatari.

4) Per a cada destinatari, la clau de sessió xifrada i alguna informació pròpia del destinatari s'agrupa en un valor del tipus *recipientInfo*.

5) Per a cada signant, es calcula el resum del missatge amb l'algorisme que el signant desitgi; si dos signants fan servir el mateix algorisme, només es calcula un cop.

6) Per a cada signant, el resum i altra informació addicional es xifra amb la clau pública pròpia de cada signant, i després es xifra de nou amb la clau de sessió establerta.

7) Per a cada signant, el resum doblement xifrat s'ajunta amb altra informació addicional del signant en una estructura de dades *signerInfo*.

8) Els resums del missatge de tots els signants, les estructures *signerInfo* i *recipientInfo* de tots els destinataris i signants s'ajunten amb el missatge xifrat en una estructura *signedAndEnvelopedData*.

El procés que el receptor ha de seguir per a processar la informació és:

a) Desxifrar la clau de sessió amb la clau privada pròpia.

b) Desxifrar el contingut del missatge, és a dir, la informació dels signants i el propi missatge amb la clau de sessió.

c) Desxifrar els resums, primer amb la clau de sessió i després amb la clau pública de cada signant.

d) Confrontar els resums obtinguts amb els calculats pel mateix destinatari utilitzant el mateix algorisme de *hash*.

e) Llegir el missatge desxifrat.

6.1.5. Els *digestedData*

L'objecte *digestedData* consisteix en qualsevol tipus de contingut i el seu resum corresponent. La finalitat d'aquest objecte és garantir la integritat del contingut i ser l'entrada de processos de creació d'objectes de tipus *envelopedData*.

La informació que porta una estructura de dades *digestedData* és la següent:

digestedData	Descripció
version	Versió de la sintaxi que s'està utilitzant.
digestalgorithmid	Identificador de l'algorisme de *hash.*
contentinfo	Contingut a resumir.
digest	Resum del contingut.

El procés de creació és el següent:

1) Es calcula un resum del contingut amb l'algorisme que es desitgi.

2) El contingut i el resum s'agrupen en una estructura de tipus *digestedData*.

6.1.6. Els *encryptedData*

Aquest objecte consisteix en qualsevol tipus de dades xifrades. No conté cap referència al destinatari (pot ser que no n'hi hagi) ni cap referència a la clau que s'ha fet servir per a xifrar, que s'assumeix que es gestiona des d'altres mitjans. La finalitat és xifrar contingut localment; per això, la clau de xifratge pot ser una simple contrasenya.

La informació que porta una estructura de dades *encryptedData* és la següent:

encryptedData	Descripció
version	Versió de la sintaxi que es fa servir.
encryptedContentInfo	Contingut xifrat.

6.2. El PKCS#10

El PKCS#10 descriu el format d'una petició de certificat. Una **petició de certificat** consisteix en un nom, una clau pública i un conjunt opcional d'atributs referents al subscriptor. Un dels atributs més utilitzats és el d'una contrasenya amb la qual el subscriptor podrà revocar el certificat.

Una petició està dividida en tres parts: la informació de la petició, l'identificador de l'algorisme de signatura i la signatura digital de la informació.

La informació de la petició s'estructura en un camp *certificationRequestInfo*:

certificationRequestInfo	Descripció
version	Versió de la sintaxi que es fa servir.
subject	Nom distingit del propietari de la clau pública.
subjectPublicKeyInfo	Informació de la clau pública.
attributes	Conjunt d'atributs amb informació addicional del subscriptor.

El procés que segueix l'autoritat de certificació en arribar-hi una petició consisteix a verificar la signatura i, en cas que sigui vàlida, construir un certificat X.509 a partir del nom i la clau pública que s'indiquen a la petició. Les altres dades que inclourà el certificat (nom de l'emissor, número de sèrie, període de validesa, identificador de l'algorisme de signatura, etc.) les determinarà l'autoritat de certificació.

Exemple de l'estructura d'un PKCS#10

```
PKCS#10: SEQUENCE (478)
   toBeSigned SEQUENCE (327)
     version INTEGER (1) 0x00 (0)
     subject SEQUENCE OF (132) CN=user, OU=Criptografia, O=UOC, C=ES
     subjectPublicKeyInfo SEQUENCE (159)
       algorithm SEQUENCE (13)
         algorithm OBJECT IDENTIFIER (9) pkcs1-rsaEncryption
         parameters TYPE (2) with NULL (0)
       subjectPublicKey BIT STRING (141) Encapsulates TYPE (140) with
            rSAPublicKey SEQUENCE (137)
               modulus INTEGER (129) 0x00A1BA3CA9B379D12BD8C34F99D01A3AD3418A2...
               exponent INTEGER (3) 0x010001 (65537)
     attributes [0] IMPLICIT SET OF (25) SEQUENCE (23)
       attrType OBJECT IDENTIFIER (9) pkcs9-challengePassword
       attrValue TYPE (12) with SET OF (10) CHOICE (10) PrintableString (8) "contrasenya"
   signatureAlgorithm SEQUENCE (13)
     algorithm OBJECT IDENTIFIER (9) pkcs1-md5WithRSAEncryption
     parameters TYPE (2) with NULL (0)
   signature BIT STRING (129) 00 4936BDAD2A39E99EFC2125EDD99FCF15D3F9...
```

6.3. El PKCS#12

El PKCS#12 descriu una sintaxi per a la transferència d'informació d'identitat personal, que pot incloure claus privades, certificats, ampliacions o altres secrets. Les aplicacions que admeten aquest format permeten l'intercanvi segur de claus entre diversos repositoris.

El PKCS#12 permet la transferència directa d'informació personal sota diversos modes de confidencialitat i integritat. Hi ha quatre combinacions de modes de confidencialitat i d'integritat segons s'utilitzin polítiques de clau pública o de contrasenyes. Els modes privats xifraran la informació personal per a protegir-la de l'exposició pública i els modes d'integritat garanteixen que la informació no sigui falsificada.

Dins dels modes privats trobem:

a) Mode privat de clau pública: la informació personal s'empaqueta i es xifra en la plataforma d'origen amb la clau pública del destí.

b) Mode privat amb contrasenya: la informació personal es xifra amb un nom d'usuari i contrasenya.

Dins dels modes d'integritat hi ha:

a) Mode d'integritat de clau pública: la integritat es garanteix mitjançant la signatura digital del contingut amb la clau privada de la plataforma d'origen. Aquesta forma es verificarà en el destí.

b) Mode d'integritat amb contrasenya: la integritat es garanteix amb un codi d'autenticació del missatge –MAC– derivat de la contrasenya.

L'estàndard permet la utilització dels quatre modes, encara que des del punt de vista de la seguretat, el més aconsellable és fer servir els modes de clau pública, tant per a proporcionar privadesa com per a garantir la integritat. El parell de claus criptogràfiques de la plataforma utilitzades per al transport del PKCS#12 no s'han de confondre amb les claus personals d'usuari que es volen transportar d'un dispositiu a un altre.

D'altra banda, no sempre és possible fer servir els modes de clau pública per al transport segur de dades personals. Es pot donar el cas que en el moment de l'exportació d'unes dades no se'n conegui la plataforma de destinació ni, per tant, la clau pública. En aquests casos utilitzarem els modes de contrasenya.

Ús dels modes de contrasenya

Els navegadors web importen i exporten certificats i claus privades amb un PKCS#12 de mode contrasenya.

La informació del PKCS#12 s'estructura de la manera següent:

PKCS#12	Descripció
version	Versió de la sintaxi que es fa servir.
authSafe	Estructura de dades PKCS#7 que té el format de *signedData* si s'utilitza mode d'integritat de clau pública o *data* si s'utilitza mode d'integritat amb contrasenya.
macData	Camp opcional que hi és present si s'utilitza mode d'integritat amb contrasenya.
mac	Estructura PKCS#7 *digestData*.
macSalt	Llavor aleatòria per a fer el resum.
iterations	Aquest camp no es fa servir; només hi és per raons històriques.

El camp *authSafe* del PKCS#12 conté l'embolcall de la informació que es vol transferir. Està format per estructures de dades PKCS#7 que garanteixen la integritat i contenen, al seu torn, una altra estructura PKCS#7 ennierada amb la informació xifrada de les dades per transportar. Els tipus de dades possibles per al PKCS#7 confidencial són:

- *data*: si no fem servir el xifratge per a amagar les dades del PKCS#12.
- *encryptedData*: si utilitzem el xifratge per contrasenya.
- *envelopedData*: si emprem el xifratge per clau pública.

La informació xifrada es crea a partir de la codificació (*data*), el xifratge (*encryptedData*) o l'empaquetament (*envelopedData*) de les dades en clar disposades en una estructura anomenada *safeContents*. El *safeContents* està format per un conjunt de bosses segures (*safeBags*) que poden portar elements simples d'informació (certificat, llistes de revocació de certificats, clau privada, etc.). Les bosses segures disposen d'atributs opcionals perquè els usuaris puguin assignar un àlies als seus objectes segurs i, d'aquesta manera, reconèixer-los quan la informació s'importa d'un entorn a un altre.

L'estàndard PKCS#12 s'ha implementat tant en programari com en maquinari i és el format més utilitzat per a la transferència de claus privades entre aplicacions.

Exemple de l'estructura d'un PKCS#12

```
PKCS#12: SEQUENCE (4403)
   version INTEGER (1) 0x03 (3)
   authSafes SEQUENCE (4337)
     contentType OBJECT IDENTIFIER (9) pkcs7-data
     content [0] TYPE (4322) with
        OCTET STRING (4318) 308210DA3082035406092A864886F70D010...
   macData SEQUENCE (57)
     mac SEQUENCE (33)
        digestAlgorithm SEQUENCE (9)
           algorithm OBJECT IDENTIFIER (5) secsig-algorithm-sha1
           parameters TYPE (2) with NULL (0)
        digest OCTET STRING (20) 15A597AC8B967C00FFF8F846389C0...
     macSalt OCTET STRING (20) DC6D414C4732EDBD12FC10B4B69B...
     macIterationCount INTEGER (1) 0x01 (1) DEFAULT
```

7. Protocols que utilitzen infraestructura de clau pública

Diversos protocols utilitzen la criptografia de clau pública per a donar serveis de seguretat i, per tant, han de ser capaços de gestionar grans llistes de claus públiques en sistemes distribuïts.

Les infraestructures de clau pública són la plataforma que els serveis de seguretat d'aquests protocols utilitzen per a fer l'intercanvi de claus.

En aquest apartat veurem diferents protocols utilitzats en l'actualitat que protegeixen els intercanvis de dades. Les aproximacions varien segons el nivell en què es vol donar el servei: el nivell d'aplicació (S/MIME, XML), el nivell de transport (TLS/SSL, SSH), el nivell de xarxa (IPsec) o el nivell físic (caixes negres que xifren totes les dades que surten per un enllaç).

7.1. IPsec

El terme *IPsec* (*IP Security Protocol*) es refereix a un conjunt de mecanismes dissenyats per a protegir el trànsit en el nivell de xarxa IP (IPv4 o IPv6).

L'**IPsec** ofereix els serveis de seguretat següents: integritat, autenticació de les dades d'origen, protecció contra reenviaments i confidencialitat (confidencialitat de les dades i protecció parcial contra l'anàlisi de trànsit).

L'IPsec utilitza els mecanismes AH i ESP per a proporcionar els seus serveis de seguretat:

a) AH (***authentication header***) està concebut per a assegurar la integritat i l'autenticació dels datagrames IP sense tenir en compte el xifratge a les dades. Consisteix a afegir un camp addicional als datagrames IP tradicionals de manera que el receptor el pugui utilitzar per a verificar les dades incloses en el paquet.

b) ESP (***encapsulation security payload***) està dissenyat per a assegurar la confidencialitat, però també pot proporcionar autenticitat a les dades. El principi de funcionament de l'ESP és generar un nou datagrama a partir del paquet IP original en el qual les dades, i eventualment la capçalera, estiguin xifrades.

L'**IKE** (***Internet key exchange***) és un protocol desenvolupat específicament per a IPsec amb l'objectiu de proporcionar mecanismes d'intercanvi de claus i autenticació. L'IKE permet l'intercanvi de claus de Diffie-Hellman i utilitza el model de PKI per a identificar i validar les claus criptogràfiques. A més, també s'utilitza la infraestructura de clau pública en els mecanismes d'autenticació que inclouen signatures digitals.

Com que l'IPsec actua en el nivell de xarxa, tota la capa de seguretat que proporciona és completament transparent per a totes les aplicacions que l'utilitzen. Els principals usos d'IPsec són:

- Connexió de dues xarxes privades distants a través d'una xarxa insegura com Internet. En aquest cas, s'estableix una xarxa privada virtual (VPN) entre dues passarel·les segures.

- Protecció de l'accés a Internet per a usuaris externs, com usuaris mòbils. La connexió IPsec es fa entre l'usuari mòbil i la passarel·la d'accés a Internet.

- Connexió extrem a extrem entre dos usuaris que no confien en la xarxa que els separa.

Per cada mecanisme de seguretat d'IPsec, hi ha dos modes de funcionament:

1) En el **mode transport**, només es protegeixen les dades provinents del protocol del nivell superior i transportades per datagrames IP. Aquest mode només el poden utilitzar els equips finals.

2) En el **mode túnel**, la capçalera IP també es protegeix (autenticació, integritat i/o confidencialitat). La capçalera nova es reemplaça altra vegada al final del túnel per l'autoritat de la capçalera original. El mode túnel el fan servir equips finals i passarel·les segures, i permet protegir l'enllaç de l'anàlisi de trànsit.

7.2. SSL

L'SSL (*secure sockets layer*) és un protocol, originalment desenvolupat per Netscape i àmpliament acceptat per la comunitat Internet, que ofereix comunicacions confidencials i autèntiques entre un client i un servidor.

L'SSL corre sobre el nivell de transport TCP/IP i per sota de protocols com HTTP o IMAP. Està format per dos subprotocols:

HTTP

S'anomena HTTPS la utilització d'HTTP sobre SSL.

- **Protocol de registre**: defineix el format utilitzat per a la transmissió de dades.

- **Protocol d'establiment**: defineix l'intercanvi de missatges entre client i servidor perquè estableixin un context segur de comunicació.

El protocol d'establiment facilita les accions següents:

- Autenticació mútua de client i/o servidor.
- Selecció dels algorismes criptogràfics i xifres que tant client com servidor suporten.
- Generació de secrets compartits.
- Establiment d'una connexió SSL xifrada.

L'SSL utilitza la infraestructura de clau pública fonamentalment per a autenticar les entitats durant l'establiment de la connexió. Tant el client com el servidor disposen d'un parell de claus criptogràfiques que utilitzen per a signar unes dades aleatòries. Les dades, la signatura i el certificat corresponent a la clau privada utilitzada (i, opcionalment, tota la cadena de certificats fins al certificat de l'autoritat de certificació arrel) s'envien a l'altra entitat perquè en verifiqui la signatura. La verificació de la signatura ha de tenir en compte que la cadena de certificats sigui vàlida i que cap d'aquests certificats estigui contingut en una llista de revocació. Si la verificació és correcta, l'entitat pot assegurar la identitat de l'interlocutor.

7.3. S/MIME

L'S/MIME (*secure multipurpose Internet mail extensions*) és una especificació de correu electrònic segur dissenyada per a afegir serveis d'autenticació, integritat, no-repudi d'origen i confidencialitat als missatges electrònics basats en el format estàndard d'Internet MIME.

L'S/MIME adopta les normes i formats definits en el PKCS#7, de manera que la seva integració en les aplicacions de correu electrònic és relativament senzilla. Les dades que es volen protegir tenen l'estructura d'una entitat MIME; els objectes segurs PKCS que resulten després del procés criptogràfic s'incrusten en el missatge de correu electrònic per mitjà d'un embolcall MIME. S/MIME ofereix formats per a sobres digitals, signatures digitals i sobres digitals autenticats.

7.4. Seguretat en XML

L'XML (*extensible mark-up language*) és una especificació desenvolupada pel World Wide Web Consortium (W3C) que defineix una sintaxi i unes regles sobre l'ús d'etiquetes per a estructurar la informació.

Els **estàndards de seguretat XML** defineixen el vocabulari i les regles de processament necessàries per a oferir serveis de seguretat a les noves tecnologies que utilitzen aquest llenguatge.

Les característiques dels estàndards XML són:

- Estan dissenyats per a oferir la flexibilitat i ampliabilitat pròpies d'XML. Permeten que la seguretat s'apliqui a documents XML, elements XML, només al contingut d'un element, o bé en documents binaris arbitraris.

- La seguretat s'aplica extrem a extrem, ja que està associada al contingut i no pas al canal de transport.

- Es reutilitzen les tecnologies de seguretat existents a tot arreu on sigui possible. Per exemple, s'utilitzen certificats X.509 que, en aquest cas, es codifiquen en format text.

Els estàndards bàsics de la seguretat XML són els següents:

WEB

Per a obtenir més informació sobre els estàndards XML, podeu consultar el portal de World Wide Web Consortium:

http://www.w3c.org

- **XML Signature** (**XMLDsig**): defineix mecanismes per a suportar la creació i verificació de signatures digitals sobre documents XML.

- **XML Encryption** (**XMLEnc**): ofereix solucions de confidencialitat. És una especificació anàloga a l'objecte *envelopedData* del PKCS#7, amb la particularitat que la confidencialitat es pot aplicar amb gran nivell de granularitat sobre el document XML. Aquesta característica permet xifrar porcions de missatges de protocols XML que han de ser encaminats per processadors intermedis.

- **XML Key Management** (**XKMS**): defineix protocols per a la gestió de claus públiques (registre, revocació, localització i validació). Permet que els usuaris administrin els seus certificats d'infraestructura de clau pública sense que hagin de conèixer-ne la tecnologia.

- **Security Assertion Markup Language** (**SAML**): especifica un vocabulari per a compartir asseveracions d'autenticació i autorització entre diferents entitats independents i, d'aquesta manera, permet els serveis d'enregistrament únic (*single sign-on*, SSO).

- **XML Access Control Markup Language** (**XACML**): defineix un vocabulari per a expressar condicions i regles d'autorització.

Resum

En aquest mòdul hem descrit el funcionament d'una infraestructura de clau pública i hem vist les estructures de dades i els protocols que utilitzen aquesta tecnologia. Els aspectes que hem tractat són els següents:

1) Hem estudiat els components que formen una infraestructura de clau pública per tal d'entendre quins serveis ens ofereix.

2) Hem fet referència als models de confiança i hem vist com s'establien les relacions de confiança entre entitats d'un mateix grup i entre diferents organitzacions.

3) Hem descrit els processos associats al cicle de vida de les claus criptogràfiques i dels certificats.

4) Hem detallat les característiques de l'element principal de la infraestructura de clau pública: el certificat digital. Hem vist com estan estructurades les dades del certificat i com es pot ampliar la informació que s'hi conté per tal d'adaptar-lo a futures necessitats.

5) Hem explicat el format dels missatges PKCS que porten informació associada a serveis de seguretat.

6) Finalment, hem fet referència a protocols de xarxa, transport i presentació que utilitzen la infraestructura de clau pública per a donar serveis de seguretat.

Activitats

1. FESTE és una autoritat de certificació espanyola en què participen, entre altres, notaris i corredors de comerç. Busqueu quines són les polítiques de certificació (CP) dels certificats que emet i quina és la declaració de pràctiques de certificació (CPS).

2. Visiteu la pàgina web http://www.openvalidation.org i proveu la funcionalitat dels serveis d'OCSP amb el vostre navegador.

3. Ceres és una autoritat de certificació que ofereix els seus serveis perquè l'FNMT (Fàbrica Nacional de Moneda i Timbre) pugui proporcionar certificats digitals a tots els ciutadans espanyols per a la resolució via Internet d'algunes gestions amb l'Agència Tributària (per exemple, la declaració de renda). A partir de la seva pàgina principal, http://www.cert.fnmt.es, seguiu tots els passos necessaris per a l'obtenció d'un certificat digital.

4. Segons hem vist, els navegadors web fan servir un model per llista de confiança. Cerqueu dins els navegadors que utilitzeu habitualment on està aquesta llista i en quines autoritats de certificació arrel confieu quan navegueu per Internet. Mireu si és possible afegir o eliminar autoritats de certificació d'aquesta llista.

Vegeu el model per llista de confiança en el subapartat 3.4 d'aquest mòdul.

Exercicis d'autoavaluació

1. Cal comprovar les llistes de revocació de certificats per a saber si un certificat ha caducat?

2. Volem recuperar la clau de signatura d'un usuari de la nostra empresa que s'ha jubilat. És possible que la CPS ho permeti?

3. Indiqueu els passos que cal seguir per a verificar la signatura digital d'una estructura PKCS#7.

4. Els PKCS són estructures de dades que porten informació criptogràfica. Feu una taula que indiqui en quins PKCS podeu trobar una clau pública, una clau privada o una clau simètrica.

5. El perfil de certificat X.509 especificat per PKIX defineix una ampliació *extKeyUsage* per a indicar les aplicacions a les quals es dirigeix l'ús del certificat. Llisteu el valor que aquesta ampliació pot prendre i la relació que té amb l'ampliació *keyUsage*.

6. El certificat d'una autoritat de certificació expira al final del 2005. Aquesta autoritat de certificació emet certificats de signatura digital amb una validesa de dos anys. Fins a quina data podrà emetre certificats? Fins quan ha de continuar donant serveis l'autoritat de certificació?

7. Quines operacions es poden fer amb un parell de claus criptogràfiques quan el certificat de xifratge corresponent ha expirat?

8. Llisteu totes les autoritats de certificació possibles per les quals una aplicació pugui considerar invàlid un certificat.

Solucionari

1. La informació sobre la data d'expiració d'un certificat no cal buscar-la en les llistes de revocació de certificats, ja que es pot trobar en el mateix certificat, dins del camp *validity*.

2. Les CPS no permeten la recuperació de claus de signatura per dos motius. D'una banda, no té sentit tenir-ne còpies de seguretat, ja que la pèrdua d'una clau de signatura no suposa cap inconvenient per a interactuar amb dades signades amb anterioritat. D'altra banda, el fet de tenir una còpia de la clau privada impediria que es pogués utilitzar en serveis de no-repudi.

3. Els passos que cal seguir per a verificar la signatura d'un PKCS#7 són:

a) Desxifrar els resums de cada signant amb la clau pública d'aquest.

b) Comparar que els resums coincideixen amb els que s'obtenen en fer els càlculs amb les dades que s'autentiquen.

4. PKCS en què es pot trobar una clau pública, una clau privada o una clau simètrica:

PKCS	Clau pública	Clau privada	Clau simètrica
7	x		x
8		x	
10	x		
12	x	x	

5. Els usos definits en el perfil de PKIX per a l'ampliació *extKeyUsage* són els següents:

extKeyUsage	**Descripció**	**Valors de *keyUsage***
serverAuth Autenticació de servidor	Es fa servir per a l'autenticació del servidor en TLS	*digitalSignature*, *keyEncipherment* o *keyAgreement*
clientAuth Autenticació de client	Es fa servir per a l'autenticació del client en TLS	*digitalSignature* i/o k*eyAgreement*
codeSigning Signatura de codi	Es fa servir per a signar codi descarregable. És necessària per a utilitzar *authenticode* (ActiveX, *applets* de Java, *javascripts*, etc.)	*digitalSignature*
emailProtection Protecció de correu electrònic	Permet l'ús per a protegir missatges de correu electrònic	*digitalSignature*, *nonRepudation* i/o *keyEncipherment* o *keyAgreement*.
timeStamping Segellat de temps	Permet l'ús per a enllaçar el *hash* d'un objecte a un temps provinent d'una font de temps acordada.	*digitalSignature* i *nonRepudiation*

6. Els certificats que emet una autoritat de certificació només tenen validesa dins el període de validesa del mateix certificat de l'autoritat de certificació. Com que l'autoritat de certificació expira al final del 2005 i emet certificats amb una validesa de dos anys, l'autoritat de certificació haurà de deixar d'expedir certificats al final del 2003.

Els certificats de l'autoritat de certificació són vàlids fins al final del 2005, i fins a aquesta data l'autoritat de certificació haurà de continuar oferint serveis als seus clients, si bé no per a l'expedició de nous certificats, sinó per tal de poder tractar casos com la revocació de certificats emesos.

7. Si el certificat de xifratge expira, la clau pública no es pot utilitzar per a xifrar més dades. En aquest cas, el període d'expiració del certificat limita la vida útil de xifratge de la clau pública. D'altra banda, la clau privada sí que es podrà fer servir més enllà de la data de caducitat del certificat: el titular de la clau la podrà utilitzar per a desxifrar informació xifrada quan el certificat encara era vàlid.

8. Les condicions que poden fer que un certificat es consideri invàlid (i per tant, que caldria comprovar si féssim una aplicació basada en certificats) són:

- El certificat ha caducat.
- La signatura digital del certificat és criptogràficament invàlida.
- L'autoritat de certificació que ha generat el certificat no és de la nostra confiança.
- El certificat ha estat revocat.
- Hi ha una ampliació crítica no reconeguda per l'aplicació.

Glossari

autoritat de certificació *f* Autoritat de confiança que emet certificats.
sigla: CA

autoritat de registre *f* Entitat que s'encarrega de les tasques més administratives de l'autoritat de certificació, com ara la confirmació de la identitat dels usuaris, la verificació que l'usuari posseeix la clau privada associada a la clau pública que es vol certificar, la publicació dels certificats en directoris, etc.
sigla: RA

autoritat de segellat de temps *f* Autoritat que proporciona una prova d'existència de certes dades abans d'un instant de temps determinat.
sigla: TSA

autoritat de validació *f* Autoritat que proporciona informació sobre l'estat de revocació dels certificats.
sigla: VA

CA *f Vegeu* autoritat de certificació.

certificat digital *m* Estructura de dades que vincula una identitat amb una clau pública. El certificat és emès per una autoritat de certificació.

CP *f Vegeu* política de certificats.

CPS *f Vegeu* declaració de les pràctiques de certificació.

CRL *f Vegeu* llista de revocació de certificats.

declaració de les pràctiques de certificació *f* Declaració de les pràctiques que una autoritat de certificació utilitza per a emetre certificats.
sigla: CPS

entitat final *f* Propietari d'un certificat que no és una autoritat de certificació.

LDAP *m* Acrònim de *light weight directory access protocol*. Protocol d'accés a un directori.

llista de revocació de certificats *f* Llista emesa per les autoritats de certificació en la que es publiquen tots aquells certificats que han estat revocats.
sigla: CRL

OCSP *m* Acrònim d'*online certificate status protocol*. Protocol per a determinar l'estat de revocació d'un certificat.

PKCS *m* Acrònim de *public-key cryptography standards*. Conjunt d'especificacions desenvolupades en els laboratoris RSA per tal d'establir una norma comuna a la indústria sobre els formats de les dades utilitzades en criptografia de clau pública.

PKI *f* Acrònim de *public key infrastructure*. Conjunt de maquinari, programari, persones, polítiques i procediments necessaris per a crear i gestionar certificats digitals basats en criptografia de clau pública.

política de certificats *f* Conjunt de normes que indiquen l'aplicabilitat d'un certificat a una aplicació o comunitat.
sigla: *CP*

RA *f Vegeu* autoritat de registre.

SCVP m Acrònim de *simple certificate validation protocol*. Protocol per a determinar la ruta de certificació d'un certificat i el seu l'estat de revocació.

TSA *f Vegeu* autoritat de segellat de temps.

VA *f Vegeu* autoritat de validació.

XML *m* Acrònim de l'expressió anglesa *extensible mark-up language*. Especificació que defineix una sintaxi i unes regles sobre l'ús d'etiquetes per a estructurar la informació.

Bibliografia

Nash, Andrew; Duancem, Joseph; Brink, James E. (2001). *PKI implementing and managing e-security*. Nova York (etc.): Osborne / McGraw-Hill.

Adams, Carlisle; Lloyd, Steve (2003). *Understanding PKI* (2a. ed.). Boston: Addison-Wesley.

IETF-PKIX Working Group. Public-Key Infrastructure (X.509).
<http://www.ietf.org/html.charters/pkix-charter.html>

RSA Security. "PKCS: Public Key Criptography Standards".
<http://www.rsasecurity.com/rsalabs/pkcs/>

World Wide Web Consortium (W3C). "XML encryption syntax and processing".
<http://www.w3.org>

Aplicacions de la criptografia

Josep Domingo Ferrer
Jordi Herrera Joancomartí

P03/05024/02266

UOC
www.uoc.edu

Índex

Introducció

Sense voler ser exhaustius, en aquest mòdul didàctic presentem les següents aplicacions de la criptografia:

- **L'autentificació i identificació**. Mitjançant protocols criptogràfics, és possible autentificar i identificar dues parts que es volen comunicar electrònicament.

- **La compartició de secrets**. Passem revista a les solucions criptogràfiques del problema de la compartició de secrets. La idea és dificultar l'extorsió fent possible que els secrets realment importants (codis d'activació de míssils, d'obertura de caixes fortes, etc.) no estiguin a les mans d'una sola persona o entitat, sinó que en calguin diverses per a recuperar-los.

- **Les situacions de desconfiança mútua**. Hi ha situacions de desconfiança mútua que plantegen problemes de seguretat no trivials quan es duen a terme de manera virtual. Per exemple, la signatura electrònica de contractes, el correu electrònic certificat o el càlcul segur a múltiples bandes. La criptografia proporciona solucions perquè tots els participants tinguin les garanties necessàries.

- **Els pagaments sense rastre**. Les signatures tapades són uns protocols de signatura digital que permeten al comprador efectuar pagaments per Internet sense que la botiga en sàpiga la identitat ni el banc en quin concepte ha despès els diners.

- **Els drets intransferibles**. Amb el protocol criptogràfic adequat, una autoritat pot donar drets a un client per utilitzar uns determinats serveis i alhora assegurar-se que no podrà transferir aquests drets a tercers sense el concurs de l'autoritat.

- **Les votacions electròniques**. Efectuar una votació mitjançant una xarxa com Internet (sense acudir físicament a un col·legi electoral) és aparentment difícil de conjugar amb els requisits d'anonimat i de secret de vot. Revisarem algunes solucions criptogràfiques que s'han proposat.

Objectius

Els materials didàctics d'aquest mòdul han de permetre que l'estudiant assoleixi els objectius següents:

1. Entendre que les aplicacions de la criptografia no es limiten a proporcionar secret i autenticitat.

2. Conèixer els esquemes de compartició de secrets.

3. Aprendre com es poden resoldre de manera segura situacions de desconfiança mútua en les quals els participants es comuniquen per mitjà d'una xarxa com Internet.

4. Ser conscients del problema que hi ha amb el rastre en els pagaments electrònics i adonar-se que això té solució criptogràfica.

5. Entendre com es poden concedir drets intransferibles per utilitzar uns determinats. serveis

6. Comprendre els requisits i el funcionament dels sistemes de votació electrònica.

1. Autentificació i identificació

1.1. El protocol de tres passos de Shamir

El protocol de tres passos de Shamir va ser inventat per A. Shamir, tot i que no va arribar a ser publicat mai.

El **protocol de tres passos de Shamir** permet establir una comunicació segura entre dues persones* sense cap intercanvi de claus previ, ni públiques ni privades.

*** Per exemple, l'Anna i en Bernat.**

Per tal de poder dur a terme el protocol, cal tenir una funció de xifratge simètrica; és a dir, és el mateix xifrar un missatge amb una clau A i el resultat tornar-lo a xifrar amb una clau B, que xifrar el missatge primer amb la clau B i el resultat xifrar-lo amb la A:

$$E_A(E_B(M)) = E_B(E_A(M)).$$

Passem a descriure el protocol de tres passos de Shamir en el qual l'Anna (amb la seva clau secreta A) vol fer arribar a en Bernat (que té una clau secreta B) el missatge M:

1) L'Anna xifra el missatge M amb la seva clau secreta A, i el resultat que s'obté, $C_1 = E_A(M)$, l'envia a en Bernat.

2) En Bernat xifra amb la seva clau secreta B el missatge que ha rebut de l'Anna, és a dir, $C_2 = E_B(C_1) = E_B(E_A(M))$, i retorna aquest resultat a l'Anna.

3) L'Anna utilitza la seva clau secreta A i la commutativitat del xifratge per a obtenir $E_B(M)$ de la manera següent:

$$D_A(E_B(E_A(M))) = D_A(E_A(E_B(M))) = E_B(M),$$

on $D_A(\ )$ indica la funció de desxifratge. Ara l'Anna envia el resultat obtingut, $C_3 = E_B(M)$, a en Bernat.

4) Finalment, en Bernat només ha d'utilitzar la seva clau secreta B per tal d'obtenir M fent $M = D_B(E_B(M))$.

Un cop hem vist com funciona el protocol de tres passos de Shamir és lògic que ens preguntem quins criptosistemes poden ser útils per tal d'implemen-

tar-lo. Amb això volem dir no solament quins sistemes de xifratge conserven la commutativitat amb les claus, sinó quins ens ofereixen una seguretat elevada per a aplicar aquest protocol amb garanties:

a) Per exemple, si ens fixem en el xifratge de Vernam, que seria el que ens aportaria la seguretat incondicional, observem que la commutativitat es compleix, ja que en aquest cas tant el xifratge com el desxifratge és la suma XOR amb la clau, i per tant:

Vegeu el xifratge de Vernam en el subapartat 2.1 del mòdul "Introducció a la criptografia" d'aquesta assignatura.

$$E_A(E_B(M)) = (M \oplus B) \oplus A = (M \oplus A) \oplus B = E_B(E_A(M)).$$

Noteu que la suma XOR també és commutativa.

Així, el protocol de tres passos de Shamir entre A i B fent servir el xifratge de Vernam genera els missatges xifrats següents:

- $C_1 = M \oplus A$,
- $C_2 = M \oplus A \oplus B$,
- $C_3 = M \oplus B$.

El problema en aquest cas no és la commutativitat del criptosistema (que sí que en té) sinó la seguretat del protocol. En efecte, un atacant en té prou de prendre nota dels tres missatges xifrats que s'intercanvien l'Anna i en Bernat, 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 com la que hi ha a continuació:

$$C_1 \oplus C_2 \oplus C_3 = (M \oplus A) \oplus (M \oplus A \oplus B) \oplus (M \oplus B) = M.$$

b) El mateix Shamir va proposar un altre esquema commutatiu molt més segur. Aquest sistema és semblant al criptosistema RSA i això implica que la seva seguretat recau en la dificultat del càlcul del logaritme discret. El criptosistema es basa en el fet de prendre un nombre primer p tal que $p - 1$ té un factor primer gran. La clau de xifratge serà un valor e tal que $\text{mcd}(p - 1, e) = 1$. La clau de desxifratge d es calcula a partir de la de xifratge, de tal manera que $de \equiv 1 \bmod (p - 1)$. Per tant, per xifrar un missatge M farem $C = M^e \bmod p$ i per desxifrar-lo, $M = C^d \bmod p$. Si apliquem aquest criptosistema al protocol descrit, veurem que obtenim la seguretat desitjada.

El criptosistema RSA s'explica al subapartat 4.1 del mòdul "Xifres de clau pública" d'aquesta assignatura.

Una vegada que s'ha presentat aquest sistema d'intercanvi d'informació, cal no oblidar, però, un detall clau quan s'intenta obtenir una comunicació segura entre dos interlocutors. Si ens fixem en el protocol, veiem que el missatge M sempre circula de forma xifrada entre A i B; per tant, *a priori*, caldria pensar que la seguretat del missatge està garantida. El problema amb què ens trobem és el mateix que en el cas de la clau pública. Com sabem que, quan comença la comunicació, l'Anna està realment parlant amb en Bernat i no pas amb algú altre?

Com ja hem destacat en altres mòduls, aquest problema es refereix a la identificació dels interlocutors. Un dels protocols d'identificació a què es pot recórrer és el de Fiat-Shamir.

Vegeu l'apartat 3 del mòdul didàctic "Fonaments de criptografia" d'aquesta assignatura.

1.2. El protocol d'identificació de Fiat-Shamir

El protocol d'identificació de Fiat-Shamir va ser creat per A. Fiat i A. Shamir l'any 1986. Un any més tard els mateixos autors, amb la incorporació de Feige, van modificar el protocol creant una de les proves de coneixement nul d'identificació més conegudes.

Vegeu el concepte de proves de coneixement nul al subapartat 1.3 d'aquest mòdul didàctic.

El **protocol d'identificació de Fiat-Shamir** requereix una autoritat certificadora en la qual s'han de registrar tots els usuaris que vulguin poder ser identificats pel sistema. L'avantatge més gran que presenta aquest esquema és que la interacció de l'usuari amb el centre de certificació només és necessària una sola vegada.

Suposem que en Pep (P) es vol identificar a en Vicenç (V). La primera cosa que ha de fer en Pep és registrar-se a l'autoritat certificadora per tal d'obtenir les seves claus (pública i privada). Per a fer-ho executa el protocol següent:

1) L'autoritat certificadora tria un $n = pq$, on p i q són dos primers secrets.

2) La clau pública de P és v, un residu quadràtic mòdul n. És a dir, existeix una x tal que $x^2 \equiv v \bmod n$ té solució i $v^{-1} \bmod n$ existeix.

3) La clau privada és la s més petita tal que $s \equiv \sqrt{v^{-1}} \bmod n$.

Una vegada en Pep ja ha obtingut les seves claus públiques i privades, es podrà identificar a en Vicenç per mitjà del protocol següent:

1) P tria un valor aleatori $r < n$, calcula $x = r^2 \bmod n$ i envia el resultat x a V.

2) V envia un bit aleatori b a P.

3) P envia a V:

$$\begin{cases} r; & \text{si } b = 0. \\ y = rs \bmod n; & \text{si } b = 1. \end{cases}$$

4) Si $b = 0$, V valida que $x = r^2 \bmod n$, i per tant sap que P coneix l'arrel quadrada de x. Si $b = 1$, V valida que $y^2 = xv^{-1} \bmod n$, i per tant sap que P coneix $\sqrt{v^{-1}}$.

Això és només una iteració del protocol, el que s'anomena una *acreditació*. Si s'executa t vegades la probabilitat que P enganyi V és $1/2^t$.

Una modificació d'aquest protocol és la que presentem tot seguit. Les millores van encaminades, d'una banda, a aconseguir més d'una acreditació en cada iteració del protocol, i, de l'altra, a incloure en la informació que s'intercanvien la identitat de l'usuari que es vol identificar.

El protocol modificat funciona de la manera que especifiquem tot seguit:

1) Primer hi ha la part de certificació a l'autoritat:

a) L'autoritat certificadora tria un $n = pq$, on p i q són dos primers secrets, i una funció *hash* $f(\ )$ que pren valors a $\mathbb{Z}_n$.

b) L'autoritat certificadora crea un identificador I per a cada usuari. Aquest identificador portarà incorporada la identitat de l'individu.

c) L'autoritat certificadora calcula k valors enters:

$$v_i = f(I,i) \text{ per a } i = 1, ..., k.$$

d) L'autoritat certificadora tria $t \leq k$ índexs i_j tals que els corresponents v_{i_j} són residus quadràtics mòdul n. Per cada un d'aquests calcula l'arrel quadrada més petita, és a dir, $s_{i_j} \equiv \sqrt{v_{i_j}^{-1}} \bmod n$ per a $i = 1, ..., t$.

e) L'autoritat certificadora envia a l'usuari l'identificador I, els valors s_{i_j} i els índexs corresponents i_j per a $j = 1, ..., t$.

2) Una vegada fet el protocol de certificació, l'autorització es basa en el fet que en Pep amb identificador I demostri a en Vicenç que coneix els valors s_{i_j} per a $j = 1, ..., t$ sense mostrar-los-hi. Vegem com es fa:

a) P envia el valor del seu identificador I i els índexs i_j per a $j = 1, ..., t$.

b) V calcula els valors v_{i_j} a partir de la funció *hash* f i de l'identificador I, tal com l'autoritat certificadora ha fet: $v_{i_j} = f(I,i_j)$ per a $j = 1, ..., t$.

c) P tria un valor aleatori $r < n$, calcula el valor $x \equiv r^2 \bmod n$ i envia el resultat a V.

d) V tria t bits aleatòriament $(b_1, ..., b_t)$ i els envia a P.

e) P retorna a V el valor següent:

$$y \equiv r \prod_{b_j=1} s_{i_j} \bmod n.$$

f) *V* comprova que es compleixi la igualtat següent:

$$x \equiv y^2 \prod_{b_j=1} v_{i_j} \bmod n.$$

En aquest protocol també cal repetir *h* vegades els darrers quatre passos (del **c** al **f**) per tal d'obtenir més seguretat; tot i que, a diferència del cas anterior, aquí una sola iteració ja redueix a $1/2^t$ la probabilitat d'engany amb èxit. Per tant, si repetim el procés *h* vegades la probabilitat esdevé $1/2^{th}$.

Aquest tipus de protocol que acabem de descriure s'engloba dins del que es coneixen com a *proves de coneixement nul*.

1.3. Proves de coneixement nul

Com podem saber que algú altre sap alguna cosa sense saber el que sap? Això que pot semblar un joc de paraules s'ha convertit en un dels temes de recerca en criptografia. El problema es troba en el fet de demostrar que sabem un secret sense dir-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.

* En anglès, *zero-knowledge proofs*.

En els protocols de les proves de coneixement nul, hi intervenen dues parts; en el nostre cas, en Pep (*P*), que és el **provador**, el que vol demostrar que sap el secret, i en Vicenç (*V*), el **verificador**, que s'ha de convèncer que el provador coneix el secret.

Podem donar dues característiques principals del que ha de complir una prova de coneixement nul:

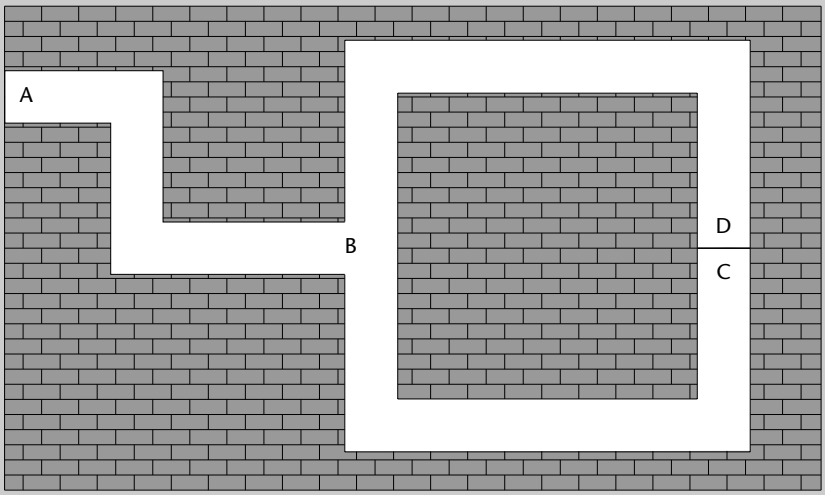
1) La probabilitat que el provador enganyi el verificador ha de ser molt petita. És a dir, si el provador no coneix el secret que diu que coneix, la probabilitat que la prova de coneixement nul s'executi correctament és molt petita.

2) Un cop s'ha fet la prova de coneixement nul, el verificador no té cap informació sobre el secret que el provador coneix. En particular, el verificador no pot provar a una tercera persona, ni per mitjà d'una prova de coneixement nul, que coneix el secret.

Per tal d'aclarir les coses observem l'exemple següent proposat per J.J. Quisquater i L. Guillou, i que ajuda a entendre el procés bàsic d'una prova de coneixement nul. Suposem que tenim una cova, tal com mostra la figura de la pàgina següent.

Com s'hi pot veure, la cova consta d'una sola entrada, la qual es bifurca més endavant en dos camins que fan la volta i es tornen a trobar. La unió dels dos camins està separada per una porta que només s'obre amb una paraula secreta.

Gràfic de la cova



En Pep coneix la paraula clau que obre la porta i vol que en Vicenç es convenci que la sap sense haver-la-hi de dir; és a dir, en Pep vol provar que sap la clau (serà el provador) i en Vicenç ho ha de verificar (serà el verificador). Així, decideixen fer el protocol següent:

1) En Vicenç es queda a l'entrada de la cova (punt A de la figura), mentre que en Pep entra a dins i tria un dels dos camins fins a arribar a la porta; és a dir, serà al punt C o D segons la tria que hagi fet.

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

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

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.

Però, què passaria si en Pep no conegués la clau que obre la porta? La primera vegada que féssim 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 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 una probabilitat 0,999023 que en Pep sap la paraula que obre la porta només cal que facin la prova deu vegades.

L'exemple de la cova...

... il·lustra com funciona una prova de coneixement nul, tot i que òbviament per al nostre propòsit n'hi hauria prou de fer entrar en Pep per la dreta i fer-lo sortir per l'esquerra.

En general, les proves de coneixement nul funcionen d'aquesta manera, és a dir, són iteratives, de manera que a cada iteració hi ha una probabilitat del 50% d'encertar.

A més, aquest tipus de protocols utilitzen la tècnica anomenada *challenge & response*, en la qual el verificador dóna al provador una informació que ell ha generat de manera aleatòria, per tal que el provador la completi utilitzant el secret que coneix. Aquesta tècnica també s'anomena sovint *cut & choose*, ja que fa referència al típic protocol de repartir un pastís entre dues persones, en el qual una fa les parts (talla) i l'altra les tria.

Des d'un punt de vista més formal, les proves de coneixement nul es basen en la transformació del problema, del qual volem demostrar que sabem la solució, en un altre d'isomorf. Aleshores, es resol aquest nou problema utilitzant la solució de l'anterior.

Exemple de prova de coneixement nul

A continuació veiem un exemple concret d'una prova de coneixement nul. Aquesta prova va ser proposada per Chaum, Evertse i van de Graaf i demostra el coneixement del logaritme discret sense necessitat de revelar-lo. Òbviament, la prova té sentit tenint en compte que el càlcul del logaritme discret és difícil. El seu funcionament és el que exposem tot seguit.

En primer lloc el sistema està format per tres paràmetres públics (p, g, b): p és un nombre primer gran, b és un nombre enter, tal que $b < p$, i g és un enter $1 < g$ generador del grup multiplicatiu $\mathbb{Z}_p$. Recordem que P vol provar a V que coneix x amb $1 < x < p - 1$ tal que és solució de l'equació $b \equiv g^x \bmod p$. El protocol que efectuen és el següent:

1) P tria un valor aleatori r, on $0 < r < p$, i envia h a V, on $h \equiv g^r \bmod p$.

2) V tria un bit c, tal que $c \in \{0, 1\}$ i l'envia a P.

3) P calcula $y = r + cx \bmod p - 1$ i l'envia a V.

4) V comprova que $g^y \equiv hb^c \bmod p$.

El protocol consisteix en la repetició n vegades dels passos descrits anteriorment.

Igual que en l'exemple de la cova, en aquest cas, com que P no coneix quin bit triarà V en el segon pas, la probabilitat que té d'enganyar V és d'1/2. Per tant, V pot determinar el grau de credibilitat del protocol fixant el nombre (elevat) de vegades en què s'han de repetir els passos i aconseguir que la probabilitat d'engany sigui $1/2^n$.

No oblidem que les proves de coneixement nul són difícils d'aconseguir, ja que moltes vegades, en cada iteració, es va revelant informació del secret. Aquesta informació, encara que és del tot insuficient per a obtenir el secret, és obtinguda pel verificador. És per aquesta raó que en molts casos en comptes de proves de coneixement nul es parla de **proves de revelació mínima***.

* En anglès, *minimum disclosure proofs.*

Pel que fa a l'aplicació de les proves de coneixement nul, un dels camps en els quals tenen més importància és en el de l'autentificació. El tradicional sis-

tema del número d'identificació personal (PIN) comença a ser insuficient per a certes aplicacions, ja que tant si es guarda el PIN en clar com si es guarda com a imatge d'una funció *hash* en algun moment l'usuari l'ha d'introduir en clar i és aleshores quan pot ser interceptat.

El protocol d'identificació de Fiat-Shamir és en essència una prova de coneixement nul, ja que acompleix els requisits que hem descrit en aquest subapartat. En el futur, la implantació de sistemes de targetes intel·ligents, capaces d'executar les instruccions i els protocols, farà que les proves de coneixement nul siguin utilitzades àmpliament per als processos d'identificació.

Recordeu que hem descrit el protocol d'identificació de Fiat-Shamir al subapartat 1.2 d'aquest mòdul didàctic.

2. Esquemes de compartició de secrets

Hi ha situacions en què un secret no pot ser guardat per una sola persona, perquè el risc que comporta és massa elevat. Una situació quotidiana seria el problema de guardar una clau secreta. Si la guardem en un lloc determinat, i en aquest lloc passa algun incident la podem perdre. Per a solucionar això podem guardar la mateixa clau en llocs diferents, però això implica una reducció de la seguretat, ja que les probabilitats que algú la trobi són més elevades. Finalment, el que sembla més segur és partir la clau en diferents trossos de manera que només amb uns quants pugui ser recuperada. D'aquesta manera, si se'n perdessin alguns encara es podria recuperar la clau.

Codis secrets

És clar que els codis que governen el llançament de míssils nuclears no poden dependre d'un sol individu, ja que si aquest patís algun trastorn... El que es fa és repartir el codi secret entre diferents persones per tal que sigui necessari l'acord d'un determinat nombre d'aquestes per a poder obtenir el codi total.

Els sistemes que s'utilitzen per a resoldre els problemes que acabem de presentar s'anomenen **esquemes de compartició de secrets**. Un tipus habitual de compartició de secrets és aquell en què es pretén repartir un secret entre n participants, de tal manera que si m qualssevol participants ajunten els seus fragments poden recuperar el secret, tenint en compte que qualsevol agrupació de menys de m fragments no obté cap informació del secret. Aquest tipus d'esquemes s'anomenen **esquemes de llindar** i es representen per esquemes de llindar (m,n).

Així, doncs, un esquema de compartició de secrets de llindar (m,n) està format per n usuaris $u_1, ..., u_n$. Cada usuari té el seu fragment corresponent s_i del secret S. A més, cal tenir en compte el següent:

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

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

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

Hi ha diferents maneres d'obtenir esquemes de compartició de secrets de tipus llindar:

1) Els anomenats **esquemes vectorials** van ser introduïts per G.R. Blakley i es basen en el següent: suposem que volem crear un esquema de compartició de secrets vectorial de llindar (m,n). El que farem és definir el secret que volem compartir com un punt P de l'espai m-dimensional. Els fragments del secret seran hiperplans de dimensió $m - 1$. Aquests hiperplans els construirem de tal manera que la intersecció de m qualssevol doni el punt P. D'aquesta mane-

ra, si reunim m hiperplans podrem trobar el punt intersecció de tots aquests i, per tant, reconstruir el secret.

Aplicació de l'esquema vectorial

Suposem que la combinació d'una caixa forta la formen tres nombres: 12, 13 i 14. A l'entitat bancària hi ha cinc delegats que comparteixen la clau i només en calen tres per a obrir la caixa. A cada delegat se li dóna una de les equacions següents que corresponen a plans de $\mathbb{R}^3$:

- $-x + y - z = -13$.
- $2x - y + z = 25$.
- $x - z = -2$.
- $x + 2y - 2z = 10$.
- $x + y - z = 11$.

És fàcil veure que si calculem el punt d'intersecció de qualsevol dels tres plans ens dóna el valor (12,13,14), que és el nombre secret.

2) Un altre esquema per compartir secrets és el proposat per A. Shamir i es basa en la **interpolació polinòmica**. Suposem que volem un esquema de llindar (m,n) per poder repartir el secret S. Escollim un nombre primer p públic, on $p > n$ i $p > S$. També escollim un polinomi de grau $m - 1$, que té com a terme independent el secret; és a dir:

$$P(x) = S + P_1x + P_2x^2 + \ldots + P_{m-1}x^{m-1} \bmod p,$$

on els coeficients P_i són aleatoris i secrets. Finalment, es trien n valors enters aleatoris inferiors a p, $\{x_1, \ldots, x_n\}$. Cada participant rep com a fragment del secret un parell $\{x_i, P(x_i)\}$, on $P(x_i)$ és l'avaluació del polinomi $P(x)$ en el punt x_i.

A continuació veiem que, per tal de recuperar el secret, n'hi ha prou amb m participants. En efecte, si pretenem calcular el polinomi $P(x)$ resulta que tenim com a incògnites els m coeficients del polinomi, ja que els valors de les variables els aporten els participants, les seves potències es poden calcular i l'avaluació del polinomi en els punts també la donen els participants. Així, s'obté el sistema d'equacions següent:

$$\begin{aligned}
P(x_1) &= S + P_1x_1 + P_2x_1^2 + \ldots + P_{m-1}x_1^{m-1} \bmod p.\\
P(x_2) &= S + P_1x_2 + P_2x_2^2 + \ldots + P_{m-1}x_2^{m-1} \bmod p.\\
&\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots\\
P(x_m) &= S + P_1x_m + P_2x_m^2 + \ldots + P_{m-1}x_m^{m-1} \bmod p.
\end{aligned}$$

Aquest és un sistema de m equacions amb m incògnites que sempre té una solució única, ja que hi intervé el determinant de Vandermonde.

Els esquemes de compartició de secrets, però, no estan exempts de determinats atacs que fan que esquemes tan simples com els presentats fins ara no siguin útils en segons quins entorns. Per exemple, ens podríem preguntar què

passaria si un dels participants donés un valor aleatori en comptes del seu fragment. La cosa certa és que el secret no es recuperaria i, encara més, no sabríem qui n'ha estat el culpable. També resulta que 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. Per tal de resoldre aquests problemes hi ha esquemes de compartició de secrets més elaborats que resisteixen aquest tipus d'atacs, però el seu estudi s'escapa de l'abast d'aquesta assignatura. 

3. Situacions de desconfiança mútua

3.1. Compromís de bit

L'Anna i en Bernat es telefonen per anar al cinema, però no es posen d'acord en la pel·lícula que volen veure. L'Anna n'ha triat una i en Bernat una altra. Finalment decideixen fer-ho a cara o creu. El problema és que com que parlen per telèfon, òbviament, és molt fàcil que en Bernat, que tira la moneda, faci trampa, ja que l'Anna no pot veure què ha sortit realment a la moneda i en Bernat pot dir sempre que ha sortit el contrari.

Per a solucionar això, el que ha de fer l'Anna és comprometre's a un valor (cara o creu) i enviar a en Bernat el compromís del valor a què s'ha compromès de manera que en Bernat no pugui veure quin és aquest valor del compromís. En Bernat llança la moneda i comunica el resultat a l'Anna. Aleshores, l'Anna obre el seu compromís i es comprova si ha encertat o no. Evidentment, a fi que l'Anna tampoc no pugui fer trampa caldrà assegurar que no ha pogut canviar el valor un cop s'ha compromès.

Traduint aquest esquema a un entorn matemàtic i suposant que l'Anna es compromet al valor d'un sol bit, tenim el que s'anomena **compromís de bit**. L'Anna es vol comprometre a un cert bit b amb en Bernat, sense que aquest conegui *a priori* el valor de b. Aleshores l'Anna genera el compromís de bit que representarem per $C(b)$ i l'envia a en Bernat. Aquest compromís té les característiques següents:

1) En Bernat, a partir del compromís $C(b)$, que és el que guarda, no pot obtenir el valor b.

2) L'Anna pot obrir el compromís $C(b)$ per tal de mostrar que efectivament b era el valor a què s'havia compromès.

3) L'Anna no pot obrir el mateix compromís $C(b)$ per tal de mostrar un altre valor $b' \neq b$.

Per a il·lustrar com pot funcionar un esquema d'aquest tipus, vegem l'exemple següent, que utilitza clau simètrica. El primer que fem és crear el compromís:

a) B genera un valor aleatori r i l'envia a A.

b) A utilitza una funció de xifratge $E(\ )$ amb clau k per tal de xifrar el valor b a què es vol comprometre juntament amb el valor aleatori r que ha rebut de B. Aleshores, envia a B el valor $C(b) = E_k(b,r)$.

A continuació efectuem l'obertura del compromís:

a) A envia la clau k a B.

b) B desxifra el compromís $C(b)$ utilitzant la clau rebuda i comprova que juntament amb el valor b compromès hi ha el seu valor aleatori r.

Si ens hi fixem, en aquest exemple es compleixen les condicions que hem esmentat anteriorment:

1) Si l'esquema de xifratge utilitzat és bo, B no pot obtenir el valor de b a partir de $C(b) = E_k(b,r)$, ja que no coneix la clau k.

2) Per a obrir el compromís, només cal que A enviï la clau k a B.

3) A no pot obrir el compromís $C(b)$ per a obtenir un altre valor $b' \neq b$, ja que quan A ha creat el compromís hi ha afegit el valor aleatori r generat per B i, si el mètode de xifratge és bo, és molt difícil que A pugui trobar una altra clau $k' \neq k$ que desxifri $C(b)$, obtenint un altre valor b', però mantenint intacte el valor r.

Un esquema de compromís de bit més interessant és el creat per G. Brassard, C. Crépeau i D. Chaum, que va ser presentat l'any 1988. El funcionament es basa en el següent: com a paràmetres públics del sistema tenim (p,g), on p és un primer tal que $p - 1 = kq$ amb q primer i g, un generador de $\mathbb{Z}_p$:

a) A es vol comprometre al valor b, on $1 < b < p - 1$. El compromís el calcula com $C(b) = g^b \bmod p$, i envia el resultat $C(b)$ a B.

b) Per a obrir el compromís, A envia directament el valor b a B. Aleshores, B comprova que el compromís sigui correcte validant que $g^b \bmod p = C(b)$.

Si ens fixem en les propietats, veiem que aquest compromís de bit també compleix les tres següents:

1) B no pot obtenir cap informació del valor $C(b)$, ja que per a obtenir b ha de calcular el logaritme discret:

$$b = \log_g C(b) \bmod p.$$

2) Efectivament, A pot obrir el compromís $C(b)$ donant el valor b a B per tal que comprovi el següent:

$$g^b \bmod p = C(b).$$

3) És clar que A no pot obrir el compromís $C(b)$ per a obtenir $b' \neq b$, ja que si ho pogués fer tindríem:

$$\frac{C(b)}{C(b')} = \frac{g^b}{g^{b'}} = g^{b-b'} = 1 \bmod p.$$

Tenint en compte la tria dels valors g i p ens queda que:

$$b - b' = 0 \bmod (p - 1),$$

i com que $1 < b < p - 1$ i $1 < b' < p - 1$ només pot ser que $b = b'$. De totes maneres, això és així tenint en compte que g és un generador de $\mathbb{Z}_p$. Per tant, si B vol estar segur que A no l'enganya ha de poder validar que g és un generador. Per aconseguir-ho, A pot utilitzar una prova de coneixement nul.

3.2. Transferència inconscient

Els protocols de transferència inconscient permeten dur a terme l'esquema següent: suposem que en Bernat vol conèixer un secret que té l'Anna, però no vol que l'Anna sàpiga si ell el sap o no. Aleshores, l'Anna i en Bernat executaran el protocol de transferència inconscient, al final del qual l'Anna haurà enviat a en Bernat el secret amb probabilitat 1/2. És a dir, en Bernat té 1/2 de probabilitat de rebre el secret i l'Anna no sabrà si en Bernat l'ha rebut o no.

Aquest concepte que sembla molt complicat i de poca utilitat veurem que té diversos camps d'aplicació. La idea va ser proposada per T. Rabin el 1981 i l'esquema que va presentar es basa en el fet que calcular arrels quadrades és equivalent a factoritzar. Vegem com funciona el protocol:

1) El secret que té A són els valors p i q en què factoritza un cert nombre $N = pq$. A envia N a B.

2) B tria un valor aleatori $x \in \mathbb{Z}_N$ i envia el seu quadrat $z = x^2 \bmod N$ a A.

3) A calcula $\{\pm x, \pm y\}$, les quatre arrels quadrades de z, gràcies al fet que coneix la factorització de N; en tria una a l'atzar i l'envia a B.

4) B comprova quina de les arrels ha rebut. Si ha rebut $\pm x$ no tindrà més informació de la que tenia quan ha començat el protocol; mentre que si rep $\pm y$ podrà calcular mcd$(x - y, N)$, que serà p o bé q, i, conseqüentment podrà factoritzar N.

Fixem-nos que com que A només coneix z no sap quina de les arrels té B i, per tant, la probabilitat que li enviï una arrel diferent de la que té és 1/2.

Com ja hem dit abans, aquest protocol per si sol potser no té gaire interès, però és la base d'altres esquemes com poden ser la signatura de contractes o el correu electrònic certificat.

Fixeu-vos que...

... $(x - y)(x + y) = x^2 - y^2 = z - z = 0 \bmod N$, i per tant $N = p \cdot q$ divideix $(x - y)(x + y)$, amb la qual cosa o bé p divideix $x - y$ o bé ho fa q.

En particular, en molts casos el que farem servir és una modificació del protocol de transferència inconscient. Aquesta modificació permet a A "enviar" dos missatges i a B rebre'n un dels dos amb probabilitat 1/2. Aquesta variant és coneguda com a **transferència inconscient 1-2**.

Vegem com es pot aplicar la transferència inconscient 1-2 en l'intercanvi de secrets. Suposem que l'Anna i en Bernat volen intercanviar $2n$ nombres secrets de m bits cada un, que representarem per a_i, on $1 \leq i \leq 2n$, els de A, i b_i, on $1 \leq i \leq 2n$, els de B. Llavors:

1) A divideix els seus $2n$ nombres secrets en n parells, per exemple (a_{2j-1},a_{2j}) per a $j = 1, ..., n$. Aleshores, A 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}; però A no sap quin dels elements ha rebut B*.

*** Recordeu que cada element del parell té un 50% de probabilitat de ser enviat.**

2) Simultàniament al pas anterior, B fa exactament el mateix amb els seus $2n$ nombres: 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 nombres a_i i b_i per a $i = 1, ..., 2n$; després, el segon bit, i així fins al final. Si A vol enganyar B no donant-li l'altre element de cap parell, només té la probabilitat d'$1/2^n$ d'aconseguir-ho, perquè B ja té un element de cada parell i A no sap quin és. Simètricament, es pot aplicar el mateix si B vol enganyar A.

3.3. Signatura de contractes

A l'hora de signar contractes en un entorn no presencial com pot ser una xarxa informàtica, ens trobem bàsicament amb dos problemes: la validesa de la signatura digital i la simultaneïtat de la signatura.

Recordeu que hem estudiat les signatures digitals al mòdul "Signatures digitals" d'aquesta assignatura.

Ja hem vist la validesa que pot tenir una signatura digital des del punt de vista de la seguretat i n'hem fet algunes consideracions. Ara bé, quan tenim un contracte en un entorn no presencial, qui el signarà primer? En el cas que no hi hagi confiança, cap de les dues parts no voldrà iniciar la signatura, perquè podria resultar que un cop signat la part contrària no ho fes. Per això necessitem un protocol per a signar contractes simultàniament. Vegem com funciona:

1) A genera aleatòriament $2n$ claus DES (que, òbviament, podrien ser de qualsevol altre xifratge de bloc) K_i^a per a $i = 1, ..., 2n$, i n parells de missatges (L_j^a,R_j^a) per a $j = 1, ..., n$. Aleshores, A xifra cada missatge amb una clau diferent: $P_j^a = E_{K_j^a}(R_j^a)$ per a $j = 1, ..., n$, i $Q_j^a = E_{K_{n+j}^a}(L_j^a)$ per a $j = 1, ..., n$.

Vegeu el criptosistema DES al subapartat 2.1 del mòdul "Xifres de clau compartida: xifres de bloc" d'aquesta assignatura.

2) B fa el mateix (genera les claus i els missatges) i, per tant, obté $P_j^b = E_{K_j^b}(R_j^b)$ per a $j = 1, ..., n$, i $Q_j^b = E_{K_{n+j}^b}(L_j^b)$ per a $j = 1, ..., n$.

3) Al contracte que han de signar s'hi afegeix una clàusula per la qual s'especifica que el contracte només es considerarà signat en el cas que A pugui desxifrar P_j^b i Q_j^b per a algun $1 \leq j \leq n$ i B pugui desxifrar P_j^a i Q_j^a per a algun $1 \leq j \leq n$.

4) A i B intercanvien les $2n$ claus secretes* utilitzant el protocol d'intercanvi de secrets de la transferència inconscient.

* A **divideix les seves claus en** n **parells** (K_j^a, K_{n+j}^a) **i** B **també ho fa en** n **parells** (K_j^b, K_{n+j}^b).

3.4. Correu electrònic certificat

Una altra situació de desconfiança mútua la trobem quan l'Anna vol enviar un correu electrònic a en Bernat i es vol assegurar que aquest el rep. Dit d'una altra manera, el que es vol obtenir és el símil del que amb el correu ordinari es coneix com a correu certificat (o, més concretament, correu amb justificant de recepció), però en l'àmbit del correu electrònic.

La manera d'aconseguir-ho és molt semblant a la tècnica de signatura de contractes que acabem d'estudiar. El que es pretén és que B no pugui llegir el missatge que li envia A fins que A no hagi rebut el justificant de recepció de B: 

1) A genera aleatòriament $n + 1$ claus DES (que, òbviament, podrien ser de qualsevol altre xifrador de bloc) K_i^a per a $i = 0, ..., n$, i en calcula n més a partir de les n inicials de la manera següent: $K_{n+i}^a = K_0^a \oplus K_i^a$ per a $i = 1, ..., n$. D'aquesta manera, A obté un total de $2n + 1$ claus.

2) A xifra el missatge que vol enviar amb la clau K_0^a: $C = E_{K_0^a}(M)$.

3) A xifra un missatge S, conegut també per B, utilitzant les $2n$ claus restants. Després envia a B els textos xifrats $C_i = E_{K_i^a}(S)$ per a $i = 1, ..., 2n$.

4) B tria $2n$ claus K_i^b per a $i = 1, ..., 2n$ i també genera $2n$ textos xifrats del missatge S, $D_i = E_{K_i^b}(S)$ per a $i = 1, ..., 2n$, i els envia a A.

5) A i B intercanvien les $2n$ claus secretes* utilitzant el protocol d'intercanvi de secrets de la transferència inconscient.

* A **divideix les seves claus en** n **parells** (K_i^a, K_{n+i}^a) **i** B **també ho fa en** n **parells** (K_i^b, K_{n+i}^b).

En finalitzar el protocol, A tindrà un justificant de recepció si pot desxifrar un parell (D_i, D_{i+n}) per a alguna $1 \leq i \leq n$. Per la seva banda, si B obté un parell de claus (K_i^a, K_{i+n}^a) ja pot desxifrar el missatge C, ja que la clau K_0^a la pot obtenir de la relació següent: $K_0^a = K_{n+i}^a \oplus K_i^a$.

Cal destacar que el servei de justificant de recepció en correu electrònic va més enllà que el de correu convencional, ja que en alguns esquemes més elaborats en què s'inclouen funcions *hash* i criptografia de clau pública es pot

incorporar un resum del correu en el justificant de recepció de manera que vagi lligat unívocament al contingut del correu.

3.5. Càlcul segur a múltiples bandes

El càlcul segur a múltiples bandes és un concepte pel qual un conjunt de n participants cooperen per a efectuar un determinat càlcul. Per tant, les entrades seran un conjunt de dades $(i_1, ..., i_n)$ tals que la dada i_j correspon al participant j. Com a sortida s'obté el valor de l'operació que ha de conèixer tothom. El principal requeriment de seguretat d'aquests esquemes prové del fet que els participants no han de saber quin és el valor d'entrada donat per un altre usuari.

D'una manera molt simple es pot donar una de les aproximacions a la solució. En el cas en què hi hagi una tercera part de confiança de la qual tothom es fia, aquesta tercera part és la que pot efectuar el càlcul. Com que tothom hi confia, tothom està segur que una vegada cada usuari li hagi lliurat el seu fragment d'informació, no el mostrarà a cap altra part.

El que s'intenta en els esquemes de càlcul segur a múltiples bandes és poder prescindir d'aquesta tercera part de confiança.

Vegem un exemple de càlcul segur a múltiples bandes en el cas concret de dos participants. Aquest protocol va ser proposat per C. Yao l'any 1982 i es coneix com el **problema del milionari**. La situació és la següent: dos milionaris volen saber qui és més ric, però no volen revelar el valor de les seves fortunes a ningú.

Per tal de simplificar una mica el problema (de fet, seria equivalent a fitar la fortuna que tenen) transformarem el cas en un altre d'equivalent, en què l'Anna i en Bernat volen saber qui té més edat sense dir quina tenen. Suposarem que tots dos són honestos i que utilitzen les seves edats reals. L'Anna té i anys, en Bernat j i cap dels dos no en té més de 100, és a dir, es verifica que $1 \leq i$ i $j \leq 100$. També necessitarem un esquema de clau pública en què hi haurà un parell de claus per a l'Anna (E_A,D_A) i un altre per a en Bernat (E_B,D_B). Òbviament, en Bernat coneix la clau pública de l'Anna, E_A, i l'Anna la d'en Bernat, E_B.

El protocol funciona de la manera que explicitem a continuació:

1) B tria un nombre aleatori gran x i el xifra amb la clau pública de A, $k = E_A(x)$.

2) B envia a A el valor $k - j$.

3) A amb la seva clau privada calcula el valor:

$$y_u = D_A(k - j + u); \text{ per a } 1 \leq u \leq 100.$$

A tria un primer *p* gran (la mida de *x* i *p* s'ha acordat prèviament de manera que la mida de *p* és més petita que la de *x*) i calcula 100 valors z_u, tals que verifiquen la relació següent:

$$z_u \equiv y_u \bmod p; \text{ per a } 1 \leq u \leq 100.$$

A més, per a tot $1 \leq u \leq 100$ amb $u \neq v$ es verifica que:

$$|z_u - z_v| \geq 2; \text{ per a } 0 < z_u < p - 1.$$

Si alguna de les desigualtats no es compleix, *A* tria un altre primer.

4) *A* dóna a *B* la seqüència ordenada següent:

$$z_1, \ldots, z_i, z_{i+1} + 1, z_{i+2} + 1, z_{100} + 1, p.$$

Recordeu que *i* és l'edat de l'Anna.

5) *B* comprova si el valor *j*-èsim de la seqüència que ha rebut és congruent amb *x* mod *p*. Si ho és, *B* es convenç que $i \geq j$; en cas contrari $i < j$.

6) *B* revela la seva conclusió a *A*. Arribats en aquest punt, es pot tornar a executar el protocol canviant els papers de *A* i *B*.

La conclusió que treu *B* en el cinquè pas és correcta, ja que:

a) Si $i \geq j$, aleshores el valor z'_j de la seqüència que *A* envia a *B* en el quart pas és $z'_j = z_j$. Per tant, tenim que $z'_j \equiv y_j \bmod p$. Com que $y_j = D_A(k - j + j) = D_A(k)$ i $k = E_A(x)$, ens queda que $y_j = D_A(E_A(x)) = x$ i, per tant: $z'_j \equiv x \bmod p$.

b) Si $i < j$, aleshores el valor z'_j verifica que: $z'_j = z_j + 1 \not\equiv z_j \equiv y_j = x \bmod p$.

3.6. Càlcul sobre dades xifrades

El càlcul sobre dades xifrades sorgeix com a necessitat per a resoldre el problema següent: suposem que l'Anna vol fer un determinat càlcul $f(\ )$ sobre certes dades *x* que té, però resulta que no té suficient potència de càlcul per a dur a terme el procés. En Bernat sí que té prou capacitat computacional i, per tant, no té cap problema a calcular $f(x)$ a partir del valor *x*, però l'Anna no vol que en Bernat obtingui cap mena d'informació de les seves dades *x*.

Tot i que en principi es podria veure com un problema semblant a la computació a múltiples bandes, no és així. La diferència principal que hi ha entre

la computació a múltiples bandes i el càlcul sobre dades xifrades és que, en el primer cas, hi intervenen totes les parts per a fer el càlcul, mentre que en el segon cas hi ha una part passiva que és la que delega el càlcul.

Una solució genèrica per a aquest problema és el que s'anomenen *homomorfismes de privacitat*. Els homomorfismes de privacitat van ser introduïts per R.L. Rivest, L. Adleman i M.L. Dertouzos l'any 1978. Un **homomorfisme de privacitat** (**HP**) és un esquema de xifratge que conserva certes operacions. És a dir, suposem que E_k és un HP que preserva la suma. Això vol dir que, donats dos valors x i y, és el mateix fer una suma i després xifrar les dades que xifrar les dades i fer la suma. Per tant:

$$E_k(x + y) = E_k(x) + E_k(y).$$

Els homomorfismes de privacitat tenen la peculiaritat que com més operacions permeten conservar, menys robustos es tornen criptogràficament parlant. En particular, se sap que si un HP preserva la relació d'ordre, aleshores és insegur contra atacs només amb text xifrat. És més, un HP que conservi la suma serà insegur per a atacs amb text en clar escollit.

Des d'un punt de vista computacional i de delegació de càlcul ens interessaran els HP que conservin les operacions bàsiques d'un cos, és a dir, $\{+, -, \cdot, \div\}$. Per tant, pel que hem esmentat anteriorment, podrem trobar HP que resisteixin atacs de text en clar conegut, però no atacs de text en clar escollits, ja que volem conservar la suma. Actualment no es coneix cap HP que preservi les quatre operacions de cos i mantingui un nivell de seguretat acceptable.

L'HP més segur i que preserva més operacions de tots els que es coneixen és el creat per J. Domingo Ferrer (1997), i que descrivim a continuació:

- Els paràmetres públics són d, un enter positiu i, m, un enter aleatori gran ($\approx 10^{200}$).

- La clau secreta és $k = (m^*, r)$, on m^* és un divisor petit de m i $r \in \mathbb{Z}_m - \{0\}$.

- L'espai de text en clar és $T = \mathbb{Z}_{m^*}$ i el de text xifrat, $T' = (\mathbb{Z}_m)^d$.

- Les operacions de text en clar són $F = \{+_{m^*}, -_{m^*}, \cdot_{m^*}\}$ i les operacions de text xifrat a F' són com a F, però component per component.

- La funció de xifratge segueix un procés concret. Dividim a de manera aleatòria i secreta en $(a_{.1}, ..., a_{.d})$, on tenim que $\sum_{j=1}^{d} a_{.j} = a \bmod m^*$ i aleshores calculem:

$$E_k(a) = (a_{.1}r \bmod m, ..., a_{.d}r^d \bmod m).$$

- El procés de desxifratge ha de seguir uns passos concrets. Donada la clau $k = (m^*,r)$, calculem:

$$b_j r^{-j} \bmod m = a_{.j}; \text{ per a } 1 \leq j \leq d,$$

  i finalment sumem mod m^* per a trobar a.

Pel que fa a la seguretat d'aquest criptosistema, es pot demostrar que si $d > 1$ i el nombre n de parells de text en clar i text xifrat que coneix l'atacant és més petit que el paràmetre de seguretat $s = \log_{m^*} m$, aleshores la probabilitat de trencar l'HP (intentant endevinar-ne la clau) és com a màxim $\pi^2(m^*)^{n-s} / 6$.

També es pot demostrar que aquest HP preserva la suma, la resta i el producte. Totes les operacions en l'espai de text xifrat es fan com si tractéssim polinomis de variable independent r. La suma i la resta es fan component per component, mentre que el producte és el producte creuat sumant els coeficients dels mateixos graus en r. En el cas de la divisió, el seu valor es pot deixar indicat com a fracció de dos nombres.

4. Diners electrònics

Actualment, la manera més habitual de portar a terme transaccions monetàries per mitjà d'Internet és fent que el comprador enviï una informació determinada sobre la seva targeta de crèdit, o bé que obri un compte a un venedor amb antelació al moment de l'operació.

La crítica més important que es pot fer a la compra per Internet basada en targetes de crèdit és que no és anònima. Efectivament, és possible monitoritzar les transaccions, atès que la identitat del client s'estableix cada vegada que fa una compra. En la vida real tenim l'alternativa de fer servir diners en metàl·lic quan volem comprar alguna cosa sense que es conegui la nostra identitat.

El terme **diners electrònics** s'usa per a englobar els protocols i les tècniques criptogràfiques que pretenen recrear el concepte de *compres en metàl·lic* per mitjà d'Internet.

Farem una aproximació general als diners electrònics basada en criptografia de clau pública. Aquesta aproximació va ser suggerida originàriament per D. Chaum (1981). Els sistemes de pagament efectuats segons la idea de Chaum ofereixen la possibilitat de l'anonimat en les transaccions.

4.1. Requisits

Els sistemes de diner electrònic haurien de complir, almenys, les propietats següents:

1) La falsificació de diners hauria de ser difícil. Es considera que un client falsifica diners electrònics si n'aconsegueix més dels que el banc li carrega en compte.

2) La despesa múltiple (és a dir, despendre dues o més vegades uns diners autèntics) hauria de ser impossible o, almenys, detectable.

3) L'anonimat del client hauria de ser preservat. Fins i tot si el banc i el venedor es confabulessin, no haurien de poder saber en què despèn els seus diners un client determinat.

4) Les operacions *on line* haurien de ser mínimes.

4.2. El sistema de pagament

Normalment, un sistema de diners electrònics consisteix en els tres protocols següents:

1) Reintegrament. Aquest protocol permet a l'usuari *U* obtenir una moneda digital del banc *B*.

2) Pagament. Amb aquest protocol, l'usuari *U* compra béns del venedor *V* a canvi de la moneda digital.

3) Dipòsit. Amb el protocol de dipòsit, el venedor *V* dóna la moneda al banc *B* perquè sigui ingressada en el compte de *V*.

En aquests protocols hem suposat que *U* i *V* tenen el mateix banc *B*. Si el banc de *V* fos $B' \neq B$, llavors hi hauria un procediment de compensació entre *B* i *B'* després del dipòsit.

Suposem ara que el banc té una clau privada, SK_B, per a signar missatges i que la clau pública corresponent, PK_B, és coneguda per tothom. Sigui $SK_B(m)$ el missatge *m* juntament amb la seva signatura sota la clau SK_B. Llavors els protocols del nostre sistema podrien ser els que expliquem a continuació.

1) Protocol de reintegrament

a) L'usuari *U* diu al banc *B* que voldria retirar 100 euros.

b) *B* retorna un bitllet digital de 100 euros que té la signatura següent:

$$SK_B(\textit{sóc un bitllet de 100 euros}, \#4527),$$

i retira 100 euros del compte de *U*. La xifra #4527 és el número de sèrie, que és diferent per a cada bitllet.

c) *U* comprova la signatura del bitllet i si és vàlida l'accepta.

2) Protocol de pagament

a) *U* paga al venedor *V* amb el bitllet.

b) *V* comprova la signatura del bitllet i si és vàlida l'accepta.

3) Protocol de dipòsit

a) *V* dóna el bitllet a *B*.

b) *B* comprova la signatura del billet i si és vàlida n'ingressa el valor al compte de *V*.

Evidentment, si l'esquema de signatura digital és segur, és impossible falsificar bitllets en els protocols anteriors. No obstant això, és molt fàcil de duplicar i de despendre diverses vegades el mateix bitllet digital (despesa múltiple). A més, també és evident que no es preserva l'anonimat, ja que el banc *B* pot relacionar el nom de *U* amb el número de sèrie que apareix en el bitllet i saber així on ha despès *U* el bitllet.

4.3. Signatures tapades

Mirem primer de resoldre el problema de l'anonimat que acabem de presentar. Per a fer-ho, ens servirem del que s'anomenen *signatures tapades**.

* En anglès, *blind signatures*.

La idea de les **signatures tapades** és que l'usuari presenta el bitllet al banc amb un cert emmascarament. En el protocol de reintegrament, el banc signa el bitllet sense veure'n els continguts. D'aquesta manera, el banc no pot determinar qui va retirar el bitllet quan un venedor el presenta durant el protocol de dipòsit.

Una analogia útil és pensar en termes de paper. L'usuari cobreix un xec amb un paper carbó i ho posa tot dins un sobre tancat. L'usuari dóna el sobre al banc. El banc signa el sobre per fora amb un bolígraf i el torna a l'usuari sense obrir-lo (de fet, en la versió digital el banc és incapaç d'obrir el sobre). Tot seguit, l'usuari extreu el xec signat de dins el sobre i el pot despendre. El banc no ha vist el xec que ha signat, de manera que no el pot associar amb l'usuari quan el xec és dipositat pel venedor. Ara bé, el banc pot verificar la signatura que porta el xec i comprovar-ne la validesa.

Però amb aquest sistema hi ha un problema greu: un usuari malintencionat pot enganyar el banc perquè signi bitllets trucats. Per exemple, un usuari podria demanar al banc un reintegrament d'1 euro i després presentar un bitllet de 100 euros perquè sigui signat. El banc signarà sense saber-ho el bitllet de 100 euros, amb la qual cosa li hauran estafat 99 euros. Tractarem d'aquest problema més endavant. De moment, donarem una construcció de signatures tapades basada en l'RSA.

Tractem més a fons dels problemes que porten associats les signatures tapades en el subapartat 4.4 d'aquest mòdul didàctic.

4.3.1. Signatures tapades RSA

Amb la signatura RSA, la signatura d'un missatge m és $s = m^d \bmod n$ si la clau pública del signatari és (n,e)*. Podem verificar la signatura comprovant si

* Recordeu que en la signatura RSA $d = e^{-1} \bmod \phi(n)$.

$s^e \bmod n \stackrel{?}{=} m \bmod n$. En el cas de les signatures tapades, l'usuari *U* vol que el banc *B* li retorni *s* sense haver de revelar *m*. Si *m* és un bitllet de 100 euros, vet aquí un possible protocol de reintegrament.

Les signatures RSA s'estudien al subapartat 2.1 del mòdul "Signatures digitals" d'aquesta assignatura.

Protocol de reintegrament

a) *U* tria un nombre aleatori *r* mod *n*.

b) *U* tapa el missatge i calculant $m' = mr^e \bmod n$.

c) *U* dóna m' al banc *B*.

d) *B* retorna una signatura per a m', per exemple $s' = (m')^d \bmod n$. Noteu que $s' = (m')^d = m^d(r^e)^d = m^d r$.

e) *B* carrega 100 euros al compte de *U*.

f) Com que *U* sap *r*, pot dividir s' per *r* per a obtenir $s = m^d$.

D'aquesta manera, hem resolt el problema de l'anonimat, ja que quan el bitllet torna a *B* no hi ha cap relació entre el bitllet i l'usuari *U* a qui va ser lliurat inicialment.

En les signatures tapades RSA,...

... els protocols de pagament i de dipòsit continuen essent els mateixos del subapartat 4.2 d'aquest mòdul.

4.4. Com evitar l'engany al banc

Les signatures tapades resolen el problema de l'anonimat, però encara queden per a resoldre dos problemes més:

- L'usuari pot enganyar el banc fent-li signar allò que no correspon (com ara un bitllet de 100 euros fent-li creure que és d'1 euro).
- Els bitllets es poden duplicar i despendre diverses vegades (despesa múltiple).

Dues solucions possibles per al primer problema són les següents:

1) Es pot tenir tan sols una denominació per clau pública; és a dir, *B* tindria diverses claus públiques $PK1_B$, $PK2_B$, ..., de tal manera que la signatura verificable amb PKi_B només valdria per a bitllets de *i* euros.

2) Una altra possibilitat és l'anomenat *procediment de "remenar i triar"*, el procés del qual segueix els passos següents:

a) *U* fabrica 100 bitllets de 20 euros.

b) *U* tapa tots els bitllets (amb uns nombres aleatoris *r*).

c) *U* dóna els bitllets tapats al banc *B*.

d) *B* n'agafa un a l'atzar i demana a *U* que destapi la resta (revelant-ne els nombres *r* corresponents). Abans de retornar la signatura del bitllet que ha triat, *B* s'assegura que tots els bitllets revelats eren correctes.

Amb aquest darrer procediment, *U* només té una probabilitat 1/100 de fer trampa sense ser detectat. Augmentant el nombre de bitllets tapats que ha de fabricar *U*, podem reduir arbitràriament la probabilitat que una trampa de *U* quedi sense detectar.

4.5. Diners electrònics *on line*

Fins ara hem aconseguit resoldre els problemes de l'anonimat i de l'engany al banc. Queda el problema de la despesa múltiple del mateix bitllet, que resoldrem amb els diners electrònics *on line*.

En la **versió *on line* dels sistemes de diner electrònic** es demana que el banc *B* registri tots els bitllets que ha rebut en una base de dades. Durant el protocol de pagament, el venedor *V* transmet el bitllet a *B* i demana si aquell mateix bitllet havia ja estat rebut amb anterioritat. Si es fa servir per primera vegada, *V* l'accepta; altrament, el rebutja.

Tot i que es tracta d'una solució senzilla, requereix molta comunicació, atès que el protocol de pagament s'assembla molt a una transacció per targeta de crèdit, en la qual el venedor *V* espera autorització abans de cloure el tracte. D'altra banda, la mida de la base de dades que ha de mantenir *B* pot arribar a ser problemàtica.

La solució *on line* impedeix la despesa múltiple. Ara bé, és possible detectar la despesa múltiple sense verificació *on line*.

4.6. Diners electrònics *off line*

La idea dels diners electrònics *off line* és la següent: durant el protocol de pagament, *U* és obligat a escriure una **tira aleatòria d'identificació** (**TAI**) sobre el bitllet. La TAI ha de complir tres requisits:

- Que sigui diferent per a cada pagament del bitllet.

- Només *U* ha de poder crear una TAI vàlida.
- Dues TAI diferents sobre el mateix bitllet haurien de permetre a *B* recuperar el nom de *U*.

Així, si *B* rep dos bitllets idèntics amb valors de TAI diferents, llavors *U* ha fet trampa i *B* pot identificar-lo. Si *B* rep dos bitllets idèntics amb els mateixos valors de TAI, llavors el trampós és el venedor *V*. Si *H* és una funció *hash* unidireccional, un sistema possible de diners *off line* és el que presentem tot seguit.

1) Protocol de reintegrament

a) *U* prepara 100 bitllets de 20 euros amb el format següent:

$$m_i = (\textit{sóc un bitllet de 20 euros}, \#4257i, y_{i,1}, y'_{i,1}, y_{i,2}, y'_{i,2}, \ldots, y_{i,K}, y'_{i,K}),$$

on $y_{i,j} = H(x_{i,j})$, $y'_{i,j} = H(x'_{i,j})$, amb $x_{i,j}$ i $x'_{i,j}$ triats aleatòriament amb la restricció següent:

$$x_{i,j} \oplus x'_{i,j} = \text{nom de } U;\ \forall i, j.$$

b) *U* tapa tots els m_i transformant-los en missatges aleatoris m'_i i envia aquests darrers al banc *B*.

c) *B* demana a *U* que destapi 99 dels 100 bitllets tapats.

d) Quan *U* destapa els bitllets, també revela els $x_{i,j}$ i $x'_{i,j}$ adequats.

e) Per als bitllets destapats, *B* comprova no tan sols que són de 20 euros, sinó també que $y_{i,j} = H(x_{i,j})$, $y'_{i,j} = H(x'_{i,j})$ i $x_{i,j} \oplus x'_{i,j} =$ nom de *U*.

f) *B* retorna una signatura per a l'únic bitllet tapat (posem per cas m'_{13}).

g) *U* recupera la signatura s_{13} sobre m_{13}.

A partir d'ara, prescindirem de l'índex $i = 13$ per simplicitat. El protocol de pagament es modifica per forçar *U* a produir una TAI sobre el bitllet. La TAI serà la concatenació de les x_j o x'_j per a $j = 1, \ldots, K$. Quina x_j o x'_j sigui depèn d'un repte aleatori que llança el venedor *V*.

2) Protocol de pagament

a) *U* dóna (m,s) a *V*.

b) *V* verifica la signatura del banc *B* sobre el bitllet i si és vàlida respon amb una tira aleatòria de bits de longitud *K*: b_1, ..., b_K (repte).

c) Si $b_j = 0$ l'usuari *U* revela x_j; altrament, revela x'_j.

d) *V* comprova que $y_j = H(x_j)$ o que $y'_j = H(x'_j)$, segons el cas. Si les comprovacions funcionen per a tot *j*, *V* accepta el bitllet.

Es compleixen les propietats de la TAI enunciades al principi del subapartat. La probabilitat que en un pagament diferent es produeixi la mateixa TAI és 2^{-K}, atès que *V* tria el repte aleatòriament. Només *U* pot produir una TAI vàlida ja que la funció *H* és unidireccional. Finalment, dues TAI diferents sobre el mateix bitllet revelen el nom de *U*, atès que si dues TAI són diferents hi ha d'haver un índex *j* per al qual tinguem tant x_j com x'_j.

3) Protocol de dipòsit

a) *V* porta el bitllet (*m*,*s*,TAI) al banc *B*.

b) *B* en verifica la signatura i comprova si (*m*,*s*) ja havia estat dipositat.

c) Si el bitllet ja era a la base de dades, *B* compara les TAI dels dos bitllets. Si són diferents, l'usuari *U* ha despès el bitllet dues vegades; si són iguals, és el venedor *V* qui intenta dipositar la moneda dos cops.

5. Concessió de drets intransferibles

En un sistema informàtic distribuït, les entitats que requereixen identificació són els ordinadors, els usuaris i els processos. Quan una d'aquestes entitats demana un servei d'una altra entitat, el terme *client* designa la primera entitat, i el terme *servidor*, la segona. Se sol suposar que els servidors són fiables en el sentit que només proporcionen el servei després d'haver comprovat que el pretès client té el dret d'obtenir-lo.

Targeta VISA

La utilització d'aquesta targeta és personal i intransferible, per la qual cosa l'establiment pot sol·licitar qualsevol document que acrediti la personalitat de l'usuari [...].

Considerem un escenari distribuït consistent en una gran xarxa amb una autoritat central, un conjunt de servidors fiables que donen accés a certs recursos i una comunitat de clients*. Imposem els requisits de funcionament següents:

*** Per exemple, podem pensar que l'autoritat és un banc i que parlem d'una xarxa de caixers automàtics (servidors).**

- **Independència**: l'autoritat dóna d'alta clients i els concedeix drets sense haver-ne d'informar els servidors, els quals no guarden informació d'accés sobre els clients.

- **Control d'accés**: per a poder dur a terme el control d'accés, els servidors necessiten només una llista certificada dels drets disponibles.

- **Intransferibilitat**: sense el concurs de l'autoritat, un client no pot transferir part dels seus drets a un altre client.

Cal remarcar que els esquemes convencionals de control d'accés no compleixen el requisit d'independència. La intransferibilitat de drets és trivial si obliguem cada servidor a guardar informació sobre qui pot accedir a un determinat servei. Ara bé, aconseguir alhora independència i intransferibilitat no és tan senzill.

Per exemple,...

... el mecanisme de matriu d'accés requereix que el servidor guardi una matriu que diu quins drets té cada client.

L'**esquema proposat per J. Domingo Ferrer** (1994) compleix els tres requisits esmentats anteriorment i ofereix una seguretat tal que:

a) El fet que un client que no comparteix cap dret amb ningú usurpi un dret, és tan difícil com resoldre un logaritme discret.

b) Si es fa servir un criptosistema de clau pública segur, la compartició de drets entre clients no compromet els seus drets no compartits.

c) Si es fa servir un criptosistema de clau pública segur, l'única manera que un client té de transferir alguns dels seus drets a un altre és per mitjà de l'autoritat.

Noteu que la intransferibilitat es refereix a la impossibilitat de fer transferències parcials de drets: la penalització que pateix un client *c* per transferir

alguns drets a c' és que tots els drets de c passen a c'. La transferència total de drets o alienació és inevitable: n'hi ha prou que c reveli la seva identitat (clau secreta, paraula de pas, PIN, etc.) a c'.

5.1. Situació inicial

Un servidor s'anomena **servidor independent de client** si no guarda informació d'accés protegida sobre els seus clients potencials (llistes d'accés ni capacitats).

Siguin p i α dos nombres públics, en què p és un primer gran i α és un generador de $\mathbb{Z}_p^*$. Prenem $p \gg (mn)^2$, on m és el nombre de clients i n és el nombre de drets. Considerem qualsevol criptosistema de clau pública segur (per exemple, l'RSA amb mòdul $N > p$).

Sigui $E(\ )$ la transformació de xifratge amb la clau pública i $D(\ )$ la transformació de desxifratge amb la clau privada. Per a l'RSA, $E(x) = x^e \bmod N$ i $D(y) = y^d \bmod N$.

Suposem que la transformació $E(\ )$ és pública, mentre que $D(\ )$ només és coneguda pels servidors. Així, doncs, la raó per a suposar que els servidors són fiables és que han de guardar la transformació privada $D(\ )$.

5.2. L'esquema i les seves propietats

Sigui *auth* l'autoritat central de la xarxa. En presència de diversos clients $c_0, \ldots, c_{m-1}$, cal trobar una manera de concedir el mateix dret a més d'un client, tot fent que l'expressió numèrica y del dret sigui única (els drets també han de ser independents del client). L'algorisme següent dóna una solució per a aquest problema. Per a donar a un client c els seus drets inicials $y_0, \ldots, y_{n-1}$, l'autoritat *auth* fa aquest **algorisme de concessió de drets**:

1) Suposant que els t primers drets entre els $y_0, \ldots, y_{n-1}$ han estat concedits a algun altre client en el passat, tria $n - t$ enters aleatoris x_i, $t \leq i \leq n - 1$ sobre $\mathbb{Z}_{p-1}$.

2) Tria un nombre aleatori a sobre $\mathbb{Z}_{p-1}$, tal que $\mathrm{mcd}(a, p - 1) = 1$.

3) Genera n enters aleatoris r_i sobre $\mathbb{Z}_{p-1}$, tals que $\mathrm{mcd}(r_i, p - 1) = 1$, per a $0 \leq i \leq n - 1$.

4) Troba n nombres z_i no nuls sobre $\mathbb{Z}_{p-1}$, per a $0 \leq i \leq n - 1$, tals que:

$$x_0 + r_0 = az_0 \bmod (p - 1),$$
$$\vdots$$
$$x_{n-1} + r_{n-1} = az_{n-1} \bmod (p - 1), \qquad (5.1)$$

on, per a $0 \leq i \leq t - 1$, x_i és tal que $y_i = \alpha^{x_i} \bmod (p - 1)$*. Per a calcular z_i cal trobar z_i resolent la i-èsima equació fent servir que a té invers a $\mathbb{Z}_{p-1}$.

*** L'autoritat guarda els logaritmes dels drets ja concedits.**

5) Calcula $y_i := \alpha^{x_i} \bmod p$, per a $t \leq i \leq n - 1$ i afegeix aquests nombres juntament amb llur significat –és a dir, allò a què el dret y_i dóna accés– a la llista pública i certificada de drets disponible tant per als servidors com per als clients.

6) Dóna els nombres z_i, $E(r_i)$, per a $0 \leq i \leq n - 1$ a c de manera pública.

7) Dóna el nombre a a c de manera confidencial. a és la identitat del client c.

Algunes remarques addicionals sobre la concessió de drets són les següents:

a) Si el mateix dret es dóna a dos clients diferents, c i c', l'autoritat els dóna parells, $(z_i,E(r_i))$ i $(z'_i,E(r'_i))$, diferents, ja que cada parell depèn de la identitat del client (a o a').

b) És possible donar en públic un dret addicional $y_n = \alpha^{x_n} \bmod p$ a un client que té drets $y_i = \alpha^{x_i} \bmod p$, per a $0 \leq i \leq n - 1$ i un nombre secret a. Amb l'algorisme de concessió de drets, només cal que *auth* triï un enter aleatori r_n sobre $\mathbb{Z}_{p-1}$, tal que $x_n + r_n = az_n \bmod (p - 1)$. Els $(z_n,E(r_n))$ resultants són donats finalment en públic al client.

c) Per a revocar drets cal que l'autoritat publiqui una nova llista certificada de drets; llavors, per a cada client c, *auth* publica nous nombres z_i, $E(r_i)$ corresponents als drets que manté el client.

d) Per a cada client c, es compleix el teorema següent (**teorema 1**): si l'autoritat *auth* ha completat l'algorisme de concessió de drets per a un client c, llavors c és capaç de mostrar possessió dels seus drets $y_0, \ldots, y_{n-1}$ (o d'un subconjunt d'aquests) a qualsevol servidor de la xarxa, que no necessita saber res de c prèviament. N'hi ha prou que el client demostri coneixement d'un sol logaritme, independentment del valor n. Per a un client que no comparteix cap dret, robar-ne un sembla tan difícil com resoldre un logaritme discret.

Demostració del teorema 1

1) Correcció. Per a obtenir accés al servei representat pel dret y_i, un client c i un servidor segueixen el protocol que especifiquem tot seguit.

Protocol d'accés a un servei

a) c dóna al servidor enters A ($\neq 1$), z_i i E_i ($\neq 0$) que satisfan l'equació següent:

$$y_i\,\alpha^{D(E_i)} = A^{z_i} \bmod p. \qquad (5.2)$$

b) *c* demostra el seu coneixement de $\log_\alpha A$ sobre $\mathbb{Z}_{p-1}$ (això es pot provar en coneixement nul amb els protocols 1 o 2 de D. Chaum i altres (1988)). Remarquem que la identitat *a* fou donada a *c* en el darrer pas de l'algorisme de concessió de drets, i és immediat a partir de les equacions 5.1 que $A := \alpha^a \bmod p$ satisfà l'equació 5.2 quan $E_i = E(r_i)$ i es fa servir el mateix z_i a la *i*-èsima equació 5.1 i a l'equació 5.2.

c) El servidor comprova que z_i i E_i satisfan l'equació 5.2. Si aquesta comprovació i la demostració del pas **b** han tingut èxit, el servidor concedeix accés al servei representat per y_i.

2) Seguretat. L'equació 5.2 és verificable pel servidor atès que y_i és public i certificat. Tot seguit mostrarem que el protocol d'accés a un servei fa molt difícil que un client *c* usurpi un dret y_i que no posseeix. La usurpació només és possible si *c* pot proporcionar al servidor una tripla, z_i, E_i, A, que satisfaci l'equació 5.2. Però si *c* pot trobar aquesta tripla, llavors pot calcular el logaritme discret de y_i com:

$$x_i = \log_\alpha y_i = D(E_i) + az_i \bmod (p - 1).$$

Per tant, la usurpació és almenys tan difícil com calcular el logaritme discret de y_i, i això sembla tan complicat com el problema del logaritme discret en general. Observeu que la dificultat de trobar $D(\ )$ no és rellevant, perquè implícitament suposem que l'usurpador *c* coneix $a = \log_\alpha A$ i $D(\ )$.

La demostració conclou i el servidor no ha necessitat cap informació prèvia sobre *c*.

Per tal d'obtenir accés simultani...

... als drets representats per $y_0, \ldots, y_n$, es pot seguir el mateix protocol, tot considerant n equacions en comptes de l'única equació 5.2 del pas **a** i fent que el servidor faci n comprovacions al pas **c**. El pas **b** queda igual.

Altres propietats de l'esquema

- **Propietat de seguretat de la compartició de drets**. Si es fa servir un criptosistema de clau pública segur, no és factible per a un client *c* de determinar la identitat d'un altre client *c'* –i robar-li així els drets no compartits– aprofitant que *c* i *c'* comparteixen un dret o un grup de drets.

- **Propietat d'intransferibilitat dels drets**. Si es fa servir un criptosistema de clau pública segur i els clients no alienen les seves identitats, no és factible per a un client *c* transferir un dret a un altre client *c'* servint-se del fet que *c* posseeix el dret.

Nota

Pel seu caràcter tècnic, deixem com a exercici d'autoavaluació la demostració de la propietat de seguretat de la compartició de drets.

Demostració de la propietat d'intransferibilitat dels drets

Gràcies a l'ús d'aleatorització i al xifratge subsegüent dels nombres aleatoris, cap dels enters x_i, r_i situats al membre esquerre de les equacions 5.1 no és conegut per un client *c* que posseeix un dret y_i. La possessió del dret només vol dir que l'autoritat ha emès dos nombres públics, z_i, E_i, que relacionen la identitat *a* de *c* amb el dret y_i segons l'equació 5.2. Ara bé, perquè *c* pugui transferir y_i a un altre client *c'*, li cal trobar un parell z'_i, E'_i tal que:

$$x_i + D(E'_i) = a'z'_i \bmod (p - 1) \qquad (5.3)$$

c pot eliminar la incògnita x_i de l'equació 5.3 i de l'equació anàloga amb els seus valors E_i, a, z_i, amb la qual cosa obté l'equació següent:

$$D(E'_i) - D(E_i) = a'z'_i - az_i \bmod (p - 1) \qquad (5.4)$$

Sense conèixer $D(\ )$ ni a' (se suposa que no hi ha alienació de *c'* en *c*), sembla difícil de trobar valors E'_i, z'_i que satisfacin l'equació 5.4.

L'única manera de transferir un dret y_i és per mitjà de l'autoritat. En efecte, de manera anàloga a la demostració del Teorema 1, *c* demostra a l'autoritat la possessió de y_i; això implica que *c* s'ha d'autentificar demostrant coneixement de la seva identitat *a*. El receptor *c'* s'autentifica a l'autoritat demostrant

possessió de la seva identitat a' en coneixement nul. Finalment, en funció de la política de seguretat vigent, l'autoritat pot decidir transferir a c' el dret y_i.

5.3. Recapitulació i aplicacions

Tal com acabem de mostrar, l'esquema proposat per a concedir drets intransferibles és molt flexible, ja que la gestió dels clients es pot fer independentment dels servidors i no cal canviar el secret a d'un client per a concedir-li o revocar-li drets.

Si diem que dues entitats són mútuament dependents quan hi ha alguna informació secreta que comparteixen, hem mostrat que les dependències funcionals entre les diferents entitats del sistema són les de la taula següent:

Dependències funcionals				
Depèn de	**Autoritat**	**Client**	**Servidor**	**Dret**
Autoritat	–	Sí	No	Sí
Client	Sí	–	No	No
Servidor	No	No	–	No
Dret	Sí	No	No	–

Les úniques dependències que hi ha són entre una comunitat de clients i l'autoritat que els va donar llur identitat, i també entre un conjunt de drets i l'autoritat que els publica i els certifica en una llista.

Per tant, veiem que, a més de ser independents del client, els servidors són també independents de l'autoritat. Això suggereix estendre l'esquema proposat de tal manera que diverses autoritats (cadascuna amb la seva comunitat de clients i llista de drets) comparteixin el mateix conjunt de servidors.

Exemples d'aplicació de la concessió de drets

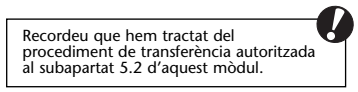
A continuació us oferim uns quants exemples d'aplicació de l'esquema de concessió de drets intransferibles.

1) Considereu un sistema de control d'accés per a la seu d'una empresa, en la qual:

- L'autoritat és l'empresa, representada per un ordinador de control connectat a una xarxa d'àrea local.
- Els clients són els empleats representats per les seves respectives targetes intel·ligents.
- Els servidors són dispositius especials connectats a la xarxa d'àrea local i situats a prop de les portes dels diversos edificis i cambres de la seu.
- Els drets permeten l'accés a les diferents cambres en diverses franges horàries.
- La llista certificada de drets és un fitxer públic signat digitalment que manté l'ordinador de control i que és baixat pels servidors de porta cada vegada que detecten que ha estat actualitzat.

Quan es contracta un empleat, se li dóna una targeta intel·ligent inicialitzada amb un nombre secret a (juntament amb alguns drets inicials, seguint l'algorisme de concessió de drets). Les actualitzacions dels nombres z_i, E_i corresponents als drets de cada empleat es difonen regularment per mitjà de les pàgines web o dels butlletins interns de l'empresa. Per comoditat, aquests nombres també es troben disponibles en un fitxer públic guardat a l'ordinador de control; els nombres que afecten els drets d'un empleat es poden carregar a la targeta intel·ligent de l'empleat inserint-la en qualsevol ordinador connectat a la xarxa.

Si un empleat vol entrar en una cambra, insereix la seva targeta al servidor de porta i s'enceta el protocol d'accés a un servei entre la targeta i el servidor. Per tal que un empleat pugui transferir un dret a un altre empleat, tots dos han de recórrer a l'ordinador remot i seguir el procediment de transferència autoritzada. L'ordinador de control decideix sobre la transferència segons la política de seguretat de l'empresa i manté un registre de les transferències que ha autoritzat amb finalitat d'auditoria.



En aquest exemple, els servidors de porta poden ser microordinadors de baix preu que contenen un rellotge, sistema de circuits per a obrir la porta, un petit processador, una memòria ROM normal per a implementar el protocol d'accés a un servei, una ROM protegida per a guardar de manera segura una clau privada, una memòria RAM per a guardar la llista actual de drets i una connexió de xarxa.

Les targetes intel·ligents consten d'un processador, una memòria ROM normal per a implementar el protocol d'accés a un servei i el de transferència autoritzada, una ROM protegida per a guardar la identitat a i una memòria de lectura/escriptura per a guardar-hi els nombres actuals z_i, E_i per cada dret y_i.

La concessió de drets i l'alta d'empleats poden ser duts a terme per l'autoritat independentment dels servidors. Tant la concessió com la revocació de drets són procediments públics.

2) A tall d'exemple amb diverses autoritats, considereu una xarxa de caixers automàtics compartida per diverses entitats emissores de targetes de crèdit:

- Els emissors de targetes són les autoritats, cadascun amb la seva comunitat de clients (els usuaris de les targetes) i un conjunt de servidors compartits (els caixers automàtics).

- Els drets permeten efectuar diverses operacions bancàries.

Gràcies a la simplicitat i a la independència de la funcionalitat del servidor, el nostre esquema és molt apropiat per a un escenari tan complex.

6. Eleccions electròniques

Si volem traslladar el mecanisme electoral del món físic (les paperetes, els sobres, la mesa electoral, el desplaçament del votant al col·legi, etc.) al món virtual (cada elector vota des del seu ordinador connectat a Internet), els problemes de seguretat que apareixen no són gens trivials.

Les **eleccions electròniques** poden ser vistes com l'exemple típic de càlcul segur a múltiples bandes. Recordem que la formulació general d'aquest problema és que hi ha m participants, cadascun dels quals té la seva entrada privada, x_i. Es vol calcular el resultat d'una funció n-dimensional $f(x_1, ..., x_m)$ sense que els participants s'hagin de revelar mútuament les seves entrades x_i.

En el cas de les eleccions electròniques, els participants són els votants, les seves entrades són un valor binari, la funció que es calcula és una suma i el resultat és el recompte de vots.

En general, la finalitat és que un protocol d'eleccions electròniques compleixi els requisits següents:

1) Només poden votar les persones autoritzades.

2) Ningú no pot votar més d'una vegada.

3) El secret del vot és preservat.

4) Ningú no pot duplicar el vot d'algú altre.

5) El recompte de vots es fa correctament.

6) Qualsevol hauria de poder comprovar la correcció del recompte.

7) El protocol hauria de ser tolerant a fallades, en el sentit que hauria de funcionar correctament fins i tot en presència d'un cert nombre de participants deshonestos.

8) Hauria de ser impossible obligar un votant a revelar què ha votat*.

*** Per exemple, per evitar la compra de vots.**

Habitualment, en un protocol d'elecció no és desitjable fer participar tots els votants V_i en el procés de càlcul. Per tant, suposem que hi ha n col·legis

electorals, C_1, ..., C_n, la tasca dels quals és recollir els vots i calcular-ne el recompte.

6.1. El protocol d'elecció de Merritt

En l'esquema de M. Merritt (1983), cada col·legi publica una clau pública, E_i, i manté secreta la clau privada corresponent. Per tal d'emetre el seu vot v_j, cada votant V_j tria un nombre aleatori s_j i calcula:

$$E_1(E_2(\ ...\ E_n(v_j,s_j)\ ...\ )) = y_{n+1,j}. \tag{6.1}$$

L'ús del subíndex $n + 1$ quedarà justificat tot seguit. El pas següent és publicar els valors y. Començant pel col·legi C_n i acabant per C_1, cada centre C_i fa el següent: per a cada $y_{i+1,j}$, C_i tria un valor aleatori $r_{i,j}$ i publica $y_{i,\pi_i(j)} = E_i(y_{i+1,j}, r_{i,j})$. $\pi_i(j)$ és la imatge de j per una permutació aleatòria π_i dels enters $[1 \ldots n]$. El col·legi C_i guarda la permutació en secret.

Al final tenim l'equació següent:

$$\begin{gathered} y_{1,\pi_1(\pi_2(\ ...\ \pi_n(j)\ ...\ ))} = \\ = E_1(E_2(\ ...\ E_n(y_{n+1,j},r_{n,j})\ ...\ r_{2,\pi_3(\ ...\ \pi_n(j)\ ...\ )})\ r_{1,\pi_2(\ ...\ \pi_n(j)\ ...\ )}). \end{gathered} \tag{6.2}$$

En aquest punt, comença el cicle de verificació. Per cada vot rebut, es fan dues iteracions de desxifratge en l'ordre següent:

$$C_1 \rightarrow C_2 \rightarrow \ ... \ \rightarrow C_n.$$

Per a cada j, la primera iteració extreu $y_{n+1,j}$ a partir de l'equació 6.2. Per a cada j, la segona iteració extreu v_j de $y_{n+1,j}$ (equació 6.1). Els valors desxifrats són anunciats i el recompte es calcula com la suma dels vots v_j. Noteu que cada iteració requereix desxifrar amb les claus privades de tots els col·legis; per tant, el recompte requereix la participació de tots els col·legis.

Clarament, se satisfan els requisits 1 i 2. També se satisfà el requisit 3, ja que en revelar els vots es manté secreta la connexió entre el vot i el votant que l'ha emès. Per a reconstruir aquesta connexió caldria conèixer totes les permutacions π_i.

El requisit 4 no se satisfà, perquè el votant V_1 pot copiar el votant V_2, per exemple, emetent la mateixa cadena de caràcters xifrada (vegeu l'equació 6.1).

Els requisits 5 i 6 se satisfan amb l'ús de cadenes aleatòries:

a) Per a cada vot, cada col·legi comprova durant la primera iteració de desxifratge que la seva cadena aleatòria apareix en el valor desxifrat. Així, els

col·legis s'asseguren que tots els seus textos xifrats (corresponents als diferents vots) es tenen en compte.

b) Per a cada vot v_j, cada votant j cerca la seva cadena s_j per assegurar-se que el seu vot és comptat. Si els s_j són prou llargs i aleatoris, la probabilitat que n'hi hagi dos d'iguals és negligible. Noteu que, per a verificar la correcció de l'elecció, cal la cooperació de tots els votants, la qual cosa és negativa en grans eleccions.

El requisit 7 requereix un matís. Si ens preocupa que la presència de participants deshonestos pugui trencar el secret de vot, llavors el protocol de Merritt és perfecte. Efectivament, fins i tot en el cas que hi hagi $n - 1$ col·legis deshonestos, no aconseguiran saber qui ha emès un determinat vot. Per a trencar el secret, cal conèixer totes les permutacions π_i. No obstant això, si ens preocupa la tolerància a fallades, el protocol és nefast; només que un dels col·legis falli (per exemple, per avaria informàtica), tot el sistema cau i cal repetir tot el procés d'eleccions.

El requisit 8 no se satisfà. El votant pot ser obligat a revelar v_j i s_j i, si intenta mentir sobre el seu vot, serà descobert perquè els valors declarats no encaixaran amb el text xifrat $y_{n+1,j}$.

6.2. Altres protocols

Una proposta recent de Cramer i altres (1996) aconsegueix satisfer el requisit 4 (còpia de vots impossible). Aquesta possibilitat no requereix que tots els votants cooperin en la verificació del recompte (millor solució per al requisit 6) i proporciona tolerància a fallades (requisit 7). Malauradament, l'extensió i la complexitat tècnica de la proposta no fan aconsellable incloure-la en aquest text.

Lectura recomanada

Per a obtenir una panoràmica més àmplia sobre eleccions electròniques, podeu consultar l'obra següent:

J. Borrell (1996). *Estudi i desenvolupament d'un esquema criptogràfic per realitzar votacions segures sobre una xarxa local* (Tesi doctoral). Barcelona: Universitat Autònoma de Barcelona.

Resum

En aquest mòdul didàctic hem fet referència a **protocols d'autentificació i d'identificació**. En particular, hem descrit el **protocol de tres passos de Shamir**, que no necessita cap intercanvi de claus públiques ni privades, però que requereix un sistema de xifratge simètric respecte de la clau. També hem vist el **protocol d'identificació de Fiat-Shamir**, que és una prova de coneixement nul. De fet, però, una descripció més acurada de les proves de coneixement nul s'ha fet en el subapartat 1.3, al qual hem descrit el **protocol de D. Chaum, J.H. Evertse i J. van de Graaf** que demostra la possessió del logaritme discret amb una prova de coneixement nul.

Hem descrit les característiques principals dels **esquemes de compartició de secrets** i els camps d'aplicació que té. Hem vist com funcionen els esquemes vectorials, com també els d'interpolació polinòmica proposats per Shamir.

Pel que fa a les **situacions de desconfiança mútua**, hem estudiat diversos protocols com el **compromís de bit** i la **transferència inconscient**, que, tot i no tenir una aplicació directa, són una peça fonamental en els protocols d'intercanvi de secrets, la signatura de contractes o el correu electrònic certificat. D'altra banda, hem descrit l'escenari del **càlcul segur a múltiples bandes** tot donant com a exemple el problema del milionari. Hem donat també la definició d'**homomorfisme de privacitat** i hem descrit la utilització que té per al càlcul sobre dades xifrades.

Els **sistemes de diner electrònic** haurien d'impossibilitar la falsificació, fer difícil la despesa múltiple, preservar l'anonimat del client i minimitzar les operacions *on line*. Els sistemes de targeta magnètica actuals no satisfan els dos darrers requisits. En canvi, hem presentat una solució criptogràfica que compleix tots els requisits anteriors.

En una xarxa de servidors amb múltiples autoritats, cadascuna de les quals té una comunitat de clients, hem mostrat com les autoritats poden concedir i revocar drets intransferibles als seus clients de manera pública. Un client no pot transferir a un altre part dels seus drets sense haver tingut en compte l'autoritat pertinent. L'esquema proposat té l'avantatge que la gestió dels drets per part de l'autoritat no requereix comunicació secreta amb els servidors.

Finalment, hem enumerat els requisits que ha de complir un **sistema d'elecció electrònica** (votació per mitjà d'Internet) si vol oferir les mateixes garanties que un sistema d'elecció convencional (amb paperetes i sobres). A tall d'il·lustració, hem esbossat un protocol relativament senzill que satisfà alguns dels requisits enumerats.

Activitats

1. Visiteu la pàgina web http://www.digicash.com. Hi trobareu informació sobre sistemes de pagament comercials semblants al que hem descrit, en el sentit que preserven l'anonimat del client.

2. Visiteu la pàgina web http://www.mastercard.com. Cerqueu-hi informació sobre el sistema SET (*Secure Electronic Transactions*), que, amb el suport de VISA i MasterCard, es perfila com l'estàndard *de facto* en pagaments electrònics. En la seva versió actual, el sistema SET no proporciona anonimat per al client.

3. Proposeu un exemple d'aplicació de l'esquema de drets intransferibles diferent dels proposats al text.

Recordeu que hem vist un exemple d'aplicació de l'esquema de drets intransferibles al subapartat 5.3 d'aquest mòdul.

4. Analitzeu com es compleixen els requisits 1 a 8 en les eleccions convencionals. Quins avantatges i quins inconvenients tindria la "democràcia directa", que consisteix en la votació via Internet de totes les decisions importants per al ciutadà?

Recordeu que hem estudiat els requisits que ha de complir un protocol d'eleccions electròniques a l'apartat 6 d'aquest mòdul.

Exercicis d'autoavaluació

1. Donats els paràmetres següents, reproduïu el protocol d'identificació de Fiat-Shamir:
- L'identificador de l'usuari és $I = 41$.
- El valor n triat per l'autoritat certificadora val $n = 7 \cdot 13 = 91$.
- La funció *hash* utilitzada és $f(I,i) \equiv I^i \bmod n$.
- El nombre de valors que l'autoritat certificadora calcula és $k = 10$.
- Els índexs triats són un total de $t = 4$, i corresponen als valors $(i_1,i_2,i_3,i_4) = (2,4,8,10)$.

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

3. Tant a l'Anna com a en Bernat els ha tocat el sorteig dels cecs, que reparteix fins a quatre milions. L'Anna ha tingut més sort i ha obtingut quatre milions mentre que en Bernat només n'ha rebut dos. Cap d'ells no vol dir la quantitat que li ha tocat però volen saber a qui dels dos els ha tocat més. Desenvolupeu el protocol del problema del milionari per tal que tots dos puguin saber qui ha guanyat més sense saber quant li ha tocat a l'altre.

Nota

Per a resoldre aquest exercici feu servir com a sistema de clau pública l'RSA amb $n = 55$ i el parell de claus $(D_A = 3, E_A = 27)$.

4. Comproveu que el xifratge RSA és un homomorfisme de privacitat que conserva el producte.

5. En aquest mòdul didàctic hem esmentat dues solucions per a evitar l'engany al banc en un sistema de pagament. La primera és tenir una signatura per cada valor de bitllet (denominació). La segona és el procediment de "remenar i triar". Cerqueu un avantatge de cada solució respecte de l'altra.

6. En el mecanisme de detecció de despesa múltiple *off line*, el venedor V pot intentar dipositar dos bitllets idèntics amb diferents TAI, per mirar d'inculpar algun client. Per a aconseguir-ho, li caldria endevinar una TAI per al segon bitllet que, juntament amb la TAI del primer bitllet, donés el nom d'algun client. Quina probabilitat té V de sortir-se'n?

7. Demostreu la propietat de seguretat de la compartició de drets de l'esquema de drets intransferibles.

Vegeu la propietat de seguretat de la compartició de drets de l'esquema de drets intransferibles que hem tractat al subapartat 5.2 d'aquest mòdul.

8. Quina és la funció dels enters aleatoris s_j i $r_{i,j}$ al protocol d'elecció de Merritt?

Solucionari

Exercicis d'autoavaluació

1. Amb les dades donades es pot confeccionar la taula següent:

Índex i	$v_i \equiv I^i \bmod n$	$v_i^{-1} \bmod n$	$s_i \equiv \sqrt{v_i^{-1}} \bmod n$
1	41	20	–
2	43	36	6
3	34	83	–
4	29	22	29
5	6	76	–
6	64	64	8
7	76	6	–
8	22	29	22
9	83	34	–
10	36	43	15

Com que $(i_1,i_2,i_3,i_4) = (2,4,8,10)$, tenim que l'usuari haurà rebut de l'autoritat certificadora l'identificador $I = 41$ i els valors $(s_2,s_4,s_8,s_{10}) = (6,29,22,15)$ que només ell coneix i que determinaran la seva identitat.
Per tant, quan l'usuari es vulgui identificar haurà d'executar el protocol següent:
1) P enviarà a V, $I = 41$ i $(i_1,i_2,i_3,i_4) = (2,4,8,10)$.
2) V utilitza la funció *hash* $f(I,i) \equiv I^i \bmod n$ per a calcular v_{i_j}:
- $v_{i_1} = v_2 = 41^2 = 43 \bmod 91$.
- $v_{i_2} = v_4 = 41^4 = 29 \bmod 91$.
- $v_{i_3} = v_8 = 41^8 = 22 \bmod 91$.
- $v_{i_4} = v_{10} = 41^{10} = 36 \bmod 91$.

3) P tria un valor aleatori, posem per cas $r = 58$ i envia a V el valor $x = 58^2 \bmod 91 = 88$.
4) V tria un vector binari aleatori, per exemple, $(b_1,b_2,b_3,b_4) = (0,1,1,1)$ i l'envia a P.
5) P utilitza aquest vector aleatori per a calcular $y = rs_4s_8s_{10} = 58 \cdot 29 \cdot 22 \cdot 15 = 51 \bmod 91$.
6) Finalment, V comprova que $x = y^2v_4v_8v_{10}$. En efecte, $88 = 51^2 \cdot 29 \cdot 22 \cdot 36 = 88 \bmod 91$.
Al final d'aquest procés P ha pogut enganyar V amb una probabilitat d'$1/2^t = 1/2^4$. Si volem disminuir aquesta probabilitat repetirem el procés des del pas **3** fins al **6** les vegades que vulguem.

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 els nombres aleatoris, que poden ser $x_1 = 8$ i $x_2 = 7$. D'aquesta manera, el polinomi és $P(x) = 7x^2 + 8x + 11 \bmod 13$.
Per a generar els fragments prenem 5 valors menors que p i calculem les seves imatges pel polinomi $P(x)$. Prenent com a valors {1, 2, 3, 4, 5} tindrem:
- $P(1) = 7 + 8 + 11 = 0 \bmod 13$.
- $P(2) = 28 + 16 + 11 = 3 \bmod 13$.
- $P(3) = 63 + 24 + 11 = 7 \bmod 13$.
- $P(4) = 112 + 32 + 11 = 12 \bmod 13$.
- $P(5) = 175 + 40 + 11 = 5 \bmod 13$.

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

3. De l'enunciat es desprèn que $i = 4$, $j = 2$, $1 \leqslant u \leqslant 5$, $n = 55$ i $(D_A = 3,E_A = 27)$. Llavors:
1) B tria un valor aleatori, per exemple, $x = 37$, i el xifra amb la clau pública de l'Anna, $E_A = 27$, és a dir, $k = 37^{27} \bmod 55 = 38$.
2) B envia a A, $k - j = 38 - 2 = 36$.
3) A amb la seva clau privada $D_A = 3$ calcula:
- $y_1 = (38 - 2 + 1)^3 \bmod 55 = 37^3 \bmod 55 = 53$.
- $y_2 = (38 - 2 + 2)^3 \bmod 55 = 38^3 \bmod 55 = 37$.
- $y_3 = (38 - 2 + 3)^3 \bmod 55 = 39^3 \bmod 55 = 29$.
- $y_4 = (38 - 2 + 4)^3 \bmod 55 = 40^3 \bmod 55 = 35$.

4) A tria un primer $p < x = 37$, per exemple, $p = 31$ i calcula:
- $z_1 = 53 \bmod 31 = 22$.
- $z_2 = 37 \bmod 31 = 6$.
- $z_3 = 29 \bmod 31 = 29$.
- $z_4 = 35 \bmod 31 = 4$.

Com podem veure, el valor de p triat és correcte perquè es compleix que $0 < z_u < p - 1$ i $|z_u - z_v| \geq 2$.

5) A envia a B la seqüència següent:

$$z_1, z_2, z_3, z_4, p,$$

que en el nostre cas és:

$$22, 6, 29, 4, 31.$$

6) B comprova si $z_2 \stackrel{?}{=} x \bmod p$. En aquest cas, com que $6 = 37 \bmod 31$, B pot concloure que $j = 2 \leqslant 4 = i$ i, per tant, sap que ell ha guanyat menys diners que l'Anna.

4. Sigui $n = pq$, $E(x) = x^e \bmod n$ i sigui d tal que $ed \equiv 1 \bmod \phi(n)$. Hem de comprovar que:

$$E(x \cdot y) \stackrel{?}{=} E(x)E(y).$$

Vegeu la funció totient d'Euler $\phi(n)$ al subapartat 1.2 del mòdul "Xifres de clau pública" d'aquesta assignatura.

Així, trobem que:

$$E(x \cdot y) = (x \cdot y)^e \bmod n = x^e y^e \bmod n = \\ = (x^e \bmod n)(y^e \bmod n) = E(x) \cdot E(y).$$

5. Tenir una signatura per a cada denominació és més simple de càlcul que no pas per mitjà del procediment de "remenar i triar" (que requereix generar molts bitllets, tapar-los i destapar-los). En canvi, l'avantatge del procediment de "remenar i triar" és que permet generar bitllets de qualsevol valor (i en qualsevol divisa); adoneu-vos que tenir una signatura per a cada denominació implica disposar d'un nombre finit de denominacions, amb la qual cosa tenim un problema a l'hora de tornar canvi.

6. Perquè dues TAI diferents siguin acceptables, per a cada índex j o bé tenim el mateix valor o bé tenim $x_j \neq x'_j$ amb $x_j \oplus x'_j =$ nom de U. El més senzill per a V és fer que la TAI falsa difereixi de l'autèntica en només un índex (posem $j = 1$). Llavors, si a la TAI autèntica hi havia x_1, V ha d'endevinar x'_1 tal que $x_1 \oplus x'_1 =$ nom de U per a algun usuari U. Si hi ha m_1 usuaris (amb noms diferents) i la llargada dels nombres x_j, x'_j i dels noms d'usuari és m_2 bits, la probabilitat que té V d'inculpar en fals un usuari és $m_1/2^{m_2}$. Noteu que hi ha m_1 noms d'usuari vàlids (casos favorables) sobre un total de 2^{m_2} noms d'usuari possibles (casos possibles).

7. Quan els clients c i c' comparteixen un dret y_i, no és probable que comparteixin el membre esquerre de cap de les equacions 5.1, atès que al logaritme x_i del dret s'hi han sumat nombres aleatoris diferents r_i, r'_i. La probabilitat de triar a $\mathbb{Z}_{p-1}$ nombres aleatoris diferents en les equacions (5.1) per a m clients i n drets és:

$$\frac{(p-1)(p-2)\ldots(p-mn)}{(p-1)^{mn}},$$

que tendeix a 1 si $p - 1 \gg (mn)^2$. Per tant, l'única igualtat que es pot establir en termes d'exponents quan c i c' comparteixen un dret resulta de les equacions 5.1 i és:

$$D(E(r_i)) - D(E(r'_i)) = az_i - a'z'_i \bmod (p-1).$$

Ara bé, a l'equació anterior c coneix $E(r_i)$, $E(r'_i)$, z_i, z'_i i la seva identitat a. Per a poder trobar a' (identitat de c'), c necessitaria conèixer $D(E(r_i)) - D(E(r'_i))$ (que no és 0, ja que hem justificat que $r_i \neq r'_i$), la qual cosa és difícil perquè la transformació $D(\ )$ només és coneguda pels servidors. Finalment, per a apropiar-se dels drets de c', no hi ha cap més opció per a c que trobar a'.

8. El valor s_j serveix per a tapar el vot, és a dir, perquè no es pugui trobar v_j encriptant totes les opcions de vot i mirant quina és la que dóna $y_{n+1,j}$. Recordeu que els valors $y_{n+1,j}$ s'anuncien públicament, amb la finalitat que només el propi votant (que sap s_j) pugui reconèixer el seu vot i sàpiga que s'està tenint en compte (prèviament al recompte). Els valors $r_{i,j}$ compleixen una funció semblant una vegada que es publiquen els vots (final del recompte); si no hi fossin, quan es revelen els vots es podria saber a quin votant j correspon un vot a base d'anar provant quin $y_{n+1,j}$ dóna el $y_{1,*}$ publicat pels col·legis.

Glossari

Alienació: transferència de la pròpia identitat (i, per tant, de tots els drets) d'un client a un altre client.

Compromís de bit: 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 mostrar b a B.

DES: *Data Encryption Standard*. Criptosistema de xifratge de bloc que xifra blocs de dades de 64 bits de llargada per mitjà d'una clau de 56 bits i l'acció de caixes S.

Despesa múltiple: despesa per part del client dels mateixos diners electrònics dues o més vegades.

Diners electrònics: tècniques i protocols criptogràfics que pretenen recrear el concepte de *compres en metàl·lic* en una xarxa oberta com Internet.

Esquema de compartició de secrets: 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 de llindar (m,n): esquema de compartició de secrets en què el secret es divideix en n trossos i se'n necessiten m per tal de recuperar-lo. A més, $m - 1$ trossos no dóna cap informació del secret.

Esquema de compartició de secrets d'interpolació polinomial: esquema de compartició de secrets en què 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 vectorial: esquema de compartició de secrets en què els fragments del secret són hiperplans d'un espai vectorial en el qual la intersecció d'un cert nombre d'hiperplans és el secret.

Falsificació: alteració d'una cosa per tal d'enganyar. Un client falsifica diners electrònics si aconsegueix més diners electrònics dels que el banc li carrega en compte.

Homomorfisme de privacitat: esquema de xifratge en què la funció xifradora $E(\ )$ conserva una o més operacions, és a dir, suposant $*$ una operació genèrica, $E(x * y) = E(x) * E(y)$.

HP: Homomorfisme de privacitat.

Intransferibilitat: qualitat d'un dret que fa que un client, sense el concurs de l'autoritat i sense alienació, no pugui transferir-lo a un altre client.

Operació *off line*: pagament electrònic efectuat sense necessitat d'accedir a la base de dades del banc per a completar el pagament.

Operació *on line*: durant el protocol de pagament, accés a una base de dades del banc (de bitllets electrònics ja despesos) necessari per a poder cloure el pagament amb èxit.

PIN: número d'identificació personal.

Prova de coneixement nul: protocol criptogràfic pel qual un participant A demostra a un altre B el coneixement d'alguna informació sense revelar-ne cap detall.

Prova d'informació mínima: nom que es dóna de vegades a les proves de coneixement nul, perquè s'argumenta que en general les proves de coneixement nul sempre donen informació, encara que sigui molt poca, i insuficient per a obtenir el secret del provador.

Provador: part encarregada de demostrar que coneix una informació en una prova de coneixement nul.

Revocació: retirada d'un dret a un client per l'autoritat.

RSA: criptosistema de clau pública, creat per Rivest, Shamir i Adleman i basat en el problema de la factorització.

Signatura tapada: presentació d'un bitllet al banc amb un cert emmascarament durant el protocol de reintegrament. El banc signa el bitllet sense veure'n els continguts (bitllet

tapat). D'aquesta manera, el banc no pot determinar qui va retirar el bitllet quan un venedor el presenta durant el protocol de dipòsit.

TAI: tira aleatòria d'identificació.

Tolerància a les fallades: propietat d'un protocol de funcionar correctament fins i tot en el cas de participants deshonestos (fallades intencionals) o de participants avariats (fallades accidentals).

Sistema de preservació de l'anonimat: sistema de pagament que permet, fins i tot en cas de confabulació entre el banc i el venedor, que aquests no sàpiguen en què despèn els diners un client determinat.

Transferència inconscient: protocol criptogràfic pel qual *A* emet un missatge en direcció a *B*. Al final del protocol pot ser que *B* rebi el missatge amb una probabilitat d'1/2, però *A* no sabrà mai si *B* l'ha rebut.

Verificador: part encarregada de verificar una prova de coneixement nul.

Bibliografia

Bibliografia bàsica

Beker, H.; Piper, F. (1982). *Cipher systems, the protection of the communications*. Londres: Northwood Books.

Borrell, J. (1996). *Estudi i desenvolupament d'un esquema criptogràfic per realitzar votacions segures sobre una xarxa local* (Tesi doctoral). Barcelona: Universitat Autònoma de Barcelona.

Chaum, D. (1981). "Untraceable electronic mail, return addresses, and digital pseudonyms". *Communications of the ACM* (vol. 24, pàg. 84-88). Nova York: Association for Computing Machinery.

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

Cramer, R.; Franklin, M.; Schoenmakers, B.; Yung, M. (1996). "Multi-authority secret-ballot elections with linear work". *Advances in Criptology-Eurocrypt'96*. Berlín: Springer-Verlag.

Domingo Ferrer, J. (1994). "Untransferable rights in a client-independent server environment". *Advances in Criptology - Eurocrypt'93*. Berlín: Springer-Verlag.

Domingo Ferrer, J. (1997). "Multi-application smart cards and encrypted data processing". *Future Generation Computer Systems* (vol. 13, juny 1997, pàg. 65-74).

Merritt, M. (1983). *Cryptographic Protocols* (Tesi doctoral). Geòrgia: Georgia Institute of Technology.

Bibliografia complementària

Goldwasser, S.; Micali, S.; Rackoff, C. (1985). "The Knowledge Complexity of Interactive Proof Systems". *Proceedings of the 17th ACM Symposium on Theory of Computing* (pàg. 291-304). Nova York: The Association for Computing Machinery.

Goldwasser, S.; Micali, S.; Rackoff, C. (1989). "The Knowledge Complexity of Interactive Proof Systems". *SIAM Journal on Computing* (vol. 18, núm. 1, pàg. 186-208). Filadèlfia: SIAM Publications.