L'aritmètica en el xifrat i la signatura de missatges*

Amparo López Villacampa

Agraeixo a la Societat Catalana de Matemàtiques i al seu president Dr. Joan Girbau la possibilitat de parlar d'un tema que m'ha interessat en els darrers anys.

Per tal d'evitar des del principi falses expectatives vull començar amb unes consideracions prèvies. El contacte amb la informàtica, a mi que pateixo precisament de bitmania, m'ha fet adonar que la potència de càlcul dels ordinadors amplia el nostre domini del que és finit i delimitat: és cert que tan finit és 2 como 10^{324}, però és obvi que el disseny d'algorismes eficients, el manegament de grans nombres exigeix atendre aspectes matemàtics que poden ésser irrellevants per números petits. Per tant no es pot excloure que calguin nous punts de vista teòrics; i això és de gran interés per al matemàtic, sobretot si no va descaminat el meu convenciment que la informàtica marcarà el desenvolupament futur de la matemàtica de manera anàloga a com la mecànica quàntica va impulsar l'anàlisi funcional.

Des d'aquí, i davant d'un auditori majoritàriament matemàtic, no pretenc altra cosa que cridar l'atenció sobre la criptografia i la seva relació amb la teoria de cossos finits.

I. El concepte de sistema criptogràfic clàssic

Començaré per recordar la noció ben coneguda de llenguatge lliure sobre un alfabet A: si A és un conjunt finit i $A^n = \{(a_1, ..., a_n) \mid a_i \in A\}$ designa les seqüències de longitud n d'elements de A, $\bigcup_{n>0} A^n$ té estructura de monoide amb l'operació de juxtaposició: $(a_1, ..., a_n) \circ (b_1, ..., b_n) = (a_1, ..., a_n, b_1, ..., b_n)$. Aquest monoide s'anomena el monoide, o llenguatge, lliure sobre A, i el designarem —per concessió a l'anglès— *FM(A)*.

Ni els llenguatges naturals ni els llenguatges formals de programació són llenguatges lliures, perquè les seves regles ortogràfiques, gramaticals i sintàctiques els donen una estructura que els fa más rígids: per exemple, certes seqüències són impossibles, la forma d'unes altres queda determinada (en castellà després de «q» sempre hi ha «u» i per tant la «u» és redundant) i la freqüència d'una lletra en el conjunt de seqüències primitives depèn de la lletra (distribució no uniforme de les freqüèn-

* Conferència organitzada per la Societat, pronunciada el 16-3-89.

cies de les lletres). Si he volgut assenyalar aquí aquest fet obvi, és per la seva radical importància en el trencament de xifrats.

Dit això, vaig a formalitzar el concepte de sistema criptogràfic clàssic. Siguin A i B conjunts finits (usualment $A = B =$ conjunt de les 26 lletres de l'alfabet romà) i sigui $P \subset FM(A)$ el conjunt d'expresions lingüístiques amb sentit. P s'anomena el conjunt de missatges.

Sigui K un conjunt finit al qual anomenarem conjunt de claus.

Un sistema criptogràfic clàssic és una família uniparamètrica $\{S_k | k \in K\}$ d'aplicacions invertibles:

$$S_k: FM(A) \to FM(B)$$

Si $M \in P$, es diu que $S_k(M)$ és el seu xifrat o criptograma amb la clau k.

La família $\{S_k | k \in K\}$ se suposa, en criptografia teòrica, coneguda públicament, i això no és un mer conveni sinó la traducció d'una regla tècnica de seguretat: «La seguretat d'un sistema no ha de dependre de res que, un cop conegut, no es pugui canviar fàcilment». Naturalment, en la pràctica, sobretot a l'àmbit militar, s'intenta mantenir secret també el sistema, perquè això dificulta la tasca dels criptoanalistes.

Dos usuaris del sistema criptogràfic que vulguin comunicar-se entre ells, es posen d'acord en una clau comú k la qual determina S_k i S_k^{-1}. Si un d'ells vol enviar a l'altre el missatge M, el xifra fent $S_k(M) = C$ i l'altre quan rep C el desxifra aplicant S_k^{-1} : $S_k^{-1}C = S_k^{-1}S_k(M) = M$.

La innocent frase «es posen d'acord en una clau k» té prou contingut com perquè li dediquem la nostra atenció. Aquest posar-se d'acord s'ha de fer per un mitjà segur, per exemple el correu certificat, i si pensem en l'ús comercial de la criptografia, en el qual hi haurà molts usuaris del mateix sistema, i en la necessitat de canviar sovint (fins i tot diàriament) les claus, es fa palesa la importància del problema de distribució de claus. S'haurà de trobar una solució millor que no enviar contínuament cartes certificades a tort i a dret! Hi tornaré més endavant.

Encara una observació: com a matemàtics potser ens podem quedar satisfets dient que k determina S_k i S_k^{-1}, però és obvi que en dissenyar un sistema criptogràfic és essencial que la clau k permeti obtenir S_k i S_k^{-1} ràpidament i fàcil.

II. Una mica d'història dels xifrats i del trencament de xifrats

Amb la pretensió de passar i fer passar una estona agradable, vull ara treure a la llum alguns dels sistemes criptogràfics que s'han fet servir històricament. Per a ferho els descriuré breu però matemàticament, indicant també la seva criptoanàlisi, és a dir, la possibilitat de trencar-los i els mètodes de fer-ho; en altres paraules, els mètodes per a recuperar S_k ó S_k^{-1} sense conèixer prèviament la clau k.

En tota aquesta secció, el alfabets A i B seran el conjunt dels 26 caràcters de l'alfabet llatí, conjunt que identificaré sovint amb Z_{26} (conjunt de les classes de restes mòdul 26).

Xifrats monogràfics: són aquells en els quals les aplicacions S_k queden determinades pels valors que assignen a les lletres de l'alfabet.

Xifrat per substitució

Sigui $K = \mathbf{Sim}_{26}$ el grup simètric ó grup de permutacions d'un conjunt de 26 elements; òbviament card $K = 26!$ Si $\varphi \in K$, es defineix

$$S\varphi(a_1, ..., a_r) = (\varphi(a_1), ..., \varphi(a_r)), \text{ on } (a_1, ..., a_r) \in Z_{26}^r \text{ i } r \geq 1.$$

La llegenda atribueix aquest xifrat a Juli Cèsar en el cas particular en que φ es restringeix a les permutacions del tipus $\varphi(x) = a + x$, on $a \in Z_{26}$, és a dir a les permutacions pertanyents a la representació de Cayley de Z_{26} a $\mathbf{Sim}_{26}$.

Però és evident que en aquest sistema certes propietats del missatge $M = (a_1, ..., a_r)$, en particular la distribució de freqüències de lletres i parells de lletres, són conservades per S_φ; i això el fa extremadament vulnerable. Si algú coneix un missatge xifrat C prou llarg i disposa de taules de freqüències de lletres i parells de lletres de l'idioma dels missatges, pot deduir-ne fàcilment el missatge M tal que $S_\varphi(M) = C$.

Xifrats periòdics

La debilitat del sistema anterior va portar als criptògrafs a idear un nou sistema que es va mantenir imbatut durant uns 300 anys, fins a la publicació del llibre de l'oficial prusià Friederick Kasiski el 1863. Vaig a descriure'l.

Sigui $r > 1$ un número natural, que anomenarem període del xifrat.

Sigui $K = \{(\varphi_1, ..., \varphi_r) \in (\mathbf{Sim}_{26}) \mid \varphi_i \neq \varphi_j \text{ si } i \neq j\}$, així que

$$\text{card } K = V_{26}^r = 26!(26! - 1) ... (26! - r + 1),$$

Si $k = (\varphi_1, ..., \varphi_r) \in K$, es defineix

$$S_k(a_1, ..., a_r, a_{r+1}, ..., a_{rt+s}) = (\varphi_1(a_1), ..., \varphi_r(a_r), \varphi_1(a_{r+1}), ..., \varphi_s(a_{rt+s})).$$

L'avantatge d'aquest sistema no radica tant a l'augment del nombre de claus com al fet que trenca la distribució de freqüències de les lletres al missatge M. Per això és molt suggestiva l'anàlisi de Kasisky, que de fet es basa en una idea simple: determinar prèviament el període r, per tractar després cada seqüència $(\varphi_i(a_i), \varphi_i(a_{r+i}), ..., \varphi_i(a_{rt+i}))$, $i = 1, 2, ..., r - 1$ com un xifrat per substitució.

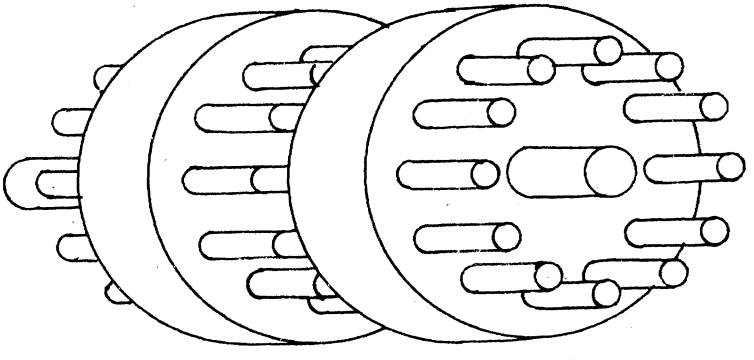
Però, com determinar el període? Es busquen al criptograma $C = S_k(M)$ els grups repetits de 3 o més caràcters. En gairebé tots els casos, aquests grups repetits al criptograma C provenen de grups repetits al missatge M. Això assegura que, en aquests casos, el nombre de caràcters entre un grup i la seva repetició ha d'ésser un múltiple del període r. Per tant, factorizant el nombre de caràcters entre cada dos grups iguals i observant la freqüència amb la qual es repeteix cada factor hom pot

normalment identificar r com el factor de major freqüència. Per acabar, es comprova la correcció de l'estimació de r observant si la distribució de freqüències de caràcters en les seqüències $(\varphi_i(a_i), \varphi_i(a_{r+i}), \ldots, \varphi_i(a_{rt+i}))$, $i = 1, 2, \ldots, r-1$ coincideix amb la distribució de freqüències, altament no uniforme, característica del xifrat per substitució.

És evident que si r és molt gran (comparable amb la longitud del missatge) l'anàlisi anterior es fa irrealitzable, però no és pas menys cert que el manegament de les claus, i els processos de xifrat i desxifrat, augmenten la seva complexitat, fins que els fa inviables en la pràctica. Davant d'aquest problema apareixen dues possibles solucions: 1) construir màquines capaces de realitzar efectivament xifrats periòdics de gran període, i 2) generalitzar els sistemes anteriors a sistemes de xifrat aperiòdic. Ambdues solucions es van assajar en la pràctica, com veurem tot seguit.

Màquines de rotors. L'Enigma alemany

Una màquina de rotors consta essencialment d'un conjunt d'anells dentats o rotors, cadascun d'ells en contacte amb el següent, muntats al voltant d'un eix; i que, com els seu nom indica, poden rotar canviant les seves posicions relatives. Cada anell té a cada extrem 26 contactes o dents, de manera que unint cada contacte d'un extrem amb un contacte de l'altre extrem es produeix una permutació $\varphi \in \mathbf{Sim}_{26}$.



(A la figura es dibuixen alguns dels contactes d'una màquina de dos rotors, però no les connexions interiors de cada rotor.)

Les rotacions possibles pels rotors constitueixen un grup finit; per exemple el generat pel gir d'amplitud $2\pi/26$ radians. En aquest cas, si σ denota la permutació cíclica produïda per l'avançament d'una posició (gir de $2\pi/26$) i $\varphi_1, \ldots, \varphi_s$ són les permutacions produïdes pels s rotors, la màquina amb les seves diferents posicions realitza les permutacions

$$\sigma^{-j_s}\varphi_s\sigma^{j_s}, \ldots, \sigma^{j_1}\varphi_1\sigma^{-j_1}\ j_1, \ldots, j_s \in \{1, \ldots, 26\}.$$

Si fins aquí he descrit com les màquines de rotors realitzen una gran família de permutacions, ara em vaig a detenir en una d'elles: l'Enigma alemany, utilitzada pels alemanys a la segona guerra mundial, que va ésser completament desxifrada pels anglesos —pel grup de Bletchey Park format per criptoanalistes i matemàtics, entre ells Alan M. Turing, sens dubte un dels pares de la informàtica.

A la seva versió més primitiva, anterior a la guerra, l'Enigma constava d'un teclat alfabètic, tres rotors, un reflector i un plafó de 26 bombetes, una per cada lletra.

Quan es premia una lletra, la lletra enviava un senyal elèctric a un contacte de la part davantera del primer rotor, sortia pel contacte de l'altra banda, unit a l'anterior per un cable, i patia transformacions anàlogues al segon i tercer rotors. El senyal després de travessar el tercer rotor arribava al reflector: un sistema de cables que unia per parells els contactes de la part posterior del tercer rotor i per tant establia una permutació R producte de 13 transposicions. El senyal així transformat pel reflector tornava a recórrer els tres rotors en sentit invers i arribava al plafó de les bombetes, encenent-ne una.

Després d'haver xifrat una lletra, el primer rotor avançava automàticament una posició (gir de $2\pi/26$ radians); d'aquesta manera un cop xifrades 26 lletres tornava a la seva posició inicial, però aleshores, automàticament, el segon rotor avançava una posició. D'aquesta manera la màquina proporcionava 26^3 permutacions diferents.

Matemàticament, si φ_1, φ_2, φ_3 denoten les permutacions que es produeixen a l'interior dels rotors respectius, σ és el cicle $(1, 2, \ldots, 26)$ i R la permutació produïda pel reflector, el procés descrit es formula

$$\alpha_0, \ldots, \alpha_s \to \beta_0, \ldots, \beta_s,$$

on si $i < 26^3$ i $i = i_1 + 26i_2 + 26^2 i_3$ amb $i_1, i_2, i_3 < 26$,

$$\beta_i = (\sigma^{-i_1}\varphi_1\sigma^{i_1})(\sigma^{-i_2}\varphi_1\sigma^{i_2})(\sigma^{-i_3}\varphi_1\sigma^{i_3})R(\sigma^{i_3}\varphi_1\sigma^{-i_3})(\sigma^{i_2}\varphi_1\sigma^{-i_2})(\sigma^{i_1}\varphi_1\sigma^{-i_1})(\alpha_i).$$

Com que R és producte de transposicions i per tant $R^2 =$ identitat, és immediat comprovar que $C(\beta_0, \ldots, \beta_s) = (\alpha_0, \ldots, \alpha_s)$, és a dir, l'operació de desxifrat coincideix amb el xifrat sempre que coincideix la posició inicial dels rotors. Aquesta posició inicial és part de la clau diària, mentre que $\varphi_1, \varphi_2, \varphi_3$ i R es mantenien fixes —però no públiques—: el canviar-les hauria significat canviar les màquines!

Deixo per altra ocasió el detallar la versió militar de l'Enigma —tenia cinc rotors dels quals cada dia n'utilitzaven 3, i un plafó de connexions que efectuava sis o set permutacions de parells de lletres tant a l'entrada com a la sortida; ambdues coses formaven part de la clau diària— i el com la van trencar els anglesos. Només diré que el fet abans esmentat que $C^2 =$ identitat, encara que era extremadament còmode pels processos de xifrat i desxifrat, va ésser un punt dèbil que els anglesos van saber explotar.

Xifrats de clau fluent o aperiòdics

Qualsevol conjunt de n elements diferents $\varphi_1, ..., \varphi_n \in \mathbf{Sim}_{26}$ constitueix una clau per a xifrar un missatge $M = (a_1, ..., a_m)$ de longitud $\leq n$ mitjançant el criptograma $(\varphi_1(a_1), ..., \varphi_m(a_m))$. Però és palès que aquest sistema és inviable malgrat que disposem d'algun «mecanisme» que faciliti el manegament de claus tan llargues com calgui i els processos de xifrat i desxifrat. Aquest «mecanisme» tots l'hem vist en alguna pel·lícula d'espies: el receptor del missatge xifrat es procura el llibre-clau en el qual estan assenyalats una pàgina i una línia determinades, arriba a casa, escriu el criptograma C i per sota de cadascuna de les seves lletres col·loca els successius caràcters que apareixen en el llibre a partir de la pàgina i la línia assenyalades, obtenint així dues seqüències $(\alpha_1, ..., \alpha_n)$ i $(\beta_1, ..., \beta_n)$. Identificant les lletres de l'alfabet amb els elements de Z_{26} obté ara les seqüències $(a_1, ..., a_n)$ i $(b_1, ..., b_n) \in Z_{26}$ i recupera el missatge M fent simplement $(a_1 - b_1, ..., a_n - b_n) \in Z_{26}$. El llibre-clau és així la font d'on brolla la clau fluent i el seu cabal no s'esgota fins que s'han acabat els caràcters del llibre.

En aquest sistema, l'anàlisi de les freqüències dels caràcters en el text xifrat $C = S_k M$ no dóna cap informació sobre el missatge M; però la seva criptoanàlisi es pot realitzar amb èxit com va fer veure Friedman [4], car donada l'estructura de l'idioma no tots els parells $\{(a, c-a) | a \in M, c-a \in k\}$ dels que pot prevenir el caràcter $c \in C$ són igualment probables, de manera que molts d'ells es poden descartar.

Només si la clau fluent k consisteix a una successió aleatòria d'elements de Z_{26} que es fa servir només una vegada (xifrat amb «cinta de només un cop») el sistema és completament resistent a qualsevol intent de trencar-lo.

Xifrats per blocs

Les unitats sobre les quals actuen les transformacions d'aquests sistemes no són els caràcters, sinó les seqüències o blocs de caràcters d'una longitud predeterminada. Més formalment: sigui *A el conjunt de les lletres de l'alfabet romà,* $r > 1$ un natural i $B = A^r$ (seqüències de r lletres).

Un xifrat per blocs de longitud r és un sistema criptogràfic l'alfabet base del qual és B, és a dir, les seves aplicacions S_k estan definides de $FM(B)$ a $FM(B)$.

Si ara $M = (a_1, ..., a_n)$ és un missatge escrit amb l'alfabet A, es divideix en blocs de longitud r: $(a_1, ..., a_r)(a_{r+1}, ..., a_{2r})$ ... (Si eventualment $n = rt + s$ amb $0 < s < r$ s'afegeix al final del missatge M la lletra «x» $r - s$ cops) i es xifra cadascun dels blocs.

Potser momentàniament un estaria temptat de repetir sobre l'alfabet $B = A^r$ els diferents tipus de xifrats ja considerats a l'apartat anterior, però n'hi haurà prou amb posar esment al més senzill d'ells, el xifrat per substitució, per fer-nos-en desistir.

Com que card B = card $A^r = 26^r$, una taula de distribució de freqüències dels elements de B exigeix 26^r entrades, i no cal que r sigui gaire gran perquè això sigui irrealitzable, fins i tot amb ordinador. La possibilitat d'obrir el sistema per anàlisi de freqüències desapareix, i el xifrat per substitució resulta ara segur davant de la criptoa-

nàlisi. Però no tot són avantatges: una clau és una permutació de 26^r elements i resulta també immanejable en la pràctica. Davant d'aquest problema, els criptògrafs han ideat sistemes de xifrat per substitució en els quals el conjunt de claus K és algun subconjunt manejable, però resistent a l'anàlisi, de $\mathbf{Sim}_{26^r}$. En aquesta tasca l'aritmètica ha vingut a ajudar-los: s'injecta $B = A^r$ en alguna estrcutura X d'aritmètica modular, es pren X com a nou alfabet i s'obté una família d'aplicacions de X, fàcil i econòmicament manejables amb l'ordinador. Vaig a donar un parell d'exemples molt emprats.

1. Si φ és l'aplicació de $A = Z_{26}$ a Z_2^5 que assigna a cada $0 \leq a < 26$ la seva expressió en base 2 $(a_4a_3a_2a_1a_0)$, aleshores φ definieix de manera canònica una inclusió $A^r \to Z_2^{5r}$.

Aquesta inclusió és precisament la forma usual de transformar una seqüència de caràcters alfabètics en una seqüència de bits.

2. Sigui n un natural. Sigui m el màxim natural tal que $25(1 + 10^2 + \ldots + 10^{2m}) < n$. L'aplicació

$$\begin{array}{rcl} \psi\colon Z_{26}^{m+1} \approx A^{m+1} & \to & Z_n \\ (a_1, \ldots, a_{m+1}) & \to & a_1 10^{2m} + \ldots + a_{m+1} \end{array}$$

és injectiva.

Com a exemple de sistema criptogràfic per blocs, descriuré el D.E.S. (Data Encryption Standard) creat per la I.B.M. i adoptat en 1977 per l'oficina de pesos i mesures dels U.S.A. Aquest sistema actua sobre blocs de 64 bits i està dissenyat de tal manera que les operacions de xifrat i desxifrat en un ordinador són molt ràpides —xifrat d'alta velocitat—. Per això és el sistema normalment emprat a l'àmbit comercial i possiblement també en el militar.

En el seu origen hi ha la idea de Shannon, exposada el 1949 [7], que és possible construir un sistema criptogràfic resistent a l'anàlisi mitjançant components simples individualment febles.

El conjunt de claus K pel D.E.S. és Z_2^{64}, però vuit dels bits són redundants, i així pròpiament $K = Z_2^{56}$.

1. Cada clau del D.E.S. dóna lloc a 16 subclaus de 48 bits cadascuna, segons el següent procés:

Si $k \in Z_2^{64}$ s'eliminen de k els bits que ocupen els llocs $8r$, $r = 1, \ldots, 8$, obtenint-se així $\varphi(k) \in Z_2^{56}$. Com que $Z_2^{56} = Z_2^{28} \times Z_2^{28}$ podem escriure $\varphi(k) = (m_1, m_2)$, $m_1, m_2 \in Z_2^{28}$.

Sigui $s \in \mathbf{Sim}_{28}$ el cicle que corre un lloc cap a l'esquerra i ψ una aplicació sobre seqüències de 56 bits que elimina 8 d'ells, de manera que $\psi\colon Z_2^{56} \to Z_2^{48}$.

Ara les subclaus vénen donades:

$$\begin{array}{rcll} b_i & = & \psi(s^i(m_1), s^i(m_2)), & i = 1, 2 \\ b_{2+i} & = & \psi(s^{2+2i}(m_1), s^{2+2i}(m_2)), & i = 1, \ldots, 6 \\ b_9 & = & \psi(s^{15}(m_1), s^{15}(m_2)), & \\ b_{9+i} & = & \psi(s^{15+2i}(m_1), s^{15+2i}(m_2)), & i = 1, \ldots, 6 \\ b_{16} & = & \psi(m_1, m_2) & \end{array}$$

2. Actuació del D.E.S. sobre Z_2^{64}.

Comencem per definir les «peces», totes elles conegudes públicament (llurs taules es poden trobar a [1]).

— Una permutació $P_I \in \mathbf{Sim}_{64}$ i que per tant actua sobre Z_2^{64} transposant els seus bits.

— Una permutació $P \in \mathbf{Sim}_{32}$ i que per tant actua sobre Z_2^{32}.

— Una aplicació d'expansió E: $\{1, ..., 32\} \to \{1, ..., 48\}$ i que per tant defineix una aplicació de Z_2^{32} a Z_2^{48}.

— Una aplicació S: $Z_2^{48} \to Z_2^{32}$ no lineal que actua mitjançant $(s_1, ..., s_8)$ on les s_i: $Z_2^6 \to Z_2^4$ són aplicacions del tipus $s_i\,(a_1a_2a_3a_4a_5a_6) = \varphi^i_{a_1a_6}(a_2a_3a_4a_5)$ on $\varphi^i_{a_1a_6} \in \mathbf{Sim}_{16}$ (cada s_i queda definida per 4 permutacions que actuen com un xifrat per substitució).

Un cop disposades les peces, anem a emboetar-les: si $b \in Z_2^{48}$ i $(a, b) \in Z_2^{32} \times Z_2^{32} \approx$ $\approx Z_2^{64}$, es defineix T_k: $Z_2^{64} \to Z_2^{64}$ per la fórmula $T_k(a, b) = (b, a + PS(E(b) + k))$. Observi's que una comprovació immediata demostra que: $T_k^{-1} = \sigma T_k \sigma$ on $\sigma(a, b) = (b, a)$.

Finalment es defineix l'operació de xifrat: si $M \in Z_2^{64}$, el seu xifrat amb la clau k és:

$$(P_I^{-1} \circ T_{k16} \circ T_{k15} \circ ... \circ T_{ki} \circ P_I)(M) = C$$

on $k_1, ..., k_{16}$ són les 16 subclaus produïdes per la clau k.

Com que $T_k^{-1} = \sigma T_k \sigma$, l'operació de desxifrat resulta ésser:

$$(P_I^{-1} \circ \sigma \circ T_{k1} \circ T_{k2} \circ ... \circ T_{k16} \circ \sigma \circ P_I)(C) = M$$

cosa que fa veure que les operacions de xifrat i desxifrat, encara que no són idèntiques —això faria el sistema més feble davant de la criptoanàlisi— són molt similars.

L'interessat en l'aspecte d'implementació del D.E.S. en un ordinador pot llegir [1].

Quant a la seva criptoanàlisi només diré que ja en 1977 Diffie i Hellmann [2] van afirmar que podia sucumbir davant d'un atac per cerca exhaustiva de la clau entre les 2^{56} possibles.

Potser per això, a l'ús actual del D.E.S. es tendeix a canviar la clau diàriament, però aquí apareix l'important problema abans esmentat de la distribució de claus. Aquest problema i les seves solucions ocuparan la nostra atenció seguidament.

Criptografia no convencional

El problema de distribució de claus perd en gran part la seva virulència si la distribució es pot realitzar per canals no especialment protegits sense que es perdi seguretat per aquest motiu. Dues possibilitats s'obren:

1. Idear un mètode que permeti a dos usuaris enviar-se pels canals ordinaris de comunicació la informació suficient per a adoptar una clau comuna, sense que aquesta informació reveli la clau a tercers.

2. Idear un mètode en el qual les operacions de xifrat siguin públiques i les de desxifrar romanguin secretes.

Ambdues possibilitats s'han realitzat efectivament, basant-se sobretot en «funcions de direcció única», és a dir funcions fàcils de calcular, però d'inversa computacionalment irrealitzable. En donaré un exemple de cadascuna.

1. Distribució pública de claus

Diffie i Hellmann van proposar el 1976 un mètode basat en la dificultat de calcular el logaritme discret en un cos finit F_p amb p primer.

$F_p^* = F_p \setminus \{0\}$ és aleshores un grup cíclic, isomorf per tant a Z_{p-1}. Sigui $\alpha \in F_p^*$ un generador o element primitiu de l'esmentat grup (de passada diré que no hi ha cap mètode general de trobar els elements primitius de F_p; sí, però, en alguns casos particulars).

Es defineix l'exponencial de base α, que denotarem per $\exp_\alpha$, com l'isomorfisme

$$Z_{p-1} \to F_p^*$$

$$r \to \alpha^r$$

Per definició, la seva funció inversa és el logaritme discret en base α, que designarem $\log_\alpha$.

Parem atenció a l'aspecte computacional d'ambdues funcions. Si s és el natural més gran tal que $s \leq \log_2 p$, calculant la sèrie α, α^2, α^{2^2}, ..., α^{2^s} per successives potenciacions quadràtiques i multiplicant tots els elements de la sèrie que corresponen als uns de l'expressió binària de p, es pot obtenir α^r per a qualsevol $1 \leq r \leq p$. Això demostra que per a calcular $\exp_\alpha(r)$ calen com a màxim de l'ordre de $2 \log_2 p$ productes en F_p (òbviament si per una α fixa volem calcular $\exp_\alpha(r)$ per moltes r diferents primer calcularem la sèrie α, α^2, α^{2^2}, ..., α^{2^s} d'un cop per sempre i després n'hi haurà prou amb calcular $\log_2 p$ productes per cada r).

En canvi, malgrat els esforços que s'hi han dedicat, els millors algorismes per a calcular logaritmes discrets requereixen de l'ordre de $\exp(\sqrt{\log p \log\log p})$ operacions amb bits. Tanmateix si $p-1$ té només factors petits, existeixen algorismes molt més ràpids.

Per tant, mentre no es trobin millors algorismes (potser la teoria de la complexitat computacional demostrarà algun dia que no és possible de trobar-ne), la funció $\exp_\alpha$ resulta ésser una funció de direcció única.

Després d'aquestes nocions algebraiques, passo a descriure l'algorisme de Diffie i Hellmann.

Es fixa un primer p pel qual el logaritme discret sigui computacionalment irrealitzable (en particular p ha d'ésser prou gran, per exemple de 200 dígits) i un element primitiu α de F_p.

Cada usuari i escull aleatòriament un número x_i tal que $2 \leq x_i \leq p-2$, que manté secret, i calcula $\alpha^{x_i} = y^i$.

Si dos usuaris volen compartir una clau comuna, cadascun envia el propi y a l'altre. La clau comuna és ara $K = \alpha^{x_i x_j} = y_j^{x_i} = y_i^{x_j}$, clau que cadascun dels dos usuaris re-

cupera a partir del número que li ha enviat l'altre i el número secret propi. La seguretat del sistema radica en que el coneixement per tercers de y_i o y_j no els permet de conèixer llurs logaritmes discrets.

2. Sistemes de clau pública

Un sistema criptogràfic de clau pública no actua sobre un alfabet ordinari, sinó sobre una estructura d'aritmètica modular M. Però allò que el distingeix essencialment de la criptografia convencional és l'escissió de la clau en dues: una de xifrat i una de desxifrat, de manera que el coneixement d'una d'elles no dóni informació sobre l'altra.

Més formalment: sigui K un conjunt, $\varphi : K \to K$ una aplicació bijectiva. El sistema consta de dues famílies uniparamètriques $\{E_k \mid k \in K\}$, $\{D_k \mid \in K\}$ d'aplicacions bijectives de M a M tals que:

1. Per qualsevol $k \in K$, E_k i $D_{\varphi(k)}$ són inverses l'una de l'altra.
2. E_k i $D_{\varphi(k)}$ són fàcilment calculables mitjançant algorismes si es coneixen k i $\varphi(k)$, respectivament.
3. A partir de l'algorisme que calcula E_k és computacionalment irrealitzable obtenir $D_{\varphi(k)}$. En altres paraules, no es pot calcular la $\varphi(k)$ a partir de la k en un temps raonable.

Quin és el seu funcionament en la pràctica? Cada usuari disposa de les seves dues claus $(k, \varphi(k))$; la primera la publica en un llistí mentre que la segona la manté secreta. Si un altre usuari li vol enviar un missatge M, el xifra mitjançant E_k i li envia $E_k(M)$; el receptor és l'únic que el pot desxifrar mitjançant $D_{\varphi(k)}$.

Els sistemes de clau pública donen també solució a un altre problema que vull esmentar: la producció de signatures. Com que la clau la coneix tothom, el receptor d'un missatge no té cap mitjà d'assegurar-se que el remitent és qui diu ser i no algú altre, mentre que en els sistemes critogràfics clàssics, on el mètode de codificar és secret, el conèixer-lo ja era una evidència de la identitat de l'emissor. Ara, en canvi, ens caldrà posar una signatura, que és essencialment una cosa que pot ésser produïda per un sol individu i en canvi pot ésser reconeguda per tothom, com el segell del regne que feia servir el rei a Hamlet per a lacrar les cartes. Indicarem com es pot aconseguir això en un sistema de clau pública: si l'usuari A vol enviar un missatge M firmat a B, primer de tot li aplica la seva clau de desxifrat secreta, $D_{\varphi(k_A)} = S$, i a continuació aplica a S la clau de xifrat pública B, i obté així $C = E_{\varphi(k_B)}(S) = C$. Quan B rep C li aplica la seva clau de desxifrat secreta $D_{\varphi(k_B)}$ per tal d'obtenir S i finalment li aplica la clau pública de xifrat del remitent calculant $E_{\varphi(k_A)}(S)$, i així obté el missatge M i la prova exhibible que és A qui li ha enviat.

El sistema R.S.A.

Rivest, Shamir i Adleman van crear el 1978 un sistema criptogràfic de clau pública basat en el fet que, encara que és computacionalment fàcil —tests de primali-

tat— trobar primers grans, per exemple de 100 dígits, factoritzar el producte de dos primers d'aquesta mena resulta prohibitiu en temps d'ordinador. Vaig a descriure'l.

Cada usuari escull aleatòriament dos primers p, q d'uns 100 dígits i calcula $N = pq$. Si $\Phi(N) = (p-1)(q-1)$ és el valor de la funció d'Euler a N, escull aleatòriament un número e tal que $2 \leq e \leq \Phi(N) - 1, 2^e > N$, i e i $\Phi(N)$ siguin primers entre ells. Calcula aleshores d tal que $ed \equiv 1(\Phi(N))$.

Ara fa públic el parell (N, e) i manté secreta tota la resta, en particular d.

Les operacions de xifrat i desxifrat són:

$$\begin{aligned} E_{(N,e)}: \; & Z_N^* \to Z_N^* \\ & P \to P^e \\ D_{(N,d)} \; & Z_N^* \to Z_N^* \\ & C \to C^d \end{aligned}$$

on Z_N^* designa el grup de les unitats de Z_N.

Aquestes dues operacions són inverses l'una de l'altra, ja que

$$D_{(N,d)} \circ E_{(N,e)}(P) = P^{ed} = P^{1+\lambda\Phi(N)} = P$$

per la congruència d'Euler.

S'observa que, coneguts e i d, la E i la D són fàcils i ràpides de calcular, i que e, sense conèixer $\Phi(N)$ —cosa que exigeix haver factoritzat prèviament N—, no dóna cap informació sobre d. Tanmateix s'han de prendre una sèrie de precaucions, de cara a evitar que puguin aplicar-se tècniques especialment ràpides de factorització: $(p-1)$ i $(q-1)$ han de tenir factors grans,el màxim comú divisor de $p-1$ i $q-1$ ha d'èsser petit, i p i q han de tenir expressions decimals amb diferent nombre de dígits.

I per acabar una dada pràctica: almenys als U.S.A., el sistema R.S.A. s'utilitza habitualment per a enviar les claus del sistema D.E.S. Així es combinen els avantatges d'ambdós sistemes: l'alta velocitat de xifrat i desxifrat del D.E.S. i la no necessitat de transmetre claus per altres canals de R.S.A.

Bibliografia

[1] A. K. Dewdney. *Creación y rotura de códigos,* Investigación y Ciencia, gener 1989.

[2] W. Diffie i M. E. Hellmann. *Exhaustive cryptanalysis of the N.B.S. data encription standard,* Computer, vol. 10, n. 6, juny 1977.

[3] W. Diffie i M. E. Hellmann. *Privacy and autentication: An Introduction to Cryptography.* Proceedings of the IEEE, vol. 67, n. 3, març 1979.

[4] W. F. Friedman. *Military Cryptanalysis.* Washington DC: U.S. Government Printing Office, 1944.

[5] R. L. Rivest, A. Shamir i L. Adleman. *On digital signatures and public key cryptosystems.* Commun. A.C.M. vol. 21, n. 2, febrer 1978.

[6] K. H. Rosen. *Elementary number theory and its applications.* Addison Wesley, 1984.
[7] C. E. Shannon. *Communication theory of secrecy systems.* Bell Syst. Tech. J., vol. 28, octubre 1949.