

Facultat d'Informàtica de Barcelona
Examen final de Criptografia
20 de juny de 2000

Problema 1. Sigui $a > b > 0$ enters positius amb $b \neq 0$. Recordeu que l'algorisme d'Euclides consisteix a posar $r_0 = a, r_1 = b$ i, mentre $r_k > 0$, fer la divisió euclidiana

$$r_{k-1} = r_k q_k + r_{k+1}, \quad 0 \leq r_{k+1} < r_k,$$

fins a arribar a un r_k que és zero.

Sigui n el nombre de divisions euclidianes necessàries per executar completament l'algorisme; és a dir, r_n és l'últim reste no nul. Demostreu que $n < 2 \log_2(a) + 1$.

Indicació: comenceu per demostrar que $r_{k+2} < r_k/2$ per a tot $k \geq 0$ i després demostreu la desigualtat que es demana considerant per separat els casos en que n sigui parell o senar.

SOLUCIÓ: Seguint la indicació. La seqüència de restes $r_0 > r_1 > r_2 \dots$ és estrictament decreixent, per definició de divisió euclidiana. Si $r_{k+1} \leq r_k/2$ aleshores $r_{k+2} < r_{k+1} \leq r_k/2$; si pel contrari $r_{k+1} > r_k/2$ aleshores a la divisió euclidiana $r_k = r_{k+1} q_{k+1} + r_{k+2}$ necessàriament $q_{k+1} = 1$ i, per tant, $r_{k+2} = r_k - r_{k+1} < r_k - r_k/2 = r_k/2$. Això demostra el que es deia a la indicació: que $r_{k+2} < r_k/2$.

Sigui $n = 2n_0$ parell. Aleshores $r_2 < r_0/2 = a/2, r_4 < r_2/2 < a/4, r_6 < a/8, \dots, r_{2n_0} < a/2^{n_0}$. Prenent logaritmes (i com que el logaritme és una funció creixent), tenint en compte que $r_{2n_0} \geq 1$, i fent servir les propietats de la funció logaritme, es té $0 \leq \log_2(r_{2n_0}) < \log_2(a/2^{n_0}) = \log_2(a) - n_0$. Per tant, $n_0 < \log_2(a) \Rightarrow n = 2n_0 < 2 \log_2(a)$.

Suposem ara que $n = 2n_0 + 1$ és un nombre senar. Aleshores $r_3 < r_1/2 = b/2, r_5 < b/4, \dots, r_{2n_0+1} < b/2^{n_0}$ i fent com abans s'obté la desigualtat $n_0 < \log_2(b)$. Per tant, com que $b < a$ es té que $n_0 < \log_2(a)$ i que $n = 2n_0 + 1 < 2 \log_2(a) + 1$.

Problema 2. El *Teorema del Nombre Primer* assegura que per a tot $x > 1$

$$\frac{x}{\ln(x)}$$

és una bona aproximació pel nombre de primers a l'interval $(1, x)$, on $\ln(x)$ és la funció logaritme neperià.

(a) Basant-vos en aquest resultat calculeu la probabilitat de que un enter senar de n bits escollit aleatòriament sigui primer.

(b) Si una implementació d'un test de primalitat probabilístic requereix $t = n^3 10^{-11}$ segons a decidir sobre la primalitat d'un enter de n bits calculeu el temps de càlcul esperat per a una funció que generi un nombre primer aleatori de n bits pel procediment de generar enters senars i aplicar-los el test de primalitat fins que en troba un de primer.

Observacions: (1) Recordeu que un "enter de n bits" és un que pertany a l'interval $[2^{n-1}, 2^n)$. (2) Podeu fer servir l'aproximació $\ln(2) \simeq 0.693$. (3) Si X és una variable aleatòria que pren els valors x_i amb probabilitats $P\{X = x_i\} = p_i$ l'esperança és $E(X) = \sum p_i x_i$ (i el temps que tardarà l'algorisme a trobar un primer és una variable aleatòria). (4) Recordeu de l'anàlisi la fórmula de sumar una sèrie següent: $1 + 2x + 3x^2 + 4x^3 + \dots = 1/(1-x)^2$, que val per tot nombre amb $|x| < 1$.

SOLUCIÓ: (a) El nombre de primers de n bits serà, aproximadament,

$$\frac{2^n}{\log(2^n)} - \frac{2^{n-1}}{\log(2^{n-1})} = \frac{2^n}{n \log 2} - \frac{2^{n-1}}{(n-1) \log 2} = \frac{2^{n-1}}{\log 2} \left(\frac{2}{n} - \frac{1}{n-1} \right) = \frac{2^{n-1}}{\log 2} \left(\frac{n-2}{n(n-1)} \right).$$

L'últim quocient entre parèntesi es pot aproximar per $1/n$ quan l'enter n és gran, de manera que el nombre de primers admet com a aproximació $2^{n-1}/n \log 2$. Com que el nombre d'enters senars de n bits és $2^{n-2} = (2^n - 2^{n-1})/2$ la probabilitat de que en escollir-ne un sigui primer és, aproximadament,

$$\frac{2^{n-1}/n \log 2}{2^{n-2}} = \frac{2}{n \log 2} \sim \frac{1}{0.347 n}.$$

(b) Diguem p a la probabilitat anterior. La probabilitat d'encertar el primer al primer intent és p , la d'encertar-lo al segon és $(1-p)p$ (al primer no s'encerta i al segon sí) la d'encertar-lo al tercer és $(1-p)^2 p$ i la d'encertar-lo al k -èsim intent és $(1-p)^{k-1} p$. En el primer cas l'algorisme fa només un test de primalitat, i tardarà temps t , en el segon cas fa dos tests, i tardarà temps $2t$, si ha de fer k intents fa k tests i tardarà temps kt . El temps de càlcul esperat és, doncs,

$$\begin{aligned} E &= p \cdot t + (1-p)p \cdot 2t + (1-p)^2 p \cdot 3t + \dots + (1-p)^{k-1} p \cdot kt + \dots \\ &= pt(1 + 2(1-p) + 3(1-p)^2 + 4(1-p)^3 + \dots) \end{aligned}$$

Aplicant la suma de la sèrie al sumatori de la dreta, que correspon al valor $x = 1-p$ de la variable x , el seu valor és $1/(1 - (1-p))^2 = 1/p^2$, per tant

$$E = \frac{pt}{p^2} = \frac{t}{p} = \frac{n^3 10^{-11}}{1/0.347 n} = \frac{n^4}{3.47 \times 10^{10}} \text{ segons.}$$

Per exemple, el temps de càlcul esperat per un primer de 1024 bits és d'uns 40 segons, i per un primer de 2048 bits és de 16×40 segons: uns 10 minuts.

Problema 3. Siguin p un nombre primer i g un generador del grup multiplicatiu \mathbb{Z}_p^* . Les claus privada i pública del sistema ElGamal són, respectivament,

- un element $r \in \mathbb{Z}_{p-1}$ escollit aleatòriament,
- l'element $u = g^r \pmod{p} \in \mathbb{Z}_p^*$.

Per firmar un document amb hash $h \in \mathbb{Z}_{p-1}$ es calcula

$$f_1 = g^k \pmod{p}, \quad f_2 = k^{-1}(h - r f_1) \pmod{p-1}$$

on $k \in \mathbb{Z}_{p-1}^*$ és un element escollit aleatòriament. Per verificar la firma es comprova la identitat

$$u^{f_1} f_1^{f_2} \equiv g^h \pmod{p}.$$

Hi ha altres “equacions de firma” i “equacions de verificació” que funcionen igual de bé (en algunes es demana que la clau privada r sigui invertible mòdul $p-1$). La taula següent conté sis equacions de firma i verificació de les quals una és incorrecta, digueu quina i justifiqueu perquè.

firma	verificació
$f_2 = k^{-1}(f_1 - rh)$	$u^h f_1^{f_2} \equiv g^{f_1}$
$f_2 = r^{-1}(f_1 - kh)$	$u^{f_2} f_1^h \equiv g^{f_1}$
$f_2 = r^{-1}(h - k f_1)$	$u^{f_2} f_1^{f_1} \equiv g^h$
$f_2 = rh + k f_1$	$u^h f_1^{f_1} \equiv g^{f_2}$
$f_2 = rk + h f_1$	$u^k f_1^{f_2} \equiv g^h$
$f_2 = r f_1 + kh$	$u^{f_1} f_1^h \equiv g^{f_2}$

SOLUCIÓ: El que no funciona és el penúltim, ja que a l'esquerra queda

$$u^k f_1^{f_2} = g^{rk} g^{k(rk+hf_1)} = g^{rk+rk^2+khf_1}$$

que en general no serà igual a g^h (per exemple prenent $k = -1$, que és un valor possible, dona $g^{-hg^{-1}}$, que no és g^h en general). En tots els demés cassos va bé. Per exemple, l'últim és

$$u^{f_1} f_1^h = g^{rf_1} g^{kh} = g^{f_2}$$

i el primer és

$$u^h f_1^{f_2} = g^{rh} g^{kk^{-1}(f_1-rh)} = g^{f_1}$$

i els demés es comproven de manera anàloga.

Problema 4. Expliqueu l'algorisme de firma-verificació del sistema RSA. Compareu-lo amb els algorismes corresponents del sistema ElGamal pel que fa al temps que tardaran cadascun en executar-se (es suposa que el nombre de bits del mòdul de RSA és el mateix que el nombre de bits del primer de ElGamal).

SOLUCIÓ: En el sistema RSA amb mòdul $n = pq$, exponent públic u i exponent privat $r = u^{-1} \pmod{\varphi(n)}$ la firma d'un missatge amb hash $h \in \mathbb{Z}_n^*$ és

$$f = h^r \pmod{n}$$

i la verificació es fa comprovant la identitat

$$h = f^u \pmod{n}.$$

Signi ℓ el nombre de bits amb que es treballa; per fer-nos una idea un valor de ℓ habitual a la pràctica és 1024.

El nombre de bits de r és el mateix que el del mòdul n ; en canvi el nombre de bits de l'exponent públic u pot ser petit si es tria la opció de fixar el valor d'aquest exponent, per exemple prenent $u = 65537$, el qual és habitual a la pràctica. Una firma RSA requereix una exponenciació modular amb exponent de ℓ bits, que són aproximadament $1.5\ell \sim 1500$ multiplicacions. La verificació requereix una exponenciació modular amb exponent petit, per exemple si $u = 65537$ tot plegat es redueix a 17 multiplicacions.

La firma ElGamal requereix per calcular f_1 una exponenciació modular i per calcular f_2 només un parell de productes i un invers: en total representen unes $1.5\ell = 1500$ multiplicacions, igual que en RSA. En canvi per verificar la firma es requereixen tres exponenciacions modulares i alguns productes, que vol dir unes $4.5\ell = 4500$ multiplicacions: molt més que el sistema RSA.

Problema 5. Perquè les claus dels sistemes criptogràfics de clau pública (asimètrics) tenen més bits que les dels sistemes de clau secreta (simètrics)?

SOLUCIÓ: En un sistema de clau secreta, si està ben dissenyat, la única manera de trobar la clau és la força bruta: s'han d'anar provant totes les claus possibles fins que es troba la que es busca.

En canvi en un sistema de clau pública la informació sobre la clau privada està tota continguda en la clau pública i per trobar-la es té, a més de la opció de la força bruta, la possibilitat de calcular-la resolent un problema difícil: la factorització en el cas de RSA i el logaritme discret en el cas de ElGamal. Com que tant per l'un com per l'altre problema es coneixen algorismes subexponencials, es poden resoldre molt més fàcilment que per força bruta i és per això que les claus han de ser més llargues que per clau secreta per tal d'assolir el mateix grau de seguretat.

Problema 6. Expliqueu les diferències que hi ha entre criptografia simètrica (de clau secreta) i criptografia asimètrica (de clau pública).

SOLUCIÓ: En criptografia simètrica tots els participants en un protocol fan servir la mateixa clau; en criptografia de clau pública cada participant té el seu propi parell de claus: la privada, que manté en secret, i la pública, que ha de posar a l'abast de tothom. En particular quan es fa servir un sistema asimètric no cal un canal segur per acordar claus com a pas previ a poder utilitzar el sistema.

La criptografia de clau pública permet fer firmes digitals que garanteixen el no-repudi, cosa que amb criptografia simètrica és impossible.

La criptografia de clau pública necessita claus més llargues i els seus algorismes de xifrat-desxifrat són més lents que els de la criptografia simètrica; és per això que, a la pràctica, per enviar missatges xifrats es fan servir tots dos tipus de criptografia alhora, aprofitant els avantatges de cadascuna.

Problema 7. Expliqueu què són els certificats de claus públiques i les autoritats certificadores.

SOLUCIÓ: Un certificat de clau pública és un document digital que serveix per assegurar que una certa clau pública pertany a una persona o entitat. Conté la informació següent:

- versió,
- número de sèrie,
- identificació de l'autoritat certificadora que emet el certificat,
- algorisme emprat per firmar,
- període de validesa: inici i final,
- identificació de la persona o entitat propietaria de la clau pública,
- clau pública,
- camps opcionals (per exemple usos possibles i limitacions d'aquesta clau),
- firma digital de totes les dades anteriors per part de l'autoritat certificadora.

Una autoritat certificadora és una empresa o organisme que es dedica a garantir que una clau pública pertany a una persona o entitat. La manera de garantir-ho és firmar certificats de claus públiques a tots els usuaris del sistema criptogràfic que ho demanin (després d'haver-los identificat). D'aquesta manera cada usuari només ha d'assegurar-se que el seu software conté les claus públiques autèntiques de les autoritats certificadores. A partir d'aquí, farà servir aquestes claus públiques per verificar els certificats que li siguin presentats pels demés usuaris.

Problemes 1,2: 1.5 punts, 3,4,5: 1 punt, 6,7: 2 punts.

La solució serà a www-ma2.upc.es/quer/quer.html, clicar Criptografia FIB.

Les notes sortiran el dia 27 de juny al racó de l'estudiant.

Reclamacions: per escrit a la bustia del departament MAII o per e-mail fins el 30 de juny.