

Curs 9 + 10

Atacuri asupra sistemelor criptografice cu cheie publică

$$\underline{\text{Def}} \quad \alpha \in \mathbb{R}$$

$$\alpha_0 = \alpha, p_0 = a_0, q_0 = 1$$

$$p_i = a_0 a_1 \dots a_{i-1}, q_i = a_i$$

$$a_i = \lfloor \alpha_i \rfloor$$

$$\alpha_{i+1} = \frac{1}{\alpha_i - a_i}$$

Intregii a_0, a_1, a_2, \dots

$$p_{i+1} = a_i p_{i-1} + p_{i-2}, i \geq 2$$

formază fracția continuă a lui α . Fracțiile $\frac{p_i}{q_i}$ se

$$q_{i+1} = a_i q_{i-1} + q_{i-2}, i \geq 2$$

numesc convergențe, $\frac{p_i}{q_i} \rightarrow \alpha$

$$\gcd(p_i, q_i) = 1 \quad \forall i$$

Important

$$p, q \in \mathbb{Z}$$

$$|\alpha - \frac{p}{q}| \leq \frac{1}{2q^2} \Rightarrow \exists i \quad p = p_i, q = q_i \quad \text{adecăt} \\ \frac{p_i}{q_i} \text{ este un convergent}$$

Presupuneri

$$N = p q, q < p < 2q$$

$$d \text{ "mic"} \quad d < \frac{1}{3} N^{\frac{1}{4}}$$

$$e \text{ public}, e < \phi = (p-1)(q-1)$$

Există K :

$$ed - K\phi = 1$$

$$\Rightarrow \left| \frac{e}{\phi} - \frac{K}{d} \right| = \frac{1}{d\phi}$$

$$\phi \approx N \text{ decarece } |N - \phi| = |p + q - 1| < 3\sqrt{N} \quad (2)$$

Observăm că $\frac{e}{N}$ close approximation to $\frac{\kappa}{d}$.

$$\left| \frac{e}{N} - \frac{\kappa}{d} \right| = \left| \frac{ed - N\kappa}{dN} \right| = \left| \frac{ed - \kappa\phi - N\kappa + \kappa\phi}{dN} \right| =$$

$$= \left| \frac{1 - \kappa(N - \phi)}{dN} \right| \leq \left| \frac{3\kappa\sqrt{N}}{dN} \right| = \frac{3\kappa}{d\sqrt{N}} < \frac{1}{3d^2} < \frac{1}{2d}$$

$$\text{Dacă } e < \phi \Rightarrow \kappa < d < \frac{1}{3}N^{\frac{1}{4}} \quad \frac{\kappa}{d} < \frac{1}{3d}N^{\frac{1}{4}}$$

$$\frac{3\kappa}{d\sqrt{N}} < \frac{3}{3d} \cdot \frac{N^{\frac{1}{4}}}{N^{\frac{1}{2}}} = \frac{1}{d}N^{-\frac{1}{4}} < \frac{1}{3d^2}$$

$\gcd(\kappa, d) = 1 \Rightarrow \frac{\kappa}{d}$ fractie simplificată.

Dacă $\frac{\kappa}{d}$ convergent pt $\frac{e}{N}$!

Exemplu

$$N = 9449868410449$$

$$e = 6792605526025$$

$$\text{aflăm că } d < \frac{1}{3}N^{\frac{1}{4}} < 584$$

$$d = \frac{e}{N} = 1, \frac{2}{3}, \frac{3}{4}, \frac{5}{7}, \frac{18}{25}, \frac{23}{32}, \frac{409}{569}, \frac{1659}{2308}$$

$$\text{Găsim că } \boxed{d = 569} \quad \text{verificând că } (m^e)^d \equiv m \pmod{N}$$

rezulta din M

la metoda de

Atacuri lattice

Teorema Coppermith $f \in \mathbb{Z}_1[x]$ polinom număr de grad d.

Dacă $\exists x_0$ cu $f(x_0) = 0 \pmod N$ și $|x_0| \leq N^{\frac{1}{d} - \varepsilon}$
atunci x_0 poate fi găsit în timp poly($\log N, \frac{1}{\varepsilon}$) pentru d fix.

(Atacul Hastad)

Folosirea unui exponent mic

Amen 3 useri cu cheile $(N_1, 3), (N_2, 3), (N_3, 3)$; RS

Cineva trimite mesajul m .

Eva vede

$$c_1 = m^3 \pmod{N_1}$$

$$c_2 = m^3 \pmod{N_2}$$

$$c_3 = m^3 \pmod{N_3}$$

Eva constată că $\gcd(N_i, N_j) = 1$ pentru toți i, j ;
(altfel găsește factori comuni, factorizează și deschidează!)

(Lema chineză)

găsește soluție simultană

$$x = c_i \pmod{N_i} \quad \forall i = 1, 2, 3$$

Deci $x = m^3 \pmod{N_1 N_2 N_3}$. Dar $m^3 < N_1 N_2 N_3$
(deoarece $m < N_i \forall i$)

$$\text{Deci } m = \sqrt[3]{x}$$

Exemplu

$$N_1 = 323, N_2 = 299, N_3 = 341$$

$$\text{Eva are } c_1 = 50, c_2 = 268, c_3 = 1$$

$$\text{Calculažă } X = 300763 \bmod N_1 N_2 N_3$$

$$m = \sqrt[3]{300763} = 67^1$$

Pentru a evita acest atac, se face padding cu random bits!

Vom vedea că nici asta nu e suficient.

De exemplu, adăugăm adresa userului la mesaj:

$$c_i = (i \cdot 2^h + m)^3 \bmod N_i$$

(Atacul Hastad)

$$\text{În general } g_i(x) = (i \cdot 2^h + x)^e \bmod N_i \quad i=1, \dots, k$$

Stim că $\exists m$ cu $g_i(m) = 0 \bmod N_i$, și vrem să recuperăm x .

$$N = N_1 N_2 \dots N_k$$

Cu lumență chineză găsim $t_i \in \mathbb{Z}$.

$$g(x) = \sum_{i=1}^k t_i g_i(x)$$

$$\stackrel{?}{\in} g(m) = 0 \bmod(N)$$

Scăreec găzduit, de gradul d , Coppersmith

putem găsi m în timp polinomial

dacă $k > e$ decarece

$$m < \min N_i < N^{\frac{1}{k}} < N^{\frac{1}{e}}$$

Stacul Franklin - Reiter

Alice are cheia (N, e) în RSA.

Bob trimite mesajele m_1, m_2 ; unde

$$m_1 = f(m_2) \text{ mod } N; f \text{ polinom public.}$$

Săt c_1, c_2 ; Eve are nevoie să deosebească m_1 dacă e este mic.

(Exemplu) $f(x) = ax + b, e = 3$

Eve stie că m_2 este rădăcina modulo n pentru

$$g_1(x) = x^3 - c_2$$

$$g_2(x) = f(x)^3 - c_1$$

Să se calculeze $X - m_2 | g_1$ și $X - m_2 | g_2$.

$\gcd(g_1, g_2) \rightarrow$ sună algoritmii Euclid sau funcționarea
în atunci ca un factor al lui N să fie

→ sună funcționarea, găsim un polinom.

În cazul asta ($f = ax + b$) $X - m_2 = \gcd(g_1, g_2) \rightarrow m_2$

$$\Rightarrow m_1.$$

(Generalizarea lui Coppersmith)

- mai tare decât Hastad.

$N = \text{RSA-modul}, n\text{-bit}$

$m = k\text{-bit message}$

Append $n-k$ -random bits $\rightarrow m' = 2^{n-k}m + r$

Bob trimité același mesaj de donări

$$M_1 = 2 \begin{pmatrix} m \\ n-k \end{pmatrix} + R_1, \quad R_{1,2} \text{ random}$$

$$M_2 = 2 \begin{pmatrix} m \\ n-k \end{pmatrix} + R_2$$

Oscar $y_0 = r_2 - r_1$, tribute per rezultat

$$g_1(x, y) = x^{\ell} - c_1$$

$$g_2(x,y) = (x+y)^e - c_2$$

Atacatorul calculează rezultantul $h(y)$ pe $g_1(x, y)$ și $g_2(x, y)$.

Sef Rezultant:

$$g_1, g_2 \in R[x]$$

$$h = \begin{pmatrix} a_0 & 0 & \dots & 0 & b_0 & 0 & \dots & 0 \\ a_1 & a_0 & \dots & 0 & b_1 & b_0 & \dots & 0 \\ \vdots & & & & \vdots & & & \\ \vdots & & & & \vdots & & & \\ a_d & a_{d-1} & \dots & a_1 & b_d & b_{d-1} & \dots & b_0 \\ & & & & \ddots & & & \\ & & & & & a_d & & \\ & & & & & & & b_d \end{pmatrix}$$

Mr casual

mostry

$$a_i, b_j \in \mathbb{Z}[y]$$

$$h = \overline{\prod}_{(x,y)} (x-y)$$

des râcl lvi h
sunt râcl comme
ale lvi gr, ngr

y este soluție "nică" pt $h(y)$, $\deg h = e^2$

Coppersmith \Rightarrow after $r_1 - r_2$, also after m_2 can run
 F_{attack} by Brute attack

Franklin - Reiter attack

(7)

Definitions of security

Def Securitate perfectă = un adversar cu putere infinită de calcul nu află nimic despre textul clar, având textul cifrat.

Def Securitate semantică = un adversar cu putere polinomială de calcul nu află nimic despre textul clar, având textul cifrat

Pf că există o funcție calculabilă $g: P \rightarrow \{0,1\}$.

Pf că $\Pr(g(m) = 0) = \Pr(g(m) = 1) = \frac{1}{2}$

Pf că $|m| = |c|$ (codificarea păstrează lungimea)

Pf că c este cunoscut publică și y . dar nu cunoaște cheia privată x

Avantajul lui S :

$$Av(S) = \left| \Pr(S(c, x, y) = g(\text{Dec}_x(c))) - \frac{1}{2} \right|$$

Codificarea este semantică sigură $\Leftrightarrow Av(S) \leq \frac{1}{p(R)}$
pentru orice adversar S , orice funcție g și orice polinom p , pentru
"parametru de securitate" R suficient de mare.

Def Securitate polinomială \Leftrightarrow nici un adversar nu poate câștiga următorul joc:

- S primește "Enc_y(m)" și o cheie y .

- Adversarul funcționează în două stadii

Find

Adversarul produce două mesaje m_0, m_1 de lungime egală

Guess

Adversarul primește o criptare c_b al unui mesaj m_b . cu $b \in \{0, 1\}$, și trebuie să găsească b cu probabilitate mai mare ca $\frac{1}{2}$.

Obs Criptarea nu trebuie să fie deterministă pt că altfel el calculează $c_1 = \text{Enc}_k(m_1)$ și compara c_1 cu c_b !

$$\text{Av}(S) = \left| \Pr(S(\text{guess}, c_b, y, m_0, m_1) = b) - \frac{1}{2} \right|$$

Schimă este polinomială sigură $\Leftrightarrow \text{Av}(S) \leq \frac{1}{p(K)}$

✓ adversar
✓ polinomial
K suf de mare

Atacuri

Atac pasiv = chosen plaintext attack = CPA

Eva are acces la diferite mesaje criptate.
Are mașina de criptare "Enc_y(m)".

Chosen ciphertext attack = CCA1

Eva are acces la o mașină de decriptare, poate decripta un nr. polinomial de mesaje (lunch-time attack - proprietatea a faptelor să rămână). După acces trebuie să decripteze un mesaj și a folosi sazină de decriptare. Deci poate folosi "Dec" în fază "Find" dar nu și în fază "Guess".

(3)

Adaptive chosen ciphertext attack = CCA2

Eva are ~~să~~ voie să decripteze orice mesaj cifrat din afara de mesajul pe care trebuie să îl decripteze!

Obs În ziua de azi se pretinde să reziste unui atac CCA2.

Def Un algoritm de criptare cu cheie publică este "sigur" dacă este (semantic sigur) împotriva unui atac CCA2.

(Foarte greu de obținut, de aceea se lucrează cu definiția următoare)

Def Un algoritm de criptare cu cheie publică este "sigur" dacă este (polynomial sigur) împotriva unui atac CCA2.

Teorema

polynomial sigur imp. CPA	\Rightarrow	semantic sigur imp. CPA.
---------------------------------------	---------------	--------------------------------------

Dem Pp. că nu este semantic sigur. Atunci există un adversar S (algoritm), $\text{Av}(S) > \frac{1}{p^{CR}}$, și K suficient de mare, p pol. fixat.

Astăzi să nu poate fi polynomial sigur.

Construim un adversar A .

Find A produce mesaje $m_0 \neq m_1$ cu $g(m_0) \neq g(m_1)$ [oriș egal probabile, deci A încearcă până reușește!]

Guess A formează C_b cu $b \in \{0, 1\}$.

A îl foloară pe S care calculează cât poate el de bine $g^{(mb)}$.

Folosind ce a ghicit S', A compară valoarea en $g^{(mo)}$ și $g^{(m_1)}$ și găsește b.

$$\Pr(A(\text{guess}, c_b, y, m_0, m_1) = b) = \Pr(S(c, y) = g^{(\text{Decy})})$$

Obs RSA (în stare pură) nu este polinomial sigur (nup $m_1 \neq m_2$).

(Dem) Atacatorul stie că s-a criptat unul din mesajele m_1 , m_2 sau m_2 (pot fi "da" sau "nu"); "vănd" sau "cumpăr"; "fete" sau "băiat". Atacatorul calculează $c' = m_1^e \pmod{N}$.

$$c' = c \Rightarrow m = m_1 \quad \square$$

$$c' \neq c \Rightarrow m = m_2$$

Obs Proprietatea homomorfică a lui RSA:

$$(m_1 m_2)^e \pmod{N} = (m_1^e \pmod{N})(m_2^e \pmod{N}) \pmod{N}$$

Obs RSA nu este CCA2 sigur.

$$\text{Eva are să spargă } c = m^e \pmod{N}$$

Eva crează $c' = 2^e c \pmod{N}$. Eva decriptează $c' \Rightarrow m'$

$$m'^{-1} = c'^{-1} d^{-1} 2^{-1} = (2^e c)^{-1} d^{-1} 2^{-1} = 2^{m-2} \equiv m$$

Decision Diffie Hellman problem DDH:

(11)

given g^x, g^y, g^z determine whether $xy = z \pmod{\#G}$

Obs DDH este în grupul $G \Rightarrow$ Elgamal polinomial sigură împ
împreună CPA.

Cifrul Elgamal: (g^K, m^h) unde K este o alegere efemerică
 h = cheia publică.

Algoritm de rezolvare a lui DDH:

- Input g^x, g^y, g^z .
- $h := g^x$
- $(m_0, m_1) = A(\text{rand}, h)$
- $c_1 = g^y$
- Alege $b \in \{0, 1\}$
- $c_2 = m_b \cdot g^z$
- $b' = A(\text{guess}, (c_1, c_2), h, m_0, m_1)$
- If $b = b'$ return true, else false. □

TOTUȘI Elgamal nu este CCA2 sigură.

Pp. Eva are de săpt mesajul $c = (c_1, c_2) = (g^K, m \cdot h^K)$

Eva creează $c' = (c_1, 2c_2)$, și decriptează $\Rightarrow m'$

$$m' \cdot 2^{-1} = 2c_2 c_1^{-x} 2^{-1} = 2m h^K g^{-xK} 2^{-1} = m g^{xK} g^{-xK} = m!$$

Lemă

Dacă QUADRES este grea \Rightarrow Goldwasser - Micali
pentru un modul N
ESTE ~~este~~ polinomial sigură
impotriva unor atacuri CPA.

Vrem să decidem dacă $y \in Q_N$.

Find

$$m_0 = 0, m_1 = 1$$

Guess

$x = y$. Dacă ghiceste corect pt care mesaj c este un c

\rightarrow ghiceste dacă $y \in Q_N$ (adică un patrat)

Lemă

Goldwasser - Micali rezigură contra CPA2.

$$c = y^b \cdot x^2 \pmod{N}$$

$$\text{Producem } c' = c \cdot z^2 \pmod{N}, z \neq 1.$$

Descriem c' !

Predicăt greu de calculat

Definire

$f: S \rightarrow T$ one way function.

$B: S \rightarrow \{0, 1\}$ "predicăt"

B "difícil" \Leftrightarrow $(B(x) \text{ usor}) \text{ dar } (B(x) \text{ greu de calculat dat } x)$
(sau hard)

Teorema

$\{g\}$ grup ciclic de ordin q , g generator, q prim!

$$B_2 = x \pmod{q}$$

$\Rightarrow B_2$ dificil pentru funcția $f(x) = g^x$.

(13)

Cu alte cuvinte
"Paritatea este dificilă pentru logaritmul secret."

(Din) Fie $\mathcal{O}(h, g)$ un oracol care calculează
 $B_2(\log g^h)$ adică ultimul bit
al logaritmului

Vrem să arătăm că putem calcula logaritmul secret.

- Input $h = g^x$ (x necunoscut)

- $t = 2^{-1} \bmod q$

- $y = 0$, $z = 1$

- while ($h \neq 1$)

$b = \mathcal{O}(h, g)$

if $b = 1$ then $y += z$; $h /= g$;

$h = h^t$; $z = 2z$;

- Output y .

~~Exemplu $p=607$, $g=67$; $\log p=10+$, $g=69$, $h=56$~~

~~$\mathcal{O}(h, g)$~~

Asumător

Teorema

Pentru problema RSA, dat $c = m^e \bmod N$

$B_1(m) = m \bmod 2$ și $B_{p_2}(m) = 0$ dacă $m < \frac{N}{2}$, și altfel

sunt predicate dificile.

$$\text{Obs} \quad \mathcal{O}_h(c, N) = \mathcal{O}_1(c \cdot 2^e \bmod N, N)$$

$$\mathcal{O}_1(c, N) = \mathcal{O}_h(c \cdot 2^{-e} \bmod N, N)$$

(14)

Schimb de cheie și schimb de
semnatură

Schimb de cheie Diffie - Hellman

G grup ciclic

Alice - alege a - trimit g^a - calculează $(g^b)^a = g^{ab}$	Bob - alege b - trimit g^b - calculează $(g^a)^b = g^{ab}$
---	---

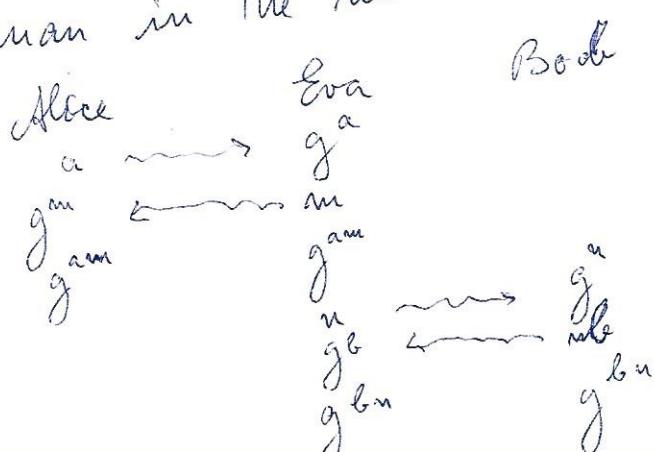
Eva interceptază g^a și g^b dar nu poate calcula g^{ab} fără un calcul de logaritmul discret.

→ Sun $G = \mathbb{F}_p^\times$ cu $|p| \approx 1024$

→ Sun $G = \langle P \rangle$ în $E(\mathbb{F}_p)$, o curbă eliptică.

În acărlă căz este suficient să transmitem 160 biti.

Atacul "man in the middle":



Pentru a evita atacul "man in the middle" avem
nevoie de un sistem sigur de numaratori
① = schema de apelare

Schemā

mesaj + cheie secretă a lui Alice = semnătură
a lui Alice

1

Bob : mesaj + cheie secretă a lui Alice = da/cu
 mesaj + semnătură + cheie publică a lui Alice = da/cu
 (2) = schema cu recuperarea mesajului

② = schema en recuperación nalgas

2

mesaj + cheie ~~secretă~~ a lui Alice = signature
 mesaj + cheie publică a lui Alice = da/mu + mesaj'

RSA poate fi folosit ca algoritm de securitate cu
recuperare de mesaj:

- Esitația folosește cheia lui secretă:
recuperare sau

$\beta \equiv m^d \pmod{N}$ (Coloseste dec !)

- Receptorul folosite încă în recuperarea mesajului

$$M = P \pmod{N}$$

Dimension aumenta redundante

Acceasta schema
permite verificarea
de la revizie
semestrială:

$$-|m|=t$$

$$- |m| = t \leq k-32$$

- $|N| = K$ unde $t \leq K-3$
 I se arunca o coroana a. 18 | m

- $|N| = R$ - a. $\approx 8 \text{ fm}$
- pad m cu zeroval a. la stanga lui m pt a produce
adaugare $(k-t)/18$ legtes
- FF || 00 || M

$$m^1 = oo \parallel o \parallel FF \parallel \dots \parallel FF \parallel oo \parallel m$$

$$- \Delta = m^d \bmod N.$$

(Folosirea funcției hash în scheme de semnatură)

Vrem să semnăm m .

Calculăm $h(m)$, apoi aplicăm ~~salad~~ RSA signing for $h(m)$

$$s = h(m)^d \pmod{N}$$

$\Rightarrow (m, s)$ pereche mesaj + semnatură

Verificarea semnaturii:

- calculăm $h' = s^e \pmod{N}$

- calculăm $h(m) = h$

- dacă $h = h'$ da, altfel nu

O subiectățire a acestei idei este

Digital Signature Algorithm

DSA

Ca varianta EC-DSA pentru curbe eliptice.

DSA = schema din appendix. Produce o pereche (r, s) de căte 160 bits

DSA = schema din appendix. Produce o pereche (r, s) de căte 160 bits

DSA = schema din appendix. Produce o pereche (r, s) de căte 160 bits

r = o funcție de o cheie efemera K ce se schimbă la fiecare mesaj

s = o funcție de

$\begin{cases} \text{mesaj} \\ \text{cheie secretă a semnatului} \\ r \\ \text{cheie efemera } K \end{cases}$

① Se alege prim q de 160 bits și prim p :

$|p|$ între 512 și 2048 bits

$$q | p-1$$

(17) ② Se generează randomă $h < p$, și calculează

$$g = h^{\frac{p-1}{q}} \quad (\text{refacem păușe când } g \neq 1)$$

$\mod p$

Astăzi $\text{ord}(g) = q$ în grupul \mathbb{F}_p^\times .

$$g^q = 1 \mod p$$

③ Odată ce avem (p, q, g) fiecare user generează cheie
lui secretă x cu $0 < x < q$. Cheia publică asociată y este

$$y = g^x \mod p$$

Scriere a mesajului m

- Calculează $h = H(m)$

- Alege cheie efemera k , $0 < k < q$

- Calculează
 $r = (g^k \mod p) \mod q$

- Calculează

$$\beta = (h + x r)^{k^{-1}} \mod q$$

- Trimit (r, β) ca semnatură a lui m .

Verificare

Calculează $h = H(m)$

$$a = h \beta^{-1} \mod q$$

$$b = r \beta^{-1} \mod q$$

$$v = (g^{a y^b} \mod p) \mod q$$

unde $y =$ cheie
publică a emisatorului

Accepted signature $\Leftrightarrow N = n$

(18)

$$g^a y^b = g^{h/s} r/s = g^{h/s} g^{r/s} = g^{(h+r/s)/s} = g^{k \text{ mod } p}$$

$$\Rightarrow (g^a y^b \text{ mod } p) \text{ mod } q = (g^k \text{ mod } p) \text{ mod } q = r \stackrel{!}{=} 0 \text{ OK}$$

Exemplu

$$q = 13, p = 4q + 1 = 53, g = 16$$

$$x = 3 \Rightarrow y = g^3 \text{ mod } p = 15$$

Dacă $H(m) = 5$, în clasa sfemei $k = 2$

$$r = (g^k \text{ mod } p) \text{ mod } q = 5$$

$$s = (h + x \cdot r) / k \text{ mod } q = 10$$

Verificarea

$$a = \frac{h}{s} \text{ mod } q = 7$$

$$b = \frac{r}{s} \text{ mod } q = 7$$

$$n = (g^a y^b \text{ mod } p) \text{ mod } q = 5$$

OK