Curs 11 + 12

DSA are o variantă pe curbe eliptice, EC-DSA.

Diferențele sunt următoarele

	DSA	EC-DSA
G :	$\langle g \rangle \leq \mathbb{F}_p^*$	$\langle P \rangle \leq E(\mathbb{F}_p)$
y :	g^x	$[x]P$
f :	$\bmod q$	$\pi_x(P) \bmod q$ (coord x)

Exemplu

Curbe eliptică $Y^2 = X^3 + X + 3$ peste $\mathbb{F}_{199}$ are $q = 197$ elemente. Deoarece 197 este prim $E(\mathbb{F}_{199})$ e grup ciclic!

Generator $P = (1, 76)$

Cheie privată $x = 29$, cheie publică $Y = [x]P = [29](1,76) = (113, 191)$

Presupunem că userul vrea să semneze un mesaj cu valoare $H(m) = 68$. Întâi produce o cheie efemeră $k = 153$ și calculează

$$r = \pi_x([k]P) = \pi_x([153](1,76)) = \pi_x((185, 35)) = 185$$

Apoi calculează

$$s = (H(m) + x \cdot r) \cdot k^{-1} \pmod q = (68 + 29 \cdot 185) \, 153^{-1} \bmod 197$$

$$= 78$$

Signatura este $(r, s) = (185, 78)$

Verificare $a = H(m)\, s^{-1} \bmod q = 68 \cdot 78^{-1} \bmod 197 = 112$

$b = r\, s^{-1} \bmod q = 185 \cdot 78^{-1} \bmod 197 = 15$

$Z = [a]P + [b]Y = [112](1,76) + [15](113,191)$

$= (111,60) + (122,140) = (185,35)$

↑ $E(\mathbb{F}_{155})$ adăugăm

$r = 185 = \pi_x(Z)$

MQV

O generalizare directă a lui Diffie Hellman:

Alice și Bob au perechi (K_p, K_s) de lungă durată:

$(A = g^a, a)$ $\quad$ $(B = g^b, b)$

Pentru a face schimb simetric de cheie secretă, generează

niște chei efemere:

$(C = g^e, c)$ $\quad$ $(D = g^d, d)$

Alice $\xrightarrow{C}$ Bob

Alice $\xleftarrow{D}$ Bob

Alice știe acum A, B, C, D, a, e

$l = \frac{1}{2} \log \#G$

Alice

$s_A = 2^l + (C \bmod 2^l)$

$t_A = 2^l + (D \bmod 2^l)$

$h_A = c + s_A\, a$

$P = (D B^{t_A})^{h_A}$

Bob

$$s_B = 2^\ell + (D \bmod 2^\ell)$$
$$t_B = 2^\ell + (C \bmod 2^\ell)$$
$$h_B = d + s_B b$$
$$P_B = (CA^{t_B})^{h_B}$$

Atunci $P_A = P_B$ este secretul împărtășit.

Alb $s_A = t_B$, $s_B = t_A$

$$\log(P_A) = \log\left((DB^{t_A})^{h_A}\right) = (d + b t_A) h_A =$$
$$= d(c + s_A a) + b t_A (c + s_A a) =$$
$$= d(c + t_B a) + b s_B (c + t_B a) =$$
$$= c(d + s_B b) + a t_B (d + s_B b)$$
$$= (c + a t_B) h_B = \log\left((CA^{t_B})^{h_B}\right) = \log(P_B) !$$

OAEP - RSA (Optimized Assymetric Encryption Padding)

sau cum se face RSA să fie CCA2 sigură.

$f: \{0,1\}^K \longrightarrow \{0,1\}^K$, one-way trap-door permutation

de exemplu $f(m) = m^e \bmod N$

Fie K_0, K_1 nr a.î. $2^{K_0}, 2^{K_1}$ prea mari pt a fi posibile; de ex $K_0, K_1 > 128$.

$$n = K - K_0 - K_1$$

$n + K_1$

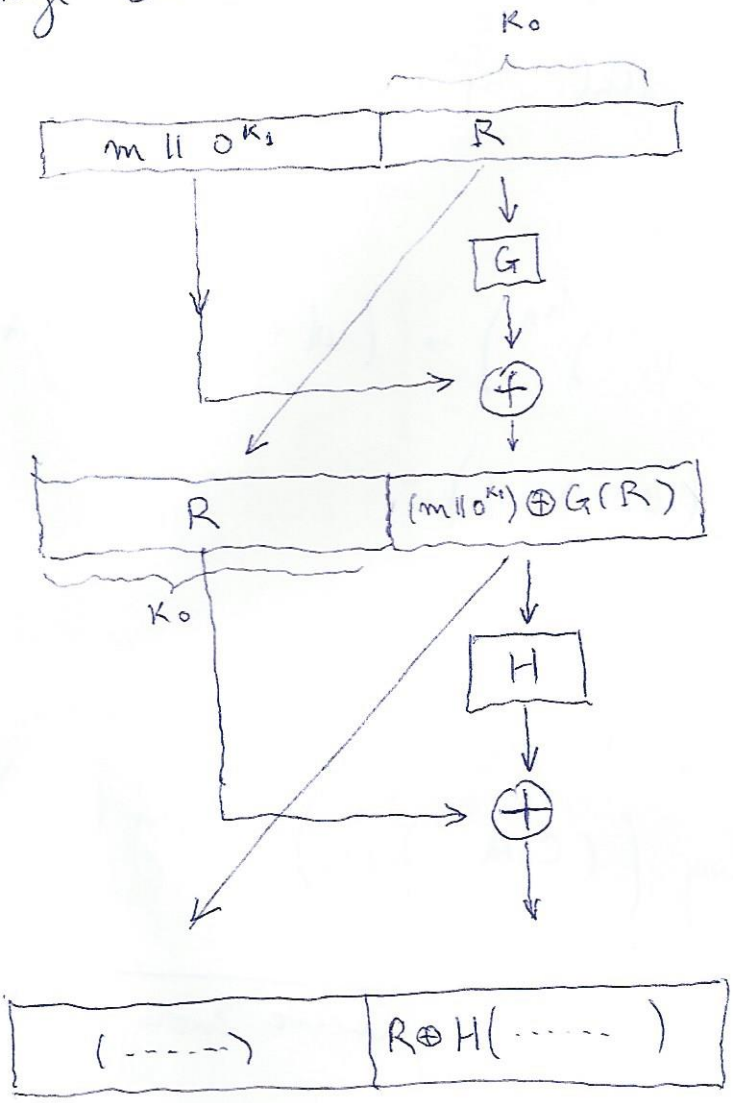
$G : \{0,1\}^{k_0} \longrightarrow \{0,1\}^{n+k_1}$

$H : \{0,1\}^{n+k_1} \longrightarrow \{0,1\}^{k_0}$

funcții hash

m mesaje, $|m| = n$.

Two stage Feistel network:



R random, $|R| = k_0$

$$Enc(m) = f\left(\left[(m \| 0^{k_1}) \oplus G(R)\right] \| \left[R + H\left((m \| 0^{k_1}) \oplus G(R)\right)\right]\right)$$

Decriptare

Aflăm $T \| (R \oplus H(T))$ unde $T = (m \| 0^{k_1}) \oplus G(R)$

Calculăm $H(T)$, deducem R din $R \oplus H(T)$

Calculăm $G(R)$, deducem m!

Teoremă RSA-OAEP este semantic sigură împotriva unui CCA2.

Cum îl salvăm pe Elgamal?

Transformarea schemelor CPA în scheme CCA2

Fie Enc o criptare sigură împ. CPA.
Știm că trebuie să fie nedeterministă, deci $Enc(m, r)$.
De exemplu la Elgamal $Enc(m, r) = (g^r, m h^r)$.

Definim $Enc'(m, r) = Enc(m \| r, H(m \| r))$, H hash function.

Decriptarea Se calculează $m' = Dec(c)$, apoi se verifică dacă $c = Enc'(m', H(m'))$. (dacă nu, returnăm $\perp$)

Deci în cazul Elgamal:

$$Enc(m, r) = \left(g^{H(m \| r)}, (m \| r)\, h^{H(m \| r)} \right)$$

Protocoale Avansate

Secrete sharing schemes (scheme cu secret partiționat...)

Un secret s trebuie partiționat între n persoane $\mathcal{P}$.

Exemplu s = codul pt lansarea bombei atomice

$n = 4$: Președintele (P), Vicepreședintele (V), Secretarul de apărare (S)

Generalul G;

Vrem ca următoarele mulțimi să poată lansa bomba:

$\{P, G\}$, $\{V, S, G\}$ dar nu submulțimi.

s_P, s_G, s_V, s_S = partea de secret pe care o primește fiecare pers

Structură de acces = colecție $\Gamma \subset \mathcal{P}(P)$ a.î. $P \in \Gamma$

$A \in \Gamma, A \subset B \Rightarrow B \in \Gamma$

Schemă de partiționare a secretului = Doi algoritmi **Share** și **Recombine**

Share (s, Γ) repartizează fiecărei persoane X o parte de secret s_X

Recombine fiecare $H = \{s_X \mid X \in G\}$

Dacă $G \in \Gamma$ return s, otherwise return nothing.

În exemplul cu președintele

$\Gamma = \{ PG, VSG, PGV, PGS, PVGS \}$

unde PG, VSG mulțimi minimale.

Un al doilea exemplu

Patru directori A, B, C, D. Oricare doi dintre ei pot colabora și deschid dulapul cu contracte.

$\Gamma = \{ AB, AC, AD, BC, BD, CD, ABC, ABD, BCD, ABCD \}$

unde AB, AC, AD, BC, BD, CD mulțimi minimale.

Ito - Nisizeki - Saito secret sharing

$m(\Gamma)$ = mulțimea mulțimilor minimale

$\forall O \in m(\Gamma)$, $l = |O|$, generăm secretele

s_i a.î. $s_1 \oplus \dots \oplus s_l = s$.

Ex 1

$s = s_1 \oplus s_2 = s_3 \oplus s_4 \oplus s_5$

$s_P = s_1$

$s_V = s_3$

$s_S = s_4$

$s_G = s_2 \| s_5$

Atenție, G trebuie să decidă în care subm. este și să scoată acel cod.

$s = s_1 \oplus s_2 = s_3 \oplus s_4 = s_5 \oplus s_6 = s_7 \oplus s_8 = s_9 \oplus s_{10}$

$= s_{11} \oplus s_{12}$, atunci

Sunt 6 descompuneri!

$s_A = s_1 \| s_3 \| s_5$

$s_B = s_2 \| s_7 \| s_9$

$s_C = s_4 \| s_8 \| s_{11}$

$s_D = s_6 \| s_{10} \| s_{12}$

$AB \rightsquigarrow s_1 \oplus s_2$

$AC \rightsquigarrow s_3 \oplus s_4$

$AD \rightsquigarrow s_5 \oplus s_6$

$BC \rightsquigarrow s_7 \oplus s_8$

$BD \rightsquigarrow s_9 \oplus s_{10}$

$CD \rightsquigarrow s_{11} \oplus s_{12}$

Replicated secret sharing

(1) Fie $A_1, \dots A_t$ multimile MAXIMALE care nu sunt în Γ

(2) Fie $B_1, \dots B_t$ complementele lui $A_1, \dots A_t$

(3) Pentru fiecare B_i se calculează un secret s_i a.î. $s = s_1 \oplus \dots \oplus s_t$

(4) Persoana P primește $s_i \Leftrightarrow P \in B_i$

Ex 1 Multimile maximale slabe sunt:

$A_1 = \{P, V, S\}$, $A_2 = \{V, G\}$, $A_3 = \{S, G\}$

Complementele sunt:

$B_1 = \{G\}$, $B_2 = \{S, P\}$, $B_3 = \{P, V\}$

$s = s_1 \oplus s_2 \oplus s_3$

$s_P = s_2 \| s_3$; $s_V = s_3$, $s_S = s_2$, $s_G = s_1$

Exemplul 2

$A_1 = \{A\}$, $A_2 = \{B\}$, $A_3 = \{C\}$, $A_4 = \{D\}$

$B_1 = \{BCD\}$ $B_2 = \{ACD\}$ $B_3 = \{ABD\}$ $B_4 = \{ABC\}$

$$S = S_1 \oplus S_2 \oplus S_3 \oplus S_4$$

$$S_A = S_2 \parallel S_3 \parallel S_4$$

$$S_B = S_1 \parallel S_3 \parallel S_4$$

$$S_C = S_1 \parallel S_2 \parallel S_4$$

$$S_D = S_1 \parallel S_2 \parallel S_3$$

Ambele sunt frumoase, dar sunt ineficiente.

Vom face Shamir „Secret Sharing", dar care are nevoie de

Coduri Reed - Solomon

- error correcting codes
- $\mathbb{F}_q$, $q = p^k$, corp finit
- n = lungimea cuvintelor din cod (cele cu care exprimăm literele)
- t = nr. de erori care pot fi corectate
- $X \subseteq \mathbb{F}_q$, $|X| = n$. Dacă chiar $\mathbb{F}_q > n$, $X = \{1, 2, \dots n\}$ altfel e orice mulțime.
- $0 \notin X$
- $\mathcal{P} = \{ f_0 + f_1 X + \dots + f_t X^t \mid f_i \in \mathbb{F}_q \}$

$|\mathcal{P}| = q^{t+1}$

Code words $C = \{ (f(x_1), f(x_2), \dots, f(x_n)) \mid f \in \mathcal{P}, x_i \in X \}$

un code-word are lungimea $n \cdot \log_2 q$.

Exemplu $q = 101$, $n = 7$, $t = 2$, $X = \{1, 2, 3, 4, 5, 6, 7\}$

Data $\hat{=}$ $f = 20 + 57X + 68X^2 \in \mathcal{P}$

$c = (44, 2, 96, 23, 86, 83, 14)$ $f(x_i) \bmod q$.

Recuperarea datelor

Primesc $c = (c_1, \dots c_n)$ dar nu cunosc polinomul.

Caut $f = \sum_{j=0}^{t} f_j X^j$

Sau $\begin{cases} c_1 = f_0 + f_1 x_1 + \dots + f_t x_1^t \\ \dots \\ c_n = f_0 + f_1 x_n + \dots + f_t x_1^t \end{cases}$ și rezolv sistemul cu det Vandermonde

Sau fac <u>interpolare Lagrange</u>

$f(X) = \sum_{i=1}^{n} c_i \delta_i(X)$ unde $\delta_i(X) = \prod_{j \neq i} \frac{X - x_j}{x_i - x_j}$

unde $\delta_i(x_i) = 1$, $\delta_i(x_j) = 0$ dacă $i \neq j$, $\deg \delta_i(X) = n - 1$.

Error detection

Dacă apare $\leq t$ erori, când calculez $f = \sum_1^t c_i \delta_i$ obțin un polinom de grad $\neq 2$!

Error correction

Berlekamp - Welch algorithm

(nu avem nevoie).

Shamir Secret Sharing

s secret, n persoane, fiecare grup de t persoane nu pot reconstitui secretul, fiecare grup de $t+1$ persoane poate reconstitui secretul

(1) Se generează polinom $f(X)$, $\deg f = t$, $f(0) = s$

$$f(X) = s + f_1 X + \dots + f_t X^t.$$

(2) Se alege $X \subseteq \mathbb{F}_q \setminus \{0\}$, ~~de exemplu~~ $|X| = n$.

(3) Persoana i primește $s_i = f(x_i)$.

Deci $(s_1, \dots s_n)$ este code-word în cod Reed-Solomon.

Dacă $t+1$ persoane pun valorile laolaltă, ei rezolvă polinomul f prin sistem de ecuații!

Altă metodă:

$$s = f(0) = \sum_{i=1}^{n} s_i \delta_i(0)$$

$Y \subseteq X$, ~~$\delta_Y =$~~ $r_{x_i, Y} = \prod_{\substack{x_j \in Y \\ x_j \neq x_i}} \frac{-x_j}{x_i - x_j}$

$\#Y > t$

$$\boxed{s = \sum_{x_i \in Y} r_{x_i, Y} \cdot s_i}$$

Commitments and oblivious transfer.

Bob și Alice joacă "piatră - hârtie - foarfece" prin telefon.

$A \longrightarrow B \quad h_A = H(R_A \| \text{HARTIE})$, R_A random string

$B \longrightarrow A$ "FOARFECE"

$A \longrightarrow B \quad R_A \| \text{HARTIE}$

Bob poate calcula singur $H(R_A \| \text{HARTIE})$

Dacă Alice vrea să trișeze, trebuie să găsească <u>repede</u> R'_A a.î.

$$H(R_A \| \text{HARTIE}) = H^*(R'_A \| \text{PIATRĂ})$$

Asta nu se poate dacă H^* este <u>rezistent la a doua preimagine</u>

Commitment = angajament!

Def O <u>schemă de commitment</u> este (binding) dacă nici un adversar nu poate câștiga următorul joc:

- Adversarul produce c, x, r a.î. $c = C(x, r)$
- Adversarul produce $x' \neq x$ și r' a.î.:

$$C(x, r) = C(x', r')$$

Def O <u>schemă de commitment</u> este <u>sigură</u> (concealing) dacă nici un adversar nu poate câștiga următorul joc:

- Adversarul produce x_0 și x_1, $|x_0| = |x_1|$
- Autoritatea generează r random și $b \in \{0, 1\}$
- Autoritatea calculează $c = C(x_b, r)$ $\rightsquigarrow$ Adversarului
- Adversarul ghicește b.

Obs Nici o schemă nu este absolut binding și concealing

Dar perfect binding $\Rightarrow$ injectivă (angajamente $\to$ valori angajate)

$\Rightarrow$ deterministă $\Rightarrow$ nu poate fi perfect concealing!

Obs Schema $H(R \| C)$, R random, H hash-function.

este computational binding și (information-theoretically concealing)

În practică apar următoarele scheme:

G ciclic de ordin q, $G = \langle g \rangle$, $g \in \mathbb{F}_p^\times$

$h \in G$, $\log_g h$ necunoscut de user.

Schemă de angajament pt $x \bmod q$:

$$B(x) = g^x$$

Pedersen: $B_a(x) = h^x g^a$, unde a ales random.

Schemă de angajament pt $x \bmod p$

$$E_a(x) = (g^a, x \cdot h^a) \quad \text{(Elgamal!)}$$

Obs $B(x)$ information-theoretically binding (dar injectivă...)

Obs E_a este c-th binding și comput. concealing.

Obs B_a comput. binding și inf. th. concealing.

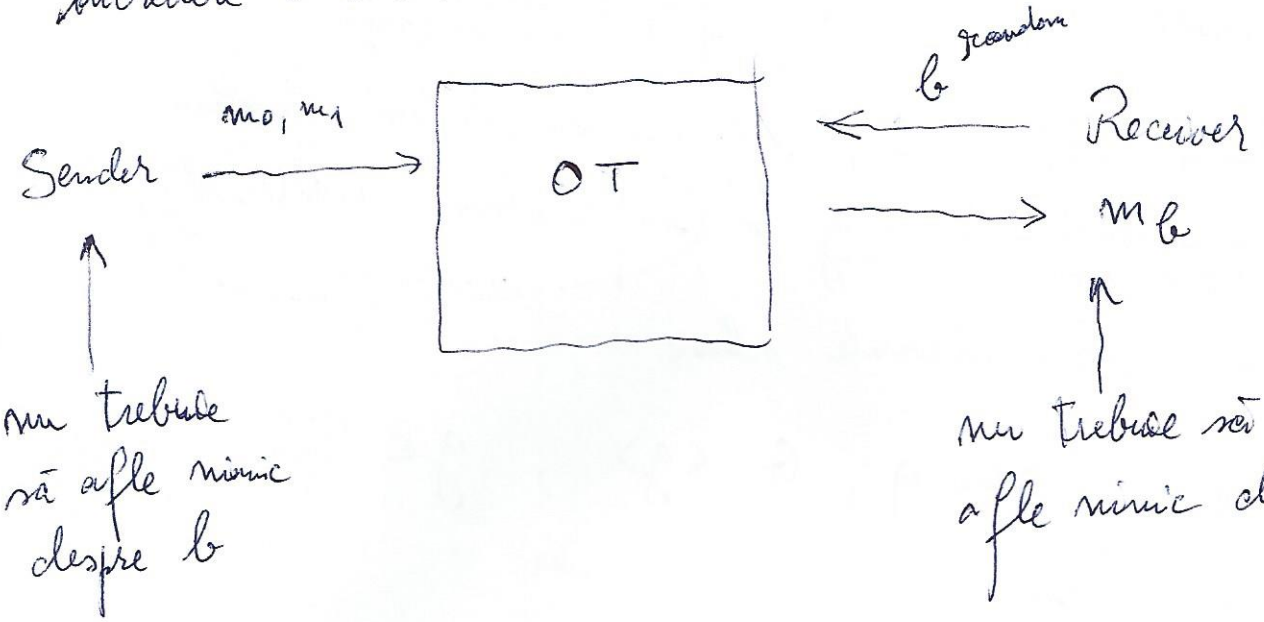
$$B(x_1)\,B(x_2) = B(x_1 + x_2)$$

Obs B și B_a homomorfisme: $B_{a_1}(x_1)\,B_{a_2}(x_2) =$

$$= B_{a_1+a_2}(x_1 + x_2)$$

Oblivious Transfer OT

Ca și la Commitments, în OT ambele părți nu au încredere una în alta



$G = \langle g \rangle$, ordin q.

Elgamal $(K_p, K_s) = (h = g^x, x)$

$H : G \to \{0,1\}^n$ hash-function.

$|m| = n \Rightarrow \quad c = (c_1, c_2) = (g^k, m \oplus H(h^k))$

k cheie efemeră.

decriptare $\quad c_2 \oplus H(c_1^x) = m \oplus H(h^k) \oplus H(g^{kx}) = m$

Ideea Receiver produce două chei publice, dar cunoaște numai o cheie secretă. Sender codifică cele două mesaje și le trimite. Receiver poate decripta doar un mesaj. Sender nu știe pe care.

~~- Sender alege $c \in G_q$, o trimite lui Receiver.~~
~~(nu cunoaște $\log_g c$!)~~