Securitatea Rețelor și Serviciilor de Comunicații

Algoritmi criptografici: Confidențialitatea datelor

Partea a 2-a

Confidențialitatea datelor folosind criptografie cu cheie secretă

Scheme de criptare cu cheie secretă bazate pe cifruri bloc

Scheme de criptare cu cifru bloc

Aplicații ale cifrurilor bloc: criptografie simetrică (cheie secretă)

- Scheme de criptare ("moduri de operare pentru confidențialitate").
- Scheme de autentificare ("moduri de operare pentru autentificare").
- Scheme de criptare autentificată: autentificare + confidențialitate.
- Alte aplicații: funcții hash, generatoare de șiruri pseudoaleatoare, etc.

Scheme standard de criptare cu cifru bloc

(NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation. Methods and Techniques, 2001.)

- Electronic Code Book (ECB).
- Cipher-Block Chaining (CBC).
- Counter (CTR).
- Cipher feedback (CFB).
- Output feedback (OFB).

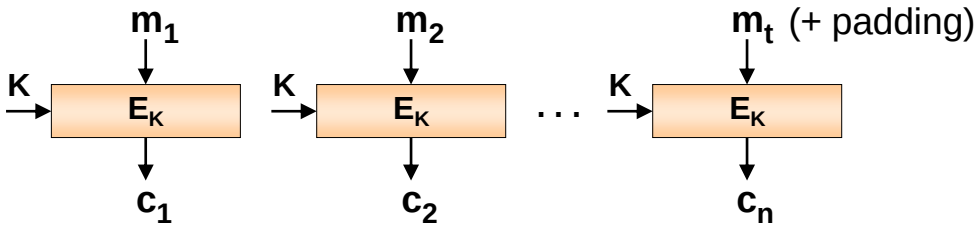
. . .

Vom discuta trei exemple mai importante: ECB, CBC, CTR.

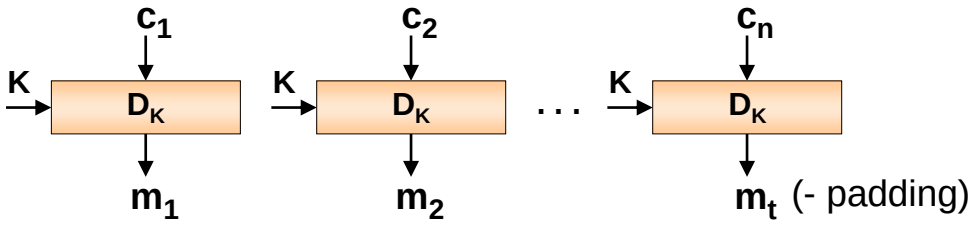
ECB nu îndeplinește IND-CPA. Celelate scheme îndeplinesc IND-CPA și, în combinație cu o schemă de autentificare, pot oferi IND-CCA.

Electronic Code Book (ECB)

Criptare



Decriptare



Parameters:

k-bit key K.

Summary:

- Encrypt n-bit plaintext blocks m_1 , ... , m_t to n-bit ciphertext blocks c_1 , ... , c_t.

- Decrypt to recover plaintext.

Encryption:

For $1 \le j \le t$

$c_j \leftarrow E_K(m_j)$

Decryption:

For $1 \le j \le t$

$m_j \leftarrow D_K(c_j)$

- ECB: Soluție simplă pentru criptarea unui șir binar de lungime finită arbitrară, $m \in \{0,1\}^*$, folosind un cifru bloc cu intrări de lungime fixă, n biți.
- Textul clar m este extins ("padding") astfel încât lungimea sa să fie multiplu de n și divizat în blocuri de n biți, $m_1, m_2, ..., m_t$.
- Fiecare bloc este criptat folosind cifrul bloc cu aceeași cheie: $c_j = E_K(m_j)$.
- Similar, simetric, pentru decriptare.

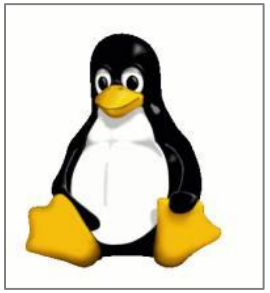
(In)Securitatea schemei ECB

Schema ECB nu îndeplinește cerința IND-CPA

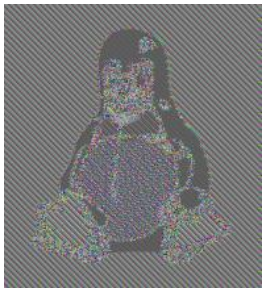
- Adversarul câștiga jocul IND-CPA cu probabilitate 1 pe baza răspunsului la interogarea: $(m_0 = 0^{2n}, m_1 = 0^n 1^n)$.
- În general, un algoritm de criptare care este determinist și nu menține informație de stare, nu poate îndeplini IND-CPA.
- Schema ECB nu poate fi folosită decât în situații particulare.

Justificare intuitivă a insecurității schemei ECB

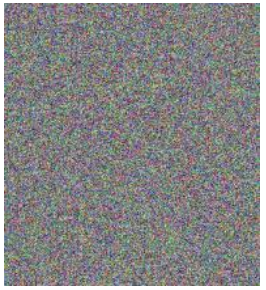
- Fiecare bloc de text clar este criptat cu aceeași funcție.
- Blocuri de text clar identice, $m_i = m_j$, produc blocuri de text cifrat identice: $E_K(m_i) = E_K(m_j)$.
- Deci textul cifrat produs de ECB dezvăluie informații privind structura textului clar.



Text clar



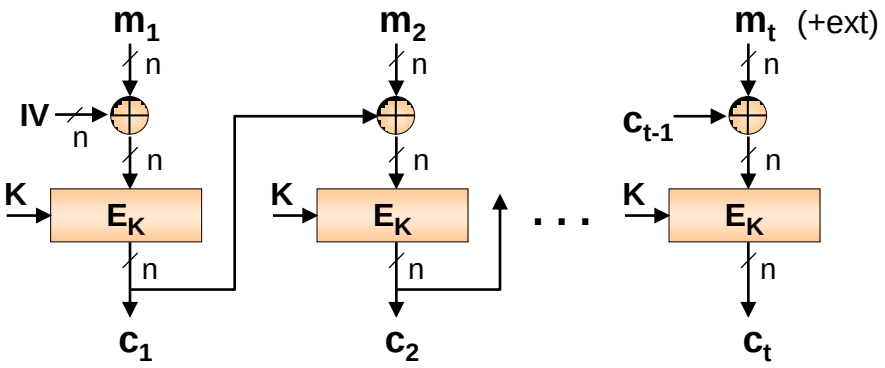
Criptare ECB



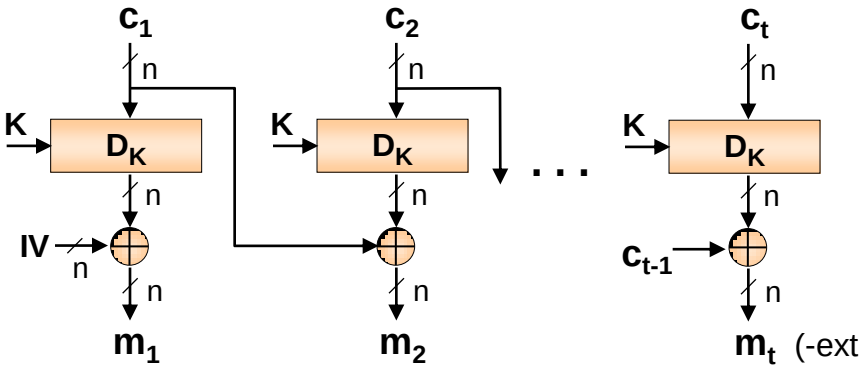
Criptare corectă (CBC, CTR, etc.)

Cipher-Block Chaining (CBC)

Criptare



Decriptare



Parameters:

- k-bit key K.
- n-bit initialization vector IV.

Summary:

- Encrypt n-bit plaintext blocks $m_1, \ldots, m_t$ to n-bit ciphertext blocks $c_1, \ldots, c_t$.
- Decrypt to recover plaintext.

Encryption:

$c_0 \leftarrow IV$

For $1 \le j \le t$

$c_j \leftarrow E_K(c_{j-1} \oplus m_j)$

Decryption:

$c_0 \leftarrow IV$

For $1 \le j \le t$

$m_j \leftarrow c_{j-1} \oplus D_K(c_j)$

- Idee: Elimină deficiențele schemei ECB modificând în mod aleator funcția de criptare aplicată fiecărui bloc de text clar: $c_j \leftarrow E_K(c_{j-1} \oplus m_j)$, $j \ge 2$.
- Pentru m_1, este necesar un vector de inițializare (IV): $c_1 \leftarrow E_K(IV \oplus m_1)$.

© Octavian Catrina

Analiza schemei CBC (1)

Criptarea textelor clare cu lungime arbitrară

- CBC are nevoie de o funcție $m \rightarrow m'$, care adaugă lui m un minim de biți astfel încât lungimea lui m' să fie multiplu al lungimii blocului (padding). Funcția trebuie să fie bijectivă pentru a recupera m la decriptare.

  Exemplu: Dacă ultimul bloc clar are $m < n$ biți, adaugă 1 bit '1' urmat de $(n-m-1)$ biți '0'; dacă ultimul bloc clar are $m = n$ biți, adaugă 1 bit '1' urmat de $(n-1)$ biți '0'.

Criptare și decriptare

- Un bloc cifrat c_j depinde de blocul clar m_j și (prin intermediul lui c_{j-1}) de toate blocurile clare precedente și de blocul IV.
- Pentru fiecare criptare trebuie ales alt bloc IV. Astfel, chiar și pentru texte clare identice se obțin texte cifrate diferite.
- Pentru a decripta blocul c_j este necesar blocul precedent c_{j-1}. În particular, pentru a decripta c_1 este necesar blocul IV.
- Blocul IV trebuie transmis ca parte a textului cifrat și *nu este secret*.
- Dezavantaj: textul cifrat este mai lung decât textul clar: padding + IV.

Analiza schemei CBC (2)

Securitatea criptării CBC

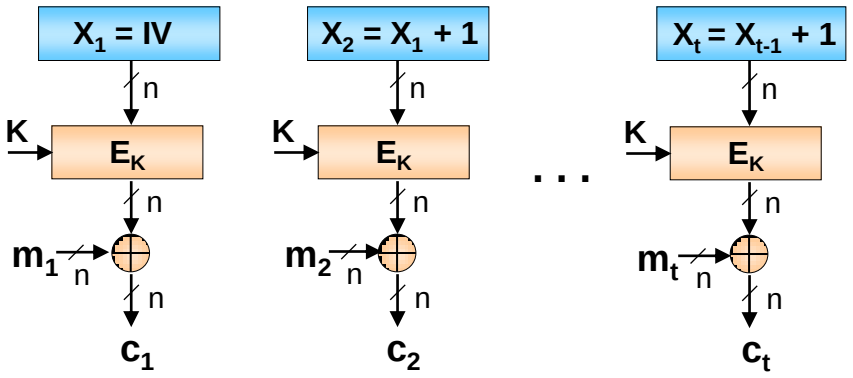
- **Teoremă**: Schema CBC îndeplinește IND-CPA dacă și numai dacă:
  (1) cifrul bloc utilizat este o permutare pseudoaleatoare (PRP) și
  (2) IV este ales în mod aleator (impredictibil) pentru fiecare text cifrat.
  Cheia trebuie schimbată după ce au fost criptate b << $2^{n/2}$ blocuri.

Alte proprietăți ale schemei CBC

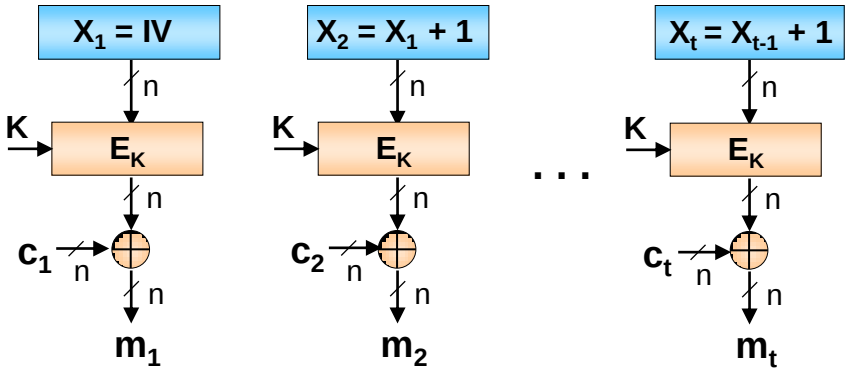
- Calcul paralel: Nu se pot cripta mai multe blocuri în paralel (trebuie calculat $E_K(c_{j-1} \oplus m_j)$). La decriptare, se poate calcula în paralel $D_K(c_j)$.
- Acces aleator la textul cifrat: Se poate decripta separat orice porțiune de text cifrat (nu e nevoie să fie decriptate blocurile precedente).
- Sincronizare: Dacă se pierd porțiuni de text cifrat, decriptarea CBC se restabilește automat după 2 blocuri cifrate consecutive. Deci CBC se poate auto-sincroniza ("self-synchronizing cipher").

Modul Counter (CTR)

Criptare



Decriptare



Parameters:

- k-bit key K.
- n-bit initial counter value IV.

Summary:

- Encrypt n-bit plaintext blocks m_1 , ... , m_t to n-bit ciphertext blocks m_1 , ... , m_t.
- Decrypt to recover plaintext.

Encryption:

$X_1 \leftarrow IV$

For $1 \le j \le t$

$c_j \leftarrow m_j \oplus E_K(X_j)$

$X_1 \leftarrow X_1 + 1$ (for example)

Decryption (same algorithm):

$X_1 \leftarrow IV$

For $1 \le j \le t$

$m_j \leftarrow c_j \oplus E_K(X_j)$

$X_1 \leftarrow X_1 + 1$ (for example)

- Idee: CTR aproximează schema one-time-pad folosind un șir secret pseudoaleator generat de cifrul bloc (în locul unui șir "perfect" aleator). Dacă IV nu se repetă, fiecare text clar este criptat cu un șir secret diferit.

Analiza schemei CTR (1)

Criptarea textelor clare de lungime arbitrară

- Nu este necesară completarea textului clar ("padding"). Dacă ultimul bloc, m_t, are lungimea $u < n$, se folosesc u biți mai semnificativi din $E_K(X_t)$.

Criptare și decriptare

- Pentru a decripta c_i, este necesar același bloc $E_K(X_i)$ ca la criptare: $c_i \oplus E_K(X_i) = m_i \oplus E_K(X_i) \oplus E_K(X_i) = m_i$.
- Blocul IV *nu este secret* și trebuie transmis ca parte a textului cifrat pentru a permite decriptarea (pentru că $X_i = IV + i - 1$).
- Schema CTR folosește același algoritm pentru criptare și decriptare. De asemenea, CTR folosește doar funcția directă, E_K, a cifrului bloc.
- Fiecare bloc de text clar, m_i, trebuie criptat cu un șir pseudoaleator secret diferit, $E_K(X_i)$. Deci valorile contorului, X_i, trebuie să fie distincte pe durata de viață a cheii. Altfel, dacă $c_i = m_i \oplus E_K(X_i)$ și $c_j = m_j \oplus E_K(X_i)$, atunci $c_i \oplus c_j = m_i \oplus m_j$. Cunoscând m_i rezultă $m_j = c_i \oplus c_j \oplus m_i$.

Analiza schemei CTR (2)

Securitatea criptării CTR

- **Teoremă**: Schema CTR îndeplinește IND-CPA dacă și numai dacă:
  (1) cifrul bloc utilizat este o permutare pseudoaleatoare (PRP) și
  (2) valorile contorului sunt distincte pe durata de viață a cheii.
  Cheia trebuie schimbată după ce au fost criptate $b << 2^{n/2}$ blocuri.

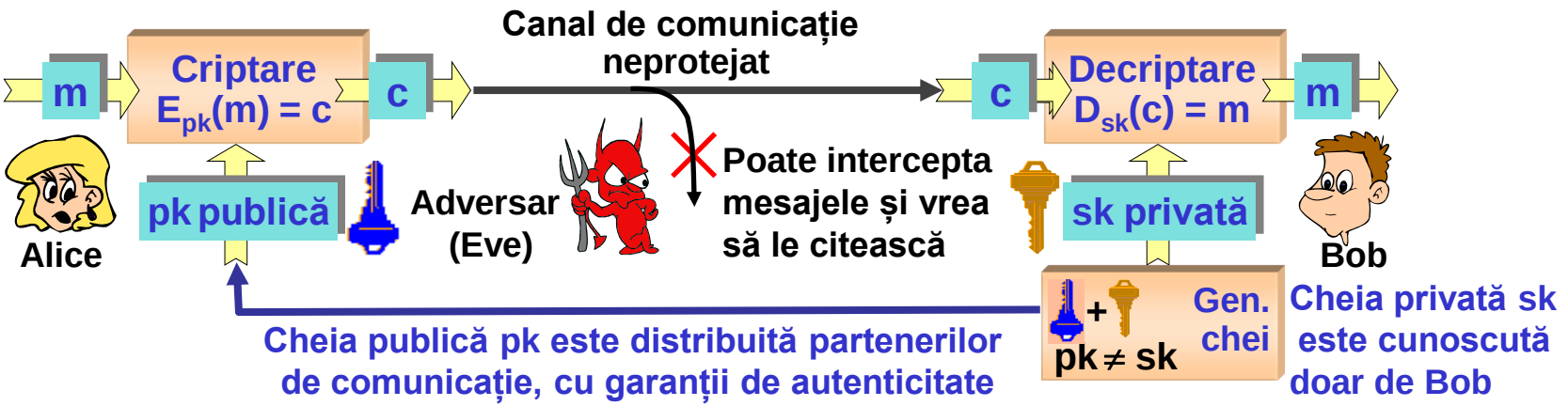
Alte proprietăți ale schemei CTR

- Calcul paralel: Se pot cripta mai multe blocuri în paralel. Se pot decripta mai multe blocuri în paralel.
- Acces aleator la textul cifrat: Se poate decripta separat orice porțiune de text cifrat (nu este nevoie să fie decriptate blocurile precedente).
- Sincronizare: CTR este un cifru sincron (synchronous stream cipher). Dacă se pierd porțiuni de text cifrat, se pierde sincronizarea și textul cifrat rămas este decriptat incorect (CTR nu se auto-sincronizează).

Confidențialitatea datelor folosind criptografie cu cheie publică

Scheme de criptare cu cheie publică

Definiții



Schemă de criptare cu cheie publică (asimetrică)

- O schemă de criptare cu cheie publică, având *mulțimea textelor clare* $\mathcal{M}$, *mulțimea textelor cifrate* $\mathcal{C}$ și *mulțimea cheilor* $\mathcal{K}$, este alcătuită din:
- *Alg. de generare a cheilor*: Intrare: parametru de securitate k. Ieșire: pereche $(sk, pk) \in \mathcal{K}$ aleasă aleator (cheia privată sk, cheia publică pk).
- *Alg. de criptare*: Intrări: $m \in \mathcal{M}$ și pk. Ieșire: text cifrat $c = E_{pk}(m) \in \mathcal{C}$. Familie de funcții inversabile $\mathcal{E} = \{ E_{pk} \mid (sk, pk) \in \mathcal{K}, E_{pk} : \mathcal{M} \to \mathcal{C} \}$.
- *Alg. de decriptare*: Intrări: $c \in \mathcal{C}$ și sk. Ieșire: $m = D_{sk}(c) \in \mathcal{M}$. Familie de funcții $\mathcal{D} = \{ D_{sk} \mid (sk, pk) \in \mathcal{K}, D_{sk} : \mathcal{C} \to \mathcal{M} \}$, $D_{sk} = E_{pk}^{-1}$, $\forall (sk, pk) \in \mathcal{K}$.

Criptarea cu cheie publică

Începutul criptografiei cu cheie publică

- Conceptele de bază au fost introduse în 1976 de Diffie și Hellman în articolul: "New directions in cryptography".
- Componentă fundamentală a criptografiei moderne: criptare, autentificarea datelor (semnătură digitală), distribuția cheilor.

Hellman și Diffie

Cerințe de bază pentru criptarea cu cheie publică

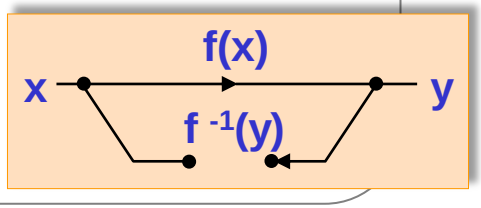
- Pentru oricare $(\mathrm{sk}, \mathrm{pk}) \in \mathcal{K}$, funcția de criptare $\mathrm{E}_{\mathrm{pk}} : \mathcal{M} \to \mathcal{C}$ și funcția de decriptare $\mathrm{D}_{\mathrm{sk}} : \mathcal{C} \to \mathcal{M}$ pot fi calculate eficient, având cheile respective.
- Recuperarea cheii private: Pentru oricare $(\mathrm{sk}, \mathrm{pk}) \in \mathcal{K}$, având cheia publică pk (deci funcția de criptare E_{pk}) nu se poate calcula (eficient) cheia privată sk asociată (deci funcția de decriptare D_{sk}).
- Recuperarea textului clar: Pentru oricare $(\mathrm{sk}, \mathrm{pk}) \in \mathcal{K}$ și $\mathrm{m} \in \mathcal{M}$, nu se poate calcula (eficient) $\mathrm{D}_{\mathrm{sk}}(\mathrm{c})$ fără a avea cheia privată sk.

- Criptarea simetrică: ambele funcții, criptare și decriptare, sunt secrete.
- Criptarea asimetrică: funcția de criptare este publică $\Rightarrow$ altfel de algoritmi.

Funcții cu sens unic

Definiție: Funcție cu sens unic (one-way function)

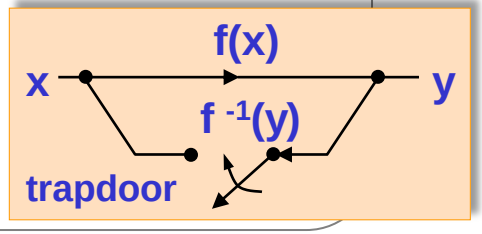
- Concept fundamental în criptografia modernă.
- O funcție $f : \mathcal{X} \to \mathcal{Y}$ se numește *funcție cu sens unic* dacă:
  - Pentru oricare $x \in \mathcal{X}$, se poate calcula eficient $f(x)$.
  - Pentru (aproape) oricare $y \in \mathcal{Y}$, nu se poate calcula (eficient) $x \in \mathcal{X}$ astfel încât $f(x) = y$.



- Pentru criptarea cu cheie publică, ne interesează în mod special *funcțiile bijective (permutările) cu sens unic*. În acest caz, există o funcție inversă, dar nu poate fi calculată (eficient).

Definiție: Permutare cu trapă secretă (trapdoor permutation)

- O funcție bijectivă $f : \mathcal{X} \to \mathcal{Y}$ se numește *permutare cu trapă* dacă:
  - Este o permutare cu sens unic.
  - Cu informații suplimentare (trapa secretă), se poate calcula eficient $x = f^{-1}(y)$ pentru oricare $x \in \mathcal{X}$.



Permutări cu sens unic (1)

Există permutări cu sens unic?

- Din 1976 și pînă în prezent *au fost identificate foarte puține funcții* care (se presupune că) îndeplinesc aceste cerințe și care pot fi utilizate în practică pentru criptografia cu cheie publică (scheme eficiente).

Problema factorizării numerelor întregi

- Fiind dat un număr întreg pozitiv n, găsiți reprezentarea sa cu factori primi: $n = p_1^{e1} \cdot p_2^{e2} \cdot \ldots p_k^{ek}$, unde p_i sunt numere prime distincte și $e_i \geq 1$.

- Varianta uzuală pentru criptografie: Fiind dat un întreg pozitiv $n = pq$, unde p și q sunt numere prime (cu reprezentare binară de lungime egală), găsiți factorii p și q.

- Cei mai eficienți algoritmi cunoscuți au complexitate subexponențială în $k = \log_2(n)$, $\Rightarrow$ problema este considerată necalculabilă.

Recorduri de factorizare		
Lungime modul n	**Data**	**Algoritm**
100 digits - 332 bits	Apr. 1991	QS
110 digits - 365 bits	Apr. 1992	QS
120 digits - 398 bits	Jun. 1993	QS
130 digits - 431 bits	Apr. 1996	G-NFS
155 digits - 512 bits	Aug. 1999	G-NFS
174 digits - 576 bits	Dec. 2003	G-NFS
200 digits - 663 bits	May 2005	G-NFS
232 digits - 768 bits	Dec 2009	G-NFS

Permutări cu sens unic (2)

Problema RSA (Rivest, Shamir Adleman)

- Fiind dat un întreg pozitiv $n = pq$, cu p, q numere prime impare, un întreg pozitiv e astfel încât $\gcd(e, (p-1)(q-1)) = 1$ și un întreg c, găsiți un întreg x astfel încât $x^e \equiv c \pmod{n}$. (gcd = greatest common divisor).

Familia de funcții RSA

$$RSA_{n,e} : Z_n \to Z_n,$$

$$RSA_{n,e}(x) = x^e \bmod n.$$

Dacă p, q, n, e au proprietățile de mai sus, $RSA_{n,e}$ este bijectivă.

Familia de funcții RSA inverse

$$RSA_{n,d} : Z_n \to Z_n,$$

$$RSA_{n,d}(x) = x^d \bmod n.$$

Dacă $ed \equiv 1 \pmod{(p-1)(q-1)}$, atunci $RSA_{n,d} = RSA^{-1}{}_{n,e}$.

- Dacă factorii p și q nu sunt cunoscuți, se presupune că problema RSA este la fel de dificilă ca problema factorizării modulului n, cu complexitate subexponențială în $k = \log_2(n)$.
- Prin urmare, funcția RSA este considerată o permutare cu sens unic. Mai mult, este o permutare cu trapă secretă, informația suplimentară fiind d.

Permutări cu sens unic (3)

Grup ciclic G de ordin q cu generator g: $G = \{ g^0, g^1, g^2, ..., g^{q-1} \}$.

Funcția exponențială discretă

$DExp_{G,g} : [0, q-1] \rightarrow G$,

$DExp_{G,g}(x) = g^x$, este bijectivă.

Funcția logaritm discretă

$DLog_{G,g} : G \rightarrow [0, q-1]$,

$DLog_{G,g}(g^x) = x$.

Problema logaritmului discret (Discrete Logarithm Problem)

- Fiind dat un grup ciclic G de ordin q, un generator g al acestui grup și un element $y \in G$, găsiți numărul întreg $x \in [0, q-1]$ astfel încât $g^x = y$.

- Pentru *anumite grupuri ciclice*, cei mai eficienți algoritmi cunoscuți pentru logaritmul discret au complexitate exponențială în $k = \log_2(q)$.
- În aceste grupuri se poate deci presupune că *exponențiala discretă este o permutare cu sens unic*. Nu se cunoaște o trapă.
- Exemple: Subgrup de ordin prim al grupului multiplicativ Z^*_p, cu p prim. Subgrup de ordin prim al punctelor de pe o curbă eliptică discretă.

  Pentru grupul Z^*_p, există algoritmi cu complexitate subexponențială.

Securitatea criptării cu cheie publică

Modele de securitate

- Adaptăm modelul general de securitate al schemelor de criptare.
- Aspect specific al criptării cu cheie publică: Adversarul cunoaște funcția de criptare, deci poate calcula orice pereche $(m_i, E_{Ke}(m_i))$.
- Modelele de atac slabe devin irelevante. Ne interesează doar atacuri cu text clar ales și cu text cifrat ales.

Cerințe de securitate

- Schemele de criptare cu cheie publică trebuie să îndeplinească cel puțin IND-CPA, însă dorim scheme care îndeplinesc IND-CCA (> IND-CPA).

- O permutare cu sens unic nu este suficientă pentru a obține IND-CPA, deoarece este un algoritm determinist.
- Este necesar un algoritm probabilistic. Exemplu: codare probabilistă a textului clar urmată de o permutare cu sens unic.

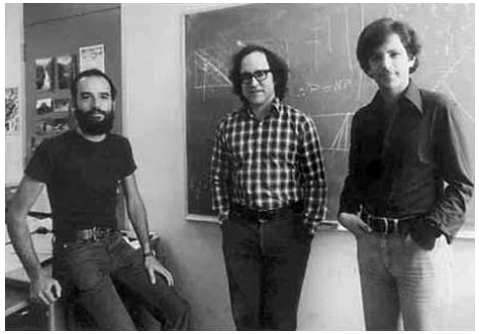
Confidențialitatea datelor folosind criptografie cu cheie publică

Criptarea RSA

Criptosistemul RSA

RSA: Primul criptosistem cu cheie publică

- Introdus în 1977 de Rivest, Shamir și Adleman.
- Primul sistem complet, conform principiilor enunțate de Diffie și Hellman: criptare și autentificare (semnătură digitală) cu cheie publică.

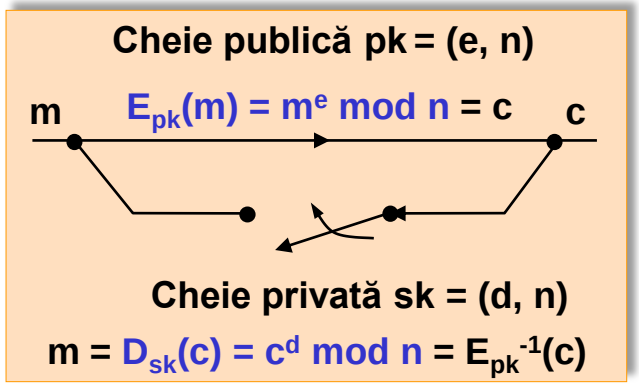


Shamir, Rivest și Adleman (1977)

- Domină primii 30 de ani ai criptografiei cu cheie publică (simplu, eficient).
- În prezent, soluțiile bazate pe problema logaritmului discret devin treptat mai eficiente (în special, criptografia cu curbe eliptice).

Criptare cu cheie publică bazată pe familia de permutări cu trapă RSA

- Criptare: $E_{pk}(m) = RSA_{n,e}(m) = m^e \bmod n.$
- Decriptare: $D_{sk}(c) = RSA_{n,d}(c) = c^d \bmod n.$



Criptarea RSA: Schema de bază

Criptare RSA: Schema de bază (fără securitate semantică)

- Alg. de generare a perechii de chei: Parametru de securitate $k = 2\ell$.
  - Alege aleator și independent numerele prime $p, q \in [2^{\ell-1}, 2^{\ell}-1]$ astfel încât $p \neq q$ și $|pq| = k$ (biți).
  - Calculează $n = pq$ și $\varphi(n) = (p-1)(q-1)$. Alege $e \in [2, \varphi(n)-1]$ aleator, a.î. $\gcd(e, \varphi(n)) = 1$. Calculează $d = e^{-1} \bmod \varphi(n)$.
  - Cheia publică (criptare): $pk = \langle n, e \rangle$.
  - Cheia privată (decriptare): $sk = \langle n, d \rangle$ sau $sk = \langle p, q, d \rangle$.
- Alg. de criptare: $E_{pk} : Z_n \to Z_n$, $E_{pk}(m) = RSA_{n,e}(m) = m^e \bmod n$.
- Alg. de decriptare: $D_{sk} : Z_n \to Z_n$, $D_{sk}(c) = RSA_{n,d}(c) = c^d \bmod n$.

- Verificarea condiției de decriptare corectă:

  $D_{sk}(E_{pk}(m)) = RSA_{n,d}(RSA_{n,e}(m)) = RSA^{-1}{}_{n,e}(RSA_{n,e}(m)) = m.$
- Algoritmul de generare a cheii alege în mod aleator o instanță a familiei de permutări $\{ RSA_{n,e} \}$. Textele clare sunt șiruri binare < k biți.

Securitatea permutării cu trapă RSA (1)

Calculul exponentului secret cunoscând p și q

- Știind p și q, exponentul d se poate calcula eficient ca în algoritmul de generare a cheii: Calculăm $\varphi(n) = (p-1)(q-1)$ și apoi folosim Algoritmul lui Euler Extins pentru a calcula $d = e^{-1} \bmod \varphi(n)$.

Cum putem inversa $RSA_{n,e}$ fără a cunoaște p și q?

- Proprietățile familiei de funcții $RSA_{n,e}$ au fost studiate extensiv.
- Se poate demonstra că problema calculării funcției $\varphi(n)$ fără a cunoaște p și q este la fel de dificilă ca problema factorizării modulului n.

  Nu există deci o metodă mai eficientă pentru a calcula $\varphi(n)$ decât aflarea factorilor p și q urmată de calculul $\varphi(n) = (p-1)(q-1)$.
- De asemenea, nu se cunoaște nici un algoritm eficient pentru a inversa $RSA_{n,e}$ (în orice punct) fără a afla mai întâi factorii p și q.

- Se presupune deci că problema inversării $RSA_{n,e}$ este la fel dificilă ca problema factorizării lui n. Nu a fost găsită însă o demonstrație riguroasă.

Securitatea permutării cu trapă RSA (2)

Principalul atac asupra funcției RSA: factorizarea modulului

- Aplicabil tuturor schemelor criptografice construite folosind RSA (criptare, autentificare). Recompensă maximă: determină cheia privată.
- Poate fi efectuat offline (fără interacțiune cu victima), alocând resurse considerabile (calcul paralel masiv folosind hardware dedicat).
- Complexitatea celor mai eficienți algoritmi de factorizare este sub-exponențială ca funcție de lungimea reprezentării binare a modulului.
  $\Rightarrow$ Atacul este contracarat utilizând un modul n suficient de mare.

Lungimea recomandată a modulului

- Biți de securitate: Măsură a securității unui algoritm criptografic, permite comparații.
- s biți de securitate $\Rightarrow$ complexitatea atacului de referință este 2^s operații.
- Pentru securitate comparabilă cu AES-128, RSA are nevoie de un modul de 3072 biți.

Biți de securitate	Cifru bloc	RSA
80	2TDEA	*k = 1024*
112	3TDEA	*k = 2048*
128	**AES-128**	***k = 3072***
192	AES-192	*k = 7680*
256	AES-256	*k = 15360*

(In)Securitatea schemei RSA de bază

Schma RSA de bază nu este o schemă de criptare sigură

- Algoritmul de criptare din schema de bază este determinist, deci nu îndeplinește IND-CPA/CCA. *Avem nevoie de un algoritm probabilist*.
- RSA are proprietăți care introduc vulnerabilități în aplicații criptografice. Algoritmul de criptare ar trebui să elimine aceste vulnerabilități.

  Ex: Au fost demonstrate atacuri CCA eficiente bazate pe homomorfismul RSA: Dacă $m = m_1 m_2 \bmod n$, atunci $RSA_{n,e}(m) \equiv RSA_{n,e}(m_1)\ RSA_{n,e}(m_2) \pmod n$.

Ipoteza RSA (RSA Assumption)

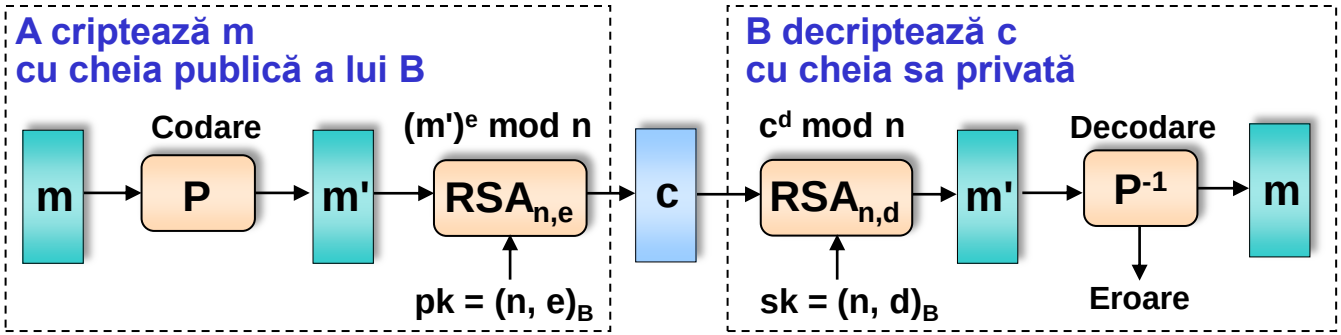
- Securitatea algoritmilor criptografici construiți folosind RSA se bazează pe ipoteza următoare: Probabilitatea ca un adversar cu resurse limitate să determine $x = RSA^{-1}_{n,e}(y)$ este neglijabilă, dacă funcția $RSA_{n,e}$ și $y \in Z_n$ sunt alese aleator și modulul n este suficient de mare.

  $\Rightarrow$ Adversarul nu poate inversa $RSA_{n,e}$ (aleasă ca în schema de bază) într-un punct aleator din întregul său domeniu. $\Rightarrow$ Putem cripta orice mesaj de lungime $< \log_2 n$ biți dacă îl codăm ca element aleator din Z_n.

Criptare probabilistă folosind RSA

Criptare RSA cu codare probabilistă a textului clar

- Construcția uzuală pentru algoritmi de criptare bazați pe RSA.
- Algoritmul de criptare RSA este construit prin *compunerea funcției RSA cu o funcție de codare pseudoaleatoare*, care transformă textul clar într-un element aleator din Z_n. Funcția de codare trebuie să fie inversabilă.



- Alg. de codare probabilist: $P : \{0,1\}^t \rightarrow Z_n$, $t < \log_2 n$ biți.
- Alg. de criptare: $c = RSA_{n,e}(P(m))$.
- Alg. de decriptare: $m = P^{-1}(RSA_{n,d}(c))$.

- Alg. de criptare probabilist (condiție necesară pentru securitate semantică).
- Compatibil cu ipoteza RSA.

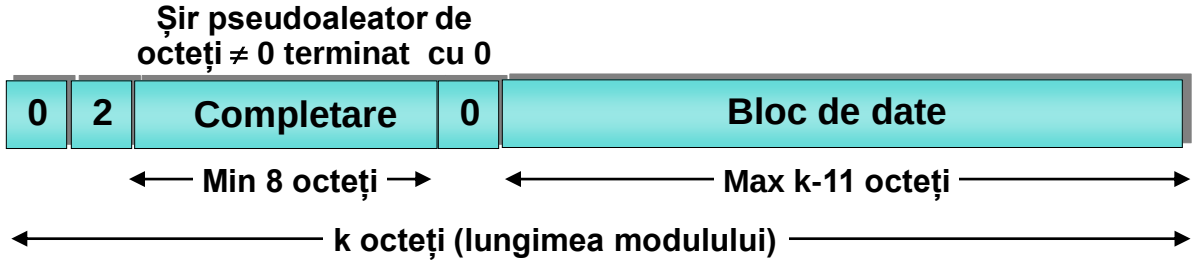
Criptare RSA cu codare PKCS #1 v1.5

Cerințe pentru algoritmii de codare folosiți în criptarea RSA

- Algoritm probabilist compatibil cu ipoteza RSA.
- Schema de criptare RSA obținută trebuie să îndeplinească IND-CCA.

O primă soluție: Criptare RSA cu codare PKCS #1 v1.5 (1993)

- Schema de criptare RSA descrisă în PKCS #1 v1.5 (standard de facto).
- Blocul de date este completat până la k octeți cu un șir pseudoaleator de octeți cu valori diferite de zero terminat cu un octet egal cu zero.

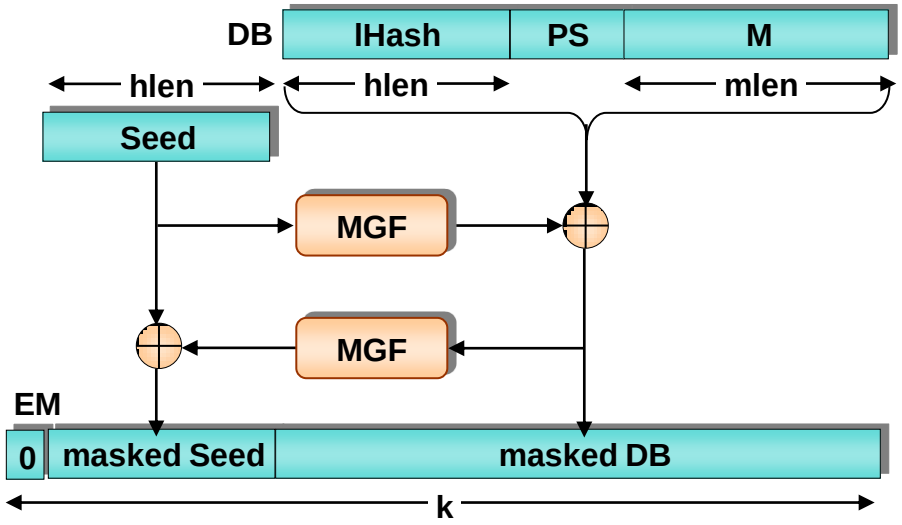


- Se presupune că criptarea RSA cu codare PKCS #1 v1.5 îndeplinește IND-CPA, dar nu există o demonstrație formală a securității sale.

Criptarea RSA-OAEP

Criptare RSA-OAEP (PKCS #1 v2, 2000)

- Schemă de criptare RSA descrisă în PKCS #1 v2, IEEE 1363-2000, etc.
- Folosește o variantă a algoritmului de codare OAEP (Optimal Asymmetric Encryption Padding) propus de Bellare și Rogaway.



MGF (Mask Generation Function): Funcție pseudoaleatoare (construită cu o funcție hash cu ieșire de hlen octeți).

Seed: Șir pseudoaleator secret.

PS (padding string): Șir de completare cu octeți = 0 terminat cu un octet = 1.

lHash: etichetă opțională.

Decodare: Se recuperează Seed din textul codat și apoi M.

- Se poate demonstra că RSA-OAEP îndeplinește IND-CCA, presupunând că ipoteza RSA este validă (și MGF este funcție pseudoaleatoare).

  Dacă un adversar (algoritm) A câștigă jocul IND-CCA cu un anumit efort și probabilitate, atunci există un algoritm care poate inversa RSA folosind A cu probabilitate și efort comparabile.