Securitatea sistemelor informatice

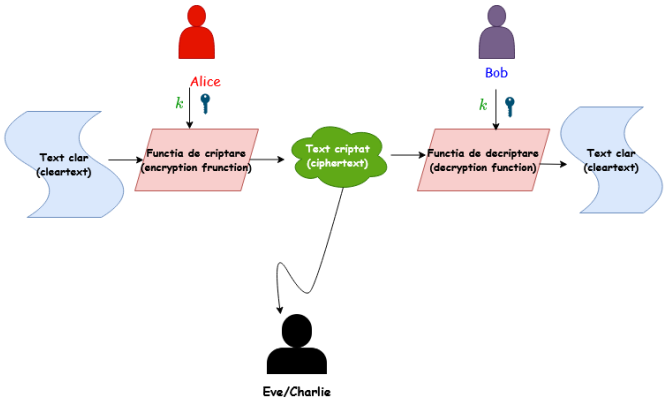
Sisteme istorice de criptare. Securitate perfecta.

Curs 2

Anul III, Informatica
2022-2023

Adela Georgescu
Facultatea de Matematica – Informatica
Universitatea Bucuresti

Criptarea simetrica (cu cheie secreta)



Criptarea simetrica (cu cheie secreta)

Definitie

Un sistem de criptare simetric definit peste $(\mathcal{K}, \mathcal{M}, \mathcal{C})$*, cu:*

- $\mathcal{K}$ = *spaţiul cheilor*
- $\mathcal{M}$ = *spaţiul textelor clare (mesaje)*
- $\mathcal{C}$ = *spaţiul textelor criptate*

este un triplet $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$*, unde:*

1. Gen: *este algoritmul probabilistic de generare a cheilor care întoarce o cheie* k *conform unei distribuţii*
2. $\mathrm{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$
3. $\mathrm{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

a.î. $\forall m \in \mathcal{M}, k \in \mathcal{K} : Dec_k(Enc_k(m)) = m$.

Criptarea simetrica (cu cheie secreta)

- K = variabilă aleatoare ce reprezintă valoarea cheii returnată de Gen
- $Pr[K = k]$ = probabilitatea cheii generate de Gen de a lua valoarea k, $\forall k \in \mathcal{K}$
- M = variabilă aleatoare ce reprezintă mesajul care se criptează
- $Pr[M = m]$ = probabilitatea ca mesajul de criptat să ia valoarea $m \in \mathcal{M}$
- expl. de probabilitate de distributie peste $\mathcal{M}$: $\mathcal{M} = \{\text{atacati azi}, \text{nu atacati}\}$ cu $Pr[M = \text{atacati azi}] = 0.2$ si $Pr[M = \text{nu atacati}] = 0.8$

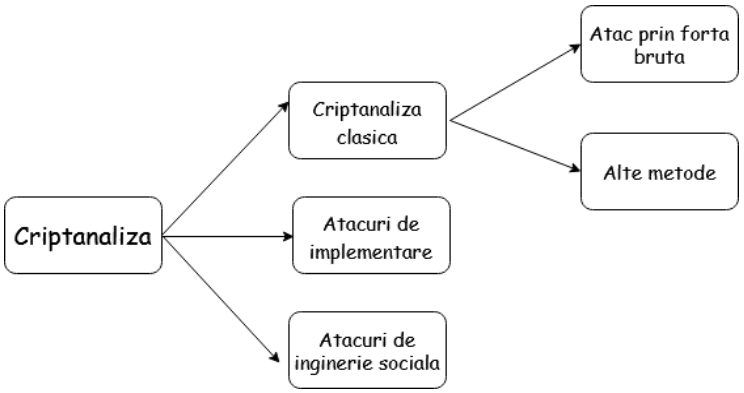
Terminologie

- Mesajul în forma originară se numește text clar;
- Expeditorul rescrie mesajul folosind un sistem de criptare, adică îl criptează și obține un text criptat;
- Destinatarul îl decriptează cunoscând metoda folosită pentru criptare;
- Procesul de determinare a cheii aferente unui sistem de criptare, cunoscând doar textul criptat (eventual și alte informații auxiliare) se numește criptanaliză;
- Decriptarea și criptanaliza au același scop: găsirea textului clar; diferența constă în faptul că la criptanaliză nu se cunoaște cheia de decriptare.

Scenarii de atac

- Atac cu text criptat (ciphertext-only attack): Atacatorul știe doar *textul criptat* - poate încerca un atac prin forță brută prin care se parcurg toate cheile până se găsește cea corectă;
- Atac cu text clar (known-plaintext attack): Atacatorul cunoaște una sau mai multe perechi *(text clar, text criptat)*;
- Atac cu text clar ales (chosen-plaintext attack): Atacatorul poate obține criptarea unor texte clare alese de el;
- Atac cu text criptat ales (chosen-ciphertext attack): Atacatorul are posibilitatea să obțină decriptarea unor texte criptate alese de el.

Criptanaliza



Sisteme de criptare istorice

Cifruri de permutare / transpoziție

Definitie

Un cifru de permutare presupune rearanjarea literelor în textul clar pentru a obține textul criptat.

Cifruri de permutare / transpoziție

- sistemul Rail Fence >>> curs
- cifruri generale de transpoziție >>> laborator

Rail Fence

	→			
	M	A	R	T
↓	E	J	I	A
	S	C	P	T

Text clar: mesaj criptat
Cheia: $k = 3$
Text criptat: MARTEJIASCPT

Cifruri de substitiuție monoalfabetice

- cifrul lui Cezar
- substituție simplă
- sistemul Cavalerilor de Malta

Cifrul lui Cezar

a	b	c	d	e	f	g	h	i	j	k	l	m
D	E	F	G	H	I	J	K	L	M	N	O	P

n	o	p	q	r	s	t	u	v	w	x	y	z
Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Text clar: mesaj criptat
Text criptat: PHVDM FULSWDW

Cifrul lui Cezar

- $\mathcal{K} = \{0, 1, \ldots, 25\}$
- $\mathcal{M} = \{a, b, \ldots, z\}^*$
- $\mathcal{C} = \{A, B, \ldots, Z\}^*$
- $\mathrm{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$

$$Enc_k(m) = m + k \pmod{26}$$

- $\mathrm{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

Generalizare - Shift cipher

- $\mathcal{K} = \{0, 1, \ldots, 25\}$
- $\mathcal{M} = \{a, b, \ldots, z\}^*$
- $\mathcal{C} = \{A, B, \ldots, Z\}^*$
- $\mathrm{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$

$$Enc_k(m) = m + k \pmod{26}$$

- $\mathrm{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

$$Dec_k(c) = c - k \pmod{26}$$

Criptanaliză - Atac prin forță brută

- $|\mathcal{K}| = 26$
- atac prin forță brută (căutare exhaustivă): încercarea, pe rând, a tuturor cheilor posibile până când se obține un text clar cu sens

Principiul cheilor suficiente: O schemă sigură de criptare trebuie să aibă un spațiu al cheilor suficient de mare a.î. să nu fie vulnerabilă la căutarea exhaustivă.

Substituția simplă

a	b	c	d	e	f	g	h	i	j	k	l	m
F	I	L	O	R	U	X	A	D	G	J	M	P

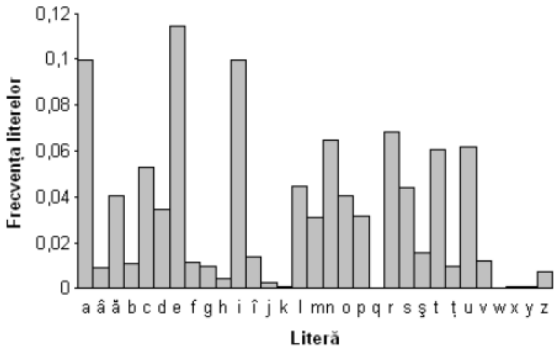
n	o	p	q	r	s	t	u	v	w	x	y	z
S	V	Y	B	E	H	K	N	Q	T	W	Z	C

Text clar: mesaj criptat
Text criptat: PRHFG LEDYKFK

Criptanaliză - Analiza de frecvență

- $|\mathcal{K}| = 26!$
- atacul prin forță brută devine mai dificil
- analiza de frecvență: determinare corespondenței între alfabetul clar și alfabetul criptat pe baza frecvenței de apariție a literelor în text, cunoscând distribuția literelor în limba textului clar
  - se cunoaște limba textului clar
  - lungimea textului permite analiza de frecvență

Criptanaliză - Analiza de frecvență



[Wikipedia]

Cifruri de substitiuție polialfabetice / poligrafice

- sistemul Playfair
- sistemul Hill
- sistemul Vigenére

Cifrul Vigenére

Text clar:	c	u	r	s	c	r	i	p	t	o	g	r	a	f
Cheie:	c	h	e	i	e	c	h	e	i	e	c	h	e	i
Text criptat:	E	B	V	A	G	T	P	T	B	S	I	Y	E	N

Cifrul Vigenére

- $\mathcal{K}$
- $\mathcal{M}$
- $\mathcal{C}$
- $\text{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$

- $\text{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

Cifrul Vigenére

- $\mathcal{K} = \{0, 1, \ldots, 25\}$
- $\mathcal{M} = \{0, 1, \ldots, 25\}^*$
- $\mathcal{C} = \{0, 1, \ldots, 25\}^*$
- $\text{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$
  Textul clar: $m_0 m_1 \ldots m_{n-1}$
  Cheia: $k_0 k_1 \ldots k_x$

$$Enc_{k_j}(m_i) = m_i + k_j \pmod{26}$$

  unde $j = i \bmod x$
- $\text{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

$$Dec_{k_j}(c_i) = c_i - k_j \pmod{26}$$

Vigenére – criptanaliza

- Pentru o cheie de **n** caractere, spatiul cheilor $|\mathcal{K}| = 26^n$
- Un text criptat c = $c_1 c_2$… poate fi impartit in **s** parti in care fiecare parte a fost criptata cu aceeasi litera din alfabet.

  pentru j ∈ $\{1,2,..,s\}$

$$c_j = m_j + k_j$$
$$c_{j+s} = m_{j+s} + k_j$$
$$c_{j+2s} = m_{j+2s} + k_j$$

- Daca se cunoaste lungimea *p* a cheii, problema se reduce la criptanaliza a *p* texte criptate cu shift cipher
- Putem face analiza de frecventa pe fiecare sir separat

Criptografia moderna

- Se bazeaza pe 3 principii moderne

Principiul 1 – orice problema criptografica necesita o definitie clasica si riguroasa – *discutat in Curs 1*

(scheme construite dupa acest principiu sunt folosite azi in TLS, SSH, IPSec)

Principiul 2 - securitatea primitivelor criptografice se bazeaza pe prezumptii clare de securitate

Principiul 3 – orice constructie criptografica trebuie sa fie insotita de o demonstratie de securitate conform principiilor anterioare

Principiul 2 – prezumptii (ipoteze) de securitate

- majoritatea constructiilor criptografice moderne **nu pot fi** demonstrate ca fiind **sigure neconditionat**
- ipotezele de securitate trebuie sa fie explicite
  - ❖ permit cercetatorilor sa valideze aceste ipoteze
  - ❖ permit comparatia intre doua scheme bazate pe ipoteze diferite de securitate
  - ❖ implicatii practice in cazul unor erori aparute in cadrul ipotezei de securitate
  - ❖ necesare pentru demonstratiile de securitate

Principiul 3 – demonstratii de securitate

- ofera o demonstratie riguroasa a faptului ca o constructie satisface definitia data in ipoteze de securitate clare
- fara o demonstratie riguroasa, intuitia ca o schema este corecta poate avea consecinte dezastruoase
- majoritatea demonstratiilor folosesc o abordare reductionista

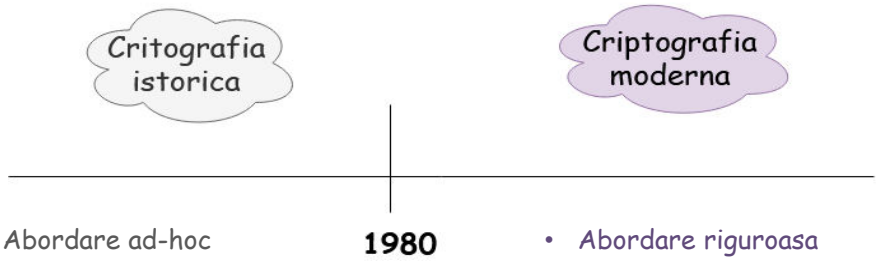
Teorema *Constructia Y este sigura conform definitiei daca prezumptia X este adevarata.*

- demonstratia va arata cum un adversar care sparge schema Y poate incalca prezumptia X.

Exemplu de ipoteza de securitate (problema dificila)

- Factorizarea numerelor mari
  - Se da un numar compus N si se cere descompunerea lui in factori primi.
  - Expl: 85 = 17 * 5
  - Astazi nu se cunoaste nici un algoritm care sa factorizeze un numar de 400 cifre intr-un timp practic
- Totusi:
  1. Un algoritm mai rapid ar putea exista
  2. Un calculator cuantic factorizeaza rapid (inca nu a fost construit dar se fac eforturi in acest sens)
  3. Criptografia "post-cuantica" este in plina dezvoltare – competitia de standardizare post-cuantica NIST, criptografia bazata pe latici etc.

Retineti



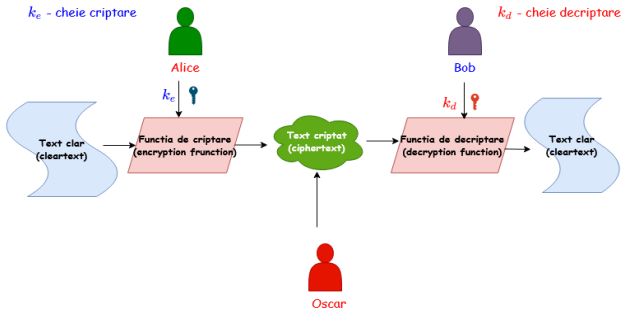
Securitate perfecta (neconditionata)

- Sistemele de criptare istorice (substitutie, permutare, Vigenere, Playfair etc.) pot fi sparte cu un efort computational foarte mic
- In cursul de azi - scheme perfect sigure care rezista in fata unui adversar cu putere computationala nelimitata
- Insa ... limitarile sunt inevitabile

Generare aleatorism

- In exemplul urmator cheile sunt generate aleator, deci avem nevoie de o sursa buna de aleatorism
- Trebuiesc folosite generatoare de numere aleatoare create in scop criptografic iar nu unele generale care nu sunt destinate aplicatiilor criptografice
- Expl.: functia **rand()** din libraria stdlib.h din *C* nu este sigura din punct de vedere criptografic
- Sunt necesare doua proprietati pentru criptografie:
  1. **Generala** - poseda proprietati statistice bune (nu poate fi reprodus)
  2. **Specifica** – este impredictibil: fiind dati n biti de iesire, este imposibil de calculat (infezabil computational) care sunt urmatorii biti

Securitate perfecta (Shannon 1949)



Ipoteza: Oscar cunoaste distributia peste M

Securitate perfecta:
1. daca Oscar afla textul criptat nu are nici un fel de informatie in plus decat daca nu l-ar fi aflat (nu schimba ceea ce stie el despre distributia peste M).

(textul criptat nu ofera nici un fel de informatie despre textul clar)

2. Oscar nu poate ghici care din doua posibile mesaje clare a fost criptat, doar vazand textul criptat

Securitate perfecta (Shannon 1949)

Definiție

O schemă de criptare peste un spațiu al mesajelor $\mathcal{M}$ este perfect sigură dacă pentru orice probabilitate de distribuție peste $\mathcal{M}$, pentru orice mesaj $m \in \mathcal{M}$ și orice text criptat c pentru care $Pr[C = c] > 0$, următoarea egalitate este îndeplinită:

$$Pr[M = m|C = c] = Pr[M = m]$$

- $Pr[M = m]$ - probabilitatea *a priori* ca Alice să aleagă mesajul m;
- $Pr[M = m|C = c]$ - probabilitatea *a posteriori* ca Alice să aleagă mesajul m, chiar dacă textul criptat c a fost văzut ;
- securitate perfectă - dacă Oscar afla textul criptat nu are nici un fel de informație în plus decât dacă nu l-ar fi aflat.

Securitate perfecta (Shannon 1949)

Definitie

O schemă de criptare (Enc, Dec) este perfect sigură dacă pentru orice mesaje $m_0, m_1 \in \mathcal{M}$ cu $|m_0| = |m_1|$ și $\forall c \in \mathcal{C}$ următoarea egalitate este îndeplinită:

$$Pr[Enc_k(m_0) = c] = Pr[Enc_k(m_1) = c]$$

unde $k \in \mathcal{K}$ este o cheie aleasă uniform.

- fiind dat un text criptat, este imposibil de ghicit dacă textul clar este m_0 sau m_1
- cel mai puternic adversar nu poate deduce nimic despre textul clar dat fiind textul criptat

One Time Pad (OTP)

- Patentat in 1917 de Vernam (mai poartă denumirea de Cifrul Vernam)
- Algoritmul:
  1. Fie $l > 0$ iar $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^l$
  2. Cheia k se alege cu distribuție uniformă din spațiul cheilor $\mathcal{K}$
  3. **Enc**: dată o cheie $k \in \{0,1\}^l$ și un mesaj $m \in \{0,1\}^l$, întoarce $c = k \oplus m$.
  4. **Dec**: dată o cheie $k \in \{0,1\}^l$ și un mesaj criptat $c \in \{0,1\}^l$, întoarce $m = k \oplus c$.

One Time Pad (OTP)

mesaj clar:	0	1	1	0	0	1	1	1	1	$\oplus$
cheie:	1	0	1	1	0	0	1	1	0	
text criptat:	1	1	0	1	0	1	0	0	1	

- avantaj - criptare și decriptare rapide
- dezavantaj - cheia foarte lungă (la fel de lungă precum textul clar)
- Este OTP sigur?

One Time Pad (OTP)

mesaj clar:	0	1	1	0	0	1	1	1	1	$\oplus$
cheie:	1	0	1	1	0	0	1	1	0	
text criptat:	1	1	0	1	0	1	0	0	1	
mesaj clar:	1	1	0	0	0	0	1	1	0	$\oplus$
cheie:	0	0	0	1	0	1	1	1	1	
text criptat:	1	1	0	1	0	1	0	0	1	

- Același text criptat poate să provină din orice text clar cu o cheie potrivită
- Dacă adversarul nu știe decât textul criptat, atunci nu știe nimic despre textul clar!

Securitatea perfecta

- Securitatea perfecta nu este imposibila dar..
  - cheia trebuie sa fie la fel de lunga precum mesajul
  - inconveniente practice (stocare, transmisie)
  - cheia trebuie sa fie folosita o singura data – one time pad
- Exercitiu: Ce se intampla daca folosim o aceeasi cheie de doua ori cu sistemul OTP ?

OTP

- Daca un adversar obtine

$$C = M \oplus K \text{ si } C' = M' \oplus K$$

- atunci el poate calcula

$$C \oplus C' = (M \oplus K) \oplus (M' \oplus K) = (M \oplus M')$$

ceea ce invalideaza proprietatea de securitate perfecta

Limitarile securitatii perfecte

Teoremă

Fie o schemă (Enc, Dec) de criptare perfect sigură peste un spatiu al mesajelor $\mathcal{M}$ și un spațiu al cheilor $\mathcal{K}$. Atunci $|\mathcal{K}| \geq |\mathcal{M}|$.

Sau altfel spus

Limitarile securitatii perfecte

Teoremă

Fie o schemă (Enc, Dec) de criptare perfect sigură peste un spatiu al mesajelor $\mathcal{M}$ și un spațiu al cheilor $\mathcal{K}$. Atunci $|\mathcal{K}| \geq |\mathcal{M}|$.

Sau altfel spus

Teoremă

Nu există nici o schemă de criptare (Enc, Dec) perfect sigură în care mesajele au lungimea n biți iar cheile au lungimea (cel mult) $n-1$ biți.

Securitate perfectă

- Primul curs: Sisteme de criptare istorice (substituție, Vigenere etc.) care pot fi sparte cu efort computațional foarte mic
- Cursul de azi: Scheme perfect sigure care rezistă în fața unui adversar cu putere computațională nelimitată
- Insă...limitările sunt inevitabile

Securitate perfectă (Shannon 1949)

Definiție

O schemă de criptare peste un spațiu al mesajelor $\mathcal{M}$ este perfect sigură dacă pentru orice probabilitate de distribuție peste $\mathcal{M}$, pentru orice mesaj $m \in \mathcal{M}$ și orice text criptat c pentru care $Pr[C = c] > 0$, următoarea egalitate este îndeplinită:

$$Pr[M = m|C = c] = Pr[M = m]$$

- $Pr[M = m]$ - probabilitatea *a priori* ca Alice să aleagă mesajul m;
- $Pr[M = m|C = c]$ - probabilitatea *a posteriori* ca Alice să aleagă mesajul m, chiar dacă textul criptat c a fost văzut ;
- securitate perfectă - dacă Oscar afla textul criptat nu are nici un fel de informație în plus decât dacă nu l-ar fi aflat.

Securitate perfectă (Shannon 1949)

Definiție echivalentă

O schemă de criptare (Enc, Dec) este perfect sigură dacă pentru orice mesaje $m_0, m_1 \in \mathcal{M}$ cu $|m_0| = |m_1|$ și $\forall c \in \mathcal{C}$ următoarea egalitate este îndeplinită:

$$Pr[M = m_0|C = c] = Pr[M = m_1|C = c]$$

unde $k \in \mathcal{K}$ este o cheie aleasă uniform.

- fiind dat un text criptat, este imposibil de ghicit dacă textul clar este m_0 sau m_1
- cel mai puternic adversar nu poate deduce nimic despre textul clar dat fiind textul criptat

Un exemplu de cifru sigur - One Time Pad (OTP)

- Patentat in 1917 de Vernam (mai poartă denumirea de Cifrul Vernam)
- Algoritmul:
  1. Fie $l > 0$ iar $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^l$
  2. Cheia k se alege cu distribuție uniformă din spațiul cheilor $\mathcal{K}$
  3. **Enc**: dată o cheie $k \in \{0,1\}^l$ și un mesaj $m \in \{0,1\}^l$, întoarce $c = k \oplus m$.
  4. **Dec**: dată o cheie $k \in \{0,1\}^l$ și un mesaj criptat $c \in \{0,1\}^l$, întoarce $m = k \oplus c$.

Un exemplu de cifru sigur - One Time Pad (OTP)

mesaj:	0	1	1	0	0	1	1	1	$\oplus$
cheie:	1	0	1	1	0	0	1	1	0
text criptat:	1	1	0	1	0	1	0	0	1

- avantaj - criptare și decriptare rapide
- dezavantaj - cheia foarte lungă (la fel de lungă precum textul clar)
- Este OTP sigur?

Teoremă

Schema de criptare OTP este perfect sigură.

- securitatea perfecta nu este imposibilă dar...
- cheia trebuie să fie la fel de lungă precum mesajul
- incoveniente practice (stocare, transmitere)
- cheia trebuie să fie folosită o singură dată - one time pad - de ce?

Exercițiu Ce se întâmplă dacă folosim o aceeași cheie de două ori cu sistemul OTP ?

Limitările securității perfecte - optimalitate OTP

Teoremă

Fie o schemă (Enc, Dec) de criptare perfect sigură peste un spatiu al mesajelor $\mathcal{M}$ și un spațiu al cheilor $\mathcal{K}$. Atunci $|\mathcal{K}| \geq |\mathcal{M}|$.

Demonstrație

Intuitie:

- *Pentru orice text criptat, se încearcă decriptarea lui cu toate cheile posibile din $\mathcal{K}$ și se obține o listă de cel mult $|\mathcal{K}|$ elemente*
- *Dacă $|\mathcal{K}| < |\mathcal{M}|$ unele mesaje nu sunt pe listă - contradicție cu securitatea perfectă (vezi definiția)*

Securitate perfectă vs. Criptografie computațională

- Am vazut scheme de criptare care pot fi demonstrate ca fiind sigure în prezența unui adversar cu putere computațională nelimitată;
- Se mai numesc și informational-teoretic sigure;
- Adversarul nu are suficientă informație pentru a efectua un atac;
- Majoritatea construcțiilor criptografice moderne $\rightarrow$ securitate computațională;
- Schemele moderne *pot fi sparte* dacă un atacator are la dispoziție suficient spațiu și putere de calcul.

Securitate perfectă vs. Criptografie computațională

- Securitatea computațională mai slabă decât securitatea informațional-teoretică;
- Prima se bazează pe prezumpții de securitate; a doua este necondiționată;
- Întrebare: de ce renunțăm la securitatea perfectă?
- Raspuns: datorită limitărilor practice!
- Preferăm un compromis de securitate pentru a obține construcții practice.

Securitate computațională

- Ideea de bază: principiul 1 al lui Kerckhoffs

  Un cifru trebuie să fie practic, dacă nu matematic, indescifrabil.

- Sunt de interes mai mare schemele care practic nu pot fi sparte deși nu beneficiază de securitate perfectă;
  1. Adversari **limitați computațional/eficienti/timp polinomial**
     Exemplu: Un atacator care realizeaza un atac prin forta bruta peste spatiul cheilor și testeaza o cheie/ciclu de ceas
     - calculator desktop - se pot testa aprox. 2^{57} chei/an
     - supercalculator - se pot testa aprox. 2^{80} chei/an
     - supercalculator, varsta universului - 2^{112} chei

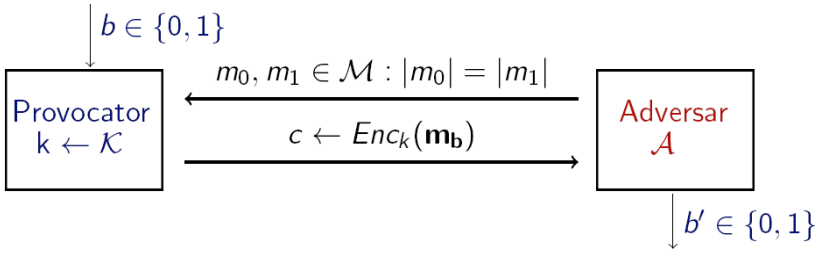
Securitate computațională

2. Adversarii pot efectua un atac cu succes cu o **probabilitate foarte mică**;
Exemplu: un adversar află textul clar cu probabilitate 2^{-60} într-un an
   - sunt șanse mai mari ca Alice și Bob să fie loviți de fulger în aceeași perioadă de timp
   - un eveniment cu prob. $2^{60}/sec.$ se produce o dată la un miliard de ani

Indistinctibilitate perfectă

- Pentru securitatea perfectă am dat două definitii echivalente, a doua sublinia ideea de indistinctibilitate: adversarul nu poate distinge între criptările a două mesaje diferite
- Vom defini indistinctibilitatea pe baza unui experiment $Priv^{eav}_{\mathcal{A},\pi}$ unde $\pi = (Enc, Dec)$ este schema de criptare
- Personaje participante: **adversarul** $\mathcal{A}$ care încearcă să spargă schema și un **provocator (challenger)**.
- Trebuie să definim capabilitățile adversarului: el poate vedea **un singur text criptat cu o anume cheie**, fiind un adversar *pasiv* care poate rula atacuri în timp polinomial, si nu are nici o alta interactiune cu Alice sau Bob

Experimentul $Priv^{eav}_{\mathcal{A},\pi}$



- Output-ul experimentului este 1 dacă $b' = b$ și 0 altfel. Dacă $Priv^{eav}_{\mathcal{A},\pi} = 1$, spunem că $\mathcal{A}$ a efectuat experimentul cu succes.
- Schema π este *perfect indistinctibilă* dacă

$$Pr[Priv^{eav}_{\mathcal{A},\pi}(n) = 1] = \frac{1}{2}$$

- Reamintim ca indistinctibilitatea perfectă este doar o definitie alternativa pentru securitatea perfectă

Securitate computațională asimptotică

- parametru de securitate n atât pentru schema de criptare cât și pentru părțile oneste și adversar
  - poate fi vazut ca lungimea cheii
  - timpul în care rulează adversarul și probabilitatea lui de succes sunt funcții de n
  - este cunoscut adversarului
  - permite utilizatorului să își aleagă nivelul de securitate dorit - este fixat la momentul inițializării schemei de criptare

Securitate computațională concretă

O schemă de criptare este (t, ϵ)-indistinctibila dacă orice adversar care rulează în timp cel mult t

$$Pr[Priv^{eav}_{\mathcal{A},\pi} = 1] \leq \frac{1}{2} + \epsilon$$

- probabilitatea de succes a adversarului $\leq \epsilon$
- adversarul ruleaza in timp $\leq t$
- dezavantaj: am dori sa avem scheme in care utilizatorul isi poate ajusta nivelul de securitate dorit

Securitate computațională asimptotică

- Se impune o nouă modalitate de a defini securitatea:

Definiție

O schemă de criptare este indistinctibila dacă pentru orice adversar PPT, exista o functie neglijabila ϵ așa încat

$$Pr[Priv^{eav}_{\mathcal{A},\pi}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$$

Neglijabil și ne-neglijabil

- în practică: ϵ este scalar și
  - ϵ ne-neglijabil dacă $\epsilon \geq 1/2^{30}$
  - ϵ neglijabil dacă $\epsilon \leq 1/2^{80}$
- în teorie: ϵ este funcție $\epsilon : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ și $p(n)$ este o funcție polinomială în n (ex.: $p(n) = n^d$, d constantă)
  - ϵ ne-neglijabiă în n dacă $\exists p(n) : \epsilon(n) > 1/p(n)$
  - ϵ neglijabilă în n dacă $\forall p(n), \exists n_d$ a.î. $\forall n \geq n_d : \epsilon(n) \leq 1/p(n)$

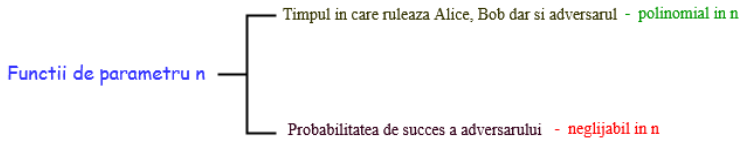
Valori concrete pentru n

Presupunem ca pentru o schema de criptare concreta, un adversar care ruleaza in timp n^3 minute reuseste sa sparga schema cu probabilitate $2^{40} * 2^{-n}$.

- Ce valori alegem pentru n la implementare?
  - pentru $n \leq 40$, atunci un adversar care rulează în 40^3 minute (adica 6 saptamani) sparge schema cu probabilitate 1
  - pentru $n = 50$, atunci un adversar care rulează în 50^3 minute (adica aproximativ 3 luni) sparge schema cu probabilitate aprox. $1/1000$ (ar putea sa nu fire acceptabil)
  - pentru $n = 500$, atunci un adversar care rulează în 200 de ani sparge schema cu probabilitate aprox. 2^{-500} (acceptabil)

Important de reținut!

- Parametrul de securitate n este public cunoscut si parte din schema
- Input-urile pentru toti algoritmii, inclusiv adversarul, sunt polinomiale in n
- Tipic, n este lungimea cheii secrete (de ex. n = 128, 256 etc.)



Criptarea simetrică - redefinită

Definiție

Un sistem de criptare simetric definit peste $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, cu:

- $\mathcal{K}$ = *spațiul cheilor*
- $\mathcal{M}$ = *spațiul textelor clare (mesaje)*
- $\mathcal{C}$ = *spațiul textelor criptate*

este un triplet $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$, *unde:*

1. $\mathrm{Gen}(1^n)$: *este algoritmul probabilistic de generare a cheilor care întoarce o cheie k conform unei distribuții*
2. Enc: *primește o cheie k și un mesaj* $m \in \{0,1\}^*$ *și întoarce* $c \leftarrow Enc_k(m)$
3. Dec: *primește cheia k și textul criptat și întoarce m sau "eroare".*

Securitatea Sistemelor Informatice

- Curs 3.2 -
Aleatorism

Adela Georgescu

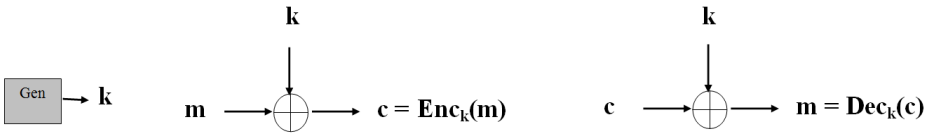
Facultatea de Matematică și Informatică
Universitatea din București
Anul universitar 2022-2023, semestrul I

Aleatorism

- Am definit ce înseamnă pentru o schemă de criptare să fie sigură (noțiunea de indistinctibilitate, curs 3), vrem să vedem o construcție
- In cadrul securității computaționale putem avea
  - chei mai scurte pentru mesaje mai lungi
  - refolosirea cheilor pentru mai multe mesaje

Aleatorism

- Incercăm criptare în stil OTP: cheia va masca mesajul dar
  - masca nu va fi doar cheia ci masca $= f(\text{cheie})$ unde f este o functie de extindere a cheii
  - pentru securitate perfecta masca trebuie sa fie perfect aleatoare
  - pentru securitate computațională, este suficient ca masca sa para aleatoare pentru un adversar PPT chiar daca nu este
- Vom avea avea nevoie întâi să definim noțiunea de *generatoare de numere pseudoaleatoare* ca element important de construcție pentru schemele de criptare simetrice



Pseudoaleatorismul

- Un șir pseudoaleator "arată" similar unui șir uniform aleator din punct de vedere al oricărui algoritm **polinomial**;
- Altfel spus: un algoritm **polinomial** nu poate face diferența între o secvență **perfect aleatoare** și una **pseudoaleatore** (decât cu probabilitate neglijabilă);
- Sau: o distribuție a secvențelor de lungime l este **pseudoaleatoare** dacă este **nedistinctibilă** de distribuția uniformă a secvențelor de lungime l;
- Mai exact: nici un algoritm polinomial nu poate spune dacă o secvență de lungime l este eșantionarea unei distribuții pseudoaleatoare sau este o secvență total aleatoare de lungime l.

Pseudoaleatorismul

- În analogie cu ce știm deja:
  - **pseudoaleatorismul** este o relaxare a **aleatorismului perfect**

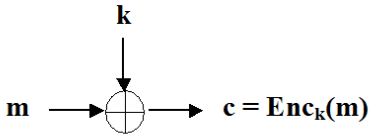
    asa cum
  - **securitatea computațională** este o relaxare a **securității perfecte**

Sistem de criptare

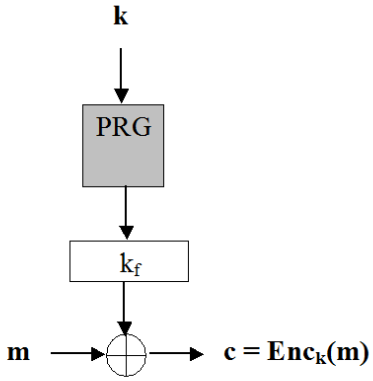
- Revenind la criptare ...
- ... aceasta presupune 2 faze:
  - **Faza 1**: se generează o secvență pseudoaleatoare de biți, folosind un generator de numere pseudoaleatoare (PRG)
  - **Faza 2**: secvența obținută se XOR-ează cu mesajul clar

Sistem de criptare bazat generator de numere aleatoare

OTP (One Time Pad)



Sistem de cripare



PRG

- Ramâne să definim noțiunea de generator de numere aleatoare sau PRG *(PseudoRandom Generator)*;
- Acesta este un algoritm **determinist** care primește o "sămânță" relativ scurtă s (*seed*) și generează o secvență *pseudoaleatoare* de biți;
- Notăm $|s| = n$, $|PRG(s)| = l(n)$
- PRG prezintă interes dacă:

$$l(n) \geq n$$

(altfel NU "generează aleatorism")

PRG

Definiție

Fie $l(\cdot)$ un polinom și G un algoritm polinomial determinist a.î. $\forall n \in \{0,1\}^n$, G generează o secvență de lungime $l(n)$.
G se numește generator de numere pseudoaleatoare (PRG) dacă se satisfac 2 proprietăți:

1. *Expansiune: $\forall n, l(n) \geq n$*
2. *Pseudoaleatorism: $\forall$ algoritm PPT $\mathcal{D}$, $\exists$ o funcție neglijabilă negl a.î.:*

$$|Pr[D(r) = 1] - Pr[D(G(s)) = 1]| \leq negl(n)$$

unde $r \leftarrow^R \{0,1\}^{l(n)}, s \leftarrow^R \{0,1\}^n$

$l(n)$ se numește factorul de expansiune al lui G

Notații

- $\mathcal{D}$ = *Distinguisher*
- PPT = Probabilistic Polynomial Time
- $x \leftarrow^R X$ = x este ales uniform aleator din X
- $negl(n)$ = o funcție neglijabilă în (parametrul de securitate) n

În plus:

- Vom nota $\mathcal{A}$ un adversar (Oscar / Eve), care (în general) are putere polinomială de calcul

Exemplu

- Consideram următorul PRG: $G(s) = s || \oplus_{i=1}^{n} s_i$
- factorul de expansiune $l(n) = n + 1$
- Consideram algoritmul D astfel: $D(w) = 1$ dacă și numai dacă ultimul bit al lui w este egal cu xor-ul tuturor biților precendenți
- Se verifica usor ca $Pr[D(G(s)) = 1] = 1$
- Daca r este uniform, atunci bitul final al lui r este uniform si deci $Pr[D(r) = 1] = \frac{1}{2}$
- $|\frac{1}{2} - 1|$ nu e neglijabilă si deci G nu este PRG

Observații

- Distributia output-ului unui PRG este departe de a fi uniformă
- Exemplificam pentru un G care dubleaza lungimea intrarii i.e. $l(n) = 2n$
- Pentru distributia uniforma peste $\{0,1\}^{2n}$, fiecare din cele 2^{2n} este ales cu probabilitate ...
- ... $\frac{1}{2^{2n}}$
- Consideram distributia output-ului lui G cand primeste la intrare un sir uniform de lungime n
- Numarul de siruri diferite din codomeniul lui G este cel mult ...
- ... 2^n
- Probabilitatea ca un sir de lungime $2n$ sa fie output al lui G este $2^n/2^{2n} = 1/2^n$

Observații

- Seed-ul unui PRG este analogul cheii unui sistem de criptare
- seed-ul trebuie ales uniform si mentinut secret
- seed-ul trebuie sa fie suficient de lung asa incat un atac prin forta bruta sa nu fie fezabil

Sistem de criptare bazat pe PRG

Definiție

Un sistem de criptare (Enc, Dec) *definit peste* $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ *se numește sistem de criptare bazat pe PRG dacă:*

1. $Enc : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$

$$c = Enc_k(m) = G(k) \oplus m$$

2. $Dec : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

$$m = Dec_k(c) = G(k) \oplus c$$

unde G *este un generator de numere pseudoaleatoare cu factorul de expansiune* l, $k \in \{0,1\}^n$, $m \in \{0,1\}^{l(n)}$

Securitate - interceptare unică

Teoremă

Dacă G *este PRG, atunci sistemul definit anterior este un sistem de criptare simetric de lungime fixă computațional sigur pentru un atacator pasiv care care poate intercepta un mesaj.*

Demonstrație intuitivă

- OTP este perfect sigur;
- Criptarea bazata pe PRG se obține din OTP prin înlocuirea *pad* cu $G(k)$;
- Dacă G este PRG, atunci *pad* și $G(k)$ sunt indistinctibile pentru orice $\mathcal{A}$ adversar PPT;
- În concluzie, OTP și sistemul de criptare bazat pe PRG sunt indistinctibile pentru $\mathcal{A}$.

Securitatea Sistemelor Informatice

- Curs 4.0 -
Noțiuni de securitate mai puternice

Adela Georgescu

Facultatea de Matematică și Informatică
Universitatea din București
Anul universitar 2022-2023, semestrul I

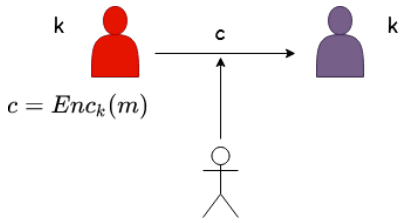
Securitate computațională

In cursurile anterioare:

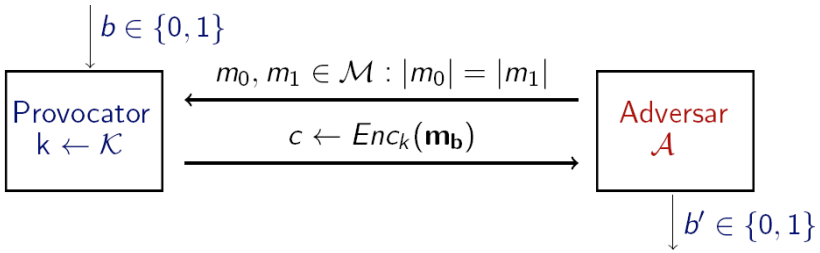
- Am definit securitate perfectă, am vazut OTP - perfect sigur și am evidențiat limitările practice
- In practică, vrem chei mai scurte și refolosirea cheii
- Am slăbit noțiunea de securitate perfectă și am obținut securitate computațională, considerand un adversar polinomial cu probabilitate neglijabilă de succes
- Am construit un sistem de criptare computațional sigur (satisface indistinctibilitatea) pentru care cheia de criptare este mai scurtă
- Insă acest sistem de criptare nu permite refolosirea cheii în siguranță

Securitate computațională

- In continuare considerăm noțiuni de securitate mai puternice care ne vor folosi pentru a obține refolosirea cheii
- Reamintim noțiunea de indistinctibilitate definită anterior, în cazul unui adversar care interceptează un singur mesaj criptat



Experimentul $Priv^{eav}_{\mathcal{A},\pi}(n)$



- Output-ul experimentului este 1 dacă $b' = b$ și 0 altfel. Dacă $Priv^{eav}_{\mathcal{A},\pi}(n) = 1$, spunem că $\mathcal{A}$ a efectuat experimentul cu succes.

Securitate - interceptare simplă

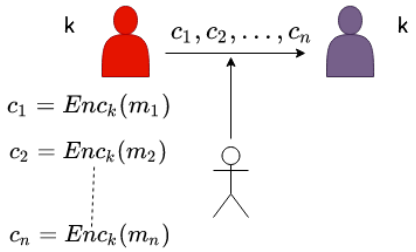
Definiție

O schemă de criptare $\pi = (Enc, Dec)$ este indistinctibilă în prezența unui atacator pasiv dacă pentru orice adversar $\mathcal{A}$ există o funcție neglijabilă negl așa încât

$$Pr[Priv^{eav}_{\mathcal{A},\pi}(n) = 1] \leq \frac{1}{2} + negl(n).$$

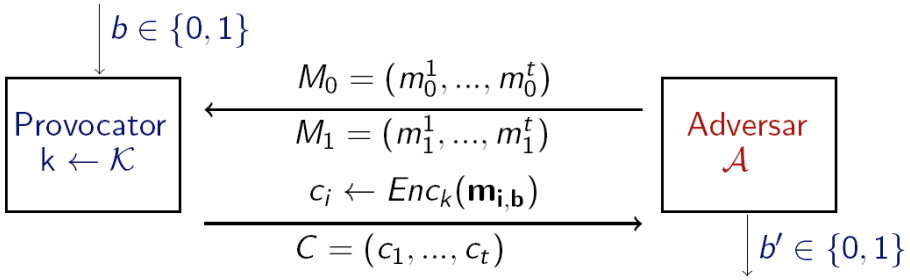
Securitate pentru interceptare multiplă

- In definiția precedentă am considerat cazul unui adversar care primește **un singur** text criptat;
- In realitate, în cadrul unei comunicații se trimit **mai multe mesaje** pe care adversarul le poate intercepta;
- Definim ce înseamnă o schemă sigură chiar și în aceste condiții.



Experimentul $Priv^{mult}_{\mathcal{A},\pi}(n)$

Experimentul $Priv^{mult}_{\mathcal{A},\pi}(n)$



- Output-ul experimentului este 1 dacă $b' = b$ și 0 altfel;
- Definiția de securitate este aceeași, doar că se referă la experimentul de mai sus.
- Securitatea pentru interceptare **simplă** nu implică securitate pentru interceptare **multiplă**!

Securitate pentru interceptare multiplă

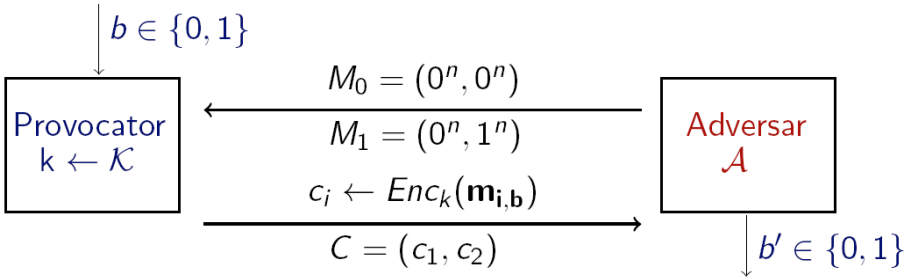
Teoremă

O schemă de criptare (Enc, Dec) unde funcția Enc este deterministă nu are proprietatea de securitate la interceptare multiplă conform cu definiția de mai sus.

Demonstrație

- Intuitiv, am vazut că schema OTP este sigură doar când o cheie este folosită o singură dată;
- La sistemul de criptare bazat pe PRG se întâmplă același lucru;
- Vom considera un adversar $\mathcal{A}$ care atacă schema (în sensul experimentului $Priv^{mult}_{\mathcal{A},\pi}(n)$)

Demonstrație



- Dacă $c_1 = c_2$, atunci $\mathcal{A}$ întoarce 0, altfel $\mathcal{A}$ întoarce 1.
- Analizăm probabilitatea ca $\mathcal{A}$ să ghicească b: dacă $b = 0$, același mesaj este criptat mereu ($m_0^1 = m_0^2$) iar $c_1 = c_2$ și deci $\mathcal{A}$ întoarce mereu 0;
- Dacă $b = 1$, atunci ($m_1^1 \neq m_1^2$) iar $c_1 \neq c_2$ și deci $\mathcal{A}$ întoarce mereu 1.

Concluzie

- $\mathcal{A}$ ghicește bitul b cu probabilitate 1 și deci schema nu este indistinctibilă la interceptare multiplă
- Pentru a obține *securitate la interceptare multipla*, avem nevoie de o schemă de criptare *probabilista*, așa încât la criptari succesive ale aceluiași mesaj să obținem texte criptate diferite

Scenarii de atac

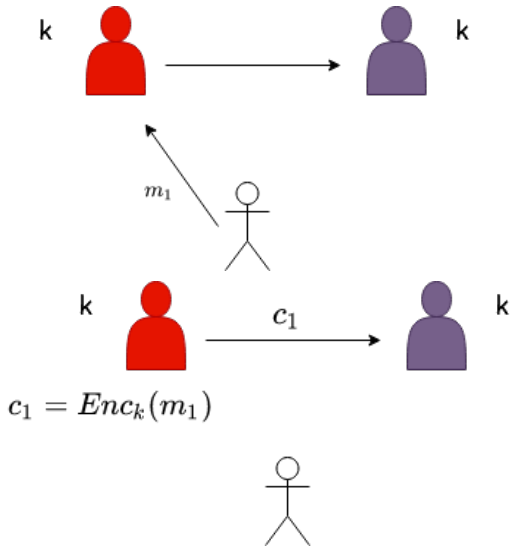
- Reamintim câteva dintre scenariile de atac pe care le-am mai întâlnit:
  - Atac cu text criptat: Atacatorul știe doar *textul criptat* - poate încerca un atac prin forță brută prin care se parcurg toate cheile până se găsește cea corectă;
  - Atac cu text clar: Atacatorul cunoaște una sau mai multe perechi *(text clar, text criptat)*;
  - Atac cu text clar ales: Atacatorul poate obține criptarea unor texte clare alese de el;
  - Atac cu text criptat ales: Atacatorul are posibilitatea să obțină decriptarea unor texte criptate alese de el.

Scenarii de atac

- Ultimele 2 scenarii de atac oferă adversarului putere crescută;
- Acesta devine un adversar **activ**, care primește abilitatea de a obține criptarea și / sau decriptarea unor mesaje, respectiv texte criptate alese de el;
- În plus, adversarul poate alege mesajele sau textele criptate în mod **adaptiv** în funcție de răspunsurile primite precedent.

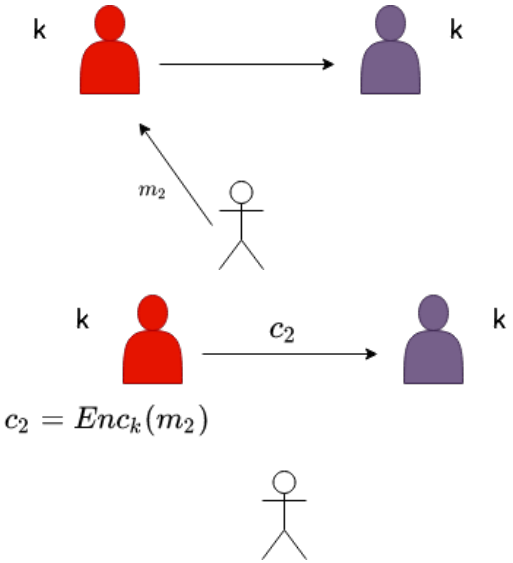
Securitate CPA

- CPA (Chosen-Plaintext Attack): adversarul poate să obțină criptarea unor mesaje alese de el;



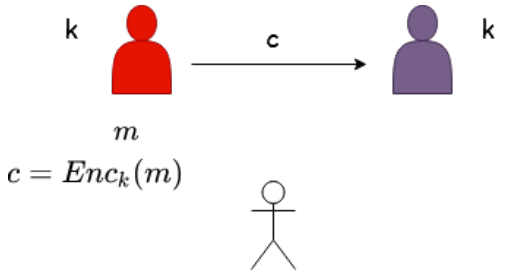
Securitate CPA

- adversarul poate cere criptarea unor mesaje alese de el repetitiv (polinomial de multe ori)



Securitate CPA

- mai tarziu adversarul observă criptarea unui mesaj necunoscut



- dorim ca adversarul să nu afle nici un fel de informație despre mesajul m

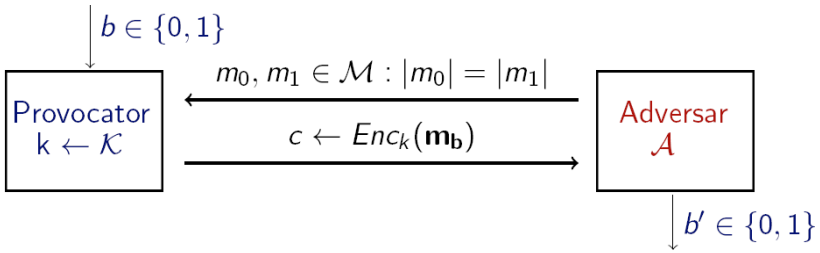
Securitate CPA

- Considerăm că securitatea este impactată dacă adversarul poate să distingă între criptările a două mesaje aleatoare;
- Vom defini securitatea CPA pe baza unui experiment de indistinctibilitate $Priv^{cpa}_{\mathcal{A},\pi}(n)$ unde $\pi = (Enc, Dec)$ este schema de criptare iar n este parametrul de securitate al schemei π;
- Personajele participante: **adversarul** $\mathcal{A}$ care încearcă să spargă schema și un **provocator (challenger)**;

Securitate CPA

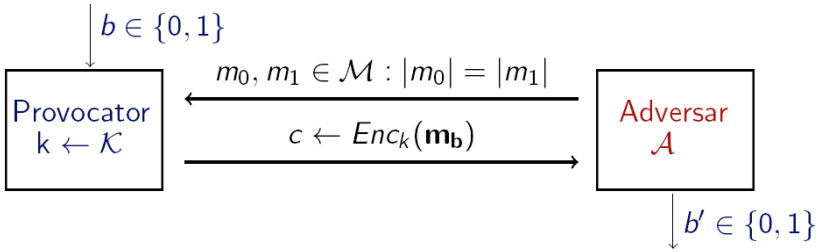
- Capabilitățile adversarului: el poate interacționa cu un **oracol de criptare**, fiind un adversar *activ* care poate rula atacuri în timp polinomial;
- Adversarul poate transmite către oracol orice mesaj m și primește înapoi textul criptat corespunzător;
- Dacă sistemul de criptare este nedeterminist, atunci oracolul folosește de fiecare dată o valoare aleatoare nouă și neutilizată anterior.

Experimentul $Priv^{cpa}_{\mathcal{A},\pi}(n)$



- Pe toată durata experimentului, $\mathcal{A}$ are acces la oracolul de criptare $Enc_k(\cdot)$!

Experimentul $Priv^{cpa}_{\mathcal{A},\pi}(n)$



- Output-ul experimentului este 1 dacă $b' = b$ și 0 altfel. Dacă $Priv^{cpa}_{\mathcal{A},\pi}(n) = 1$, spunem că $\mathcal{A}$ a efectuat experimentul cu succes.

Experimentul $Priv^{cpa}_{\mathcal{A},\pi}(n)$

Definiție

O schemă de criptare $\pi = (Enc, Dec)$ este CPA-sigură dacă pentru orice adversar PPT $\mathcal{A}$ există o funcție neglijabilă negl așa încât

$$Pr[Priv^{cpa}_{\mathcal{A},\pi}(n) = 1] \leq \frac{1}{2} + negl(n).$$

- Un adversar nu poate determina care text clar a fost criptat cu o probabilitate semnificativ mai mare decât dacă ar fi ghicit (în sens aleator, dat cu banul), chiar dacă are acces la oracolul de criptare.

Securitate CPA - al doilea război mondial

criptanaliza sistemului de criptare german Enigma de către englezi

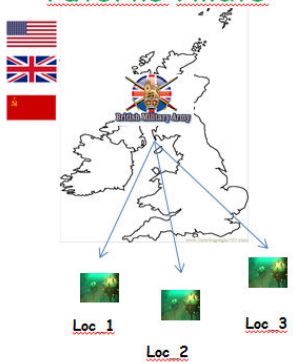
Puterile Aliate

Puterile Axei

Securitate CPA - al doilea război mondial

armata engleză a plasat mine în anumite locații...

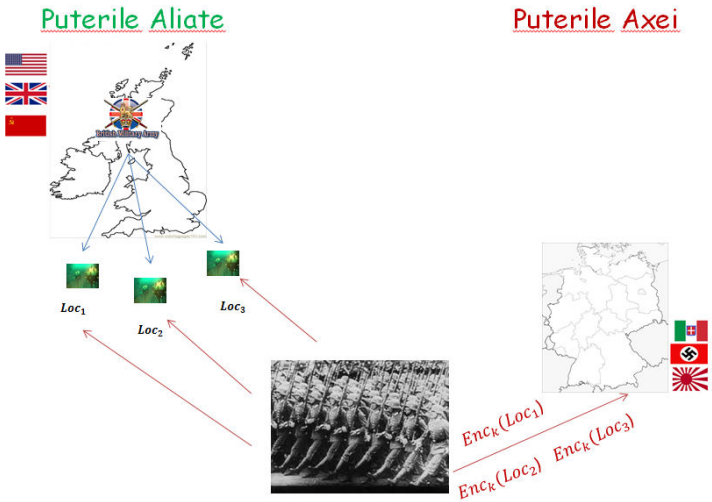
Puterile Aliate



Puterile Axei

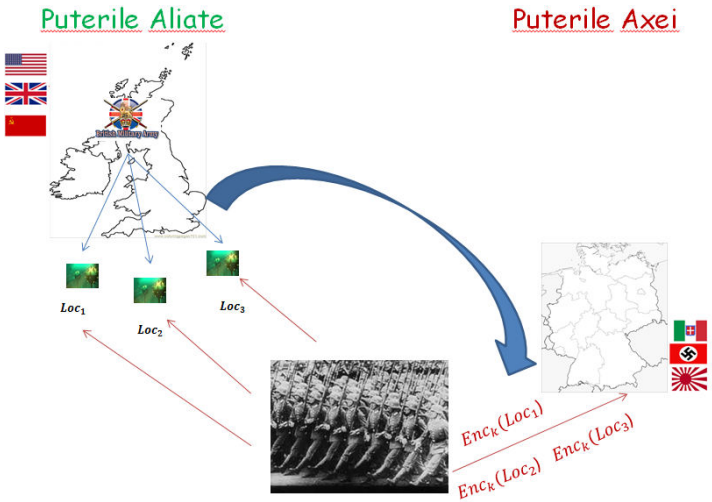
Securitate CPA - al doilea război mondial

...știind că armata germană le va găsi și va trimite locațiile lor criptate către sediu



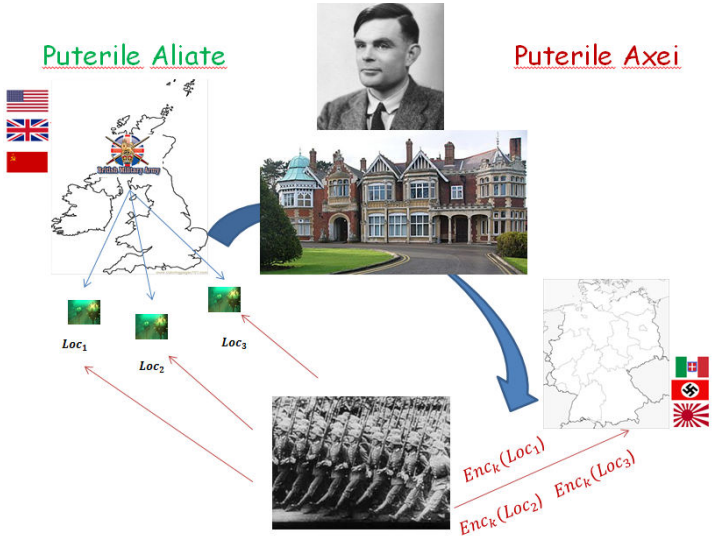
Securitate CPA - al doilea război mondial

aceste mesaje criptate au fost interceptate de către englezi ...



Securitate CPA - al doilea război mondial

... si folosite la Bletchely Park pentru criptanaliza mașinii Enigma



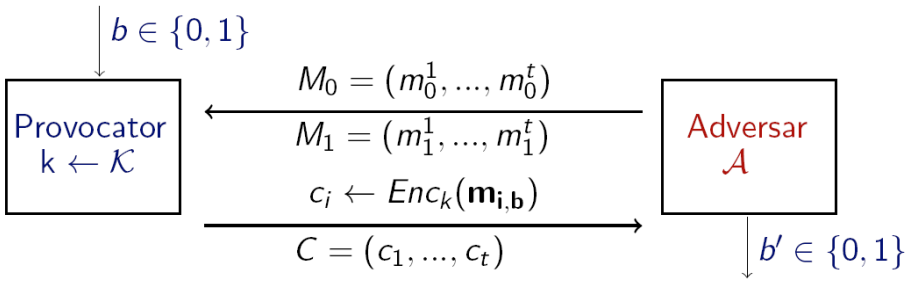
Securitate CPA

- Întrebare: Un sistem de criptare CPA-sigur are întotdeauna proprietatea de indistinctibilitate?
- Răspuns: DA! Experimentul $Priv^{eav}_{\mathcal{A},\pi}(n)$ este $Priv^{cpa}_{\mathcal{A},\pi}(n)$ în care $\mathcal{A}$ nu folosește oracolul de criptare.
- Întrebare: Un sistem de criptare determinist poate fi CPA-sigur?
- Răspuns: NU! Adversarul cere oracolului criptarea mesajului m_0. Dacă textul criptat este egal cu c, atunci $b' = 0$, altfel $b' = 1$. În concluzie, $\mathcal{A}$ câștigă cu probabilitate 1.

Securitate CPA - Criptare multiplă

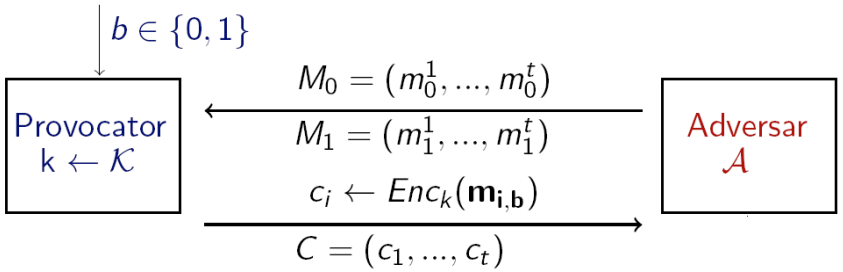
- În definiția precedentă am considerat cazul unui adversar care primește **un singur** text criptat;
- În realitate, în cadrul unei comunicații se trimit **mai multe mesaje** pe care adversarul le poate intercepta;
- Definim ce înseamnă o schemă sigură chiar și în aceste condiții.

Experimentul $Priv^{cpa}_{\mathcal{A},\pi}(n)$

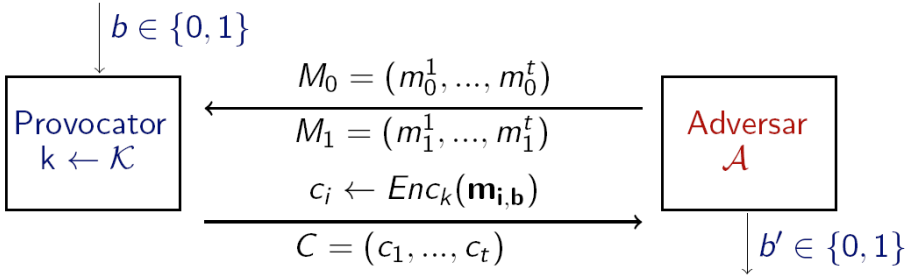


- Pe toată durata experimentului, $\mathcal{A}$ are acces la oracolul de criptare $Enc_k(\cdot)$!

Experimentul $Priv^{cpa}_{\mathcal{A},\pi}(n)$



Experimentul $Priv^{cpa}_{\mathcal{A},\pi}(n)$



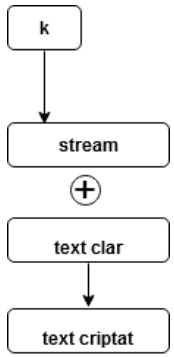
- Output-ul experimentului este 1 dacă $b' = b$ și 0 altfel;
- Definiția de securitate este aceeași, doar că se referă la experimentul de mai sus.
- Securitatea pentru criptare **simplă** implică securitate pentru criptare **multiplă**!

Recapitulare - PRG

- am definit generatoarele de numere pseudo-aleatoare, am văzut că ele sunt vulnerabile în fața unui adversar nelimitat computațional și că putem construi sisteme de criptare sigure bazate pe ele
- Intrebare PRG există?
- Răspuns: nu putem demonstra necondiționat, dar credem cu tărie că există
- Explicație: d.p.d.v. teoretic, putem construi PRG condiționat, bazat pe existența funcțiilor one-way
- In practică, construcțiile existente pentru PRG nu pot fi demonstrate ca fiind sigure, dar credem că sunt întrucât nu se cunosc algoritmi "distinguisher" ($\mathcal{D}$) eficienți $\rightarrow$ prezumpție: PRG există.

PRG-uri în practică

- Dezavantaj: PRG, asa cum le-am definit, produc tot output-ul odată și acesta este de lungime fixă
- In practică, PRG-urile sunt instanțiate cu sisteme de criptare fluide

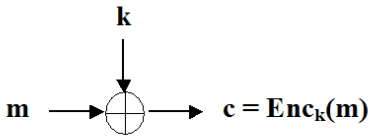


Sisteme fluide

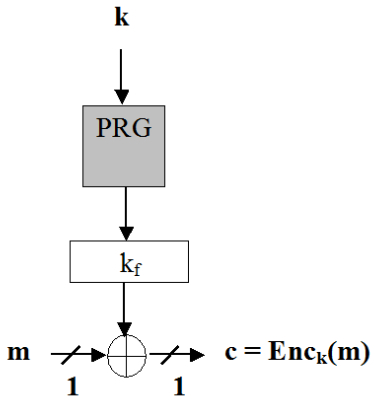
- sistemele fluide produc biții de output (pseudo-aleatori) gradual și la cerere, fiind mai eficiente și flexibile
- criptarea cu un sistem fluid presupune 2 faze:
  - **Faza 1**: se generează o secvență pseudoaleatoare de biți, folosind un generator de numere pseudoaleatoare (PRG)
  - **Faza 2**: secvența obținută se XOR-ează cu mesajul clar
- Atenție! De multe ori când ne referim la un sistem de criptare fluid considerăm doar Faza 1

Sisteme fluide

OTP (One Time Pad)



Sisteme fluide



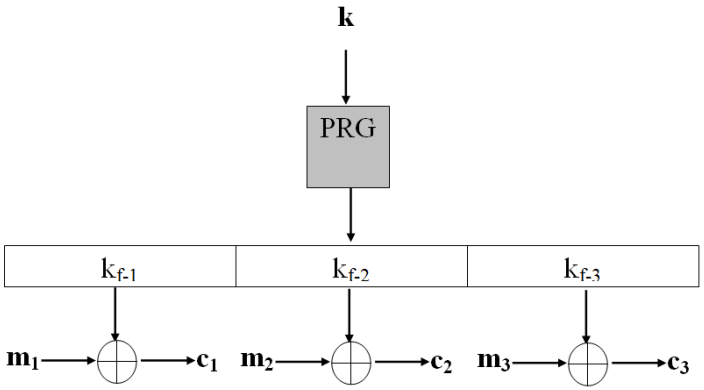
Securitate - interceptare multiplă

- Un sistem de criptare fluid în varianta prezentată este determinist: *unui text clar îi corespunde întotdeauna același mesaj criptat*;
- În consecință, utilizarea unui sistem fluid în forma prezentată pentru criptarea mai multor mesaje (cu aceeași cheie) este nesigură;
- Un sistem de criptare fluid se folosește în practică în 2 moduri: sincronizat și nesincronizat.

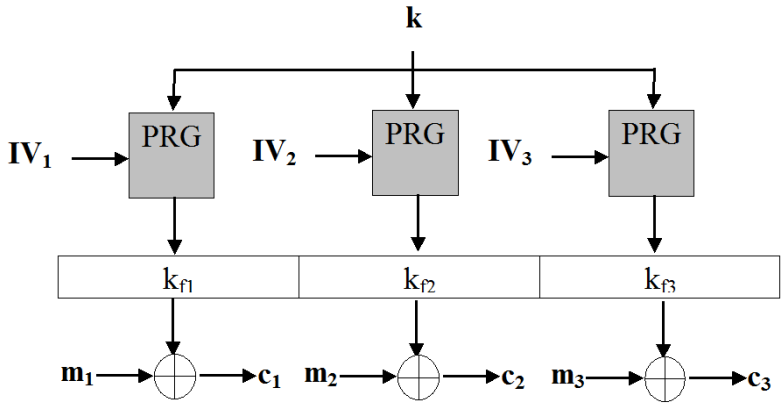
Moduri de utilizare

- modul sincronizat: partenerii de comunicație folosesc pentru criptarea mesajelor *părți succesive* ale secvenței pseudoaleatoare generate;
- modul nesincronizat: partenerii de comunicație folosesc pentru criptarea mesajelor secvențe pseudoaleatoare *diferite*.

Modul sincronizat



Modul nesincronizat



Moduri de utilizare

Modul sincronizat

- mesajele sunt criptate în mod succesiv (participanții trebuie să știe care părți au fost deja folosite)
- necesită păstrarea stării
- mesajele succesive pot fi percepute ca un *singur mesaj clar lung*, obținut prin concatenarea measajelor succesive
- se pretează unei singure sesiuni de comunicații

Modul nesincronizat

- mesajele sunt criptate în mod independent
- NU necesită păstrarea stării
- valorile $IV_1, IV_2, \ldots$ sunt alese uniform aleator pentru fiecare mesaj transmis
- valorile $IV_1, IV_2, \ldots$ (dar si IV în modul sincronizat) fac parte din mesajul criptat (sunt necesare pentru decriptare)

Proprietăți necesare ale PRG în modul nesincronizat

Fie $G(s, IV)$ un PRG cu 2 intrări:

- s = *seed*
- IV = *Initialization Vector*

PRG trebuie să se satisfacă (cel puțin):

1. $G(s, IV)$ este o secvență pseudoaleatoare chiar dacă IV este public (i.e. securitatea lui G constă în securitatea lui s);
2. dacă IV_1 și IV_2 sunt valori uniform aleatoare, atunci $G(s, IV_1)$ și $G(s, IV_2)$ sunt indistinctibile.

Exemple

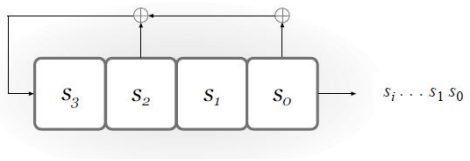
- RC4 (Ron"s Cipher 4):
  - definit de R.Rivest, în 1987
  - utilizat în WEP
  - inițial secret !
- WEP (Wired Equivalent Privacy):
  - standard IEEE 802.11, 1999 (rețele fără fir)
  - înlocuit în 2003 de WPA (Wi-Fi Protected Access), 2004 WPA2 - IEEE 802.11i

Exemple

- A5/1:
  - definit în 1987 pentru Europa și SUA
  - A5/2 definit în 1989 ca o variantă mai slabă pentru alte zone geografice
  - utilizat în rețelele de telefonie mobilă GSM
  - inițial secret !
- SEAL (Software-Optimized Encryption Algorithm)
  - definit de D.Coppersmith și P.Rogaway, în 1993
  - prezintă o implementare foarte eficientă pe procesoarele pe 32 de biți
  - versiunea curentă (SEAL 3.0) este patentată IBM

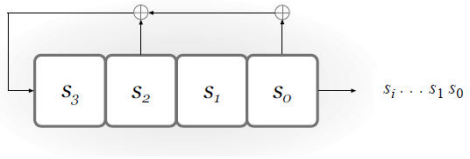
Linear-Feedback Shift Registers (LFSR)

- sunt foarte eficiente în implementari hardware
- au proprietăți statistice bune dar totuși sunt predictibile, deci nu sunt PRG-uri sigure din punct de vedere criptografic
- Mai jos este un exemplu



- Componente, în general:
  - *n registri* $s_{n-1}, ..., s_0$ - fiecare contine un singur bit
  - *n coeficienti feedback* $c_{n-1}, ..., c_0$
  - *gradul* este n

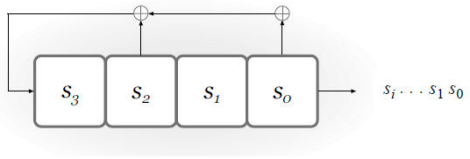
Linear-Feedback Shift Registers (LFSR)



In exemplul de mai sus avem

- $c_0 = c_2 = 1$ si $c_1 = c_3 = 0$
- fiecare bit de la ieșire este calculat după formula $c_0 s_0 \oplus ... \oplus c_3 s_3$
- la fiecare tact de ceas, LFSR scoate la ieșire valoarea din registrul s_0 iar valorile din ceilalți regiștri sunt deplasate la dreapta cu o poziție

Linear-Feedback Shift Registers (LFSR)



Pentru starea inițială (0,0,1,1), biții de la ieșire vor fi ...
(0,0,1,1)
(1,0,0,1)
...
In general

- starea LFSR constă din *n* biți (conținutul regiștrilor la un moment dat)
- există cel mult 2^n stări posibile până când output-ul LFSR-ului se repetă
- cunoscând cel mult $2n$ biti de la ieșire, un atacator poate afla starea inițială și coeficienții de feedback

RC4
Informații generale

RC4 este:

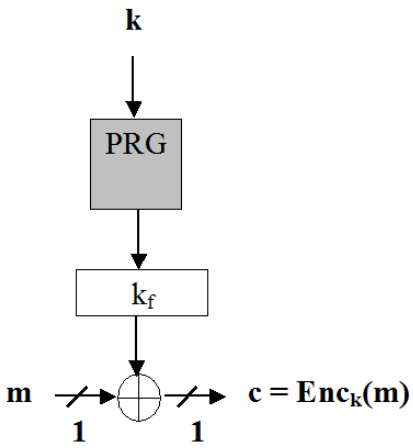
- introdus de R. Rivest la MIT (1987);
- înregsitrat ca marca a RSA Data Security;
- păstrat secret până în 1994 când a devenit public;
- utilizat în WEP, SSL/TLS.

Descriere

- RC4 este un sistem de criptare fluid pe octeți:

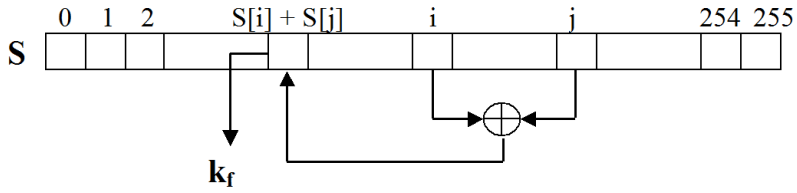
$$m \in \{0,1\}^8, c \in \{0,1\}^8$$

- Ramâne de definit PRG...



Descriere

- 2 faze:
  - inițializare: determină starea internă, fără să producă chei fluide;
  - generare de chei fluide: modifică starea internă și generează un octet (*cheia fluidă*) care se XOR-ează cu m pentru a obține c;
- Starea internă:
  - un tablou S de 256 octeți: $S[0], \ldots, S[255]$;
  - 2 indici i și j;
- Toate operațiile se efectuează pe octeți (i.e. (mod 256)).



Descriere

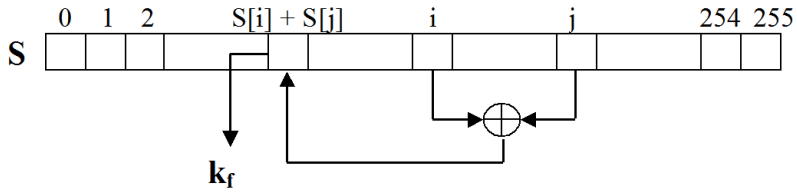
Faza 1. Inițializare

- n = numărul octeților din cheie, $1 \leq n \leq 256$
- $j \leftarrow 0$
  for $i = 0$ **to** 255 **do**
  $S[i] \leftarrow i$
  end for
  for $i = 0$ **to** 255 **do**
  $j \leftarrow j + S[i] + k[i \pmod n]$
  swap $(S[i], S[j])$
  end for
  $i \leftarrow 0$
  $j \leftarrow 0$

Descriere

Faza 2. Generarea cheii fluide

- cheia se obține octet cu octet
- $i \leftarrow i + 1$
  $j \leftarrow j + S[i]$
  swap $(S[i], S[j])$
  return $S[S[i] + S[j]]$



Descriere

Detalii de implementare:

- $5 \leq n \leq 16 \Rightarrow 40 \leq |k| \leq 256$;
- memorie: 256 octeți (pentru S) și câteva variabile *byte*;
- operații simple, rapid de executat.

Securitate

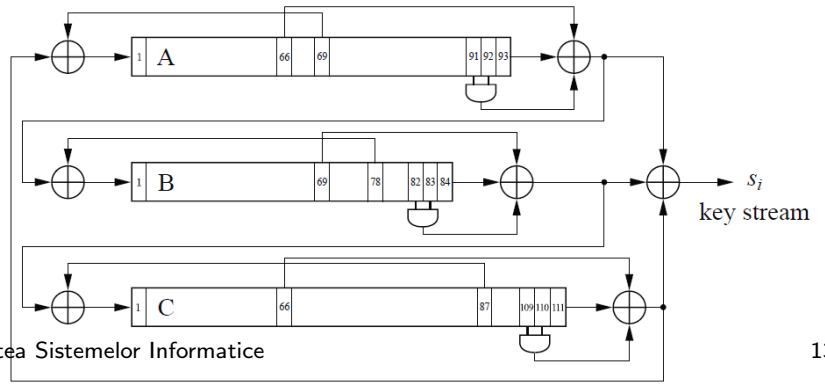
- primii octeți generați drept cheie fluidă sunt total ne-aleatori și oferă informații despre cheie (Fluhrer, Mantin and Shamir 2001)
- RC4 pe 104 biți (utilizat pentru WEP pe 128 biți) a fost spart în aprox. 1 min (algoritm al lui Tews, Weinmann, Pychkine 2001, bazat pe idea lui Klein 2005)
- un atac recent arată că pot fi determinați primii aprox. 200 octeți din textul clar criptat cu RC4 în TLS cunoscând $[2^{28} - 2^{32}]$ criptări independente (Royal Holloway, 2013)

Vulnerabilitati LFSR

- LFSR-urile sunt liniare iar liniaritatea induce vulnerabilități (sistemele liniare de ecuații permit aflarea informațiilor sensibile)
- Insă combinațiile de mai multe LFSR-uri pot produce sisteme de criptare sigure

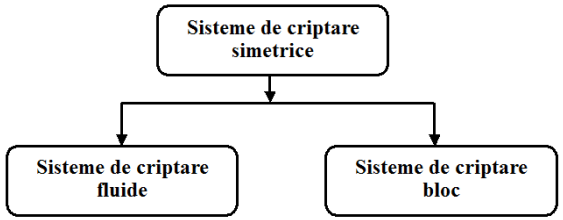
Trivium

- Trivium a fost propus în 2008, este simplu și compact hardware, constă din 3 FSR-uri (feedback shift registers) neliniare de grad 93, 84 respectiv 111
- Regiștri sunt cuplați: la fiecare tact, cel mai din stânga registru va conține o valoare calculată ca funcție aplicată unui registru din același FSR dar și unor regiștri dintr-un alt FSR
- cel mai bun atac cunoscut pentru Trivium este cel prin forță brută

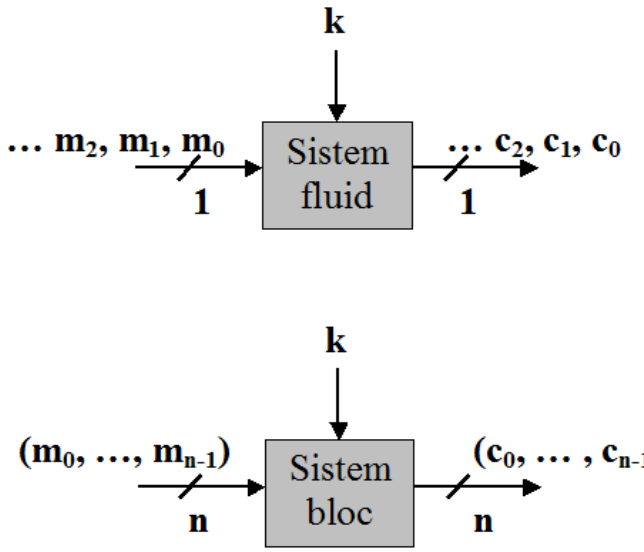


Criptografia simetrică

- Am studiat sisteme simetrice care criptează **bit cu bit** - sisteme de criptare fluide;
- Vom studia sisteme simetrice care criptează **câte n biți simulan** - sisteme de criptare bloc;



Sisteme bloc vs. sisteme fluide



Sisteme bloc vs. sisteme fluide

... d.p.d.v. al modului de criptare:

Sisteme fluide

- criptarea biților se realizează individual
- criptarea unui bit din textul clar este independentă de orice alt bit din textul clar

Sisteme bloc

- criptarea se realizează în blocuri de câte *n* biți
- criptarea unui bit din textul clar este dependentă de biții din textul clar care aparțin aceluiași bloc

Sisteme bloc vs. sisteme fluide

... d.p.d.v. *tradițional*, în practică:

Sisteme fluide

- necesități computaționale reduse
- utilizare: telefoane mobile, dispozitive încorporate, PDA
- par să fie mai puțin sigure, multe sunt sparte

Sisteme bloc

- necesități computaționale mai avansate
- utilizare: internet
- par să fie mai sigure, prezintă încredere mai mare

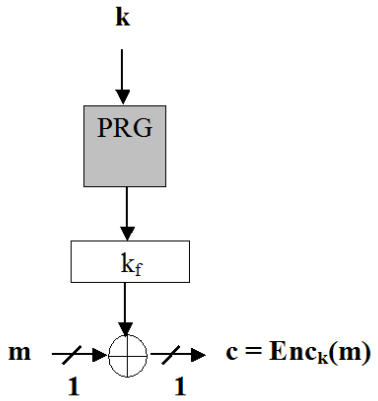
Sisteme bloc

- Introducem noțiunea de permutare pseudoaleatoare sau PRP *(PseudoRandom Permutation)*
- În analogie cu ce știm deja:
  - **PRP** sunt necesare pentru construcția **sistemelor bloc**

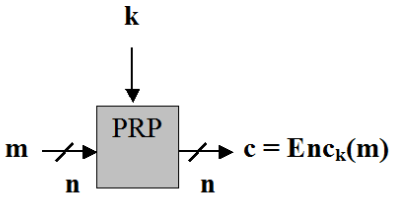
    așa cum
  - **PRG** sunt necesare pentru construcția **sistemelor fluide**

Sisteme bloc

Sisteme fluide



Sisteme bloc



PRP

- Ramâne să definim noțiunea de permutare pseudoaleatoare sau PRP *(PseudoRandom Permutation)*;
- Acesta este o funcție **deterministă** și **bijectivă** care pentru o cheie fixată produce la ieșire o **permutare** a intrării ...
- ... **indistinctibilă** față de o permutare aleatoare;
- În plus, atât funcția cât și inversa sa sunt **eficient calculabile**.

PRP

Definiție

O permutare pseudoaleatoare definită peste $(\mathcal{K}, \mathcal{X})$ *este o funcție bijectivă*

$$F : \mathcal{X} \times \mathcal{K} \to \mathcal{X}$$

care satisface următoarele proprietăți:

1. *Eficiență:* $\forall k \in \mathcal{K}, x \in \mathcal{X}, \exists$ *algoritmi determiniști polinomiali care calculează* $F_k(x)$ *și* $F_k^{-1}(x)$
2. *Pseudoaleatorism:* $\forall$ *algoritm PPT* $\mathcal{D}$, $\exists$ *o funcție neglijabilă negl a.î.:*

$$|Pr[D(r) = 1] - Pr[D(F_k(\cdot)) = 1]| \leq negl(n)$$

unde $r \leftarrow^R Perm(X), k \leftarrow^R \mathcal{K}$

Notații

- $F_k(x) = F(k, x)$
  o cheie este în general (aleator) aleasă și apoi fixată
- $Perm(X)$ = mulțimea tuturor funcțiilor bijective de la $\mathcal{X}$ la $\mathcal{X}$
- $\mathcal{X} = \{0,1\}^n$
- $\mathcal{D}$ = *Distinguisher* care are acces la *oracolul* de evaluare a funcției

PRF

- Introducem noțiunea de funcție pseudoaleatoare sau PRF *(PseudoRandom Function)*...
- ... ca o generalizare a noțiunii de **permutare pseudoaleatoare**;
- Acesta este o funcție **cu cheie** care este **indistinctibilă** față de o funcție aleatoare (cu același domeniu și mulțime de valori).

PRF

Definiție

O funcție pseudoaleatoare definită peste $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$ este o funcție bijectivă

$$F : \mathcal{X} \times \mathcal{K} \to \mathcal{Y}$$

care satisface următoarele proprietăți:

1. *Eficiență:* $\forall k \in \mathcal{K}, x \in \mathcal{X}, \exists$ *algoritm determinist polinomial care calculează* $F_k(x)$
2. *Pseudoaleatorism:* $\forall$ *algoritm PPT* $\mathcal{D}$, $\exists$ *o funcție neglijabilă negl a.î.:*

$$|Pr[D(r) = 1] - Pr[D(F_k(\cdot)) = 1]| \leq negl(n)$$

unde $r \leftarrow^R Func(X, Y), k \leftarrow^R \mathcal{K}$

Notații

- $F_k(x) = F(k, x)$
  o cheie este în general (aleator) aleasă și apoi fixată
- $Func(X, Y)$ = mulțimea funcțiilor de la $\mathcal{X}$ la $\mathcal{Y}$
- $\mathcal{X} = \{0,1\}^n$, $\mathcal{Y} = \{0,1\}^n$
  considerăm în general că *PRF păstrează lungimea*
- $\mathcal{D}$ = *Distinguisher* care are acces la *oracolul* de evaluare a funcției

$PRP \subseteq PRF$

- Întrebare: De ce *PRF* poate fi privită ca o generalizare a *PRP*?
- Răspuns: *PRP* este *PRF* care satisface:
  1. $\mathcal{X} = \mathcal{Y}$
  2. este inversabilă
  3. calculul funcției inverse este eficient

Construcții

- **PRF $\Rightarrow$ PRG**
  Pornind de la PRF se poate construi PRG
- **PRG $\Rightarrow$ PRF**
  Pornind de la PRG se poate construi PRF
- **PRP $\Rightarrow$ PRF**
  Pornind de la PRP se poate construi PRF
- **PRF $\Rightarrow$ PRP**
  Pornind de la PRF se poate construi PRP

Întrebare: Care dintre aceste construcții este trivială?

Construcții

Răspuns: **PRP $\Rightarrow$ PRF**

PRP este o particularizare a $PRF : \mathcal{X} \times \mathcal{K} \rightarrow Y$ care satisface:

1. $\mathcal{X} = \mathcal{Y}$
2. este inversabilă
3. calculul funcției inverse este eficient

Construcții

- **PRF $\Rightarrow$ PRG**
  Pornind de la PRF se poate construi PRG
- **PRG $\Rightarrow$ PRF**
  Pornind de la PRG se poate construi PRF
- **PRP $\Rightarrow$ PRF** $\surd$
  Pornind de la PRP se poate construi PRF
- **PRF $\Rightarrow$ PRP**
  Pornind de la PRF se poate construi PRP

$PRF \Rightarrow PRG$

- Considerăm $F : \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$ PRF;
- Construim $G : \mathcal{K} \rightarrow \{0,1\}^{nl}$ *PRG* sigur:

$$G(k) = F_k(0)||F_k(1)||\ldots||F_k(l-1)$$

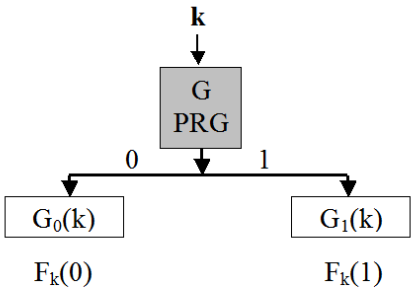
- Întrebare: De ce este G sigur?
- Răspuns: $F_k(\cdot)$ este *indistinctibilă* față de o funcție aleatoare $\Rightarrow G(k)$ este *indistinctibilă* față de o secvență aleatoare de lungime ln.
- Avantaj: Construcția este *paralelizabilă*

Construcții

- **PRF $\Rightarrow$ PRG** $\surd$
  Pornind de la PRF se poate construi PRG
- **PRG $\Rightarrow$ PRF**
  Pornind de la PRG se poate construi PRF
- **PRP $\Rightarrow$ PRF** $\surd$
  Pornind de la PRP se poate construi PRF
- **PRF $\Rightarrow$ PRP**
  Pornind de la PRF se poate construi PRP

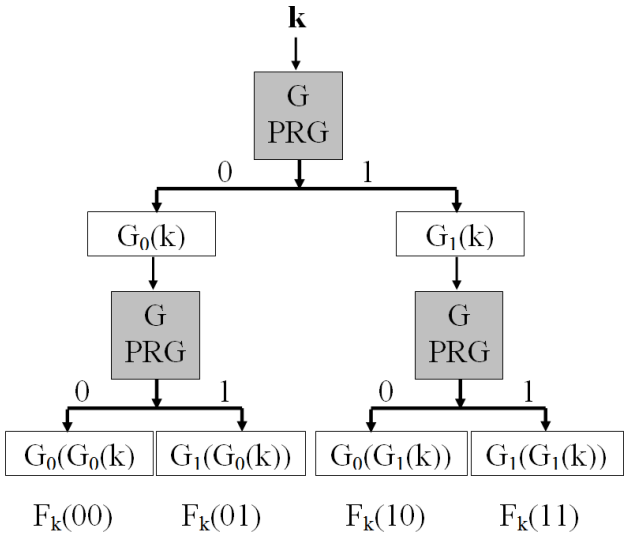
$PRG \Rightarrow PRF$

- Construcția pentru un singur bit de intrare...



$PRG \Rightarrow PRF$

- ...se poate generaliza pentru un număr oarecare de biți



PRG ⇒ PRF

- Construcția poate fi reprezentată ca un arbore binar cu cheia k rădăcină;
- Pentru un nod de valoare k', copilul stâng ia valoarea $G_0(k')$ și copilul drept ia valoare $G_1(k')$;
- Valoarea funcției $F_k(x) = F_k(x_0, \ldots, x_{n-1})$ este obținută prin parcurgerea arborelui în funcție de x ;
- Adâncimea arborelui este *liniară* în n (n);
- Dimensiunea arborelui este *exponențială* în n (2^n);
- NU se utilizează în practică din cauza performanței scăzute.

Construcții

- **PRF ⇒ PRG** $\surd$
  Pornind de la PRF se poate construi PRG
- **PRG ⇒ PRF** $\surd$
  Pornind de la PRG se poate construi PRF
- **PRP ⇒ PRF** $\surd$
  Pornind de la PRP se poate construi PRF
- **PRF ⇒ PRP**
  Pornind de la PRF se poate construi PRP

PRF ⇒ PRP

Teoremă (Luby-Rackoff 5)

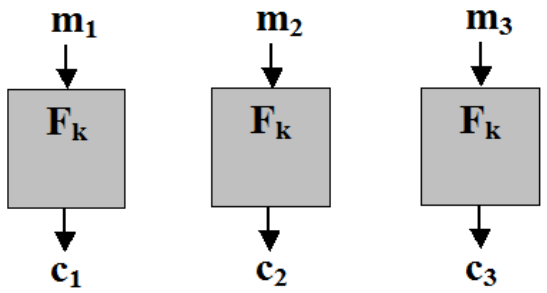
Dacă $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ *este PRF, se poate construi* $F' : \mathcal{K} \times \{0,1\}^2 \to \{0,1\}^2$ *PRP.*

- Construcția folosește runde **Feistel**, pe care le vom prezenta într-un curs ulterior.

Moduri de utilizare

- Să continuam cu ceva mai practic...
- Întrebare: Ce se întâmplă dacă lungimea mesajului clar este **mai mică** decât dimensiunea unui bloc?
- Răspuns: Se completează cu biți: **1 0 . . . 0**;
- Întrebare: Ce se întâmplă dacă lungimea mesajului clar este **mai mare** decât lungimea unui bloc?
- Răspuns: Se utilizează un **mod de operare** (ECB, CBC, OFB, CTR);
- Notăm cu F_k un sistem de criptare bloc (i.e. PRP) cu cheia k fixată.

Modul ECB (Electronic Code Book)



Modul ECB (Electronic Code Book)

- Pare modul cel mai **natural** de a cripta mai multe blocuri;
- Pentru decriptare, F_k trebuie să fie **inversabliă**;
- Este **paralelizabil**;
- Este **determinist**, deci este **nesigur**;
- Întrebare: Ce informații poate să ofere modul de criptare ECB unui adversar pasiv?
- Răspuns: Un adversar pasiv detectează repetarea unui bloc de text clar pentru că se repetă blocul criptat corespunzător;
- Modul ECB NU trebuie utilizat în practică!

Modul ECB (Electronic Code Book)

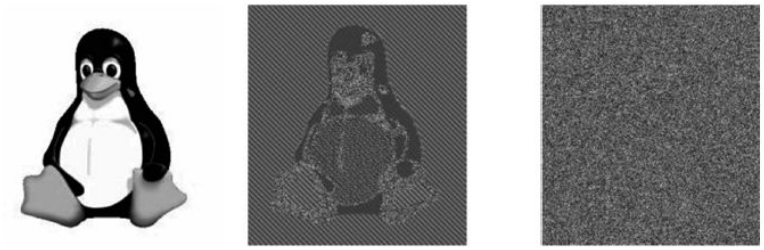
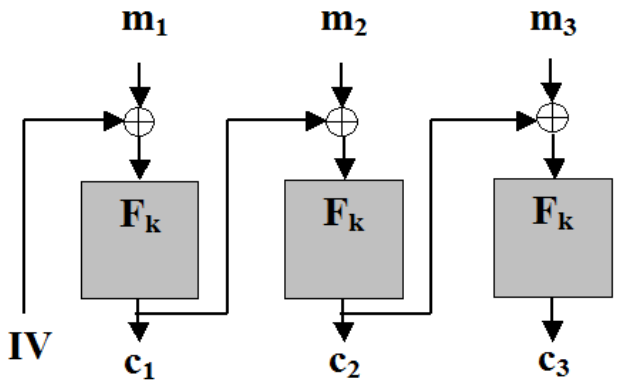


Figure: Imagine preluată de pe `https://en.wikipedia.org/`

Figura din mijloc este criptarea imaginii din stânga în modul ECB.
In dreapta este aceeași imagine criptată folosind un mod sigur.

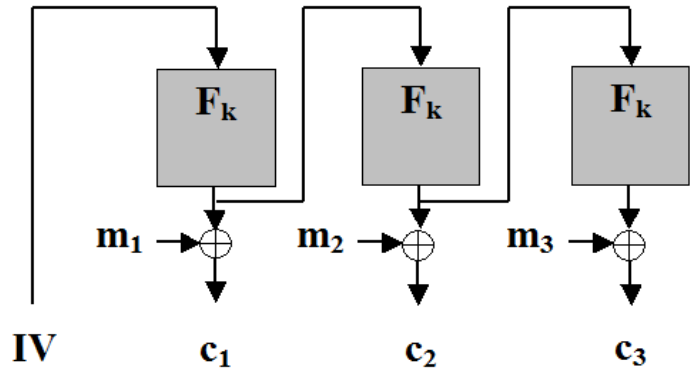
Modul CBC (Cipher Block Chaining)



Modul CBC (Cipher Block Chaining)

- IV este o ales în mod aleator la criptare;
- IV se transmite în clar pentru ca este necesar la decriptare;
- Pentru decriptare, F_k trebuie să fie **inversabliă**;
- Este **secvențial**, un dezavantaj major dacă se poate utiliza procesarea paralelă.

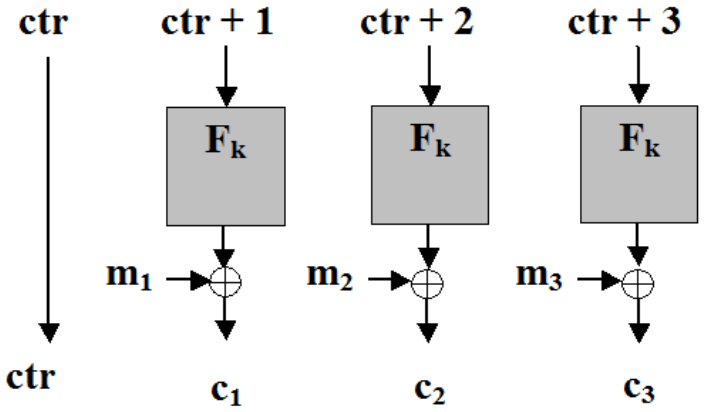
Modul OFB (Output FeedBack)



Modul OFB (Output FeedBack)

- Generează o secvență pseudoaleatoare care se XOR-ează mesajului clar;
- IV este o ales în mod aleator la criptare;
- IV se transmite în clar pentru ca este necesar la decriptare;
- F_k nu trebuie neapărat să fie inversabliă;
- Este **secvențial**, însă secvența pseudoaleatoare poate fi pre-procesată anterior decriptării.

Modul CTR (Counter)



Modul CTR (Counter)

- Generează o secvență pseudoaleatoare care se XOR-ează mesajului clar;
- *ctr* este o ales în mod aleator la criptare;
- *ctr* se transmite în clar pentru ca este necesar la decriptare;
- F_k nu trebuie neapărat să fie inversabliă;
- Este **paralelizabil**;
- În plus, secvența pseudoaleatoare poate fi pre-procesată anterior decriptării.

Modul CTR (Counter)

- Generează o secvență pseudoaleatoare care se XOR-ează mesajului clar;
- *ctr* este o ales în mod aleator la criptare si se transmite în clar pentru ca este necesar la decriptare;
- Este **paralelizabil**; F_k nu trebuie neapărat să fie inversabliă;
- În plus, secvența pseudoaleatoare poate fi pre-procesată anterior decriptării.
- CTR poate fi văzut și ca un sistem fluid nesincronizat:
  - pentru criptarea unui mesaj de lungime $l < 2^{n/4}$ blocuri, se alege un IV uniform din $\{0,1\}^{2n/4}$
  - fiecare bloc de text criptat este calculat $y_i = F_k(IV||i)$ unde i este codificat ca un string pe $n/4$ biți

Câteva considerații practice

- modurile CTR, OFB și CBC sunt CPA-sigure
- modurile CBC, OFB și CTR folosesc un IV uniform aleator - asigură faptul că F_k este mereu evaluat pe intrări diferite (previne situația în care adversarul afla informații la vederea de intrări identice)
- CTR - IV ales uniform de lungime $3n/4$ înseamnă că IV se repetă după criptarea aprox. $q(n) = 2^{2n/8}$ mesaje
- Dacă $n = 64$ atunci $q \approx 17.000.000$ ceea ce e puțin pentru zilele noastre
- Dacă $n = 128$ și vrem să folosim CTR avand garanția că IV se repetă cu probabilitate cel mult 2^{-40}, rezultă $q \approx 2^{28}$ mesaje (calculand q din $\frac{q^2}{2^{3n/4}+1} \leq 2^{-40}$)

Câteva considerații practice - IV folosit greșit

- Ce se întâmplă dacă IV se repetă?
- Pentru modurile OFB si CTR, întregul stream pseudoaleator (cu care se face xor pe mesaj) se repetă
- Dacă IV nu este uniform aleator (deci este predictibil), CTR este sigur dar CBC nu este sigur.

Securitatea Sistemelor Informatice

- Curs 5.1 -
Scheme de criptare CPA-sigure bazate pe PRF

Adela Georgescu

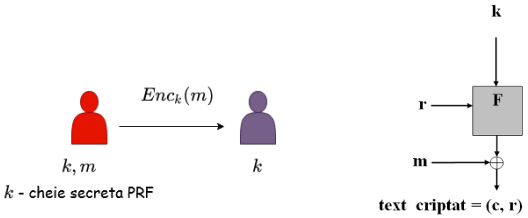
Facultatea de Matematică și Informatică
Universitatea din București
Anul universitar 2022-2023, semestrul I

Sisteme de criptare bloc - observații

- Sistemele de criptare bloc sunt instanțieri sigure ale PRP
  Pentru n suficient de mare, un PRP este indistinctibil de un PRF
- reamintim că pentru PRP avem nevoie și de invertibilitate, dar pentru un n suficient de mare, un PRP este și un PRF
- în practică, sistemele de criptare bloc sunt și PRF bune, nu doar PRP-uri bune, deci le putem folosi oricând avem nevoie de una din cele două construcții

Sistem de criptare CPA-sigur

- Sistemele de criptare bloc sunt instanțieri sigure ale PRP



- Fie F_k o funcție cu cheie
- $\mathrm{Gen}(1^n)$: alege uniform cheie $k \in \{0,1\}^n$
- $\mathrm{Enc_k(m)}$: pentru $|m| = |k|$, alege r uniform în $\{0,1\}^n$

$$Enc_k(m) = (r, F_k(r) \oplus m)$$

- $\mathrm{Dec_k(c = (c_0, c_1))}$: întoarce $c_1 \oplus F_k(c_0)$

Observații

- cheia este la fel de lungă precum mesajul - la fel ca la OTP
- dar, spre deosebire de OTP, se pot cripta mai multe mesaje cu aceeași cheie în siguranță

Sistem de criptare CPA-sigur

Teoremă

Dacă F este PRF, construcția anterioară este o schemă de criptare CPA-sigură pentru mesaje de lungime n.

Schița demonstrației

Considerăm $(\overline{\Pi} = \overline{\mathrm{Gen}}, \overline{\mathrm{Enc}}, \overline{\mathrm{Dec}})$ *care se obține din schema anterioară* $(\Pi = \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *unde* F_k *- PRF este înlocuită cu f aleatoare. Fie* $\mathcal{A}$ *- adversar PPTsi* $q(n)$ *numărul maxim de interogări ale oracolului de criptare efectuate de* $\mathcal{A}$*. Arătăm:*

1. $\mathcal{A}$ *nu poate distinge între* Π *si* $\overline{\Pi}$ *decât cu probabilitate neglijabilă adică: există o funcție neglijabilă* `negl` *așa încât:*

$$|Pr[Priv^{cpa}_{\mathcal{A},\pi}(n) = 1] - Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1]| \leq \mathtt{negl(n)}$$

Sistem de criptare CPA-sigur

2. $Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1] \leq \frac{1}{2} + \frac{q(n)}{2^n}$.
   - la fiecare criptare a lui m - interogare la oracol sau ca provocare de la Challenger - se alege $r \in \{0,1\}^n$ iar $c = (r, f(r) \oplus m)$.
   - fie $(\tilde{r}, f(\tilde{r}) \oplus m_b)$ provocarea primită de $\mathcal{A}$. Există 2 variante:
     1. valoarea $\tilde{r}$ nu este folosită niciodată ca răspuns de către oracolul de criptare $\Rightarrow Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1] = \frac{1}{2}$
     2. valoarea $\tilde{r}$ este folosită cel puțin o dată ca răspuns la interogările oracolului de criptare $\Rightarrow$ $\mathcal{A}$ poate calcula m_b. El primește răspuns de la oracolul de criptare $Enc(m) = (\tilde{r}, s)$ și calculează $f(\tilde{r}) = s \oplus m$. $Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1] \leq \frac{q(n)}{2^n}$.
   - Notăm cu Ev evenimentul de la 2. și cu $\neg Ev$ evenimentul de la 1. Atunci: $Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1] =$
   $= Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1] \wedge Ev + Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1] \wedge \neg Ev$

$$\leq \frac{q(n)}{2^n} + \frac{1}{2}$$

.

Sistem de criptare CPA-sigur

- Am obținut că $Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1] \leq \frac{q(n)}{2^n} + \frac{1}{2}$.
- Am stabilit de asemenea că $|Pr[Priv^{cpa}_{\mathcal{A},\pi}(n) = 1] - Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1]| \leq \mathtt{negl(n)}$
- Din ambele relații avem $Pr[Priv^{cpa}_{\mathcal{A},\overline{\pi}}(n) = 1] \leq \frac{q(n)}{2^n} + \frac{1}{2} + \mathtt{negl(n)}$, ceea ce încheie demonstrația.

Scenarii de atac activ

- Reamintim câteva dintre scenariile de atac pe care le-am mai întâlnit:
  - Atac cu text clar ales - chosen plaintext attack (CPA): Atacatorul poate obține criptarea unor texte clare alese de el;
  - Atac cu text criptat ales - chosen ciphertext attac (CCA): Atacatorul are posibilitatea să obțină decriptarea unor texte criptate alese de el.

Scenarii de atac activ

- In aceste scenarii de atac adversarul are putere crescută;
- Acesta devine un adversar **activ**, care primește abilitatea de a obține criptarea și / sau decriptarea unor mesaje, respectiv texte criptate alese de el;
- În plus, adversarul poate alege mesajele sau textele criptate în mod **adaptiv** în funcție de răspunsurile primite precedent.

Noțiuni de securitate

- Definim astfel 2 noțiuni de securitate:
  - CPA (Chosen-Plaintext Attack): adversarul poate să obțină criptarea unor mesaje alese de el; - discutată în cursul anterior
  - CCA (Chosen-Ciphertext Attack): adversarul poate să obțină criptarea unor mesaje alese de el și decriptarea unor texte criptate alese de el.

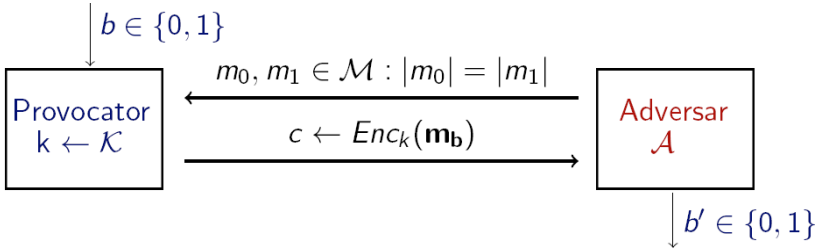
Securitate CCA

- Capabilitățile adversarului: el poate interacționa cu un **oracol de criptare** și cu un **oracol de decriptare**, fiind un adversar *activ* care poate rula atacuri în timp polinomial;
- Adversarul poate transmite către oracolul de criptare orice mesaj m și primește înapoi textul criptat corespunzător sau poate transmite către oracolul de decriptare *anumite* mesaje c și primește înapoi mesajul clar corespunzător;
- Dacă sistemul de criptare este nedeterminist, atunci oracolul de criptare folosește de fiecare dată o valoare aleatoare nouă și neutilizată anterior.

Securitate CCA

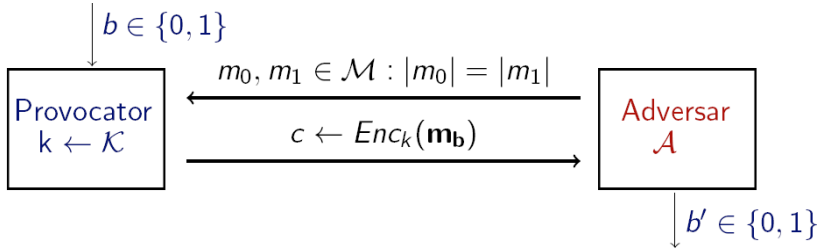
- Considerăm că securitatea este impactată dacă adversarul poate să distingă între criptările a două mesaje aleatoare;
- Vom defini securitatea CCA pe baza unui experiment de indistinctibilitate $Priv^{cca}_{\mathcal{A},\pi}(n)$ unde $\pi = (Enc, Dec)$ este schema de criptare iar n este parametrul de securitate al schemei π;
- Personajele participante: **adversarul** $\mathcal{A}$ care încearcă să spargă schema și un **provocator (challenger)**;

Experimentul $Priv^{cca}_{\mathcal{A},\pi}(n)$



- Output-ul experimentului este 1 dacă $b' = b$ și 0 altfel. Dacă $Priv^{cca}_{\mathcal{A},\pi}(n) = 1$, spunem că $\mathcal{A}$ a efectuat experimentul cu succes.

Experimentul $Priv^{cca}_{\mathcal{A},\pi}(n)$



- Pe toată durata experimentului, $\mathcal{A}$ are acces la oracolul de criptare $Enc_k(\cdot)$ și la oracolul de decriptare $Dec_k(\cdot)$ cu restricția că nu poate decripta c!

Experimentul $Priv^{cca}_{\mathcal{A},\pi}(n)$

Definiție

O schemă de criptare $\pi = (Enc, Dec)$ este CCA-sigură dacă pentru orice adversar PPT $\mathcal{A}$ există o funcție neglijabilă $negl$ așa încât

$$Pr[Priv^{cca}_{\mathcal{A},\pi}(n) = 1] \leq \frac{1}{2} + negl(n).$$

- Un adversar nu poate determina care text clar a fost criptat cu o probabilitate semnificativ mai mare decât dacă ar fi ghicit (în sens aleator, dat cu banul), chiar dacă are acces la oracolele de criptare și decriptare.

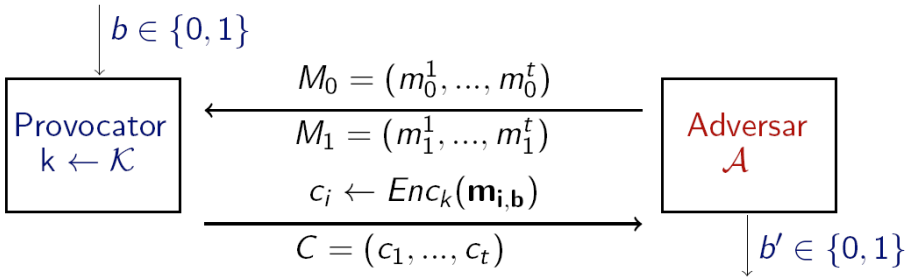
Securitate CCA

- Întrebare: Un sistem de criptare CCA-sigur este întotdeauna CPA-sigur?
- Răspuns: DA! Experimentul $Priv^{cpa}_{\mathcal{A},\pi}(n)$ este $Priv^{cca}_{\mathcal{A},\pi}(n)$ în care $\mathcal{A}$ nu folosește oracolul de decriptare.
- Întrebare: Un sistem de criptare determinist poate fi CCA-sigur?
- Răspuns: NU! Sistemul nu este CPA-sigur, deci nu poate fi CCA-sigur.

Securitate CCA - Criptare multiplă

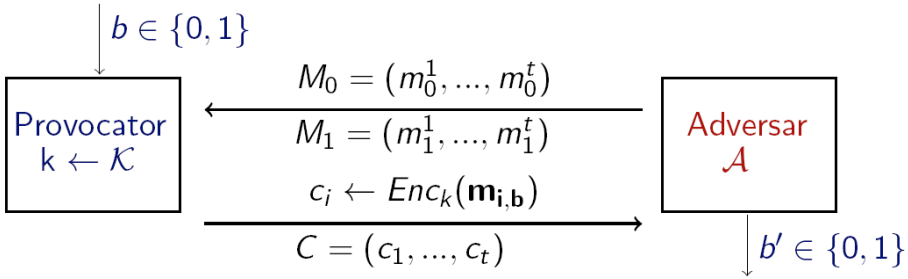
- În definiția precedentă am considerat cazul unui adversar care primește **un singur** text criptat;
- În realitate, în cadrul unei comunicații se trimit **mai multe mesaje** pe care adversarul le poate intercepta;
- Definim ce înseamnă o schemă sigură chiar și în aceste condiții.

Experimentul $Priv^{cca}_{\mathcal{A},\pi}(n)$



- Pe toată durata experimentului, $\mathcal{A}$ are acces la oracolul de criptare $Enc_k(\cdot)$ și la oracolul decriptare $Dec_k(\cdot)$ cu restricția că nu poate decripta $c_1, \ldots, c_t$!

Experimentul $Priv^{cca}_{\mathcal{A},\pi}(n)$



- Output-ul experimentului este 1 dacă $b' = b$ și 0 altfel;
- Definiția de securitate este aceeași, doar că se referă la experimentul de mai sus.
- Securitatea pentru criptare **simplă** implică securitate pentru criptare **multiplă**!

Securitate CCA

- Nici una din schemele de criptare de până acum nu sunt CCA-sigure.
- Arătăm pentru construcția anterioară, unde $Enc_k(m) = (r, F_k(r) \oplus m)$.
- Considerăm că $\mathcal{A}$ alege $m_0 = 0^n$ și $m_1 = 1^n$.
- $\mathcal{A}$ primește $c = (r, s)$, inversează primul bit al lui s și cere decriptarea textului rezultat c^* (permis deoarece $c^* \neq c$).
- Oracolul răspunde cu 10^{n-1}, și deci $b = 0$ sau cu 01^{n-1}, deci $b = 1 \Rightarrow Pr[Priv^{cca}_{\mathcal{A},\pi}(n) = 1] = 1$.
- Concluzie: orice schemă de criptare care permite ca textele criptate să fie modificate într-un mod controlat nu poate fi CCA-sigură.

Important de reținut!

- Securitate - interceptare simplă $\Rightarrow$ securitate - interceptare multiplă
- Schemele deterministe nu sunt semantic / CPA / CCA sigure
- Securitate CCA $\Rightarrow$ securitate CPA $\Rightarrow$ securitate semantică

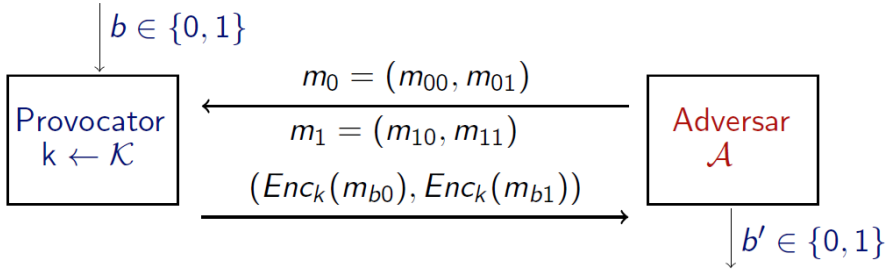
Exemplu

Fie (Enc, Dec) un sistem de criptare simetric. Se consideră sistemul de criptare (Enc', Dec') pentru mesaje de dimensiune dublă cu funcția de criptare definită astfel:

$$Enc'_k(m_1||m_2) = (Enc_k(m_2), Enc_k(m_1))$$

Arătați că sistemul nu este CCA-sigur.

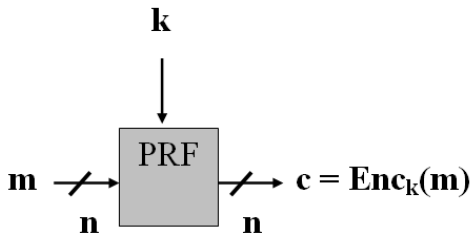
Rezolvare



$\mathcal{A}$ transmite oracolului de decriptare $(Enc_k(m_{b0}), Enc_k(m_{b0}))$ și primește $m' = (m_{b0}, m_{b0})$, deci determină b' cu probabilitate 1.

Sisteme bloc ca PRF

- Am văzut că sistemele de criptare bloc folosesc *PRF*;



Sisteme bloc ca PRF

- În criteriile de evaluare pentru adoptarea AES s-a menționat:

  The security provovided by an algorithm is the most important factor... Algorithms will be judged on the following factors...

  The extent to which the **algorithm output is indistinguishable from a random permutation on the input block**.

- Întrebare: Cum se obțin *PRF* în practică?

Paradigma confuzie-difuzie

- Se construiește funcția F, pe baza mai multor funcții aleatoare f_i de dimensiune mai mică;
- Considerăm F pe 128 biți și 16 funcții aleatoare $f_1, \ldots, f_{16}$ pe câte 8 biți;
- Pentru $x = x_1 || \ldots || x_{16}$, $x \in \{0,1\}^{128}$ $x_i \in \{0,1\}^8$:

$$F_k(x) = f_1(x_1) || \ldots || f_{16}(x_{16})$$

- Spunem că $\{f_i\}$ introduc confuzie în F.

Rețele de substituție - permutare

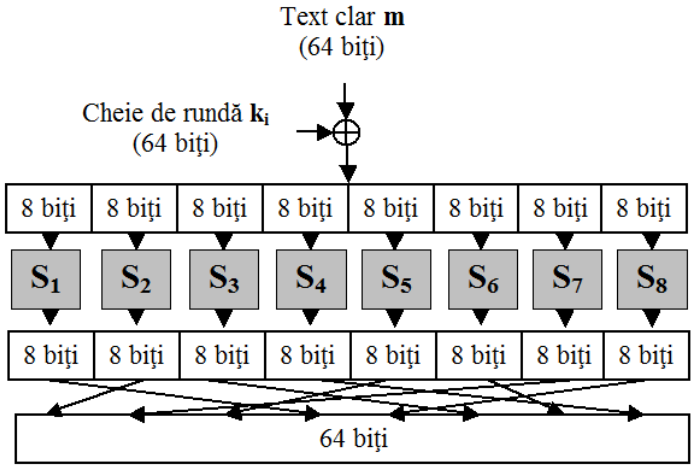
F încă nu este PRF dar

- F se transformă în PRF în 2 pași:
  - **Pasul 1**: se introduce difuzie prin amestecarea (permutarea) biților de ieșire;
  - **Pasul 2**: se repetă o rundă (care presupune *confuzie* și *difuzie*) de mai multe ori;
- Repetarea *confuziei* și *difuziei* face ca modificarea unui singur bit de intrare să fie propagată asupra tuturor biților de ieșire;

Rețele de substituție - permutare

- O rețea de substituție-permutare este o implementare a construcției anterioare de *confuzie-difuzie* în care funcțiile $\{f_i\}$ sunt **fixe** (i.e. nu depind de cheie) si se numesc permutări;
- $\{f_i\}$ se numesc S-boxes (Substitution-boxes);
- Cum nu mai depind de cheie, aceasta este utilizată în alt scop;
- Din cheie se obțin mai multe **chei de rundă** (*sub-chei*) în urma unui proces de derivare a cheilor (*key schedule*);
- Fiecare cheie de rundă este XOR-ată cu valorile intermediare din fiecare rundă.

Rețele de substituție - permutare



Rețele de substituție - permutare

- Există 2 principii de bază în proiectarea rețelelor de substituție - permutare:
  - **Principiul 1**: Inversabilitatea S-box-urilor;
    - dacă toate S-box-urile sunt inversabile, atunci rețeaua este inversabilă;
    - necesitate funcțională (pentru decriptare)
  - **Principiul 2**: Efectul de avalanșă
    - Un singur bit modificat la intrare **trebuie** să afecteze toți biții din secvența de ieșire;
    - necesitate de securitate.

Exemplu: AES - Advanced Encryption Standard

- ianuarie 1997 - NIST anunță competiția pentru selecția unui nou sistem de criptare bloc care să înlocuiască DES;
- septembrie 1997 - 15 propuneri: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent, and Twofish;
- 1998, 1999 - au loc 2 workshop-uri în urma carora ramân 5 finaliști: MARS, RC6, Rijndael, Serpent, Twofish;
- octombrie 2000 - după un al treilea workshop se anunță câștigătorul: Rijndael.
- AES este folosit in multe standarde comerciale: IPsec, TLS, IEEE 802.11i (WPA2), SSH, Skype, etc.

AES - Advanced Encryption Standard

[Google Scholar - User profiles]

[http://keccak.noekeon.org/team.html]

Rijndael = **Rij**men + **Dae**men

Descriere AES

- AES este o rețea de substituție - permutare pe 128 biți care poate folosi chei de 128, 192 sau 256 biți;
- Lungimea cheii determină numărul de runde:

Lungime cheie (biți)	128	192	256
Număr runde	10	12	14

- Folosește o matrice de octeți 4×4 numită **stare**;
- Starea inițială este mesajul clar ($4 \times 4 \times 8 = 128$);
- Starea este modificată pe parcursul rundelor prin 4 tipuri de operații: *AddRoundKey*, *SubBytes*, *ShiftRows*, *MixColumns*;
- Ieșirea din ultima rundă este textul criptat.

Securitatea sistemului AES

- Singurele atacuri netriviale sunt asupra AES cu număr redus de runde:
  - AES-128 cu 6 runde: necesită 2^{72} criptări;
  - AES-192 cu 8 runde: necesită 2^{188} criptări;
  - AES-256 cu 8 runde: necesită 2^{204} criptări.
- Nu există un atac mai eficient decât căutarea exhaustivă pentru AES cu număr complet de runde.

"It is free, standardized, efficient, and highly secure."
(J.Katz, Y.Lindell, *Introduction to Modern Cryptography*)

Rețele Feistel

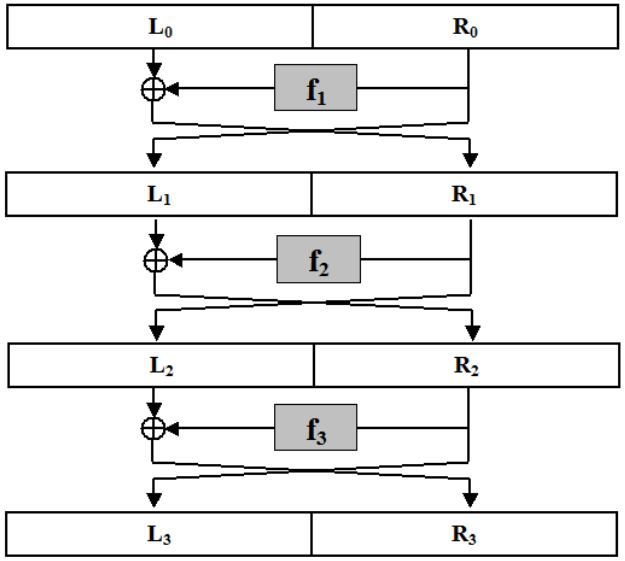
- Se aseamănă rețelelor de substituție-permutare în sensul că păstrează aceleași elementele componente: S-box, permutare, procesul de derivare a cheii, runde;
- Se diferențiază de rețelele de substituție-permutare prin proiectarea de nivel înalt;
- Introduc avantajul major că S-box-urile NU trebuie să fie inversabile;
- Permit așadar obținerea unei structuri *inversabile* folosind elemente *neinversabile*.

Horst Feistel (1915 - 1990)

[Wikipedia]

- Structurile simetrice utilizate în construcția sistemelor bloc poartă numele lui Feistel;
- Munca sa de cercetare la IBM a condus la sistemul de criptare Lucifer și mai târziu la DES.

Rețele Feistel



Rețele Feistel

- Intrarea în runda i se împarte în 2 jumătăți: L_{i-1} și R_{i-1} (i.e. *Left* și *Right*);
- Ieșirile din runda i sunt:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f_i(R_{i-1})$$

- Funcțiile f_i depind de cheia de rundă, derivând dintr-o funcție publică $\widehat{f_i}$:

$$f_i(R) = \widehat{f_i}(k_i, R)$$

Rețele Feistel

- Rețelele Feistel sunt inversabile indiferent dacă funcțiile f_i sunt inversabile sau nu;
- Fie (L_i, R_i) ieșirile din runda i;
- Intrările (L_{i-1}, R_{i-1}) în runda i sunt:

$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus f_i(R_{i-1})$$