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

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

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

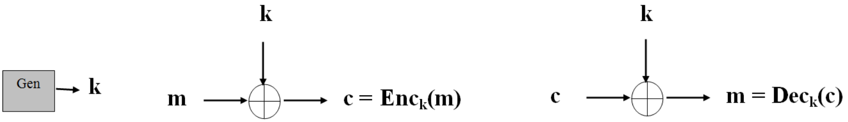
- Incercăm criptare în stil OTP: cheia va masca mesajul dar
  - masca nu va fi doar cheia ci masca $= f(\text{cheie})$ unde f este o functie de extindere a cheii

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

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**;

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);

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;

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**

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 ...

Sistem de criptare

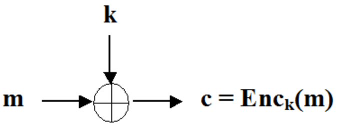
- Revenind la criptare ...
- ... aceasta presupune 2 faze:
  - **Faza 1**: se generează o secvență pseudoaleatoare de biți, folosind un generator de numere pseudoaleatoare (PRG)

Sistem de criptare

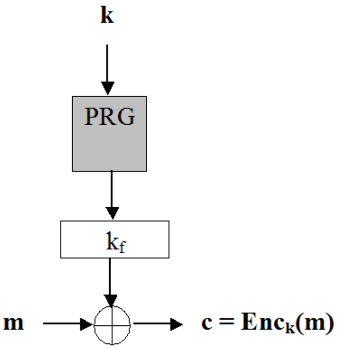
- Revenind la criptare ...
- ... aceasta presupune 2 faze:
  - **Faza 1**: se generează o secvență pseudoaleatoare de biți, folosind un generator de numere pseudoaleatoare (PRG)
  - **Faza 2**: secvența obținută se XOR-ează cu mesajul clar

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)*;

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;

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

- 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

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: $G(s) = s || \oplus_{i=1}^{n} s_i$

Exemplu

- Consideram următorul PRG: $G(s) = s || \oplus_{i=1}^{n} s_i$
- factorul de expansiune $l(n) = n + 1$

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

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 $Pr[D(G(s)) = 1] = 1$

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 $Pr[D(r) = 1] = \frac{1}{2}$

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

Observații

- Distributia output-ului unui PRG este departe de a fi uniformă
- Exemplificam pentru un G care dubleaza lungimea intrarii i.e. $l(n) = 2n$

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 ...

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 ...
- ... $\frac{1}{2^{2n}}$

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

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 ...

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

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

Observații

- Seed-ul unui PRG este analogul cheii unui sistem de criptare
- seed-ul trebuie ales uniform si mentinut secret

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;

Demonstrație intuitivă

- OTP este perfect sigur;
- Criptarea bazata pe PRG se obține din OTP prin înlocuirea *pad* cu $G(k)$;

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;

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 $\mathcal{A}$.