Capitolul 7

Sisteme de criptare cu cheie publică

7.1 Consideraţii generale

În sistemele de criptare clasice, *Alice* şi *Bob* îşi aleg o cheie secretă K care defineşte regulile de criptare (e_K) şi decriptare (d_K). În aproape toate cazurile d_K şi e_K coincideau sau se puteau deduce imediat una din alta. Astfel de sisteme sunt numite *sisteme cu cheie privată* (sau *sisteme simetrice*) deoarece publicarea lui e_K face sistemul extrem de vulnerabil.

Un punct slab al sistemelor cu cheie privată este acela că necesită o comunicare prealabilă a cheii între *Alice* şi *Bob* printr-un canal sigur, înainte de transmiterea mesajului criptat. Practic, în condiţiile cererii tot mai mari de securizare a comunicaţiilor, acest lucru este din ce în ce mai dificil de realizat.

Obiectivul sistemelor de criptare cu cheie publică este acela de a face "imposibil" (asupra acestui termen vom reveni) de obţinut cheia d_K plecând de la e_K. Astfel, regula de criptare e_K poate fi făcută publică, fiind accesibilă oricui. Avantajul constă în faptul că *Alice* (sau oricare altă persoană) poate trimite lui *Bob* un mesaj criptat cu e_K fără a intra în prealabil în contact direct. *Bob* este singura persoană capabilă să decripteze textul, utilizând informaţia sa secretă d_K.

Ideea de sistem de criptare cu cheie publică apare în 1976 şi este prezentată de Diffie şi Hellman (vezi [17])[1]. De atunci au apărut diverse astfel de sisteme, a căror securitate este bazată pe probleme calculatorii ($\mathcal{NP}$ complete). Cele mai cunoscute sisteme de criptare cu cheie publică sunt:

- Sistemul RSA: se bazează pe dificultatea descompunerii în factori primi a numerelor mari (de sute de cifre). Este sistemul cel mai larg utilizat în acest moment.

[1]Se pare că prima idee de cheie publică este schiţată în ianuarie 1970 de către britanicul James Ellis – membru în Communication-Electronic Security Group – în articolul *The possibility of non-secret encryption*. Informaţia este făcută publică de către British Government Communications Headquarters abia in 1997.

- *Sistemul El Gamal*: se bazează pe dificultatea calculului logaritmului discret într-un corp finit.

- *Sistemul Merkle - Hellman*: primul sistem definit cu cheie publică, bazat pe *problema* $\{0,1\}$ *a rucsacului.*

- *Sistemul McEliece*: este bazat pe teoria algebrică a codurilor, decodificarea unui cod liniar fiind de asemenea o problemă $\mathcal{NP}$ - completă.

- *Curbe eliptice*: Sunt sisteme de criptare care îşi desfăşoară calculele pe mulţimea punctelor unei curbe eliptice (în locul unui inel finit Z_n).

7.2 Funcţii neinversabile

O observaţie importantă este aceea că un sistem cu cheie publică nu este sigur în mod necondiţionat; oricine – putând să efectueze criptări – are posibilitatea să găsească anumite puncte slabe care să îi permită să şi decripteze mesajele. Ideea de bază folosită este aceea de *funcţie neinversabilă.* Să clarificăm puţin acest aspect.

Exemplul 7.1. *Ne putem imagina uşor străzile cu sens unic dintr-un oraş. Astfel, este uşor ca mergând pe astfel de străzi să ajungi de la punctul A la punctul B, dar este imposibil să ajungi de la B la A. În acest mod, criptarea este privită ca direcţia $A \to B$; deşi este foarte uşor de parcurs drumul în această direcţie, nu te poţi întoarce înapoi spre A (adică să decriptezi mesajul).*

Exemplul 7.2. *Să considerăm cartea de telefon a unui oraş mare*[2]*; cu ajutorul ei este foarte uşor să găsim numărul de telefon al unei anumite persoane. În schimb, este extrem de greu - practic imposibil - să afli persoana care are un anumit număr de telefon. Te afli în situaţia parcurgerii secvenţiale a (cel puţin) unui volum gros, ceea ce conduce la o creştere exagerată a timpului.*

Aceasta dă o sugestie de construcţie a unui sistem de criptare cu cheie publică. Criptarea se face independent de context, literă cu literă. Pentru fiecare literă a textului clar se alege un nume care începe cu acest caracter şi numărul de telefon al persoanei respective va constitui criptarea. Sistemul este homofonic; două apariţii diferite ale aceleiaşi litere vor fi codificate foarte probabil cu numere diferite.

De exemplu, textul clar SOLIST se poate cripta astfel:

[2]O carte de telefon expirată va duce la creşterea dificultăţii decriptării ilegale.

S	*Simion Pavel*	6394502
O	*Olaru Ştefan*	7781594
L	*Lambru Stelian*	6300037
I	*Ilie Romeo*	3134971
S	*Solovean Raluca*	6281142
T	*Tecuceanu Paul*	3359962

Deci, textul criptat va fi

639450 277815 946300 037313 497162 811423 359962.

De remarcat că metoda este nedeterministă; din acelaşi text clar se pot obţine enorm de multe texte criptate. Pe de-altă parte, orice text criptat conduce la un text clar unic.

Bob va avea la dispoziţie pentru decriptare o carte de telefon ordonată crescător după numere. Aceasta îi va permite să decripteze mesajele cu un algoritm de complexitate $\mathcal{O}(log\ n)$.

În general, o funcţie neinversabilă f trebuie să verifice două condiţii:

- Fiind dat x, $f(x)$ este uşor de calculat;
- Calculul lui x din $f(x)$ este imposibil.

De remarcat că, din punct de vedere strict matematic, nu se cunosc astfel de funcţii. A demonstra că există funcţii neinversabile este echivalent cu a demonstra relaţia $\mathcal{P} \neq \mathcal{NP}$, conjectură care stă la baza întregii teorii criptografice (a se vedea [65], [66]). De aceea, termenii folosiţi sunt relativi la complexitatea calculatorie. Astfel, o problemă este:

1. *uşoară* – dacă se poate rezolva cu un algoritm cel mult liniar;
2. *grea* – dacă se poate rezolva cu un algoritm polinomial neliniar;
3. *imposibilă* – dacă este $\mathcal{NP}$ - completă.

Am listat la început o serie de probleme $\mathcal{NP}$ - complete care stau la baza principalelor sisteme de criptare cu cheie publică.

Exemplul 7.3. *Să considerăm "problema rucsacului" (a se vedea sistemul de criptare Merkle Hellman prezentat în Capitolul* 11*). Ea constă dintr-un vector* $A = (a_1, a_2, \ldots, a_n)$ *cu* n *elemente numere întregi, pozitive, distincte, şi un număr întreg pozitiv* k*. Trebuiesc aflaţi acei* a_i *din* A *(dacă există) a căror sumă este* k*. Numele intuitiv dat problemei este evident. De exemplu, fie*

$A = (43, 129, 215, 473, 903, 302, 561, 1165, 696, 1523)$ *şi* $k = 3231$.

Se determină $3231 = 129 + 473 + 903 + 561 + 1165$*, care este o astfel de soluţie (vom da mai târziu o definiţie formală riguroasă a problemei).*

În principiu o soluţie se poate găsi parcurgând sistematic toate submulţimile lui A *şi verificând dacă suma elementelor lor este* k*. În cazul de sus, aceasta înseamnă* $2^{10} - 1 = 1023$ *submulţimi (fără mulţimea vidă), dimensiune acceptabilă ca timp de lucru.*

Ce se întâmplă însă dacă A are câteva sute de componente ? În acest caz se cunoaşte faptul că problema rucsacului este $\mathcal{NP}$ - completă.

Cu ajutorul lui A se poate defini o funcţie f astfel:

Fie $x \in [0, 2^n - 1]$; x poate fi reprezentat în binar ca un cuvânt de lungime n (completând eventual în faţă cu 0 - uri). $f(x)$ va fi numărul obţinut din A prin însumarea tuturor numerelor a_i aflate pe poziţiile marcate cu 1 în reprezentarea binară a lui x.

Formal,

$$f(x) = A \cdot B_x^T$$

unde B_x este reprezentarea binară a lui x, scrisă ca un vector coloană.

Să definim acum un sistem de criptare bazat pe problema rucsacului. Textul clar este codificat iniţial în binar şi segmentat apoi în blocuri de câte n biţi (eventual ultimul bloc este completat la sfârşit cu zerouri). Fiecare bloc rezultat este apoi criptat calculând valoarea corespunzătoare a funcţiei f.

Pentru alfabetul latin sunt suficienţi 5 biţi pentru codificarea binară a literelor şi a spaţiului. Mai exact, dacă asociem literelor A - Z reprezentările binare ale numerelor $1 - 26$, vom avea:

	–	00000	A	–	00001	B	–	00010
C	–	00011	D	–	00100	E	–	00101
F	–	00110	G	–	00111	H	–	01000
I	–	01001	J	–	01010	K	–	01011
L	–	01100	M	–	01101	N	–	01110
O	–	01111	P	–	10000	Q	–	10001
R	–	10010	S	–	10011	T	–	10100
U	–	10101	V	–	10110	W	–	10111
X	–	11000	Y	–	11001	Z	–	11010

Să considerăm un text clar; FLOARE DE COLT de exemplu. Cum fiecare caracter se codifică în 5 biţi, în fiecare bloc intră două caractere:

FL OA RE _D E_ CO LT.

Codificând cu ajutorul vectorului rucsac A definit anterior, se obţin şapte blocuri de câte 10 biţi:

0011001100 0111100001 1001000101 0000000100 0000000101 0001101111 0110010100

care conduc la textul criptat:

$$(2414, 3243, 3204, 1165, 1118, 5321, 1811).$$

Să considerăm sistemul de criptare definit în Exemplul 7.3. Dacă îl privim ca un sistem clasic (cu cheie privată), criptanalistul trebuie să afle vectorul de bază A şi apoi să rezolve problema rucsacului.

Dacă el foloseşte un atac cu text clar ales, îl va afla uşor pe A: este suficient să trimită n texte clare cu cîte un singur 1 iar restul 0. Problema apare în momentul rezolvării problemei rucsacului; aici atât *Bob* cât şi *Oscar* sunt puşi în faţa acelaiaşi probleme $\mathcal{NP}$ - complete. Ori, practic, doar *Oscar* trebuie să rezolve o problemă dificilă, nu şi *Bob*.

O altă problemă ridicată de acest sistem de criptare: este obligatoriu ca un text criptat să determine în mod unic un text clar. Aceasta înseamnă că nu trebuie să existe două submulţimi ale lui A care să aibă aceeaşi sumă. Astfel, dacă se ia $A = (17, 103, 50, 81, 33)$, textul criptat $(131, 33, 100, 234, 33)$ poate fi decriptat în două moduri: SAUNA şi FAUNA.

7.3 Trapa secretă

Pentru ca *Bob* să nu fie pus în aceaşi situaţie ca şi *Oscar*, el trebuie să dispună de un procedeu care să îi permită să transforme problema $\mathcal{NP}$ - completă publică, într-o problemă uşoară. Acest procedeu este numit *trapă secretă*. În primul exemplu, trapa secretă era cartea de telefon ordonată după numerele de telefon, nu după abonaţi. Să vedem care este trapa secretă în sistemul de criptare definit în Exemplul 7.3:

Exemplul 7.4. *Sunt clase de probleme ale rucsacului uşor de rezolvat; una din ele o formează vectorii cu creştere mare.*

Spunem că vectorul rucsac $A = (a_1, a_2, \ldots, a_n)$ *este super-crescător dacă*

$$\forall\, j \geq 2, \quad a_j > \sum_{i=1}^{j-1} a_i.$$

În acest caz, pentru a rezolva problema rucsacului este suficient să parcurgem vectorul A de la dreapta spre stânga. Cunoscând valoarea k, cercetăm întâi valoarea de adevăr a relaţiei $k \geq a_n$. *Dacă răspunsul este FALSE,* a_n *nu poate aparţine sumei pe care o căutăm. Dacă însă se obţine TRUE,* a_n **trebuie** *să fie în sumă, deoarece toate elementele* a_i *rămase nu pot depăşi în sumă pe k.*

Vom defini

$$k_1 = \begin{cases} k & \text{dacă} \quad a_n > k \\ k - a_n & \text{dacă} \quad a_n \leq k \end{cases}$$

şi repetăm procedeul pentru k_1 *şi* a_{n-1}. *Algoritmul se va opri la valoarea* a_1.

Să presupunem că avem vectorul rucsac super-crescător

$$A = (1, 3, 5, 11, 21, 44, 87, 175, 349, 701)$$

şi vrem să decodificăm mesajul 278. *Vom parcurge* 10 *paşi, sumarizaţi în Tabelul* 1.

Număr	*Componenta lui A*	*Bit rezultat*
278	701	0
278	349	0
278	175	1
103	87	1
16	44	0
16	21	0
16	11	1
5	5	1
0	3	0
0	1	0

Tabelul 1*:*

În final se obţine secvenţa binară 00110 01100 *(aflată pe ultima coloană) care - conform codificării din Exemplul 7.3 corespunde perechii de litere* FL*.*

Dacă se foloseşte însă public o astfel de informaţie, orice utilizator – inclusiv *Oscar* – poate decripta mesajele folosind un algoritm liniar. Ori s-a presupus (Capitolul 1) că, pentru orice intrus, încercarea de aflare a mesajului clar trebuie să conducă la rezolvarea unei probleme $\mathcal{NP}$ - complete.

Exemplul 7.5. *Pentru sistemul bazat pe problema rucsacului, Bob va proceda astfel: va alege un număr* m $(m > \sum_{i+1}^{m} a_i)$ *numit* **modul** *şi un număr* t*,* $cmmdc(m,t) = 1$ *numit* **multiplicator***. Există atunci un număr* s *astfel ca* $t \cdot s \equiv 1 \ (mod\ m)$*.*

Plecând de la vectorul super-crescător $A = (a_1, a_2, \ldots, a_n)$ *Bob generează vectorul* $B = (b_1, b_2, \ldots, b_n)$ *unde* $b_i = t \cdot a_i \ (mod\ m)$*.*

Vectorul B *este declarat public pentru criptare, iar* m, t *şi* s *vor forma trapa secretă a lui Bob.*

Astfel, dacă luăm $m = 1590$ *şi* $t = 43$*, vectorul super-crescător*

$$A = (1, 3, 5, 11, 21, 44, 87, 175, 349, 701)$$

devine

$$B = (43, 129, 215, 473, 903, 302, 561, 1165, 697, 1523),$$

adică cel prezentat în Exemplul 7.3. În plus, $s = t^{-1} = 37$*.*[3]

Cum se va proceda: Cel care doreşte să trimită lui Bob un mesaj criptat va folosi vectorul rucsac B *şi va cripta mesajul* x *în* $y = B \cdot B_x^T$*, conform Exemplului 7.3.*

La recepţie, Bob va calcula întâi $z = s \cdot y \ (mod\ m)$*, după care va decripta mesajul* z *folosind vectorul super-crescător* A*. Se poate arăta uşor că soluţia este chiar* x*.*

Astfel, de exemplu Alice poate cripta mesajul FL *în* 2414 *(cf. Exemplului 7.3). La primirea acestui număr, Bob va determina întâi* $s \cdot 2414 = 37 \cdot 2414 \ (mod\ 1590) = 278$*. În Exemplul 7.4 s-a văzut că decriptarea mesajului* 278 *cu vectorul* A *conduce la textul clar* FL*.*

[3]Pentru calculul inversului unui număr se poate folosi algoritmul lui Euclid extins, prezentat în Anexă.

Putem trasa acum câteva principii generale de construire a unui sistem de criptare cu cheie publică ([45]):

1. Se începe cu o problemă dificilă P; rezolvarea lui P este imposibilă în conformitate cu teoria complexităţii (nu se cunoaşte nici un algoritm determinist de complexitate polinomială care să rezolve P).

2. Se selectează o subproblemă P_1 a lui P, rezolvabilă în timp polinomial (preferabil liniar).

3. Se aplică o transformare problemei P_1 astfel încât să se obţină o problemă P_2 care să nu semene cu P_1 dar să fie foarte apropiată de problema P.

4. Se face publică problema P_2 şi se descrie algoritmul de criptare bazat pe aceasta. Informaţia referitoare la modul în care se obţine P_1 din P_2 este o trapă secretă.

5. Se construiesc detaliile sistemului de criptare, astfel încât principiile de lucru să difere esenţial pentru destinatar faţă de criptanalist; astfel, în timp ce primul va folosi trapa secretă şi va rezolva problema P_1, al doilea va trebui să rezolve problema P_2, imposibilă datorită asemănării ei cu problema P.

În funcţie de aceste principii generale, apar în detalii de construcţie multe alte probleme pe care constructorii sistemelor de criptare trebuie să le rezolve.

7.4 Securitatea sistemelor de criptare cu cheie publică

În majoritatea sistemelor de criptare, aparatul matematic folosit este bazat pe teoria numerelor, teoria funcţiilor recursive şi teoria probabilităţilor. Pe o scară mult mai restrânsă apar funcţiile eliptice, teoria automatelor, calcul neconvenţional (cuantic, molecular etc).

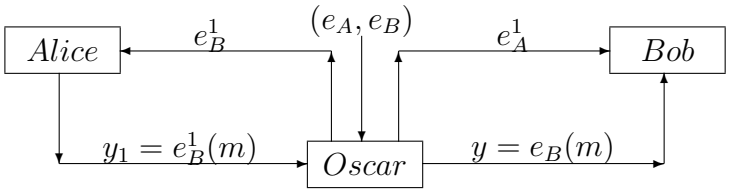
Sistemele de criptare cu cheie publică au un avantaj major faţă de sistemele clasice: aici nu mai este necesar efortul transmiterii cheii. Un contact prealabil între *Alice* şi *Bob* pentru a pune la punct detaliile sistemului de criptare este inutil.

Un sistem de criptare cu cheie publică nu oferă însă o securitate absolută. Aceasta se datorează faptului că *Oscar* are oricând posibilitatea să lanseze atacuri pasive sau active. Anume:

- Dacă un text criptat y este interceptat de criptanalist, acesta poate căuta exhaustiv un text clar x astfel ca $e_K(x) = y$. Singura apărare contra unui astfel de atac constă în gradul de complexitate al sistemului.

- Un criptanalist activ poate efectua cu succes un atac numit *man-in-the-middle* (a nu se confunda cu *meet-in-the-middle* prezentat în capitlul anterior).

  Să presupunem că *Alice* şi *Bob* doresc să stabilească un contact. Ei fac publice cheile de criptare e_A respectiv e_B. Dacă contactul este nepersonalizat, *Oscar* poate controla mesajele schimbate între cei doi, în felul următor:



  1. *Oscar* "opacizează" printr-un mijloc oarecare aceste chei, şi trimite lui *Alice* cheia e_B^1 ca din partea lui *Bob*; substituie – similar – pentru *Bob* cheia e_A cu e_A^1.
  2. Fie m mesajul pe care *Alice* vrea să îl trimită lui *Bob*. Ea va cripta şi va trimite $y_1 = e_B^1(m)$.
  3. *Oscar* interceptează mesajul (reamintim, toate canalele sunt nesigure) şi află $m = d_B^1(y_1)$.
  4. *Oscar* recriptează $y = e_B(m)$ şi trimite y lui *Bob*.

  Bineînţeles, dacă doreşte, *Oscar* poate modifica, interschimba sau întârzia mesajele interceptate.

Din această cauză, în toate sistemele de criptare cu cheie publică apare necesitatea autentificării mesajului sau a expeditorului, precum şi aceea a confidenţialităţii.

Definiţia 7.1. *Confidenţialitatea*[4] *asigură accesul la informaţie doar părţilor autorizate de a avea acest acces.*

Definiţia 7.2. *Autentificarea*[5] *este procesul prin care un calculator (program de calculator sau alt utilizator) încearcă să confirme unui destinatar că mesajul primit de acesta vine (sau nu vine) din partea sa.*

Metodele prin care se poate autentifica un expeditor uman, sunt clasificate în:

1. "*ceva ce utilizatorul este*" (de exemplu amprente digitale, de retină, de voce, secvenţă DNA, recunoaşterea semnăturii, identificatori biometrici).
2. "*ceva de utilizatorul are*" (de exemplu card ID, date de securitate soft aflate pe calculator sau telefon).

[4]conform International Standards Organization (*ISO*).
[5]de la grecescul "authentes" – autor.

3. "*ceva ce utilizatorul ştie*" (de exemplu un password, o parolă, un număr de identificare - PIN).

4. Orice combinaţie între metodele anterioare (de exemplu un card bancar cu PIN asigură o dublă autentificare).

Alt termen frecvent utilizat este cel de **integritate**. El se referă la validitatea datelor.

Definiţia 7.3. *Integritatea este siguranţa că datele la care se referă un utilizator pot fi accesate şi eventual modificate numai de cei autorizaţi să o facă.*

În general integritatea poate fi compromisă în două moduri:

1. Prin alterare intenţionată (de exemplu modificarea unui cont bancar, a unei adrese de e-mail, a unui document de identitate);

2. În mod accidental (transmisii perturbate de zgomote de canal, zgârierea harddiscului)[6].

Să presupunem că *Alice* şi *Bob* sunt doi utilizatori, cu posibile conflicte de interese. Când *Alice* trimite un mesaj lui *Bob*, ambele părţi trebuie să se asigure că:

- Mesajul nu este trimis de o terţă persoană care pretinde a fi *Alice*;

- *Bob* să nu poată obliga pe *Alice* să ţină cont de mesaje care nu-i aparţin, iar *Alice* să nu poată repudia propriile mesaje.

Într-o oarecare măsură, cele două condiţii sunt contradictorii: conform primei condiţii, *Bob* trebuie să ştie ceva despre modul de criptare al lui *Alice*, care îi va permite să autentifice mesajul, iar conform celei de-a doua condiţii, el nu trebuie să ştie prea mult. O modalitate frecvent utilizată pentru autentificarea mesajelor este folosirea codurilor de autentificare.

Exemplul 7.6. *MAC-ul (Message Authentication Code) definit în cadrul sistemului de criptare DES este o variantă prin care se poate asigura atât autenticitatea cât şi integritatea mesajului.*

Dacă se solicită şi autentificarea partenerilor, atunci se foloseşte de obicei semnătura electronică.

[6]De recuperarea informaţiei pierdută în acest mod se ocupă Teoria Codurilor detectoare şi corectoare de erori.

Exemplul 7.7. *Să presupunem că Alice vrea să trimită lui Bob mesajul m. Dacă se foloseşte un sistem de criptare cu cheie publică în care funcţiile de criptare/decriptare sunt comutative, iar $(e_A, d_A), (e_B, d_B)$ sunt perechile (cheie publică, cheie privată) ale celor doi parteneri, ei pot urma următorul protocol:*

1. *Alice trimite lui Bob $y_1 = e_A(m)$;*
2. *Bob trimite lui Alice $y = e_B(y_1)$;*
3. *Alice trimite lui Bob $d_A(y) = e_B(m)$;*
4. *Bob calculează $d_B(e_B(m)) = m$ şi află mesajul.*

Se observă că sunt verificate cele două condiţii de autentificare şi – în plus – protocolul rezistă unui atac de tip man-in-the-middle.

Dacă dorim să folosim un singur contact, Alice poate trimite mesajul $y = e_B(d_A(m))$. La recepţie, Bob va folosi propria sa cheie pentru decriptare, impreună cu cheia publică a lui Alice. Metoda merge şi pentru sisteme de criptare necomutative.

7.5 Comparaţie între criptarea simetrică şi cea cu cheie publică

Avantaje ale sistemelor de criptare cu cheie simetrică:

1. Pot transmite volume mari de date. Există implementări hard care pentru unele sisteme de criptare asigură rate de criptare de sute de mega-octeţi pe secundă (sunt şi implementări soft cu rate de mega-octeţi pe secundă).
2. Cheile sunt relativ scurte.
3. Pot fi folosite ca bază de construcţie a diverselor mecanisme de criptare, cum ar fi generatori de numere pseudo-aleatoare, generatori de funcţii de dispersie, scheme de semnătură.
4. Prin compunere pot conduce la sisteme de criptare puternice.
5. Au o istorie bogată în evenimente şi experienţă.

Dezavantaje ale sistemelor de criptare cu cheie simetrică:

1. Cheia trebuie să rămână permament secretă în (cel puţin) două locuri distincte.
2. Cu cât lungimea unui mesaj criptat este mai mare, cu atât el este mai uşor de spart.
3. În reţele mari, o gestionare a cheilor devine extrem de dificilă.
4. Necesită un canal sigur de comunicare, cel puţin pentru transmiterea cheii. Acest lucru devine dificil mai ales pentru sistemele care necesită schimbări frecvente ale cheilor de criptare/decriptare.

Avantaje ale sistemelor de criptare cu cheie publică:

1. Sistemul este ideal pentru transmiterea informaţiei prin canale nesigure.
2. Sistemele cu cheie publică sunt simplu de definit şi elegante matematic.
3. Doar cheia de decriptare trebuie ţinută secretă, la un singură adresă (destinatar).
4. În funcţie de modul de utilizare, o pereche de chei (publică, privată) poate fi păstrată o perioadă mai lungă de timp.
5. Conduc la aplicaţii de mare întindere: semnături electronice, algoritmi de autentificare, componente de comerţ electronic etc.

Dezavantaje ale sistemelor de criptare cu cheie publică:

1. Sunt semnificativ mai lente decât sistemele simetrice.
2. Sunt necesare chei de lungimi mult mai mari.
3. Nu se poate garanta securitatea absolută a nici unei scheme de criptare cu cheie publică.
4. Implementarea trebuie realizată cu foarte mare grijă. Sisteme cu grad teoretic ridicat de securitate pot fi sparte uşor printr-o implementare neglijentă.

După cum se observă, cele două clase de sisteme de criptare dispun de o serie de avantaje complementare. Acest lucru face ca ele să fie folosite combinat.

Exemplul 7.8. *Multe sisteme de criptare încep comunicarea transmiţând via un sistem cu cheie publică, cheia unui sistem simetric. În faza a doua, mesajele sunt criptate folosind sistemul simetric de criptare. Aceasta asigură o viteză mult mai mare de transmitere şi un spor de autenticitate a mesajelor.*

7.6 Exerciţii

7.1. *Justificaţi modul în care un sistem de criptare simetric asigură condiţiile de confidenţialitate, autentificare şi integritate.*

7.2. *Construiţi exemple de funcţii neinversabile.*

7.3. *Fie $f(x)$ şi $g(x)$ două funcţii neinversabile. Daţi un argument auristic pentru a arăta că nici una din funcţiile $f(x)+g(x)$, $f(x)\cdot g(x), f(g(x))$ nu este în mod obligatoriu neinversabilă.*

7.4. *Alice şi Bob aleg public un nunăr prim p. După aceea, fiecare alege şi păstrează secret câte două numere (e_A, d_A) respectiv (e_B, d_B) astfel ca $e_X \cdot d_X \equiv 1 \ (mod\, p-1)$.*

Dacă Alice doreşte să transmită lui Bob mesajul m, se va urma protocolul:

1. *Alice trimite lui Bob textul $x = m^{e_A}$;*
2. *Bob răspunde cu mesajul $y = x^{e_B}$;*
3. *Alice trimite înapoi mesajul $z = y^{d_A}$.*

Arătaţi că Bob este capabil să decripteze mesajul şi discutaţi problemele de securitate implicate.

Bibliografie

[1] Anderson R. ş.a. - *Serpent: A proposal for the Advanced Encryption Standard*, http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/serpent.pdf

[2] Atanasiu A. - *Teoria codurilor corectoare de erori*, Editura Univ. Bucureşti, 2001;

[3] Atanasiu, A. - *Arhitectura calculatorului*, Editura Infodata, Cluj, 2006;

[4] Blum L., Blum M., Shub M. - *Comparision of two pseudo-random number generators*, Advanced in Cryptology, CRYPTO 82

[5] D. Bayer, S. Haber, W. Stornetta; Improving the efficiency and reliability of digital time-stamping. Sequences II, Methods in Communication, Security and Computer Science, Springer Verlag (1993), 329-334.

[6] Biham E., Shamir A. - *Differential Cryptanalysis of DES - like Cryptosystems*, Journal of Cryptology, vol. 4, 1 (1991), pp. 3-72.

[7] Biham E., Shamir A. - *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[8] Biham E., Shamir A. - *Differential Cryptanalysis of the Full 16-Round DES*, Proceedings of Crypto92, LNCS 740, Springer-Verlag.

[9] Biham E. - *On Matsuis Linear Cryptanalysis*, Advances in Cryptology - EUROCRYPT 94 (LNCS 950), Springer-Verlag, pp. 341-355, 1995.

[10] Biryukov A., Shamir A., Wagner D. - *Real Time Cryptanalysis of A5/1 on a PC*, Fast Software Encryption - FSE 2000, pp 118.

[11] Bruen A., Forcinito M - *Cryptography, Information Theory, and Error - Correction*, Wiley Interscience 2005.

[12] Brigitte Collard - *Secret Language in Graeco-Roman antiquity* (teză de doctorat) $http://bcs.fltr.ucl.ac.be/FE/07/CRYPT/Intro.html$

[13] Cook S., *http : //www.claymath.org/millennium/P_vs_NP/Official_Problem_Description.pdf*

[14] Coppersmith D. ş.a. - *MARS - a candidate cypher for AES*,
http://www.research.ibm.com/security/mars.pdf

[15] Daemen J., Rijmen V. - *The Rijndael Block Cipher Proposal*,
http://csrc.nist.gov/CryptoToolkit/aes/

[16] Damgard I.B. - *A design principle for hash functions*, Lecture Notes in Computer Science, 435 (1990), 516-427.

[17] Diffie D.W., Hellman M.E. - *New Directions in Cryptography*, IEEE Transactions on Information Theory, IT-22, 6 (1976), pp. 644-654

[18] Diffie D.W., Hellman M.E. - *Multiuser cryptographic techniques*, AFIPS Conference Proceedings, $45(1976), 109 - 112$

[19] L´ Ecuyer P. - *Random Numbers for Simulation*, Comm ACM 33, 10(1990), 742-749, 774.

[20] Enge A. - *Elliptic Curves and their applications to Cryptography*, Kluwer Academic Publ, 1999

[21] El Gamal T. - *A public key cryptosystem and a signature scheme based on discrete algorithms*, IEEE Transactions on Information Theory, 31 (1985), 469-472

[22] Fog A. - *http://www.agner.org/random/theory*;

[23] Gibson J. - *Discrete logarithm hash function that is collision free and one way.* IEEE Proceedings-E, 138 (1991), 407-410.

[24] Heyes H. M. - *A Tutorial on Linear and Differential Cryptanalysis.*

[25] van Heyst E., Petersen T.P. - *How to make efficient fail-stop signatures*, Lecture Notes in Computer Science, $658(1993), 366 - 377$

[26] Junod P. - *On the complexity of Matsuiş attack*, in SAC 01: Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography, pp 199211, London, UK, 2001. Springer-Verlag.

[27] Kahn D. - *The Codebreakers*, MacMillan Publishing Co, New York, 1967

[28] Kelly T. - *The myth of the skytale*, Cryptologia, Iulie 1998, pp. 244 - 260.

[29] Konheim A. - *Computer Security and Cryptography*, Wiley Interscience, 2007.

[30] Knuth D. - *The art of computer Programming*, vol 2 (Seminumerical Algorithms)

[31] Lenstra, H.W. - *Factoring Integers with Eiipltic Curves*, Annals of Mathematics, vol. 126, pp. 649-673, 1987.

[32] Matsui M, Yamagishi A. - *A new method for known plaintext attack of FEAL cipher.* Advances in Cryptology - EUROCRYPT 1992.

[33] Matsui M. - *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology - EUROCRYPT 93, LNCS 765, Springer-Verlag, pp. 386-397, 1994.

[34] Matsui M. - *The first experimental cryptanalysis of the Data Encryption Standard*, in Y.G. Desmedt, editor, Advances in Cryptology - Crypto 4, LNCS 839, SpringerVerlag (1994), 1- 11.

[35] Matsui M. - *New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptalaysis*, Fast Software Encryption, LNCS 1039, Springer-Verlag, 1996, pp. 205-218.

[36] Merkle R. C., Hellman M. - *Hiding Information and Signatures in Trapdoor Knapsacks*, IEEE Trans. IT 24(5), Sept 1978, pp. 525530.

[37] Merkle R.C. - *A fast software one-way functions and DES*, Lecture Notes in Computer Science, 435 (1990), 428-446

[38] Menezes A., Oorschot P., Vanstome S. - *Handbook of Applied Cryptography*, CRC Press 1996.

[39] Preneel B., Govaerts R., Vandewalle J. - *Hash functions based on block ciphers: a syntetic approach*; Lecture Notes in Computer Science, 773 (1994), 368-378

[40] Rivest R. ş.a - *The RC6^{TM} Block Cipher*,
ftp://ftp.rsasecurity.com/pub/rsalabs/rc6/rc6v11.pdf

[41] Rivest R.L. - *The* **MD4** *message digest algorithm*; Lecture Notes in Computer Science, 537, (1991), 303-311

[42] Rivest R., Shamir A., Adleman A. - *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM, Vol. 21 (2), 1978, pages 120–126.

[43] Rosing, M - *Implementing Elliptic Curve Cryptography*, Manning, 1998

[44] D. Salmon - *Data Privacy and Security*, Springer Professional Computing, 2003

[45] Salomaa A. - *Criptografie cu chei publice*, Ed. Militară, Bucureşti 1994

[46] Schneier B. - *Applied Cryptography*, John Wiley and Sons, 1995

[47] Schneier B ş.a. - *Twofish*, http://www.counterpane.com/twofish.html

[48] Shamir, A. - *A polynomial time Algorithm for breaking the basic Merkle - Hellman cryptosystem*,

http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/C82/279.PDF

[49] Shoup, V. - *Lower bounds for discrete logarithm and related problems*, Advanced in Cryptology, EUROCRYPT 97, Springer - Verlag LNCS 1233, pp. 313-328, 1997.

[50] Selmer E.S. - *Linear Recurrence over Finite Field*, Univ. of Bergen, Norway, 1966;

[51] Sibley E.H. - *Random Number Generators: Good Ones are Hard to Find*, Comm ACM 31, 10(1988), 1192-1201.

[52] Smid M.E., Branstad, D.K. - *Response to comments on the NIST proposed digital signature standard*, Lecture Notes in Computer Science, $740(1993), 76 - 88$

[53] Stinton D., *Cryptography, Theory and Practice*, Chapman& Hall/CRC, 2002

[54] Wiener M.J. - *Cryptanalysis of short RSA secret exponents*, IEEE Trans on Information Theory, 36 (1990), 553-558

[55] Williams H.C. - *Some public-key criptofunctions as intractable as factorisation*, Cryptologia, 9 (1985), 224-237.

[56] Zeng K.G., Yang C.H., Wei D.Y., Rao T.R.N.- *Pseudorandom Bit Generators in Stream Cipher Cryptography*, IEEE Computer, 24 (1991), 8.17.

[57] *Secure hash Standard*; National Bureau of Standards, FIPS Publications 180, 1993

[58] $http://en.wikipedia.org/wiki/Enigma\_machine$

[59] $http://en.wikipedia.org/wiki/M - 209$

[60] *http://en.wikipedia.org/wiki/Caesar_ cipher# History_ and_ usage*

[61] *http://psychcentral.com/psypsych/Polybius_ square*

[62] *http://www.answers.com/topic/vigen-re-cipher*

[63] *http://en.wikipedia.org/wiki/Rosetta_ stone*

[64] *Serpent homepage*, http://www.cl.cam.ac.uk/ rja14/serpent.html

[65] *P versus NP homepage*, http://www.win.tue.nl/ gwoegi/P-versus-NP.htm

[66] http://www.win.tue.nl/ gwoegi/P-versus-NP.htm

[67] http://en.wikipedia.org/wiki/Complexity_ classes_ P_ and_ NP