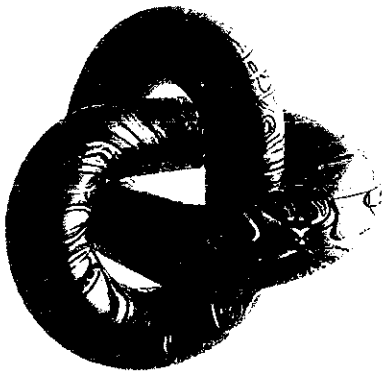


CRIPTAREA DATELOR

Gabriela- Anca PANAIT

Dan-Cristian ȚACU

LIMBAJE DE COMUNICARE

Majoritatea oamenilor de ştiinţă: lingvişti, filozofi, logicieni, specialişti în informatică şi cibernetică, admit că principala funcţie a limbii o reprezintă comunicarea şi deci transmiterea de informaţii dar odată cu acestea sunt transmise şi diferite cunoştinţe. Limbajul natural a fost izvorul din care s-au născut limbajele: literar, istoric, poetic, filozofic, logic, ştiinţific şi în final limbajele formalizate şi convenţionale şi respectiv limbajele utilizate la programarea calculatoarelor.

Informaţie şi limbaj

Savanţii acceptă în prezent ideea că informaţia este o un concept primar, care poate fi transmisă cu un minim de energie, dar cantitatea de informaţie nu depinde de valoarea acestui minim. Teoria informaţiei în prezent este considerată ca un domeniu de sine stătător, cu toate că are o istorie relativ scurtă.

Limbajul natural

Pentru a transmite o cantitate de informaţie în condiţii optime, semnalul trebuie să fie organizat după anumite principii. Practica transmiterii informaţiei este cea care impune măsuri suplimentare de redundanţă, care să limiteze perturbaţiile date de sursele de zgomote.

In plus apar intonaţiile, apoi la operele literare metafore, comparaţii, sinonime, etc, care duc la o transmitere specială a informaţiei.

Limbajul natural este unul mai direct, mai expresiv, mai plastic, mai concret, dar în acelaşi timp mai confuz, mai particular, mai intuitiv.

Din această cauză, limbajul natural nu poate fi înţeles până în prezent de calculator. In prezent se fac cercetări pentru cunoaşterea cuvintelor cele mai uzuale transmise oral sau în scris (transmise prin scanare sau microfonie, nu cele transmise normal prin tastatura calculatorului), de către calculator.

Limbajul filozofic

Cugetarea filozofică reprezintă şi ea o experienţă umană, dar având o altă optică. Experienţa filozofică se orientează pe alte căi şi anume caută cauza reală a lucrurilor şi căutând explicaţii complete şi fundamentale. O întrebare care a frământat filozofii de-a lungul timpului, a fost determinarea cauzei tuturor lucrurilor din univers şi geneza acestora. Limbajul filozofic este aparent natural, dar în realitate este unul criptic în care parabola, supoziţia şi simbolul joacă un rol decisiv. Pe parcurs, mai ales în ultimul secol, limbajul filozofic s-a orientat în mare parte spre un limbaj logic. Cu toate acestea nici acest limbaj nu poate fi folosit în discuţia cu calculatorul sau pentru transmiterea unor comenzi de programare.

Limbajul clasic al logicii

Crearea unui limbaj abstract a fost necesar după ce s-a constatat că limbajul natural are practic probleme în transmiterea şi analiza noţiunilor din domeniul cunoaşterii ştiinţifice. Aceasta deoarece în limbajul natural cuvintele exprimă anumite obiecte şi stări, dar în limbajul ştiinţific se lucrează şi cu noţiuni.

Astfel noţiunea poartă ca semnificaţie, un conţinut nu al indivizilor unei clase, ci al clasei formate dintr-o mulţime de indivizi. Câtă vreme cuvântul exprimă obiecte sau stări, este cuvânt, când exprimă clase de obiecte, devine noţiune.

Prin judecăţi şi silogisme, limbajul logic suplineşte aptitudinea limbajului natural de a exprima adecvat formele dinamice ale realităţii. Judecata în limbajul logic este o aserţiune la modul indicativ, apoi judecata logică este o expresie a timpului prezent şi în fine judecata logică a devenit expresia unei simple relaţii de apartenenţă a predicatului la subiect, apartenenţă modificată pe parcurs prin raportul de existentă.

Judecata logică clasică permite doar exprimarea indicativului prezent, a relaţiei de apartenenţă sau existenţă, afirmaţia sau negaţia, apoi totalitatea, parţialitatea şi singularitatea raporturilor dintre subiect şi predicat, la forma impersonală.

Limbajul ştiinţific

Limbajul ştiinţific este un limbaj obiectiv, impersonal, atemporal, aspaţial şi amodal. El descrie fapte şi relaţii între acestea, delimitând cu precizie obiectele fizice de cele logice şi faptele certe de cele incerte.

Apoi, limbajul ştiinţific este sobru în descrieri, schematic în demonstraţii, tinde să se matematizeze, mai ales în ultimele decenii când o mare parte din gândirea de rutină a fost transferată calculatoarelor.

Limbajul logico-matematic este unul abstract, folosind o gamă largă de relaţii şi renunţând la noţiuni, din pricina conţinutului lor ontologic. Noţiunea logico-matematică a devenit un simbol al oricărui obiect.

Transferul de informaţie în sistemele axiomatice

Ştiinţa începând cu Aristotel, are trei părţi distincte: cunoaştere, exprimare şi demonstraţie.

In multe sisteme axiomatice moderne, axiomele şi deci şi teoremele nu sunt expresia unor adevăruri reale, ci doar a unor situaţii convenţionale care se găsesc într-o strictă şi riguroasă dependentă fată de axiomele date, prin intermediul unor reguli de deducţie stabilite.

In sistemele axiomatice moderne, transferul de informaţie se face respectând doar corespondenţa dintre mecanismul de gândire şi regulile impuse pentru desfăşurarea lui corectă. Dacă gândirea se desfăşoară mecanic în conformitate strictă cu regulile care o comandă în sistemul respectiv, sistemul devine un transmiţător de informaţii.

Limbajele de programare

Din punct de vedere semiotic, majoritatea specialiştilor susţin că unele dintre limbajele de programare sunt mai apropiate de cele logice, pe când altele mai apropiate de cele naturale.

Cronologic, limbajul în cod maşină, a fost primul limbaj de programare utilizat. Pornindu-se de la descrierea semantică a algoritmului, programul cuprinde un sir finit de instrucţiuni, redactate sub forma unor secvenţe cu caractere binare. Aceasta comportă un efort foarte mare din partea programatorilor, şi în prezent este utilizată doar la programarea unor microprocesoare pentru automatele simple.

Astăzi, o mare parte din etapele necesare în programarea în limbajul maşină au fost transferate calculatorului, prin crearea unor tipuri de limbaje de programare. Aceste limbaje sunt recunoscute de calculator care este utilat cu anumite programe în acest sens, denumite compilatoare, care le transformă în limbaj maşină.

In ultimii 40 de ani, au fost realizate un număr impresionat de limbaje de programare printre care putem enumera:

- FORTRAN – un limbaj universal, unul din cele mai folosite limbaje în anii 70.
- COBOL – limbaj folosit pentru manipularea de date, operaţii contabile;
- BASIC un limbaj folosit de masa mare a utilizatorilor, nespecialişti în programare.
- PROLOG limbaj logic folosit pentru programe expert din diferite domenii: medicină (pentru diagnostic şi recomandări tratament), geologie, procese tehnologice, etc.;
- C++ limbaj orientat pe obiecte ce asigura stocarea datelor din realitate, instrumente de programare şi de modelare a unor situaţii din lumea reala
- JAVA este un limbaj modern, orientat spre obiecte şi este utilizat pentru paginile Web, fiind bine protejat împotriva viruşilor.

Date şi informaţii

Pentru a deveni informaţii, datele privitoare la obiectul de activitate trebuie prelucrate în concordantă cu cerinţele informaţionale, adică culegerea datelor şi prelucrarea lor şi apoi distribuirea lor la factorii de decizie.

Deci:

- **datele** privesc evenimentele primare, colectate pentru informare sau rezolvarea unor probleme sau situaţii;

- **informaţiile** sunt mesaje obţinute prin prelucrare datelor, calcule, sortări, clasificări.

Datele supuse prelucrării sunt introduse în calculator sub formă de cifre şi alfanumerice (litere, cifre şi alte caractere speciale).

Intrucât calculatoarele lucrează cu circuite integrate care nu cunosc decât două stări distincte, cea mai mică unitate care este prezentată în informatică este **bitul** (binary digit).

In prezent calculatoarele lucrează cu cea mai mică unitate de date ce este adresată memoriei calculatorului, şi care se numeşte **byte** sau octet, succesiune de 8 biţi.

Multipli unui byte, folosiţi în mod obişnuit în literatura de specialitate sunt:

- 1 kB = 1.024 bytes ;
- 1 MB = 1.024 kB;
- 1 GB = 1.024 MB;
- 1 TB = 1.024 GB.

SISTEME DE NUMERAŢIE. ELEMENTE DE LOGICĂ

Sistemul de numeraţie zecimal

Alfabetul sistemului zecimal, cel mai cunoscut şi utilizat în prezent este format din zece cifre : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Aceste cifre sunt prin definiţie numere consecutive, astfel $7 = 6 + 1$. Un număr în baza 10 conform relaţiei (1) poate fi scris ca o sumă de puteri ale lui 10:

$$1998 = 1 \cdot 10^3 + 9 \cdot 10^2 + 9 \cdot 10^1 + 8$$

Sistemul de numeraţie binar

Sistemul de numeraţie binar, cel mai simplu posibil inventat acum 500 de ani în China şi cel mai utilizat în reprezentarea codificată a numerelor în calculatoare, are următoarele caracteristici:

- Baza de numeraţie a sistemului este 2 şi conţine numai două simboluri, cifrele: 0 şi 1;
- Cifra cu valoarea cea mai mare este 1.

Un număr scris în baza 2, poate fi dezvoltat după puterile bazei astfel:

$$N_2 = a_n a_{n-1} \quad a_2 a_1 a_0 = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \quad + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

In continuare sunt prezentate câteva numere scrise în baza 2:
1001_2, 101010_2, 11100010_2. Ultimul număr poate fi dezvoltat după puterile lui 2 astfel:

$$11100010_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^1$$

Conversia unui număr din reprezentarea în baza *10* în reprezentarea în baza *2* se realizează în următorii paşi:

1. Se separă numărul zecimal în partea sa întreagă şi partea fracţionară.
2. Se converteşte partea întreagă la o reprezentare în baza *2*.
3. Se converteşte partea zecimală la o reprezentare în baza *2*.
4. Se combină cele două reprezentări (prin sumare), obţinându-se reprezentarea binară a numărului zecimal dat.

Se remarcă în cadrul acestei descrieri existenţa a doi paşi importanţi (este vorba de paşii *2* şi *3*).

Vom discuta pe larg aceşti paşi:.

a) Conversia unui întreg. Pentru a converti un număr scris în zecimal într-o reprezentare de tip binar, se împarte succesiv acest număr la *2* până când câtul devine *0*. Resturile obţinute în urma acestor împărţiri succesive reprezintă cifrele numărului scris în noua bază. Aceste cifre (pe care le vom numi şi biţi) sunt calculate în ordine crescătoare, bitul cu rangul cel mai puţin semnificativ fiind primul.

Exemplul 1: Să se convertească numărul $254_{(10)}$ în binar.

254	*2*
127	*0*
63	*1*
31	*1*
15	*1*
7	*1*
3	*1*
1	*1*
0	*1*

Rezultă: $254_{(10)} = 11111110_{(2)}$.

Exemplul 2: Să se convertească numărul $179_{(10)}$ în binar.

179	*2*
89	*1*
44	*1*
22	*0*
11	*0*
5	*1*
2	*1*
1	*0*
0	*1*

Rezultă: $179_{(10)} = 10110011_{(2)}$.

b) Conversia unui număr fracționar.

Pentru a realiza conversia unui număr zecimal fracționar în codul corespunzător binar, se va înmulți acesta cu *2* și se va separa apoi partea întreagă. Partea întreagă a produsului reprezintă un bit al numărului binar căutat. Procedura continuă până când partea fracționară devine nulă sau se obține precizia de reprezentare dorită.
Biții corespunzători reprezentării binare sunt determinați, prin acest procedeu, în ordine crescătoare, cel mai semnificativ fiind primul.

Exemplul 3: Să se convertească numărul $0.7109375_{(10)}$ în binar.
$0.7109375 \cdot 2 \rightarrow 1$
$0.4218750 \cdot 2 \rightarrow 0$
$0.8437500 \cdot 2 \rightarrow 1$
$0.6875000 \cdot 2 \rightarrow 1$
$0.3750000 \cdot 2 \rightarrow 0$
$0.7500000 \cdot 2 \rightarrow 1$
$0.5000000 \cdot 2 \rightarrow 1$
Rezultă: $0.7109375_{(10)} = 0.1011011_{(2)}$.
Evident că, acum, dacă vom dori conversia în binar a numărului zecimal $179.7109375_{(10)}$ acesta se va scrie:

$179.7109375_{(10)} = 10110011.1011011_{(2)}$.

Ne punem în continuare problema conversiei inverse, din reprezentarea binară în cea
zecimală. O posibilitate de a realiza acest lucru constă în scrierea sub formă de puteri a numărului reprezentat în binar.

Exemplul 4: Să se convertească numărul $10110011.1011011_{(2)}$ în zecimal.
$10110011.1011011_{(2)} = 1 \cdot 2^7 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-6} + 1 \cdot 2^{-7} = 128+32+16+2+1+1/2+1/8+1/16+1/64+1/128 = 179,7109375$

Sistemul de numeraţie octal

Utilizat în reprezentarea codificată a numerelor în calculatoare este şi sistemul de numeraţie octal, întrucât îl include pe cel binar. Are următoarele caracteristici:

- Baza de numeraţie a sistemului este 8 şi conţine opt cifre: de la 0 la 7;
- Cifra cu valoarea ce mai mare este 7. .

Un număr scris în baza 8, poate fi dezvoltat după puterile bazei astfel:

$$N_8 = a_n a_{n-1} \ldots a_2 a_1 a_0 = a_n \cdot 8^n + a_{n-1} \cdot 8^{n-1} + \ldots + a_2 \cdot 8^2 + a_1 \cdot 8^1 + a_0 \cdot 8^0$$

Sistemul de numeraţie hexazecimal

Utilizat cel mai mult în ultima vreme în reprezentarea codificată a numerelor în calculatoare este sistemul de numeraţie hexazecimal, întrucât le include pe cele binar şi octal. Are următoarele caracteristici:

- Baza de numeraţie a sistemului este 16 şi conţine 16 cifre: de la 0 la 9 şi în plus literele consecutive A, B , C, D, E şi F;
- Simbolul cu valoarea cea mai mare este F şi are valoarea 15..

Un număr scris în baza 16, poate fi dezvoltat după puterile bazei astfel:

$$N_{16} = a_n a_{n-1} \quad a_2 a_1 a_0 = a_n \cdot 16^n + a_{n-1} \cdot 16^{n-1} + \quad + a_2 \cdot 16^2 + a_1 \cdot 16^1 + a_0 \cdot 16^0$$

Realizarea a 4 siruri de transformari a unui numar dintr-un sistem de numeratie in celelalte sisteme de numeratie

Transformarea din sistemul de numeratie zecimal in sistemul de numeratie binar:

$29_{(10)} = ?_{(2)}$

29 : 2 = 14, rest *1*
14 : 2 = 7, rest *0*
7 : 2 = 3, rest *1*
3 : 2 = *1*, rest *1*

Se preia rezultatul ultimei impartiri (3 : 2 = *1*) si restul sirului de impartiri in ordine inversa (rest *1*, rest *1*, rest *0*, rest *1*).

$29_{(10)} = 11101_{(2)}$

Transformarea din sistemul de numeratie zecimal in sistemul de numeratie octal:

$29_{(10)} = ?_{(8)}$

29 : 8 = *3*, rest *5*

Se preia rezultatul ultimei impartiri (29 : 8 = *3*) si restul sirului de impartiri in ordine inversa (rest *5*).

$29_{(10)} = 35_{(8)}$

Transformarea din sistemul de numeratie zecimal in sistemul de numeratie hexazecimal:

$29_{(10)} = ?_{(16)}$

29 : 16 = *1*, rest *13*

Baza 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Baza 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Se preia rezultatul ultimei impartiri (29 : 16 = *1*) si se urmareste unde se gaseste nr. 13 in sistemul de numeratie hexazecimal.

$29_{(10)} = 1D_{(16)}$

Funcţiile logice

Matematicianul englez George BOOLE a reuşit să facă o legătură între formulele algebrice şi relaţiile logice. Pornind de la principiul că o propoziţie poate fi adevărată sau falsă, BOOLE atribuie valoarea 1 propoziţiilor adevărate şi respectiv 0 propoziţiilor false, elaborând **algebra booleană.**

Notând propoziţiile cu a, b c, etc se pot construi funcţiile logice:

- Disjuncţia(operaţie logică tradusă prin SAU) a două propoziţii, notată cu $\vee$;
- Conjuncţia(operaţie logică tradusă prin ŞI) a două propoziţii, notată cu $\wedge$;
- Negaţia unei propoziţii notată cu $\neg$.

In cazul a două propoziţii a şi b se pot obţine tabele de adevăr, pentru disjunţie, conjuncţie şi negaţie aşa cum rezultă din tabelul urmator

a	b	a∧b	a∨b	¬a
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

CODURI DE REPREZENTARE A DATELOR

Necesitatea reprezentării în calculator a unui număr mare de caractere (cifre, litere, caractere speciale) a condus la apariţia şi utilizarea unor coduri. Deoarece în calculatorul electronic orice informaţie este reprezentată în sistemul binar, apare necesitatea translatării informaţiei externe, accesibilă omului, în informaţie internă, accesibilă calculatorului, şi invers. Această translatare se realizează prin operaţia de codificare.

Codurile în care sunt reprezentate numai numere se numesc *coduri numerice*. Codurile în care sunt reprezentate numere, litere şi alte semne speciale se numesc *coduri alfanumerice*.

Dintre codurile alfanumerice, cele mai reprezentative sunt codurile ASCII, codul Gray, codul 8421 şi EBCDIC. În toate cazurile se foloseşte octetul (opt poziţii binare) pentru reprezentarea unui caracter.

Codificarea numerelor si şirurilor alfanumerice folosind codul ASCII

Un cod foarte important folosit în reprezentarea standard a simbolurilor alfabetice, şi numerice este codul *ASCII (American Standard Code for Information Interchange)*.

El are $2^8 = 256$ simboluri sursa, codificate în secvente binare de lungime 8;

Prezentam în tabelul următor simbolurile ASCII:

Caracterele din domeniul 0-31 sunt caractere neimprimabile si sunt folosite pentru controlul echipamentelor periferice (ex. imprimanta).

Cod	Simbol	Descriere
0	NUL	Null char
1	SOH	Start of Heading
2	STX	Start of Text
3	ETX	End of Text

4	EOT	End of Transmission
5	ENQ	Enquiry
6	ACK	Acknowledgment
7	BEL	Bell
8	BS	Back Space
9	HT	Horizontal Tab
10	LF	Line Feed
11	VT	Vertical Tab
12	FF	Form Feed
13	CR	Carriage Return
14	SO	Shift Out / X-On
15	SI	Shift In / X-Off
16	DLE	Data Line Escape
17	DC1	Device Control 1 (oft. XON)
18	DC2	Device Control 2
19	DC3	Device Control 3 (oft. XOFF)
20	DC4	Device Control 4
21	NAK	Negative Acknowledgement
22	SYN	Synchronous Idle
23	ETB	End of Transmit Block
24	CAN	Cancel
25	EM	End of Medium
26	SUB	Substitute
27	ESC	Escape
28	FS	File Separator
29	GS	Group Separator
30	RS	Record Separator
31	US	Unit Separator

Caracterele ASCII imprimabile sunt caracterele cu codurile cuprinse intre 32 şi 127:

Cod	Simbol	Descriere
32		Space
33	!	Exclamation mark
34	"	Double quotes (or speech marks)
35	#	Number
36	$	Dollar
37	%	Procenttecken
38	&	Ampersand
39	'	Single quote
40	(	Open parenthesis (or open bracket)
41	)	Close parenthesis (or close bracket)
42	*	Asterisk
43	+	Plus
44	,	Comma
45	-	Hyphen
46	.	Period, dot or full stop
47	/	Slash or divide
48	0	Zero
49	1	One
50	2	Two
51	3	Three
52	4	Four
53	5	Five
54	6	Six
55	7	Seven
56	8	Eight
57	9	Nine
58	:	Colon
59	;	Semicolon
60	<	Less than (or open angled bracket)
61	=	Equals

62	>	Greater than (or close angled bracket)
63	?	Question mark
64	@	At symbol
65	A	Uppercase A
66	B	Uppercase B
67	C	Uppercase C
68	D	Uppercase D
69	E	Uppercase E
70	F	Uppercase F
71	G	Uppercase G
72	H	Uppercase H
73	I	Uppercase I
74	J	Uppercase J
75	K	Uppercase K
76	L	Uppercase L
77	M	Uppercase M
78	N	Uppercase N
79	O	Uppercase O
80	P	Uppercase P
81	Q	Uppercase Q
82	R	Uppercase R
83	S	Uppercase S
84	T	Uppercase T
85	U	Uppercase U
86	V	Uppercase V
87	W	Uppercase W
88	X	Uppercase X
89	Y	Uppercase Y
90	Z	Uppercase Z
91	[	Opening bracket
92	\	Backslash
93	]	Closing bracket
94	^	Caret - circumflex

95	_	Underscore
96	`	Grave accent
97	a	Lowercase a
98	b	Lowercase b
99	c	Lowercase c
100	d	Lowercase d
101	e	Lowercase e
102	f	Lowercase f
103	g	Lowercase g
104	h	Lowercase h
105	i	Lowercase i
106	j	Lowercase j
107	k	Lowercase k
108	l	Lowercase l
109	m	Lowercase m
110	n	Lowercase n
111	o	Lowercase o
112	p	Lowercase p
113	q	Lowercase q
114	r	Lowercase r
115	s	Lowercase s
116	t	Lowercase t
117	u	Lowercase u
118	v	Lowercase v
119	w	Lowercase w
120	x	Lowercase x
121	y	Lowercase y
122	z	Lowercase z
123	{	Opening brace
124	|	Vertical bar
125	}	Closing brace
126	~	Equivalency sign - tilde
127		Delete

Codul ASCII extins conține caracterele cu codul cuprins între 128 și 255 :

Cod	Simbol	Descriere
128	€	Euro sign
129		
130	‚	Single low-9 quotation mark
131	*ƒ*	Latin small letter f with hook
132	„	Double low-9 quotation mark
133	…	Horizontal ellipsis
134	†	Dagger
135	‡	Double dagger
136	ˆ	Modifier letter circumflex accent
137	‰	Per mille sign
138	Š	Latin capital letter S with caron
139	‹	Single left-pointing angle quotation
140	Œ	Latin capital ligature OE
141		
142	Ž	Latin captial letter Z with caron
143		
144		
145	‘	Left single quotation mark
146	’	Right single quotation mark
147	“	Left double quotation mark
148	”	Right double quotation mark
149	•	Bullet
150	–	En dash
151	—	Em dash
152	˜	Small tilde
153	™	Trade mark sign
154	š	Latin small letter S with caron
155	›	Single right-pointing angle quotation mark
156	œ	Latin small ligature oe
157		
158	ž	Latin small letter z with caron
159	Ÿ	Latin capital letter Y with diaeresis
160		Non-breaking space

161	¡	Inverted exclamation mark
162	¢	Cent sign
163	£	Pound sign
164	¤	Currency sign
165	¥	Yen sign
166	¦	Pipe, Broken vertical bar
167	§	Section sign
168	¨	Spacing diaeresis - umlaut
169	©	Copyright sign
170	ª	Feminine ordinal indicator
171	«	Left double angle quotes
172	¬	Not sign
173		Soft hyphen
174	®	Registered trade mark sign
175	¯	Spacing macron - overline
176	°	Degree sign
177	±	Plus-or-minus sign
178	²	Superscript two - squared
179	³	Superscript three - cubed
180	´	Acute accent - spacing acute
181	µ	Micro sign
182	¶	Pilcrow sign - paragraph sign
183	·	Middle dot - Georgian comma
184	¸	Spacing cedilla
185	¹	Superscript one
186	º	Masculine ordinal indicator
187	»	Right double angle quotes
188	¼	Fraction one quarter
189	½	Fraction one half
190	¾	Fraction three quarters
191	¿	Inverted question mark
192	À	Latin capital letter A with grave
193	Á	Latin capital letter A with acute
194	Â	Latin capital letter A with circumflex
195	Ã	Latin capital letter A with tilde

196	Ä	Latin capital letter A with diaeresis
197	Å	Latin capital letter A with ring above
198	Æ	Latin capital letter AE
199	Ç	Latin capital letter C with cedilla
200	È	Latin capital letter E with grave
201	É	Latin capital letter E with acute
202	Ê	Latin capital letter E with circumflex
203	Ë	Latin capital letter E with diaeresis
204	Ì	Latin capital letter I with grave
205	Í	Latin capital letter I with acute
206	Î	Latin capital letter I with circumflex
207	Ï	Latin capital letter I with diaeresis
208	Ð	Latin capital letter ETH
209	Ñ	Latin capital letter N with tilde
210	Ò	Latin capital letter O with grave
211	Ó	Latin capital letter O with acute
212	Ô	Latin capital letter O with circumflex
213	Õ	Latin capital letter O with tilde
214	Ö	Latin capital letter O with diaeresis
215	×	Multiplication sign
216	Ø	Latin capital letter O with slash
217	Ù	Latin capital letter U with grave
218	Ú	Latin capital letter U with acute
219	Û	Latin capital letter U with circumflex
220	Ü	Latin capital letter U with diaeresis
221	Ý	Latin capital letter Y with acute
222	Þ	Latin capital letter THORN
223	ß	Latin small letter sharp s - ess-zed
224	à	Latin small letter a with grave
225	á	Latin small letter a with acute
226	â	Latin small letter a with circumflex
227	ã	Latin small letter a with tilde
228	ä	Latin small letter a with diaeresis
229	å	Latin small letter a with ring above
230	æ	Latin small letter ae

231	ç	Latin small letter c with cedilla
232	è	Latin small letter e with grave
233	é	Latin small letter e with acute
234	ê	Latin small letter e with circumflex
235	ë	Latin small letter e with diaeresis
236	ì	Latin small letter i with grave
237	í	Latin small letter i with acute
238	î	Latin small letter i with circumflex
239	ï	Latin small letter i with diaeresis
240	ð	Latin small letter eth
241	ñ	Latin small letter n with tilde
242	ò	Latin small letter o with grave
243	ó	Latin small letter o with acute
244	ô	Latin small letter o with circumflex
245	õ	Latin small letter o with tilde
246	ö	Latin small letter o with diaeresis
247	÷	Division sign
248	ø	Latin small letter o with slash
249	ù	Latin small letter u with grave
250	ú	Latin small letter u with acute
251	û	Latin small letter u with circumflex
252	ü	Latin small letter u with diaeresis
253	ý	Latin small letter y with acute
254	þ	Latin small letter thorn
255	ÿ	Latin small letter y with diaeresis

De remarcat modalitatea diferita de codificare a cifrelor zecimale.

În codul ASCII se codifica *caracterele*, deci – în această codificare, cifrele 0, 1...., 9 sunt considerate caractere şi prelucrate ca atare, de aceea nu se pot face operaţii aritmetice obişnuite cu ele, ci doar cu codurile ASCII asociate.

Codificarea numerelor si şirurilor alfanumerice folosind codul Gray

Un **cod Gray** este codul care îi atribuie unei mulţimi continue de întregi, sau fiecărui membru al unei liste circulare, un cuvânt de simboluri, prin care două cuvinte alăturate diferă printr-un singur simbol. Poate exista mai mult de un cod Gray pentru o lungime dată de cuvânt, iar termenul a fost folosit pentru codul binar pentru întregi nenegativi. Versiunea pe patru biţi este aceasta:

0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

Codul Gray binar pentru *n* biţi poate fi generat prin recursivitate prin prefixarea unui bit 0 în faţa codului Gray pentru *n-1* biţi, apoi prefixând un bit 1 pentru aceluiaşi cod pentru *n-1* biţi, dar reflectat (în ordine inversă).

Un cod Gray de ordin n este un sir care contine toate numerele de la 0 la 2^{n-1}, astfel incat orice doua numere consecutive din sir sa difere în reprezentarea lor binară prin exact un bit.

Exista mai multe coduri Gray distincte, cel mai des întalnit fiind asa-numitul **"binary reflected Gray code"** .

Modul de constructie este destul de intuitiv: fiecare număr nou este format din cel anterior prin modificarea celui mai putin semnificativ bit, astfel încât numărul să nu mai fi fost adăugat înainte la şir (de exemplu, pentru n = 2 şi primul număr 0, sirul obţinut va fi 0, 1, 3, 2).

Codificarea numerelor si şirurilor alfanumerice folosind codul 8421

În cazul *codurilor ponderate*, o cifră zecimală este exprimată printr-o combinaţie de 4 cifre binare, în care fiecărei cifre i se asociază o anumită pondere.

Ponderile pot fi pozitive sau negative.

Valoarea cifrei zecimale se obţine prin suma biţilor din cod, fiecare bit fiind multiplicat cu valoarea ponderii asociate.

Considerând un cod format din biţii $b0$, $b1$, $b2$, $b3$, ponderile asociate acestora fiind $p0$, $p1$, $p2$, respectiv $p3$, valoarea cifrei zecimale codificate este:

$$N = p0b0 + p1b1 + p2b2 + p3b3$$

Ponderile fiecărui bit reprezintă valoarea corespunzătoare din denumirea codului. Pentru ponderile de sus, codul are denumirea $p3p2p1p0$. În tabelul urmator se prezintă exemple de coduri ponderate de 4 biţi mai des utilizate.

Exemplu: $0101_{8421} = 8^0 + 4^1 + 2^0 + 1^1 = 5$

În cazul codului 8421, deoarece fiecare bit are ponderea numărării în binar (2^0, 2^1, 2^2, 2^3), iar cuvintele de cod reprezintă numerele succesive în sistemul binar natural, codul se mai numeşte cod binar-zecimal natural (NBCD – *Natural Binary Coded Decimal*). În mod obişnuit, acest cod se numeşte, impropriu, cod BCD.

Coduri binar-zecimale ponderate de 4 biţi.

Nr. zecimal	8421	2421	$64\bar{2}\bar{3}$	$84\bar{2}\bar{1}$
0	0000	0000	0000	0000
1	0001	0001	0101	0111
2	0010	0010	0010	0110
3	0011	0011	1001	0101
4	0100	0100	0100	0100
5	0101	1011	1011	1011
6	0110	1100	0110	1010
7	0111	1101	1101	1001
8	1000	1110	1010	1000
9	1001	1111	1111	1111

În cazul codului 2421, numit şi *cod Aiken* (după numele prof. Howard Aiken, care a realizat calculatorul MARK I), primele 5 cifre zecimale (0 – 4) au aceeaşi exprimare ca şi în codul 8421. Cifra zecimală 5 poate fi exprimată fie prin 0101, fie prin 1011. Deci, reprezentarea unor cifre zecimale nu este unică, această proprietate fiind valabilă şi pentru alte coduri. Pentru codificare s-a ales reprezentarea 1011, deoarece codul pentru cifra 5 se poate obţine atunci prin complementarea codului pentru cifra 4. Aceeaşi regulă se poate aplica pentru obţinerea codului cifrei 6 din codul cifrei 3, a codului cifrei 7 din codul cifrei 2 etc.

Codurile care au această proprietate se numesc coduri *autocomplementare*. Un cod este autocomplementar dacă cuvântul de cod al complementului faţă de 9 al cifrei *N* (deci 9 – *N*) se poate obţine din codul cifrei *N*, prin complementarea fiecăruia din cei 4 biţi. De exemplu, codul 8421 nu este autocomplementar, iar codurile 2421, 6423 , 8421 sunt autocomplementare.

Codificarea numerelor si şirurilor alfanumerice folosind codul ECBDIC

EBCDIC (Extended Binary Coded Decimal Interchange Code) este un cod ce utilizează 8 cifre binare cu care se pot realiza 2^8=256 de combinaţii. O parte din combinaţii sunt utilizate pentru codificarea anumitor comenzi.

Fiecare caracter se reprezintă prin două simboluri din sistemul de numeraţie hexazecimal.

Reprezentarea în cod EBCDIC a expresiei „REFERAT LA INFORMATICA" se prezinta astfel:

- in sistem hexazecimal:

hexazecimal	1	2	3	4	5	6	7	8	9
C	A	B	C	D	E	F	G	H	I
D	J	K	L	M	N	O	P	Q	R
E	-	S	T	U	V	W	X	Y	Z

9D 5C 6C 5C 9D 1C 3E 3D 1C 9C 5D 6C 6D 9D 4D 1C 3E 9C 3C 1C

Se face transformarea in sistemul de numeratie binar aplicand teorema (unui hexazecimal îi corespund 4 binare) si tabela de transformare:

hexazecimal	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

- în sistem binar:

10011101 01011100 01101100 01011100 10011101 00011100 00111110 00111101 00011100 10011100 01011101 01101100 01101100 10011101 01001101 00011100 00111110 10011100 00111100 00011100

Reprezentarea numerelor

Reprezentarea interna a datelor numerice se face diferentiat, in functie de tipul informatiei :

- numere intregi cu semn sau fara semn;
- numere reale.

Asupra datelor de tip numeric lucreaza operatorii aritmetici + , - , * , / , si de comparare < , > , =, #, >=, >=.

Reprezentarea numerelor intregi. Fiecare numar intreg pozitiv sau negativ este codificat ca un numar binar de lungime fixa. Lungimea secventei, binare este multiplu de 8 biti : 8,16,32... . Pentru completarea secventei de biti se adauga zerouri nesemnificative. La reprezentarea intregilor cu semn, primul bit din stanga reprezentarii indica semnul numarului, astfel: **1** pentru numar **negativ** si **0** pentru numar **pozitiv**.

Exemplu: daca se reprezinta un intreg fara semn, fie 9 acest numar, pe 16 biti atunci se obtine:

$$9_{(10)} = 1001_{(2)}\ 0000\ 0000\ 0000\ 1001.$$

Rezulta ca domeniul de reprezentare a intregilor fara semn, utilizand 8 cifre binare este 0...255, iar pentru 16 cifre binare, 0... + 65535.

Domeniul de definitie al unei date de tip numeric intreg cu semn, reprezentat pe 8 cifre binare(pe un octet sau un byte) este –128... +127, iar pe cuvinte de 16 biti este de –65536...+65535.

Reprezentarea numerelor reale. Numerele reale sunt formate din **semn**, **parte intreaga** si **parte fractionara**. Acestea pot fi reprezentate in doua moduri in virgula fixa (*binary fixed print*) sau in virgula mobila (*binary floating print*).

În reprezentarea in virgula fixa se presupune ca partea intreaga este despartita de partea fractionara printr-o virgula imaginara care se afla intr-o pozitie fixa. În acest caz sunt fixe atat numarul de pozitii ale partii intregi cat si numarul de pozitii ale partii fractionare. Acest mod de reprezentare a realilor este dezavantajos deoarece nu permit decat reprezentarea unei game restranse de numere reale.

În virgula mobila, numerele sunt reprezentate prin exponent si mantisa in asa numita notatie stiintifica. Se stie ca orice numar poate fi scris explicitand diferite puteri ale lui 10 (exponenti). În acest fel poate fi controlata pozitia virgulei zecimale, care isi schimba locatia in functie de valoarea exponentului.

Exemplul 1: **$43,7 = 437 * 10^{(-1)} = 437E\text{-}1.$** 437 este mantisa iar –1 este exponentul.

Conform acestei conventii, daca se foloseste un cuvant de 32 biti, pentru reprezentarea unui real in virgula mobila, atunci repartizarea bitilor se va face astfel :1 bit pentru semnul numarului, 1 bit pentru semnul exponentului, 7 biti pentru exponent si 23 de biti pentru mantisa.

Exemplul 2: $12,5_{(10)} = 1100,1_{(2)} = 0,11001_{(2)} * 2^4 = 11001_{(2)} * 10_{(2)}^{100}{}_{(2)},$

mantisa este 11001; exponentul este $4_{(10)} = 100_{(2)}$; bitul de semn al numarului = 0 ;bitul de semn al exponentului = 0 ; iar reprezentarea numarului este

0 0 0000100 11001 00 0000 0000 000 0000

Se poate demonstra ca domeniul de valori al unei date pe 32 biti din care 7 pentru exponent si 23 pentru mantisa este: $-10^{38} \ldots 10^{38}$, iar data va avea maxim 7 cifre semnificative.

Reprezentarea in virgula mobila permite memorarea numerelor reale de diferite dimensiuni cu o precizie foarte mare.

În functie de numarul de biti folositi pentru reprezentarea numarului exista :

- reprezentare in simpla precizie – pe 32 de biti;
- reprezentare in simpla precizie – pe 64 de biti.

Reprezentarea desenelor si sunetelor.

Desenele si sunetele sunt si ele codificate in secvente de cifre binare. Pentru codificare se stabilesc niveluri de luminozitate pentru desene sau niveluri de semnal sonor pentru sunete. Aceste niveluri se codifica prin numere intregi care pot fi reprezentate in sistem binar. Acest procedeu se numeste **digitizarea** desenelor si sunetelor.

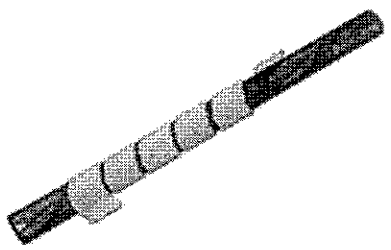
CRIPTAREA DATELOR

Criptografia reprezintă o ramură a matematicii care se ocupă cu securizarea informaţiei precum şi cu autentificarea şi restricţionarea accesului într-un sistem informatic. În realizarea acestora se utilizează atât metode matematice (profitând, de exemplu, de dificultatea factorizării numerelor foarte mari), cât şi metode de criptare cuantică. Termenul *criptografie* este compus din cuvintele de origine greacă κρυπτός *kryptós* (ascuns) şi γράφειν *gráfein* (a scrie).

Criptologia este considerată ca fiind cu adevărat o ştiinţă de foarte puţin timp. Aceasta cuprinde atât criptografia - scrierea secretizată - cât şi criptanaliza. De asemenea, criptologia reprezintă nu numai o artă veche, ci şi o ştiinţa nouă: veche pentru că Iulius Cezar a utilizat-o deja, dar nouă pentru că a devenit o temă de cercetare academico-ştiinţfică abia începând cu anii 1970. Această disciplină este legată de multe altele, de exemplu de teoria numerelor, algebră, teoria complexităţii, informatică.

Informatia (religioasa, militara, economica, etc.) a însemnat întotdeauna putere, prin urmare dorinta de a o proteja, de a o face accesibila doar unor elite, unor initiati, s-a pus din cele mai vechi timpuri.

Primele texte cifrate descoperite pâna în prezent dateaza de circa 4000 de ani si provin din Egiptul antic, dar existenta acestora probabil ca dateaza de la aparitia scrierii în toate civilizatiile umane. Exista date privind utilizarea scrierii cifrate în Grecia antica înca din secolul al V-lea i.e.n.



Pentru cifrare se folosea un baston în jurul caruia se înfasura, spirala lâng a spirala, o panglica îngusta de piele, papirus sau pergament pe care, paralel cu axa, se scriau literele mesajului. Dupa scriere panglica era derulata, mesajul devenind indescifrabil. El putea fi reconstituit numai de catre persoana care avea un baston identic cu cel utilizat la cifrare. În secolul al IV-lea i.e.n. în Grecia se cunosteau 16 scrieri cifrate.

Scytalul din Grecia Antică, probabil, ca şi această reconstrucţie modernă, a fost unul din cele mai vechi dispozitive de implementare a unui cifru.

Istoricul grec Polybius (sec II î.e.n.) este inventatorul unui tabel de cifrare patrat de dimensiune 5x5, tabel aflat la baza elaborarii unui numar mare de sisteme de cifrare utilizate si azi. În Roma antica secretul informatiilor politice si militare se facea utilizând scrierea secreta. Amintim cifrul lui Cesar, utilizat înca din timpul razboiului galic. Exista documente ce atesta existenta scrierilor secrete în Asia înca din antichitate. Astfel, literatura indiana da o serie de referinte dintre care Artha-Sastra (321-300 i.e.n.), Lalita-Vistara si Kamasutra sunt exemplele cele mai cunoscute.

Stenografia, stiinta scrierilor secrete insesizabile camuflate în texte în clar,constituie o forma particulara de secretizare. Contributia araba la dezvoltarea criptologiei, mai putin cunoscuta si mediatizata este de o remarcabila importanta. David Kahn, unul dintre cei mai de seama istoriografi ai domeniului, subliniaza în cartea sa The Codebreakers ca criptologia s-a nascut în lumea araba. Primele trei secole ale civilizatiei islamice (700-1000 e.n.) au constituit, pe lânga o mare extindere politica si militara si o epoca de intense traduceri în limba araba ale principalelor opere ale antichitatii grecesti, romane, indiene, armene, ebraice, siriene. Unele carti sursa erau scrise în limbi deja moarte, deci reprezentau în fapt texte cifrate, astfel încât traducerea lor constituie primii pasi în criptanaliza, deci originile criptologiei pot fi atribuite arabilor. Cartea lui Ahmad ibn Wahshiyyah (cca. 900 e.n.) contine 93 de alfabete ale diferitelor limbi, moarte sau vii.

Dezvoltarile criptanalizei au fost mult sprijinite de studiile lingvistice ale limbii arabe care în primele patru secole ale imperiului islamic a constituit limba oficiala unificatoare pentru un imperiu de o uriasa întindere si în acelasi timp si limba stiintifica. Arabii au preluat cunostintele matematice ale civilizatiilor grecesti si indiene. Arabii sunt cei care au introdus sistemul zecimal de numerotatie si cifrele "arabe". Termenii "zero ","cifru", "algoritm", "algebra" se datoreaza tot lor. Manuscrise recent descoperite arata ca primele scrieri în domeniul probabilitatilor si statisticii dateaza cu 800 de ani înaintea celor corespunzatoare ale lui Pascal si Fermat. Însusi termenul de "cifru" ne vine de la arabi. El provine de la cuvântul arab "sifr" care reprezinta traducerea în araba a cifrei zero din sanscrita.

Conceptul de zero a fost deosebit de ambiguu la începuturile introducerii lui în Europa, în care sistemul de numerotatie folosit era cel roman. De aceea se obisnuia sa se spuna despre cineva care vorbea neclar ca vorbeste ambiguu, ca un cifru. Acest înteles de ambiguitate a unui mesaj poarta si azi denumirea de cifru. În perioada Renasterii, odata cu trezirea interesului pentru civilizatia antica, sau redescoperit lucrarile criptografiei din antichitate. Extinderea relatiilor diplomatice dintre diferitele state feudale a determinat o puternica dezvoltare a secretizarii informatiei.

În fapt istoria criptografiei/ criptologiei urmareste îndeaproape cresterea si descresterea marilor imperii si civilizatii. Ea nu apare si nu se dezvolta decât acolo unde este putere ce trebuie protejata.

Aparitia si dezvoltarea continua a utilizarii calculatoarelor practic în toate domeniile vietii, existenta si evolutia puternica a retelelor teleinformatice la nivel national si international, globalizarea comunicatiilor, existenta unor baze de date puternice, aparitia si dezvoltarea comertului electronic, a postei electronice, constituie premisele societatii informationale în care pasim. Toate acestea indica o crestere extraordinara a volumului si importantei datelor transmise sau stocate si implicit a vulnerabilitatii acestora. Protectia în aceste sisteme vizeaza:

- eliminarea posibilitatilor de distrugere voita sau accidentala.
- asigurarea caracterului secret al comunicarii pentru a preveni posibilitatea ca persoane neautorizate sa extraga informatii.
- autentificarea informatiei în scopul prevenirii posibilitatii ca persoane neautorizate sa introduca informatii în sistem.
- în anumite situatii, cum ar fi transferurile electronice de fonduri, negocierile contractuale, este importanta existenta unor semnaturi electronice pentru a evita dispute între emitator si receptor cu privire la mesajul transmis.

Toate aceste obiective arata o largire puternica a domeniului de aplicare al criptografiei de la domeniul diplomatic, militar, politic, la cel civil cu caracter economic si social. La ora actuala, 99% din criptografie nu este utilizata pentru protejarea secretelor militare ci pentru carduri bancare, plati de taxe radio / TV, taxe de drumuri, acces autorizat în cladiri sau la calculatoare, terminale de loterie, instrumente electronice de plati anticipate. În aceste aplicatii rolul criptografiei este acela de a face furturile mai greu de realizat.

Criptarea este o metoda de codare a datelor prin intermediul unei forme algoritmice intr-o forma indescifrabila inainte de transmiterea acestora de catre statia de lucru sursa si decodate ulterior la destinatie prin aplicarea algoritmului de criptare in sens invers.

Criptarea datelor tranzitate se poate face in doua forme:
· Criptare simetrica – criptare conventionala
· Criptare asimetrica – Criptarea cu "cheie publica"

Criptarea simetrica utilizeaza urmatoarea schema:

Componente	Descriere
Text obisnuit	Datele ce sunt tranzactionate in retea.
Algoritm de criptare	Tehnici de disimulare a datelor originale, transformandu-le in date indescifrabile de catre alt utilizator.
Cheie Secreta	Utilizata pentru a identifica algoritmul de criptare. Cheia de criptare poate fi o parola ascunsa care sa permita decriptarea datelor numai de catre destinatar.
Text cifrat (Codat)	Textul este codat.
Algoritm de decriptare	Algoritmul invers care trebuie aplicat pentru decriptare la destinatie.

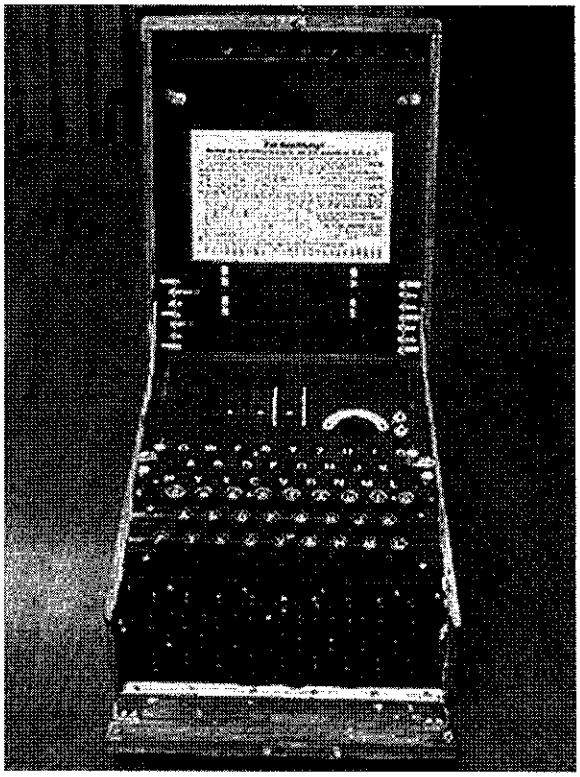
Criptarea Asimetrica sau criptarea cu cheie publica sau privata.

Aceasta tehnica de securizare a datelor consta in alocarea unei chei publice sau private unice. Astfel daca datele au fost criptate cu o cheie "1" – cheia privata, ele nu pot fi decriptate decat cu cheia "2" – cheia publica; si invers. Utilizarea criptarii asimetrice impune deasemenea autentificarea sursei, astfel ca o replica la un e-mail anterior va contine in mod automat si identitatea fostului destinatar.

In cazul in care luam ca exemplu o schema hashing, este generata o valoare hash pentru datele respective in computerul sursa si este trimisa impreuna cu aceasta la destinatie. Computerul destinatar va receptiona codul hash odata cu datele receptionate. In cazul in care cheia hash generata de computerul de

destinatie este identica celei primite de la sursa, inseamna ca date nu au suferit nici o modificare, integritatea lor a fost pastrata.

Acest cod hash, poarta denumirea de semnatura digitala cand este trimis impreuna cu datele.



Masina de codat Enigma este un dispozitiv de cifrat portabil, utilizat pentru criptarea si decriptarea mesajelor secrete. Aceasta a fost proiectata la Berlin in 1918, de inginerul german Arthur Scherbius.

Masina este bazata pe mecanismul rotorilor montati pe cilindri si intrebuintati in operatiile de cifrare si descifrare de informatii, reprezentand in mana germanilor instrumentul care a participat covarsitor la obtinerea titlului si consemnarea numelui lor ca popor invingator al celui de-al II-lea razboi mondial. In timpul desfasurarii razboiului, germanii au dezvoltat mai multe variante ale masinii, intreaga informatie comunicata radio, telegrafic era codata prin recurgerea la sistemul cu Enigma.

Spargerea codurilor maşinii Enigma de către Biuro Szyfrów, şi, ulterior, decriptarea pe scară largă a traficului Enigma la Bletchley Park, a fost un important factor ce a contribuit la victoria Aliaţilor în război.

Dezvoltarea electronicii şi a calculatoarelor numerice după al doilea război mondial au făcut posibile cifruri mult mai complexe. Mai mult, calculatoarele au permis criptarea oricărui fel de date reprezentate de calculator în format binar, spre deosebire de cifrurile clasice care criptau doar texte în limbaj scris, dizolvând utilitatea abordării lingvistice a criptanalizei în multe cazuri. Multe cifruri informatice pot fi caracterizate prin operarea pe secvenţe de biţi (uneori pe grupuri sau blocuri), spre deosebire de schemele clasice şi mecanice, care manevrează

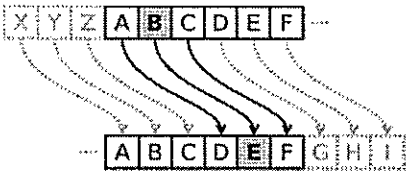
caractere tradiţionale (litere şi cifre) direct. Totuşi, calculatoarele au ajutat şi criptanaliştii, ceea ce a compensat până la un punct creşterea complexităţii cifrurilor. Cu toate acestea, cifrurile moderne bune au rămas cu un pas înaintea criptanalizei; este cazul de obicei ca utilizarea unui cifru de calitate să fie foarte eficientă (rapidă şi puţin costisitoare în ce priveşte resursele), în timp ce spargerea cifrului să necesite un efort cu multe ordine de mărime mai mare, făcând criptanaliza atât de ineficientă şi nepractică încât a devenit efectiv imposibilă.

Proiectanţii de sisteme şi algoritmi criptografici, pe lângă cunoaşterea istoriei criptografiei, trebuie să ia în consideraţie în dezvoltarea proiectelor lor şi posibilele dezvoltări ulterioare. De exemplu, îmbunătăţirile continue în puterea de calcul a calculatoarelor au mărit gradul de acoperire al atacurilor cu forţa brută la specificarea lungimii cheilor.

În principal, până la începutul secolului al XX-lea, criptografia s-a ocupat mai ales de şabloane lingvistice. De atunci, accentul s-a mutat pe folosirea extensivă a matematicii, inclusiv a aspectelor de teoria informaţiei, complexitatea algoritmilor, statistică, combinatorică, algebră abstractă şi teoria numerelor. Criptografia este şi o ramura a ingineriei, dar una neobişnuită, întrucât se ocupă de opoziţia activă, inteligentă şi răuvoitoare; majoritatea celorlalte ramuri ale ingineriei se ocupă doar de forţe naturale neutre. Se fac cercetări şi în examinarea relaţiilor dintre problemele criptografice şi fizica cuantică.

Cifrul lui Cezar

Aplicarea cifrului lui Cezar asupra unui text oarecare constă în înlocuirea fiecărei litere din textul inițial cu a 3-a litera care se află după litera curentă, în alfabet, astfel încât B din textul inițial devine E în textul criptat.



Această metodă este numită așa după Iulius Cezar, care o folosea pentru a comunica cu generalii săi.

```
Normal:     ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cifru :     DEFGHIJKLMNOPQRSTUVWXYZABC
```

Pentru a cripta un mesaj se caută fiecare literă a mesajului în linia "Normal" și se scrie litera corespunzătoare din linia "Cifru". Pentru decriptarea unui text cifrat se procedează invers.

```
Mesaj inițial:    REVISTA INTERFERENTE TEHNOLOGICE
Mesaj criptat:    UHYLVWD LQWHUIHUHQWH WHKQRORJLFH
```

În cazul cifrului lui Cezar refacerea textului inițial este relative ușoară, fie cunoaștem cu câte caractere a fost decalată fiecare literă din text, fie cunoscând faptul că în limba română literele e și a au cea mai mare frecvență numărăm caracterul care apare cel mai des in textul criptat (fie acesta ch) si decalam apoi toate caracterele cu diferenta (E-ch) sau cu diferenta (A-ch).

Sunt sanse foarte mari ca unul din textile obtinute sa fie textul inițial. Cifrul lui Cesar este un cifru cu substituție monoalfabetică fiecare caracter al textului în clar (M) este înlocuit cu un caracter corespondent în textul cifrat (C).

Cifrul Cezar este denumit după Iulius Cezar, care a folosit un alfabet cu o deplasare de trei poziții.

Deşi Cezar a fost primul care a fost folosit cifrul într-un mod în care se poate atesta, alte cifruri bazate pe substituţie se cunosc ca fiind folosite anterior. Nepotul lui Iulius Cezar, Augustus, a folosit de asemenea cifrul, dar cu o deplasare de unu:

> *Când scria încifrat, scria B în loc de A, C în loc de B, şi restul literelor pe acelaşi principiu, folosind AA pentru X.* — Suetonius, Viaţa lui Augustus.

Există dovezi cum că Iulius Cezar folosea şi sisteme mai complicate, iar un scriitor, Aulus Gellius, referă un tratat (acum pierdut) despre cifrurile lui:

> *Există chiar şi un tratat scris în mod ingenios de către gramaticianul Probus cu privire la semnificaţia secretă a literelor din compoziţia epistolelor lui Cezar.* — Aulus Gellius.

Nu se ştie cât de util era cifrul Cezar în acel timp, dar este probabil ca el să fie destul de sigur, atât timp cât numai câţiva dintre inamicii lui Cezar erau în stare să scrie şi să citească, dar mai ales să cunoască concepte de criptanaliză. Presupunând că un atacator reuşea să citească un mesaj, nu există indicii cu privire la existenţa unor tehnici de soluţionare a cifrurilor cu substituţie. Primele dovezi cunoscute sunt lucrările din secolul al IX-lea din lumea arabă, o dată cu descoperirea analizei frecvenţei.

Chiar şi în 1915, cifrul Cezar era folosit: armata rusească l-a utilizat ca înlocuitor pentru cifruri mai complicate care s-au dovedit a fi prea dificile pentru ca trupele lor să le folosească; criptanaliştii germani şi austrieci nu aveau nici o dificultate în decriptarea mesajelor lor.

Cifrurile Cezar pot fi găsite astăzi în jucăriile pentru copii. O deplasare de 13 este efectuată în algoritmul ROT13, o metodă simplă de alambicare a textului de pe unele forumuri de pe Internet, dar nu ca metodă de criptare.

Spargerea cifrului

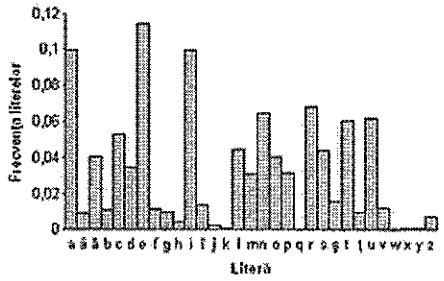
Cifrul Cezar poate fi spart uşor chiar şi având la dispoziţie numai criptotextul. Două situaţii pot fi luate în considerare:

1. atacatorul cunoaşte (sau ghiceşte) că a fost folosită un fel de substituţie simplă, dar nu neapărat o schemă Cezar

2. atacatorul ştie că s-a folosit cifrul Cezar, dar nu cunoaşte valoarea de deplasare.

În primul caz, cifrul poate fi spart folosind aceeaşi tehnică ca pentru cazul general de substituţie simplă, precum analiza frecvenţei sau cuvinte şablon.

În timpul decriptării, este foarte probabil ca atacatorul să observe regularitatea în soluţie şi să deducă că cifrul Cezar este algoritmul folosit.



Distribuţia literelor într-un text din limba română are o formă cunoscută şi predictibilă. Un cifru Cezar "roteşte" această distribuţie şi de aceea e posibilă aflarea valorii de deplasare prin analizarea graficului de frecvenţă rezultat. În limba română caracterele cele mai frecvente sunt vocalele e, a şi i în aceastaă ordine.

În al doilea caz, spargerea schemei este mult mai simplă. Deoarece numărul de deplasări posibile e limitat (31 în română), fiecare din ele poate fi testată printr-un atac prin forţă brută . O cale de a realiza acest lucru este de a scrie un fişier cu criptotextul într-un tabel cu toate deplasările posibile — tehnică numită uneori "completarea componentei normale".

Exemplul este dat pentru criptotextul "DXDHJŞUYDX".

Textul normal este imediat recognoscibil de ochi la valoarea cinci. O altă cale de a vizualiza această metodă este de a scrie sub fiecare literă alfabetul înapoi faţă de literă. Acest atac poate fi accelerat folosind şiruri cu alfabetul scris invers. Şirurile sunt apoi aliniate astfel încât criptotextul să apară pe un rând, iar astfel textul iniţial va apărea pe un alt rând.

O altă abordare a atacului prin forţă brută este identificarea literelor conform distribuţiei lor în limba în care a fost scris textul. Prin crearea graficului frecvenţelor

literelor din criptotext şi prin cunoaşterea distribuţiei obişnuite, un om poate descoperi valoarea deplasării prin observarea decalajului dintre anumite caracteristici ale graficului. Aceasta este cunoscută ca analiza frecvenţei. Şi computerele pot determina acest lucru prin măsurarea echivalenţei dintre distribuţia curentă şi distribuţia aşteptată;

Pentru texte naturale va exista doar o decriptare plauzibilă, deşi pentru texte normale foarte scurte se poate să existe mai multe versiuni. De exemplu, criptotextul UHU poate, în mod plauzibil, să fie decriptat în "ana" sau în "bob"; similar, "PR" în "ac" sau "ce".

Criptări şi decriptări multiple nu aduc nimic în plus în ceea ce priveşte securitatea. Aceasta pentru că două criptări, de exemplu deplasarea A şi deplasarea B, vor fi echivalente cu deplasarea $A + B$. În termeni matematici, criptarea cu diferite chei formează un grup.

Cifrul lui Polybius este un cifru substituţie.

Literele alfabetului latin sunt aşezate întrun pătrat de dimensiune 5x5. Literele I si J sunt combinate pentru a forma un singur caracter, deoarece alegerea finală (între I si J) poate fi uşor decisă din contextul mesajului. Rezultă 25 de caractere aşezate într-un pătrat 5x5. Cifrarea oricărui caracter se face alegând perechea potrivită de numere (intersecţia liniei şi coloanei) corespunzătoare dispunerii caracterului în pătrat.

Pătratul lui Polybius

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	IJ	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

R se înlocuieşte cu 42
E se înlocuieşte cu 15
V se înlocuieşte cu 51
I se înlocuieşte cu 24
S se înlocuieşte cu 43
T se înlocuieşte cu 44
A se înlocuieşte cu 11

```
Mesaj iniţial:    REVISTA INTERFERENTE TEHNOLOGICE

Mesaj criptat:    42155124434411 2433441542211542153344l5
44152333343134222415
```

Cifrul Vigenère

Cifrul Vigenère foloseşte un cifru Cezar cu o deplasare diferită la fiecare poziţie din text; valoarea deplasării este definită folosind un cuvânt-cheie care se repetă. Dacă o cheie este la fel de lungă ca şi mesajul şi aleasă aleatoriu, atunci acesta este un cifru care nu poate fi spart atât timp cât cheia este secretă. Cuvintele cheie mai scurte decât mesajul introduc un şablon ciclic care poate fi detectat cu o versiune statistică avansată a analizei frecvenţei.

Cifrul Vigenère este numit după Blaise de Vigenère, deşi Giovan Batista Belaso inventase primul acest cifru. Vigenère a inventat însă un cifru cu autocheie mai puternic.

Acest cifru este cunoscut deoarece deşi este uşor de înţeles şi implementat, pare pentru începători imposibil de spart; acestui fapt i se datorează descrierea **le chiffre indéchiffrable** (franceză, "cifrul indescifrabil"). În consecinţă, mulţi oameni au încercat să implementeze scheme de criptare care sunt, esenţialmente, cifruri Vigenère, doar ca să fie sparte.

Maşina Enigma este mai complexă, dar esenţialmente un cifru cu substituţie polialfabetică.

Acest cifru utilizează cifrul lui Cezar şi un anumit cuvânt cheie. Cheia dictează alegerea liniilor în criptarea şi decriptarea fiecărui caracter din mesaj.

Exemplu:

Cuvânt cheie **M O N A** M O N A M O N A
12 14 13 0 12 14 13 0 12 14 13 0

Text în clar: A S O S I T T I M P U L
Text cifrat: M G B S U H G I Y D H L

O variantă a acestui cifru este *cifrul Vigenere cu cheie în clar (cheie de încercare).* Cheia de încercare indică linia (sau liniile) de început pentru primul (sau primele caractere) ale textului în clar ca în exemplul următor. Apoi caracterele textului în clar sunt folosite drept chei pentru alegerea liniilor în criptare.

Exemplu: Reluăm exemplul anterior, dar alegem
litera M drept cheie de încercare.

Obţinem:

Cuvânt cheie M A S O S I T T I M P U
Text în clar A S O S I T T I M P U L
Text cifrat M S G G A B M B U B J F

Observaţie: Se remarcă introducerea unei reacţii în procesul de criptare, textul cifrat fiind condiţionat de conţinutul mesajului.

O altă variantă a cifrului Vigenere este *cifrul Vigenere cu autocheie (cheie cifrată).* După criptarea cu cheie de încercare, fiecare caracter succesiv al cheii în secvenţă se obţine de la caracterul cifrat al mesajului şi nu de la
textul în clar.

Exemplu:
Cuvânt cheie M M E S K S L E M Y N H
Text în clar A S O S I T T I M P U L
Text cifrat M E S K S L E M Y N H S

Observaţie: Deşi fiecare caracter utilizat drept cheie poate fi găsit din caracterul anterior al textului cifrat, el este funcţional dependent de toate caracterele anterioare ale mesajului, inclusiv de cheia de încercare.

Urmare a acestui fapt este efectul de difuziune a proprietăţilor statistice ale textului în clar asupra textului cifrat, ceea ce face ca analizele statistice să devină foarte grele pentru un criptanalist.

În baza standardelor actuale, schemele de cifrare Vigenere nu sunt foarte sigure; contribuţia importantă a lui Vigenere constă în faptul că a descoperit că pot fi generate secvenţe nerepetitive drept cheie, prin utilizarea a însuşi mesajului sau a unor părţi ale acestuia.

Cifruri de transpoziţie

Cifrurile de transpoziţie se caracterizează prin faptul că textul în clar rămâne acelaşi, doar ordinea caracterelor se schimbă.

Exemplu: Cifrul simplu cu transpunere în coloane:

- textul în clar se scrie orizontal într-o anumită formă, ca la Polybius sau ceva asemănător, iar textul cifrat se citeşte pe verticală (coloane):

V E N I
V I D I "V V V E I I N D C I I I"
V I C I

O simplă transpoziţie permite păstrarea proprietăţilor statistice ale textului în clar şi textului cifrat; o nouă transpoziţie a textului cifrat măreşte securitatea cifrului. Mulţi algoritmi moderni folosesc transpoziţia, dar consumul de memorie este mare comparativ cu substituţia, care din acest punct de vedere este mai convenabilă.

O metodă ce realizează o criptare prin permutarea caracterelor ce apar în textul iniţial se bazează pe o grilă haşurată, o grila cu n linii si n coloane, în care haşurăm $n^2/4$ casuţe astfel încât prin rotirea acestei grile de 4 ori casuţele haşurate să nu se suprapună. Ca realizare practică se va face o singură grila în care decupăm haşurile, şi pe care o rotim după dorinţă.

Criptarea se realizează prin scrierea textului iniţial sub forma unei grile (matrice) şi suprapunerea peste acesta a grilei formate. Se vor trece în textul criptat caracterele ce se găsesc sub casuţele haşurate.

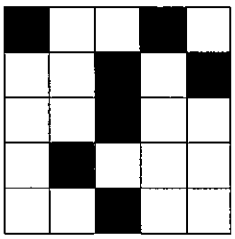
Cu ajutorul unui *text în clar* (o mică parte din textul iniţial) şi ajutorul unui calculator se poate regăsi textul iniţial. Problema se rezumă la generarea tuturor grilelor cu n linii şi n coloane, cu nunăr cunoscut de hasuri (în implementare se vor folosi valori de 0 şi 1 pt haşurare) şi ca verificare se va căuta în textul obţinut prin decriptare *textul în clar*.

De exemplu pentru a cripta textul "GRUPUL SCOLAR TRAIAN VUIA" vom parcurge următorii paşi:

1. Determinăm numărul de caractere din text şi aflăm n astfel încât n^2 să fie cel puţin egal cu numărul de caractere din text. În cazul nostru lungimea textului este 25, iar numărul n gasit este n=5.

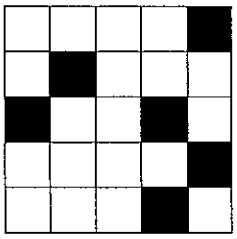
2. Vom creea o grilă cu n linii si n coloane (n=5), haşurând în acestă grilă $n^2/4$ căsuţe, astfel dispuse astfel încât prin patru rotaţii consecutive ale grilei (în sensul acelor de ceasornic) aceste haşuri să nu se suprapună.

Presupunem că aceasta este grila inițială si celelalte trei, obținute prin rotirea, de fiecare dată, cu 90^0 către dreapta.

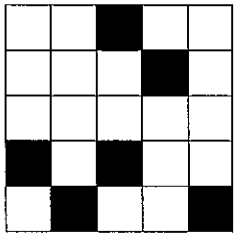


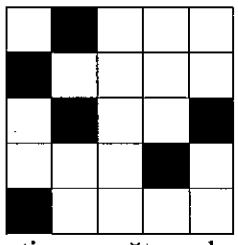
3. Vom scrie acum textul inițial sub forma unei grile cu 5 linii şi 5 coloane:

G R U P U
L _ S C O
L A R _ T
R A I AN
_ V U I A

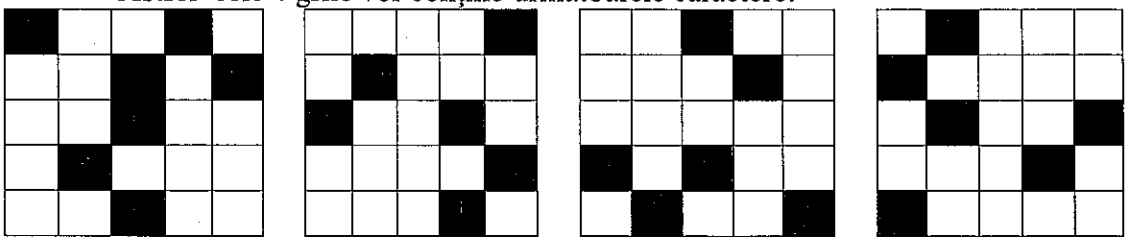


4. Vom creea acum, textul criptat adăugând, pe rând literele care coincid cu casuțele haşurate din cele 4 grile.





Astfel cele 4 grile vor conține următoarele caractere:



Scriind aceste caractere în ordinea în care acestea apar, obținem textul criptat:

GPSORAUU_L_NIUCRIVARLATA_

Textul inițial: GRUPUL_SCOLAR_TRAIAN_VUIA

Criptografie asimetrică

Criptografia asimetrică este un tip de criptografie în care utilizatorul are o pereche de chei, una publică şi una privată. Folosind cheia publică se poate cripta un mesaj care nu va putea fi decriptat decât cu cheia pereche, cea privată. Matematic, cele două chei sunt legate, însă cheia privată nu poate fi obținută din cheia publică, invers se poate.

O analogie foarte potrivită pentru proces este folosirea cutiei poştale. Oricine poate pune în cutia poştală a cuiva un plic, dar la plic nu are acces decât posesorul cheii de la cutia poştală.

Cripografia asimetrică se mai numeşte **criptografie cu chei publice**. Metodele criptografice în care se foloseşte aceeaşi cheie pentru criptare şi decriptare fac sunt metode de **criptografie simetrică** sau **criptografie cu chei secrete**.

Complementar, criptografia cu cheie secretă, cunoscută şi ca simetrică, foloseşte o singură cheie secretă utilizată atât pentru criptare cât şi pentru decriptare. Pentru a putea folosi această metodă atât receptorul cât şi emițătorul ar trebui sa cunoască cheia secretă.

Cele două mari ramuri ale criptografiei asimetrice sunt:

1.Criptarea cu cheie publică – un mesaj criptat cu o cheie publică nu poate fi decodificat decat folosind cheia privată corespunzătoare. Metoda este folosită pentru a asigura confidențialitatea.

2.Semnături digitale – un mesaj semnat cu cheia privata a emițătorului poate fi verificat de oricine are acces la această cheie, astfel asigurandu-se autenticitatea mesajului.

O analogie pentru semnăturile digitale ar fi sigilarea unui plic folosind un sigiliu personal. Plicul poate fi deschis de oricine, dar sigiliul personal este cel care verifică autenticitatea plicului.

In trecut, cheia folosită pentru criptare trebuia să fie secretă şi prestabilită folosind o metodă sigură, dar nu criptică, de exemplu, o întâlnire sau un curier sigur.

Totuşi, această metodă impunea nişte dificultăţi de ordin practic. Criptarea cu cheie publică a fost creată tocmai cu scopul de a înlătura aceste probleme – cu această metodă utilizatorii pot comunica sigur pe un canal nesigur fară să fie nevoie de o cheie prestabilită.

Criptarea cu Cheie Publica este considerată a fi cel mai important pas in domeniul criptografiei in ultimii 300-400 de ani. Metodele moderne de criptare cu cheie publică au fost prezentate public pentru prima data de catre profesorul Martin Hellman de la Universitatea Stanford si studentul Whitfield Diffie in 1976. Lucrarea lor descria un sistem criptografic in care doua persoane putea avea o comunicare sigura pe un canal de comunicare nesigur fara sa schimbe o cheie secreta.
"Smecheria" matematica din criptarea cu cheie publică se bazeaza pe existenta unor asa zise functii one-way, sau functii matematice care sunt usor de calculat in timp ce inversa lor este foarte dificil de calculat.

De exemplu, sa zicem ca folosesc doua numere: 9 si 16 si vreau sa calculez produsul. Aceasta operatie ia foarte putin timp, si gasesc imediat rezultatul, 144.

Dar, sa zicem ca ma gandesc la 144 si trebuie sa gasesc perechea de numere care impartite dau 144. Sau, calculez 3 la puterea 6 si gasesc imediat 729. Dar daca am numarul 729 si vreau sa aflu cele doua numere intregi x si y care logx * y = 729, va lua mult timp sa gasesc toate solutiile posibile si sa aleg perechea initiala.

Cele două exemple de mai sus reprezinta doua din perechile functionale ce sunt folosite in criptarea cu cheie publică, usurinta multiplicari si exponentiala, fata de dificultatea calcularii factorizarii si logaritmarii. Criptarea cu cheie publică normala foloseste doua chei ce se afla intr-o anume relatie matematica dar cunoasterea uneia dintre chei nu presupune aflarea usoara a celeilalte. O cheie este utilizata pentru criptarea textului normal, iar cealalta este folosite pentru decriptarea textului criptat. Important este faptul ca este nevoie de amandoua cheile pentru a functiona mecanismul. Pentru ca este nevoie de mai multe chei, acest tip de criptare se mai numeste si criptare asimetrica.

In criptarea cu cheie publică, una din cheii este proiectata sa fie cheia publica si poate fi publicata oriunde. Cealalta cheie este proiectata sa fie cheia privata si nu trebuie facuta cunoscuta altor persoane. Aceasta metoda poate fi folosita si pentru a dovedi cine a trimis un mesaj.

Sa zicem ca Alice cripteaza un mesaj folosind cheia sa privata, Bob primeste textul criptat pe care il decripteaza folosind cheia publica a Alice, astfel stie ca ea a criptat mesajul, iar ea nu poate nega aceasta.

Prima, si cea mai importanta, implementare a criptarii cu cheie publică este RSA, numita dupa cai trei matematicieni de la MIT care au dezvoltat-o - Ronald Rivest, Adi Shamir si Leonard Adleman. RSA este folosita in sute de produse software si poate fi folosita pentru schimbul de chei, semnaturi digitale saau criptarea unor blocuri mici de date. RSA foloseste un bloc de criptare de dimensiune variabila, iar cheia este si ea de lungime variabila. Perechea de chei este derivata dintr-un numar foarte mare, n, care este produsul a doua numere prime alese dupa anumite reguli speciale, aceste numere pot avea fiecare mai mult de 100 de cifre, numarul n avand mult mai multe cifre. Cheia publica contine numarul n si un derivat al unuia dintre factorii folositi pentru determinarea lui n; astfel un atacator nu poate determina factorii primi ai lui n (deci nu poate afla cheia privata) numai din aceste informatii din acest motiv, algoritmul RSA este atat de sigur. Unele descrieri eronate ale PKC spun ca siguranta algoritmului RSA este determinata de faptul ca este dificila factorizarea unui numar prim foarte larg, ceea ce este eronat, din motiv ca orice numar prim, indiferent de marime, are numai doi factori.
Totusi abilitatea computerelor de a factoriza numere foarte mari, astfel putand fi folosite pentru atacarea metodelor de tipul RSA, creste continuu si sistemele de data recenta pot gasi factorii primi ai numere cu aproape 140 de cifre. Protectia algoritmului RSA consta si in faptul ca utilizatorii pot creste marimea cheii anuland astfel cresterea performantei procesoarelor.

O alternativa a algoritmului RSA este metoda Diffie-Hellman care este folosita doar pentru schimbul de chei. Este folosit in combinatie cu Standardul Semnaturii Digitale al NIST (DSS - Digital Signature Standard).

Metoda RSA folosește produsul dintre două numere prime mari pentru a cripta si descripta, făcând atât criptarea cheiei publice cât și semnatura digitală. Securitatea acestei metode se bazează pe dificultatea descompunerii numerelor mari, o problemă la care nu a fost gasită o soluție practicabilă până în prezent.

Merkle-Hellman este unul dintre primele criptosisteme cu cheie publică, inventat de Ralph Merkle și Martin Hellman în 1978. Deși ideile de la baza acestuia sunt mai elegante și mai simple decât cele ale criptosistemului RSA, a fost spart. Sistemul Merkle-Hellman se bazează pe problema sumelor de submulțimi (un caz special al problemei rucsacului): dată o listă cu numere și un alt număr, care este suma unei submulțimi a listei de numere, determinați submulțimea. În general,

această problemă este considerată a fi NP-completă; dar există nişte cazuri uşoare care pot fi rezolvate eficient. Schema Merkle-Hellman este bazată pe transformarea unui caz uşor într-unul dificil, şi invers.

Schimbul de chei Diffie-Hellman

Metoda schimbului de chei Diffie-Hellman, cunoscută şi ca metoda de distribuţie a cheilor publice, poartă numele a doi specialişti de la Standford University, Whitfield Diffie şi Martin Hellman. În anul 1976, ei au inventat o metodă prin care două părţi pot cădea de comun acord să comunice prin mesaje secrete fără să fie nevoie de o terţă parte, de un schimb off-line sau de transmiterea vreunei valori secrete între ele.

Independent, Ralph Merkle a venit cu o soluţie de distribuţie a cheilor publice, numai că metoda propusă implica substanţiale cheltuieli pentru efectuarea calculelor şi a transmisiei. Varianta realizată de Diffie şi Hellman a fost numită sistemul distribuţiei cheilor publice sau al schimburilor de chei publice.

Metoda Diffie-Hellman se bazează pe conceptul perechii de chei publică privată. Protocolul începe cu fiecare parte care generează independent câte o cheie privată. În pasul următor, fiecare calculează câte o cheie publică, aceasta fiind o funcţie matematică a cheilor private respective. Urmează schimbul de chei publice. În final, fiecare dintre cele două persoane calculează o funcţie a propriei chei private şi a cheii publice a celeilalte persoane. Matematica este cea care va face să se ajungă la aceeaşi valoare, care este derivată din cheile lor private. Ele vor folosi valoarea ca pe cheie a mesajului. Diffie şi Hellman folosesc exponenţierea în aritmetica modulară pentru a calcula cheile publice şi cheia mesajului. Aritmetica modulară este ca şi aritmetica standard, cu excepţia faptului că foloseşte numere numai în intervalul 0 la N, numit modulo. Atunci când o operaţie produce un rezultat care este mai mare sau egal cu N, N este scăzut repetat din rezultat până când valoarea se încadrează în intervalul 0 la N-1 (ca şi cum s-ar împărţi la N şi se ia în seamă restul).

De exemplu, 3+4 mod 5 = 2. Dacă rezultatul este negativ, N se adaugă acestuia până când se va încadra în intervalul 0 la N-1.

De exemplu, 3-8 mod 7 = -5 mod 7 = 2.

În aritmetica modulară, exponenţierea este o funcţie într-un singur sens. Aceasta înseamnă că este uşor de calculat un număr y = gx mod N pentru o valoare

secretă x, însă este mult mai dificil să se calculeze x din y, dacă numerele sunt suficient de mari, ca de exemplu o lungime de câteva sute de cifre (noi presupunem că g şi N sunt cunoscute).

Exemplul 1

Să presupunem că p = 7, g = 3, cheia lui Alice xa = 1 şi a lui Bob xb = 2
Vom avea:

- Alice calculează cheia sa publică: ya = gxa mod p = 31 mod 7 = 3
- Bob calculează cheia sa publică: yb = gxb mod p = 32 mod 7 = 2
- Alice calculează K = ybxa mod p = 21 mod 7 = 2
- Bob calculează K = yaxb mod p = 32 mod 7 = 2

sau

K = gxa*xb mod p = 32×1 mod 7 = 9 mod 7 = 2.

Exemplul 2

Să presupunem că p = 5, g = 4, cheia lui Alice xa = 3 şi a lui Bob xb = 2

- Alice calculează cheia sa publică: ya = gxa mod p = 43 mod 5 = 4
- Bob calculează cheia sa publică: yb = gxb mod p = 42 mod 5 = 1
- Alice calculează K = ybxa mod p = 13 mod 5 = 1
- Bob calculează K = yaxb mod p = 42 mod 5 = 1

sau

K = gxa*xb mod p = 43×2 mod 5 = 4096 mod 5 = 1.

SEMNĂTURA ELECTRONICĂ

O semnătură digitală reprezintă o informaţie care îl identifică pe expeditorul unui document. Semnatura digitală este creată prin criptarea conţinutului documentului, folosind cheia criptografică a expeditorului. Aceasta face ca semnătura să fie unică atât pentru fişier cât şi pentru deţinătorul cheii. Orice modificări aduse documentului afectează semnătura, oferindu-se astfel atât integritate cât şi autentificare.

Semnăturile digitale utilizează criptarea asimetrică, în care se foloseşte o cheie pentru a crea semnătura şi o altă cheie, legată de prima, pentru a o verifica. Într-un sistem de securitate a cheii publice, toţi participanţii au nevoie de propria cheie de semnare, sau cheie privată. Cheia publică este raspandită şi identificată de către certificatele digitale. Certificatele sunt emise de terţi de încredere, cunoscuţi sub numele de autorităţi de certificare (AC), care îşi asumă responsabilitatea pentru identificarea utilizatorilor şi pentru acordarea cheilor.

AC-urile sunt deseori administrate de companii care sunt în măsură să garanteze pentru dreptul de desfăşurare a afacerii. În mod asemănător, companiile mari pot folosi AC-uri interne organizaţionale pentru a identifica personalul şi funcţia fiecăruia, în scopul autentificării tranzacţiilor de comerţ electronic.

Certificatul digital este utilizat pentru o gamă variată de tranzacţii electronice ce include e-mail, comerţul electronic, transferul electronic de fonduri. În mod obişnuit, comerţul electronic impune şi utilizarea unui certificat pentru securizarea serverului.

Certificatul digital reprezintă un instrument în stabilirea unui canal securizat pentru comunicarea informaţiilor confidenţiale. Doar criptarea nu este suficientă şi nu oferă suficiente informaţii legate de identitatea celui care trimite informaţii criptate. Fără o protecţie specială va asumaţi anumite riscuri în tranzacţiile online. Certificatul digital rezolvă această problemă, furnizând o modalitate electronica pentru verificarea identităţii unui individ. Folosit în completarea criptării, certificatul digital furnizează o soluţie completă de securizare, confirmând identitatea tuturor părţilor implicate într-o tranzacţie.

În mod asemănător, un server securizat trebuie să posede propriul certificat digital pentru a demonstra utilizatorilor că acel server este utilizat chiar de organizaţia respectivă şi informaţiile furnizate sunt legale.

O semnătură digitală pentru documentele electronice este echivalentă cu o semnătură olografa pentru documentele tipărite. Semnătura reprezintă un eşantion de date care demonstrează că o anumită persoană a scris sau a fost de acord cu acel document căruia i s-a ataşat semnătura.

De fapt, o semnătura digitală furnizează un grad mult mai mare de securizare decat semnătura olografă. Destinatarul mesajului semnat digital poate verifica atât faptul ca mesajul original aparţine persoanei a cărei semnătură a fost ataşată cât şi faptul ca mesajul n-a fost alterat, intenţionat sau accidental, de când a fost semnat. Mai mult, semnătura digitală nu poate fi negata; semnatarul documentului nu se poate disculpa mai târziu invocând faptul că a fost falsificată.

Cu alte cuvinte, semnăturile digitale permit autentificarea mesajelor digitale, asigurând destinatarul de identitatea expeditorului şi de integritatea mesajului. Ceea ce **nu asigură** semnătura digitală este **confidenţialitatea mesajului** – deoarece criptarea are loc doar la nivelul hash-ului şi nu a documentului în sine, dacă se doreşte acest lucru putându-se folosi softuri speciale – şi **momentul semnării documentului** (care este dat de data sistemului pe care a fost salvat, în condiţiile în care aceasta este în perioada de valabilitate a certificatului digital) – pentru aceasta din urmă putându-se folosi un serviciu de marcă temporală.

Cine si pentru ce foloseste semnatura electronica ?

Semnatura electronica se aplica unui Document in format electronic, redactat sau obtinut prin intermediul programelor informatice, ca de exemplu:

1. Documente MS Word (contracte, documente, notificari, retete medicale, etc.) Spread-sheet Excel (rapoarte financiare, state de plata a salariilor, note de comanda, etc.)
2. Documente Adobe Acrobat (la fei ca si cele in MS Word, insa superioare ca si aspect si securitate a integritatii informatiilor)
3. Fotografii digitale (constatari, mostre, comanda de produse, etc.)
4. Programe expert de gestiune a documentelor (documente ale administratiei publice si locale, documente care urmeaza sa fie arhivate in cadrul arhivelor electronice din cadrul unor Data Centre, etc.)
5. Documente rezultate din prelucrarea datelor contabile si fiscale (declaratii de impunere, documente de plata catre Bugetul de Stat, etc.)

Efectul produs prin aplicarea semnaturii electronice este asimilat semnaturii olografe si stampilei (daca e cazul) aplicate unui Document in forma fizica. Semnatura electronica este un instrument deosebit de util si puternic totdata, care ofera urmatoarele facilitati:

1. Permite semnatarilor unui Document sa oficializeze acest act chiar daca se afla la mare distanta geografica fata de locul unde trebuie semnat documentul.

2. Pe de alta parte, un Document semnat electronic se supune clauzei juridice de nonrepudiere, adica semnatarul nu poate sustine ca nu a semnat acel Document.

3. Prin faptul ca orice modificare, cat de mica, a documentului semnat digital duce automat la anularea semnaturii, destinatarul are convingerea ca documentul a ajuns in stare nealterata.

Orice Document semnat digital poate fi tiparit la imprimanta, putand contine informatia că documentul original e cel electronic si ca acesta poarta semnatura electronica a semnatarului.

Practic, orice domeniu de activitate are nevoie de facilitatea transmisiei electronice a informatiilor, indiferent de natura acestora. Prin aplicarea semnaturii electronice pe documentele care impun autenfificarea prin semnatura si sigiliu (celebrul SS de pe documentele tipizate), se realizeaza trecerea la noua era informationala.

Se poate imagina trecerea de la birourile prafuite, inecate de dosare, contracte necitite Inca, scrisori care asteapta de mult un raspuns, faxuri si alte hartii poate niciodate identificate, la biroul modem, unde calculatorul preia activitatea de rutiina si de organizare a fluxurilor de informatii, unde functionarul plictisit si blazat, temator in fata revolutiei tehnologice, este inlocuit de persoane dinamice, obisnuite si stimulate sa lucreze in echipa folosind eficient tehnologia inteligenta de organizare si comunicatie, care de fapt formeaza componenta vitala a organizatiei in atingerea obiectivelor propuse.

Se poate imagina noua viziune manageriala, unde rutina, dezordinea, dezorganizarea sunt inlocuite de tehnologie si inteligenta, unde locul curierilor si a tot felul de agenti sau servicii de mesagerie de multe ori scumpe si neperformante este inlocuit de noua tehnologie a comunicatiilor electronice si a gestiunii inteligente si performante a informatiei

De ce avem nevoie de un certificat digital?

Magazinele virtuale, transferurile electronice bancare şi alte servicii electronice devin, pe zi ce trece, instrumente obişnuite, convenabile şi flexibile şi pot fi utilizate chiar de acasă.

Preocupările legate de confidenţialitate şi securitate pot fi prevenite apelând la acest mijloc nou de comunicare. Doar criptarea nu este suficientă şi nu oferă suficiente informaţii legate de identitatea celui care trimite informaţii criptate.

Fără o protecţie specială va asumaţi anumite riscuri în tranzacţiile online. Certificatul digital rezolvă această problemă, furnizând o modalitate electronica pentru verificarea identităţii unui individ. Folosit în completarea criptării, certificatul digital furnizează o soluţie completă de securizare, confirmând identitatea tuturor părţilor implicate într-o tranzacţie.

În mod asemănător, un server securizat trebuie să posede propriul certificat digital pentru a demonstra utilizatorilor că acel server este utilizat chiar de organizaţia respectivă şi informaţiile furnizate sunt legale.

REPREZENTAREA NUMERICĂ A IMAGINILOR

O imagine este o suprafaţă de obicei dreptunghiulară caracterizată, la nivelul oricărui punct al ei, de o anumită culoare. Ideal, culoarea variază continuu în oricare direcţie. Din păcate, în sistemele numerice, nu se pot utiliza mărimi care variază continuu, ci doar forma discretizată a acestora.

Astfel, o imagine trebuie să fie discretizată înainte de a se pune problema reprezentării numerice. Discretizarea constă în împărţirea imaginii într-un caroiaj asemănător unei table de şah. Fiecare porţiune de imagine delimitată de acest caroiaj va fi considerată ca având o culoare uniformă – o medie a culorii existente pe această secţiune. Aceste secţiuni mai sunt denumite şi pixeli sau puncte, numărul acestora definind rezoluţia imaginii.

De exemplu, pentru o imagine oarecare care are o rezoluţie de 640x480 pixeli, înseamnă că pe suprafaţa acesteia s-a definit un caroiaj care o împarte pe orizontală în 640 de secţiuni iar pe verticală, în 480. Pasul următor îl constituie găsirea unei reprezentări pentru culoare.

Orice culoare poate fi descompusă în trei culori primare (de exemplu roşu-R, verde-G şi albastru-B), cu alte cuvinte orice imagine poate fi obţinută prin suprapunerea aditivă a trei radiaţii luminoase având aceste trei culori şi intensităţi diferite. Deci, pentru a reprezenta numeric o culoare, este suficient să se reprezinte intensităţile luminoase ale celor trei culori primare.

Dacă se alocă câte 8 biţi pentru fiecare componentă, se pot coda 256 nivele de intensitate, astfel, absenţa culorii (intensitate zero) se codifică prin valoarea 00000000 în binar sau 00 în hexazecimal, iar intensitatea maximă, prin cea mai mare valoare ce poate fi reprezentată pe 8 biţi, şi anume, 11111111 în binar sau FF în hexazecimal. Această reprezentare, însă, ţine mai mult de modalităţile tehnice de captare şi reproducere a imaginii şi mai puţin de mecanismul fiziologic de percepere a culorii. Prin diferite experimente s-a constatat că din punct de vedere al capacităţii de percepere a detaliilor, ochiul este mai sensibil la intensitatea luminoasă a culorii decât la nuanţă. Din acest motiv prezintă interes o altă modalitate de reprezentare a culorii care să ţină cont de această observaţie, un exemplu fiind reprezentarea YUV utilizată în televiziunea în culori. În cazul acestei reprezentări, componenta Y corespunde intensităţii luminoase percepute pentru respectiva culoare (coeficienţii 0,30, 0,59 şi0,11 reprezintă strălucirile relative la alb ale celor trei culori primare roşu, verde şi, respectiv, albastru). Această componentă mai este întâlnită şi sub numele de luminanţă. Componentele U şi V sunt cele care definesc nuanţa culorii, din acest motiv, sunt denumite componente de crominanţă. Acestea se calculează ca diferenţa dintre componenta roşie, respectiv albastră, şi cea de luminanţă. Avantajul reprezentării YUV este acela că separă componenta de luminanţă, pentru care ochiul

este foarte sensibil la detalii, de componentele de nuanţă pentru care sensibilitatea este mai redusă. Acest lucru face posibilă reducerea informaţiei asociate unei imagini prin utilizarea unei rezoluţii mai reduse pentru componentele de crominanţă.

În cazul televiziunii în culori se realizează o "compresie" prin limitarea benzii de frecvenţă alocate semnalelor de crominanţă (de exemplu în sistemul PAL semnalele U şi V au o bandă de 1,3MHz faţă de semnalul Y care are o bandă de 6MHz).

Reprezentarea imaginii în format necompresat

O imagine se reprezintă ca o matrice de puncte (de obicei de forma pătrată), fiecare punct fiind caracterizat de o culoare. Se porneşte de la observaţia că orice culoare poate fi obţinută prin amestecul în diferite proporţii a trei culori de bază (culori primare). În practică se utilizează ROŞU (R), VERDE (G) şi ALBASTRU (B). Intensitatea luminoasă a unei culori primare poate fi reprezentată numeric sub forma unui întreg de 8 biţi, valoarea 0 corespunzând intensităţii nule iar cea maximă (255) intensităţii maxime. În acest fel, o culoare va fi reprezentată numeric printr-un triplet de întregi pe 8 biţi (R,G,B). De exemplu culoarea GALBEN va avea o reprezentare de forma (255,255,0). În aceste condiţii imaginea se reprezintă sub forma unei matrice IM(N_x,N_y) unde N_x reprezintă numărul de puncte pe orizontală şi N_y este nuămrul de puncte pe verticală, iar elementele matricei sunt tripleţi de întregi pe 8 biţi de tip (R,G,B).

Metode şi abordări ale compresiei imaginii

În continuare se reiau câteva metode folosite în compresie, evidenţiind aplicabilitatea lor în compresia de imagini.

1. Cuantizarea scalară poate fi folosită pentru a compresa imagini, dar performanţele ei sunt mediocre. De exemplu, o imagine cu 8 biţi/pixel poate fi compresată prin cuantizare scalară eliminând cei mai nesemnificativi patru biţi ai fiecărui pixel. Aceasta conduce la o rată de compresie de 0,5, care pe lângă faptul că nu este semnificativă, determină în acelaşi timp şi reducerea numărul de culori (sau nuanţe de gri) de la 256 la doar 16. O astfel de reducere nu numai ca descreşte pe ansamblu calitatea imaginii reconstruite, dar poate chiar crea benzi de diferite culori, un efect observabil şi deranjant care este ilustrat aici.

2. *Cuantizarea vectorială* poate fi folosită cu mai mult succes pentru a compresa imagini.

3. *Metodele statistice* funcţionează mai bine când simbolurile ce trebuie compresate au probabilităţi diferite. O secvenţă de intrare în care mesajele au aceeaşi probabilitate nu se va compresa eficient. În acest sens, într-o imagine alb-negru sau color în tonuri continue, diferitele culori sau nuanţe de gri se dovedesc de multe ori a avea aproximativ aceleaşi probabilităţi. De aceea metodele statistice nu sunt o alegere bună pentru compresia unor astfel de imagini, şi sunt necesare noi abordări.

Imaginile cu discontinuităţi de culoare, în care pixeli adiacenţi au culori foarte diferite, se compresează mai bine cu metodele statistice, dar în acest caz nu este uşoară predicţia pixelilor. Metodele tradiţionale sunt nesatisfăcătoare pentru compresia de imagini, astfel încât au fost necesare abordări noi, care, deşi diferite, se bazează pe eliminarea redundanţei din imagine, folosind *principiul compresiei de imagine*. Dacă se selectează aleator un pixel dintr-o imagine, există o probabilitate mare ca vecinii săi să aibă aceeaşi culoare sau culori foarte apropiate.

Compresia de imagine este, deci, bazată pe faptul că pixelii învecinaţi sunt puternic corelaţi. Această corelare se numeşte şi redundanţă spaţială. În general, metodele de compresie pentru imagini sunt proiectate pentru un tip particular de imagine şi în continuare se prezintă câteva din aceste metode specifice. Imaginile particulare vizate sunt imagini cu două nivele, imagini cu tonuri de gri şi imagini color.

Abordarea 1. Aceasta este folosită pentru imagini cu două nivele. Un pixel dintr-o astfel de imagine este reprezentat printr-un bit. Aplicând principiul compresiei de imagine asupra unei imagini cu două nivele, înseamnă că pixelii învecinaţi ai unui pixel P tind a fi identici cu P. Astfel, are sens folosirea unei codări RLC (Run length coding) pentru a compresa o astfel de imagine. O metodă de compresie pentru o astfel de imagine o poate scana în ordinea rastrului (rând cu rând), calculând lungimile şirurilor de pixeli albi şi negri. Aceste lungimi sunt codate prin coduri de lungime variabilă şi sunt înscrise în secvenţa compresată. Un exemplu de astfel de metodă este compresia facsimil. Ar trebui accentuat faptul că aceasta este doar o abordare a imaginilor cu două nivele. În practică, detaliile metodelor particulare diferă, în funcţie de aplicaţie. De exemplu, o metodă poate scana imaginea coloană cu coloană, sau în zig-zag, sau o poate scana regiune cu regiune.

Abordarea 2 se aplică, de asemenea, pentru imagini cu două nivele. Principiul compresiei de imagine spune că vecinii unui pixel tind a fi similari lui. Se poate extinde acest principiu şi concluziona că dacă pixelul curent are culoarea c (unde c este ori alb, ori negru), atunci pixelii de aceeaşi culoare anteriori şi următori tind să aibă aceiaşi vecini imediaţi. Această abordare urmăreşte n vecini apropiaţi ai pixelului curent şi îi consideră ca un număr de n biţi. Acest număr se numeşte

contextual pixelului. În principiu pot exista 2^n contexte, dar datorită redundanţei imaginii, distribuţia lor este neuniformă. Unele contexte ar trebui să fie foarte frecvente, iar celelalte, rare. Codorul numără de câte ori a fost găsit deja fiecare context pentru un pixel de culoarea c, şi asignează corespunzător probabilităţi acestor contexte. Dacă pixelul curent are culoarea c şi contextul său are probabilitatea p, codificatorul poate folosi coduri aritmetice adaptive pentru a codifica pixelul cu acea probabilitate. Această abordare este folosită de standardul JBIG (Joint Bi-level Processing Group).

În continuare, se consideră imagini în nuanţe de gri. Un pixel dintr-o astfel de imagine este reprezentat prin n biţi şi poate avea una din 2^n valori. Aplicarea principiul compresiei de imagine asupra unei imagini în nuanţe de gri implică faptul că vecinii imediaţi ai unui pixel P tind a fi similari cu P, însă nu în mod necesar identici cu el. Astfel, nu mai poate fi folosită codarea RLE (run length encoding) pentru compresia unei astfel de imagini, ci sunt folosite următoarele două abordări.

Abordarea 3. Se separă imaginea în tonuri de gri în n imagini pe două nivele şi apoi se compresează fiecare cu un cod RLE instantaneu. Principiul compresiei de imagini, în acest caz, s-ar formula prin afirmaţia că doi pixeli adiacenţi care sunt similari în imaginea în tonuri de gri vor fi identici în cele mai multe imagini cu două nivele.

Abordarea 4. Se foloseşte contextului unui pixel pentru a prezice valoarea sa. Contextul unui pixel este dat de valorile câtorva dintre vecinii săi. Se examinează câţiva vecini ai unui pixel P, se calculează o medie A, a valorilor lor, şi se prezice că P va avea valoarea A. Principiul compresiei de imagini spune că predicţia va fi corectă în cele mai multe cazuri, aproape corectă în multe cazuri, şi complet greşită în puţine cazuri. Valoarea prezisă a pixelului P reprezintă informaţia redundantă în P, astfel încât se calculează diferenţa : $\Delta = P - A$, şi se asignează coduri instantanee de lungime variabilă pentru diferitele valori Δ. Dacă P poate lua valori de la 0 la m-1, atunci Δ va avea valori în intervalul [-(m-1),+(m-1)], şi numărul cuvintelor de cod necesare este 2m - 1. Experimente cu un număr mare de imagini sugerează că valorile lui Δ tind să fie distribuite după distribuţia Laplace. O metodă de compresie ar putea să folosească această distribuţie pentru a asigna o probabilitate fiecărei valori a lui Δ, şi apoi să se folosească codarea aritmetică pentru a coda eficient valorile Δ. Acesta este principiul metodei progresive multinivel MLP. Contextul unui pixel poate fi constituit din unul sau doi din vecinii săi imediaţi. Dacă, însă, se consideră mai mulţi pixeli vecini în obţinerea contextului, se pot obţine rezultate mai bune. Media A într-un astfel de caz ar trebui ponderată cu vecinii apropiaţi, care au o pondere mai mare. Pentru ca decodorul să poată decoda o imagine, ar trebui să poată calcula contextual fiecărui pixel. Acest lucru înseamnă că în context ar trebui să fie

incluşi doar pixelii care au fost deja codaţi. Dacă imaginea este scanată în ordinea rastrului, contextul ar trebui să conţină doar pixeli localizaţi deasupra pixelului curent sau pe acelaşi rând cu el, la stânga.

Abordarea 5. Se aplică o transformare valorilor pixelilor, şi se codează valorilor transformate. Se reaminteşte că pentru a realiza compresia, trebuie redusă sau eliminată redundanţa. Redundanţa unei imagini este cauzată de corelaţia dintre pixeli, deci transformând pixelii într-o reprezentare în care aceştia sunt decorelaţi, se elimină redundanţa. De asemenea este posibil ca o transformare să fie apreciată în funcţie de entropia imaginii. Într-o imagine puternic corelată, pixelii tind a avea valori echiprobabile, ceea ce duce la o entropie maximă. Dacă pixelii transformaţi sunt decorelaţi, anumite valori de pixeli devin mai frecvente, având astfel probabilităţi mari, în timp ce alte valori sunt rare, fapt ce conduce la o entropie mică. Cuantizând valorile transformate, se poate produce o compresie cu pierdere de informaţie, eficientă, a imaginii. Se doreşte ca valorile transformate să fie independente, deoarece codarea valorilor independente face mai simplă construirea unui model statistic. În cazul imaginilor în culori, un pixel este constituit din trei componente de culoare, roşu, verde şi albastru. Majoritatea imaginilor color sunt ori în tonuri continue, ori în tonuri discrete.

Abordarea 6. Principiul acestei abordări constă în separarea unei imagini color în tonuri continue în trei imagini în tonuri de gri şi compresia fiecăreia din ele separat, folosind abordările 2, 3 şi 4. Pentru o imagine în tonuri continue, principiul compresiei de imagini implică faptul că pixelii adiacenţi au culori similare, dacă nu chiar identice. Totuşi, culori similare nu înseamnă valori similare ale pixelilor. Se consideră, de exemplu, valori pe 12 biţi ale pixelilor, în care fiecare componentă de culoare este exprimată în patru biţi. Astfel, cei 12 biţi 1000|0100|0000 reprezintă un pixel a cărui culoare este o mixtură de opt unităţi de roşu (aproape 50%, din valoarea maximă de 15 unităţi), patru unităţi de verde (circa 25%), şi deloc albastru. Se consideră doi pixeli adiacenţi cu valorile 0011|0101|0011 şi 0010|0101|0011. Aceştia au culori similare, din moment ce doar componentele lor roşii diferă printr-o unitate. Cu toate acestea, când se consideră ca numere de 12 biţi, cele două numere 001101010011 şi 001001010011 sunt diferite, pentru că diferă într-un bit cu pondere semnificativă. O caracteristică importantă a acestei abordări este folosirea unei reprezentări tip luminanţă – crominanţă, YUV, în loc de reprezentarea comună RGB. Avantajul acestei reprezentări este că ochiul este sensibil la modificări mici ale luminanţei, dar nu şi la ale crominanţei. Aceasta permite pierderea unei cantităţi considerabile de date în componentele de crominanţă, fără o pierdere vizibilă de calitate.

Abordarea 7. O abordare diferită este necesară pentru imaginile în tonuri discrete. Se reaminteşte că o astfel de imagine conţine regiuni uniforme care pot apărea de mai multe ori într-o imagine. Un exemplu îl constituie o pagină scrisă la calculator care constă din text şi icoane. Fiecare caracter de text şi fiecare icoană este o regiune, şi fiecare regiune poate apărea de mai multe ori în imagine. O modalitate posibilă de compresie a unei astfel de imagini este scanarea sa, identificarea regiunilor, şi găsirea regiunilor care se repetă. Dacă o regiune B este identică cu o regiune A deja găsită, atunci B poate fi compresată prin înregistrarea unui pointer corespunzător lui A în secvenţa compresată. Metoda descompunerii în blocuri este un exemplu de implementare a acestei abordări.

Abordarea 8. Se împarte imaginea în regiuni (care se suprapun sau nu) şi se compresează prin procesarea părţilor una câte una. Se presupune că următoarea parte de imagine neprocesată este partea cu numărul n. Se încearcă regăsirea ei în părţile $1 \div n-1$, care au fost deja procesate. Dacă partea n poate fi exprimată, de exemplu, ca o combinaţie a unor părţi anterioare scalate şi rotite, atunci doar cele câteva numere care specifică combinaţia trebuie salvate, şi partea n poate fi ignorată. Dacă partea n nu poate fi exprimată ca o combinaţie de părţi deja procesate, aceasta este procesată şi salvată în secvenţa compresată. Această abordare este baza diferitelor metode fractale pentru compresia de imagini. Se aplică principiul compresiei de imagine asupra părţilor de imagine, în loc de pixelii individuali. Aplicat în acest fel, principiul afirmă că imaginile ce urmează a fi compresate au un anumit 14 volum de auto-similaritate, adică părţi de imagine sunt identice sau similare cu întreaga imagine sau cu alte părţi.

Transformări folosite în compresia imaginilor

Conceptul matematic de transformare este important în multe domenii, printre care şi cel al compresiei de imagini. O imagine poate fi compresată prin transformarea pixelilor săi (care sunt corelaţi) într-o reprezentare unde aceştia sunt decorelaţi. Compresia este obţinută dacă valorile noi sunt mai mici, în medie, decât cele originale. Compresia cu pierdere de informaţie poate fi obţinută prin cuantizarea valorilor transformate. Decodorul primeşte valorile transformate din secvenţa compresată şi reconstruieşte datele originale (exacte sau aproximate), prin aplicarea transformării inverse. Transformările discutate în această secţiune sunt ortogonale.

Termenul de decorelare se referă la faptul că valorile transformate sunt independente unele de altele. Ca urmare, ele pot fi codate independent, ceea ce face mai simplă construirea unui model statistic. O imagine poate fi compresată, dacă reprezentarea sa are redundanţă. Redundanţa în imagini derivă din corelarea pixelilor. Dacă se transformă imaginea într-o reprezentare în care pixelii sunt decorelaţi, se elimină redundanţa şi imaginea a devenit în totalitate compresată.

Compresia JPEG (Joint Photographers Experts Group)

Domeniul compresiei (codării) de imagini este legat de minimizarea numărului de biţi necesari pentru a reface o imagine, cu aplicaţii în special în transmisia şi stocarea imaginilor. Aplicaţiile din domeniul transmisiilor de imagini se întâlnesc în televiziunea radiodifuzată, comunicaţiile spaţiale, radar şi sonar, reţele de telecomunicaţii, transmisii fax, teleconferinţe etc.

Compresia imaginilor este esenţială din punct de vedere al memorării (stocării) imaginilor în aplicaţii de imagistică medicală, în tehnica video digitală, pentru realizarea documentelor multimedia etc. Noile tehnologii de compresie a imaginilor oferă o soluţie posibilă pentru integrarea aplicaţiilor de imagini şi video digitale.

Ratele de compresie au ajuns în prezent până la 1:100, depinzând de calitatea imaginii refăcute. Tehnica de compresie nu este suficientă pentru a putea rezolva problemele care apar în aplicaţiile multimedia. Pentru a putea realiza portabilitatea aplicaţiilor de imagini şi secvenţe video digitale pe mai multe sisteme, este necesară implementarea unor standarde pentru compresia datelor multimedia. Aceste standard stabilesc modalităţile de stocare şi transmisie a datelor compresate

în vederea posibilităţii utilizării lor. Cel mai utilizat standard de compresie a imaginilor statice este standardul JPEG, creat de Joint Photographics Experts Group. Metoda de compresie este de tip "cu pierdere", fiind concepută astfel încât să se profite de limitările în percepţia video a ochiului uman. Acest standard permite setarea raportului calitate/compresie şi lucrează cu aceleaşi nivele de culoare, în număr de 24 (16,7 milioane deculori), indiferent de numărul total de culori din imagine. În momentul de faţă este unul dintre cele mai frecvent întâlnite formate de fişiere grafice.

Formatul JPEG este recomandat pentru afişarea de imagini redate cu o foarte mare varietate de culori sau pentru imagini de precizie fotografică. JPEG foloseşte o tehnică de compresie variabilă, care are drept rezultat obţinerea de fişiere foarte mici în comparaţie cu alte formate.

Standardul JPEG se bazează pe transformarea informaţiei primare din domeniul timp în domeniul frecvenţă. Este cunoscut faptul că imaginile sunt puternic corelate spaţial, adică un pixel de imagine conţine informaţii şi despre pixelii vecini. Corelaţia spaţială ce caracterizează imaginile reprezintă redundanţă din punct de vedere informaţional şi se diminuează prin transformări matematice care au rolul de a concentra energia imaginii în cât mai puţine elemente.

Transformările matematice din domeniul timp în domeniul frecvenţă nu reprezintă în sine compresie de date. Abia operaţiunile ce urmează, şi anume, cuantizarea şi codarea entropică reprezintă compresie de date. Reducerea redundanţei spaţiale se face atât pentru imaginea sursă originală, cât şi pentru eroarea reziduală, aşa cum se va vedea în cele ce urmează. La refacerea imaginilor, după ce acestea au fost compresate JPEG, cantitatea de informaţie este mai mică decât cea iniţială, fără o afectare vizibilă a calităţii. Prin transformarea imaginii din domeniul timp, (pixeli), în domeniul frecvenţă se reţin doar componentele de joasă frecvenţă ale imaginii. Componentele de frecvenţă înaltă pot fi reduse, fără o afectare deranjantă a percepţiei vizuale a imaginii. Evident că acest lucru este determinat de gradul de compresie acceptat. JPEG este o metodă sofisticată de compresie cu pierdere pentru imagini color sau alb-negru (cu scară de gri).

Un avantaj al JPEG este faptul că foloseşte mulţi parametri, permiţând astfel utilizatorului să regleze cantitatea de date pierdute şi, de asemenea, rata de compresie. Momentan reprezintă cel mai bun standard existent în materie de compresie a imaginilor statice. Standardul este implementat atât în format software cât şi hardware pentru a satisface necesităţile de prelucrare în timp real a aplicaţiilor multimedia. Creat iniţial pentru compresia imaginilor statice, standardul a fost extins şi pentru secvenţele video. Standardul realizat pentru secvenţe video se numeşte M-JPEG (Motion JPEG). Practic în cazul secvenţelor video digitale fiecare cadru este

considerat ca o imagine fixă şi compresat cu standardul JPEG. Metoda nu este cea mai eficientă din punctul de vedere al mărimii ratei de compresie, dar oferă o alternativă pentru compresia video digitală. Adesea, ochiul uman nu distinge nici o degradare a imaginii chiar la o rată de compresie de 10:1 sau 20:1.

Există patru moduri principale de operare specificate de standardul JPEG :
- modul de bază, în care fiecare componentă a imaginii este codată printr-o singură scanare stânga-dreapta, respectiv sus-jos;
- codarea expandată DCT cu pierderi, în care se realizează o codare progresivă a spectrelor imaginii de intrare;
- codarea fără pierderi, în care imaginea este codată astfel încât se garantează reproducerea exactă la decodare;
- codarea ierarhică, în care imaginea este codată la rezoluţii multiple.

Compresia progresivă JPEG poate fi obţinută folosind trei tipuri de algoritmi:
- un algoritm progresiv de selecţie spectrală;
- un algoritm progresiv de aproximări succesive;
- un algoritm progresiv combinat.

Compresia secvenţială JPEG fără pierderi

Standardul JPEG permite şi folosirea unui algoritm de compresie fără pierderi, respectiv un algoritm de compresie predictivă în locul transformării DCT.

Compresia JPEG ierarhică

Compresia ierarhică JPEG permite o reprezentare progresivă a imaginii decodate, într-un mod similar cu algoritmul progresiv, dar, în plus faţă de acesta, permite imagini codate cu rezoluţii multiple. Se creează astfel, un set de imagini compresate, începând cu imagini de rezoluţie mică şi continuând cu imagini a căror rezoluţie creşte către rezoluţia imaginii originale. Acest proces se numeşte subeşantionare sau codare piramidală.

Criptarea imaginilor vizuale

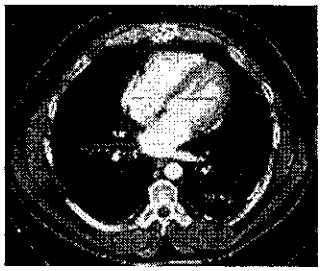
Ideea de criptografie vizuală, inspirată de procedurile criptografice bazate pe scheme de prag (scheme de partajare a secretului), este invenţia fascinantă a lui Moni Naor şi Adi Shamir şi a fost introdusă la "EUROCRYPT '94".

În forma ei cea mai simplă, ea implementează un sistem criptografic indestructibil, cu deosebirea că nu este nevoie de echipamente de calcul sofisticate pentru decriptare.

Criptografia vizuală reprezintă o idee fascinantă a doi autori de largă recunoastere în lumea criptografiei computaţionale. Deşi aparută recent, ea a căpătat o dezvoltare impresionantă. Au fost supuse cercetării metode dintre cele mai diverse, plecând de la cele clasice şi mergând până la scheme criptografice vizuale bazate pe proprietăţile de absorbţie a luminii ale diferitelor materiale, scheme criptografice bazate pe structuri generale de acces, scheme de criptare vizuală a imaginilor color s.a.

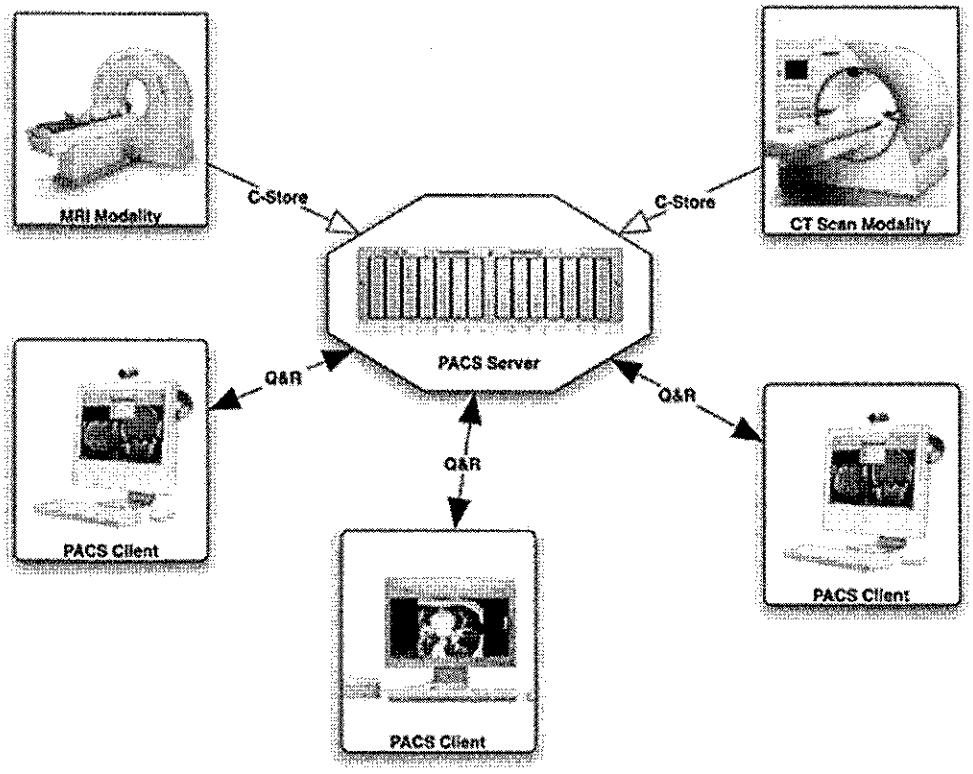
În efortul comun de a desluşi consecinţele şi aplicaţiile acestei noi paradigme întâlnim nume sonore în criptografia computaţională, dar şi în informatică în general cum ar fi: Adi Shamir, D.R. Stinson, Frank Stajano, C. Blundo, A. De Santis, D. Naccache, E. Biham şi mulţi alţii. Există, de asemenea, suficiente pagini de Web întreţinute de către pasionaţi ai criptografiei vizuale dintre care aş aminti numai pagina lui D.R. Stinson şi cea a grupului de lucru pe probleme de criptografie vizuală al studenţilor Universitţii Catolice din Leuven , Belgia, grup condus de nu mai puţin celebrul profesor Bart Preneel.

PACS



În imagistica medicală, **P**icture **A**rchiving and **C**ommunication **S**ystems (PACS) este o combinaţie hardware şi software dedicat pentru stocare pe termen scurt şi lung, recuperare, management, distribuţie şi prezentare de imagini. Imaginile electronice şi rapoartele sunt transmise digital prin intermediul PACS; aceasta

elimină necesitatea de a descărca manual, prelua sau transmite studiile realizate. Formatul universal pentru stocare şi transfer de imagini PACS este DICOM (**D**igital **I**maging and **C**ommunications in **M**edicine). Datele non-imagine, cum ar fi documentele scanate, pot fi încorporate folosind formate standard din industria de consum cum ar fi PDF (Portable Document Format), o dată ce a fost încapsulat în DICOM. Un PACS constă în patru componente majore: modalităţile de tratare a imaginii, cum ar fi CT şi MRI, o reţea securizată pentru transmiterea de informaţii pentru pacient, staţii de lucru pentru interpretarea şi vizualizarea imaginilor şi arhive pentru depozitarea şi regăsirea de imagini şi rapoarte. Combinat cu tehnologia disponibilă şi în curs de dezvoltare web, PACS are capacitatea de a oferi acces în timp util şi eficient la imagini, interpretări şi datele conexe . PACS-ul pune capăt barierelor fizice şi de timp asociate cu extragerea tradiţională pe bază de imagine de film, distribuţie şi afişare.



Compresia imaginilor binare

Ne propunem o introducere în metodele de compresie a imaginilor binare. De aceea tratarea subiectului nu se pretinde a fi exhaustiva. Urmărim familiarizarea elevilor cu diversele procedee de codare exacta dezvoltate până în prezent. Vor fi luate în considerare trei metode de codare exactă a imaginilor binare: (a) White Block Skipping (WBS), (b) Run Length Coding (RLC) si (c) Block Coding (BC). Prima metoda, WBS, va fi analizată din punctul de vedere al factorului de compresie ăi al modelului de Lanţ Markov asociat. A doua metodă, RLC, va cuprinde analiza atât a Codării Lungimii Şirurilor Independent de Culoare cât şi Codarea Lungimii Şirurilor Funcţie de Culoare. Şi în cazul RLC va fi dezvoltat modelul Lanţului Markov asociat. Pentru a treia metodă va fi calculat factorul de compresie maxim, corespunzator unei codări absolut optimale, iar apoi va fi implementat codul Huffman asociat statisticii preluate dintr-o imagine şi din mai multe imagini. Cele trei metode studiate vor fi apoi comparate din punctul de vedere al factorului de compresie. În final se vor face considerente asupra altor metode de codare a imaginilor binare şi asupra standardelor existente.

A comprima un mesaj înseamnă a păstra numai acei parametri care sânt esenţiali pentru destinatar. Ceilalţi parametri nu se transmit, respectiv nu se stochează.

Sistemele de compresie pot fi grupate în două categorii:

- sisteme care utilizează procedee care conservă entropia sursei, dar reduc redundanţa (codare exactă).
- sisteme care utilizează procedee care reduc entropia sursei (o parte din informaţie se pierde).

Imaginile binare sânt semnale discrete, bidimensionale, cu suport finit care pot lua doar două valori (negru = "0" şi alb = "1"). Aceste imagini se întâlnesc în practică sub formă de pagini tipărite, scrisori, documente, ziare, hărţi geografice, hărţi meteorologice, fişe cu amprente etc. Ca urmare transmisia şi stocarea unor astfel de imagini este întâlnită astăzi în diverse domenii de activitate începând cu uzuala, de acum, transmisie prin fax şi terminând cu bazele de date de amprente utilizate în criminalistică. Datorită volumului mare de memorie pe care îl necesită o imagine binară (un format standard de scrisoare de 8.5 x 11 inch eşantionat la 200 puncte/inch contine aprox. 3,7 Mbiţi) apar avatajoase diversele metode de compresie, care reduc timpul de transmisie, lărgimea de bandă necesară transmisiei şi necesarul de capacitate de stocare.

Compresia exactă este posibilă datorită redundanţei existente în imaginile binare. Astfel, se constată existenţa unei corelaţii bidimensionale puternice între

elementele (pixeli) alăturate ale imaginii.Unele imagini binare sânt caracterizate de zone întinse de nivel "1" (alb). De aici apare ideea utilizata de WBS, adică "sărirea blocurilor de alb".

Tehnica RLC exploatează eficient corelaţia orizontală a elemetelor imaginii prin gruparea lor în şiruri negre şi şiruri albe, rezultând deci "codarea lungimii şirurilor".

Metoda BC exploatează în plus şi corelaţia verticală a elementelor imaginii prin gruparea acestora în blocuri, de unde denumirea de "codarea pe bloc".

Factorul de compresie obţinut printr-un procedeu de codare care reduce redundanţa sursei este definit astfel:

$$Q = N_O / N_C$$

unde:

N_O este numarul total de biţi din imaginea originală

N_C este numarul total de biţi din aceaşi imagine după codare.

Notind cu "b" rata de bit, adica numarul mediu de biţi pe pixel după codare, atunci, factorul de compresie poate fi definit şi în funcţie de rata de bit:

$$Q = 1 / b$$

Relaţiile de mai sus aplicate asupra aceleiaşi imagini şi aceluiaşi procedeu de codare, conduc la acelaşi rezultat numeric.

Limita superioară a factorului de compresie este:

$$Q_{MAX} = 1 / H$$

unde H este entropia sursei de semnal, care poate fi evaluată apriori doar aproximativ, utilizând un model matematic al sursei.

Tehnica White Block Skipping (WBS)

a) Descriere

WBS este o metodă foarte simplă, recomandată pentru compresia imaginilor binare care conţin mult alb.

Fiecare linie a imaginii este divizată în blocuri de N pixeli. Pentru blocurile formate numai din pixeli albi se atribuie cuvântul de cod "0". Pentru toate celelalte blocuri se atribuie un cuvânt de cod de lungime N+1 având primul bit "1", urmat de N biţi ce constituie pattern-ul digital al blocului respectiv.

Rata de bit pentru metoda WBS cu blocuri de dimensiune N este dată de expresia:

$$b(N) = (1 - P_N + 1 / N) \text{ biţi / pixel}$$

unde P_N este probabilitatea ca un bloc cu N pixeli să conţină numai pixeli albi.

b) Modelul Lanţului Markov

Se presupune ca imaginea binară este generată de o sursă Markov de ordinul întîi. După cum se cunoaşte, un proces Markov de ordinul întîi este în mod complet descris din punct de vedere statistic de densitatea de probabilitate de ordinul al doilea. Aşadar culoarea (alb sau negru) a unui pixel din imagine va fi statistic dependentă numai de culoarea pixelului precedent din aceeaşi linie.

Tehnica Run Length Coding (RLC)

Metoda RLC presupune codarea şi transmiterea lungimii şirurilor de pixeli consecutivi de aceeaşi culoare dintr-o linie de imagine. Atunci când codarea lungimii şirurilor nu depinde de culoarea şirului curent (alb sau negru) , alocând acelaşi cod pentru şiruri de lungime egală şi culori diferite, metoda se numeste RLC Independent de Culoare. Când însă, codul alocat depinde de culoarea şirului metoda se numeşte RLC Funcţie de Culoare.

Tehnica Block Coding (BC)

Această metodă presupune împarţirea imaginii în blocuri bidimensionale de m x k pixeli. Deoarece fiecare pixel din cei mk pixeli ai unui bloc poate să ia două valori, se obţin $N = 2^{mk}$ combinaţii posibile, deci N tipuri de blocuri. Deci imaginea poate fi considerată ca fiind generată de o sursă cu $N = 2^{mk}$ simboluri.

Deoarece nu toate tipurile de blocuri sânt egal probabile se justifică implementarea unui cod Huffman adaptat statistic probabilitatilor de apariţie ale celor N simboluri ale sursei. Construcţia şi utilizarea codului Huffman este dificil de utilizat în practica pentru dimensiuni ale blocurilor mai mari decât 3x3 (tabela de cod de dimensiune $2^9 = 512$).

Prin împarţirea imaginii în blocuri, metoda BC exploateaza atât corelaţia orizontală cât şi corelaţia verticală a pixelilor, ceea ce determină, pentru anumite imagini, factori de compresie mai mari în comparaţie cu metodele precedente.

Alte metode de codare

În practică se utilizează şi alte metode de codare, a caror implementare este mai complexă, dar care determină factori de compresie mai ridicaţi. Ca metode exacte au fost dezvoltate Predictive Coding (PC) si Line-to-Line Run-Length Difference Coding (LLRLDC). Metoda PC se bazează pe calculul unor predictori şi transmiterea sau stocarea doar a erorilor de predicţie. Metoda LLRLDC exploatează corelaţia existentă între lungimile şirurilor adiacente din două linii succesive. Astfel în loc să se codeze lungimile şirurilor de pixeli de aceeasi culoare, se codează diferenţele între lungimile a două şiruri adiacente.

Metodele prezentate până acum, permit, aşa cum s-a subliniat, în condiţiile unei transmisii lipsite de erori, reproducerea exactă la recepţie a imaginii binare originale. Există însă metode care se bazează pe o codare aproximativă. Se obţine astfel o îmbunătăţire a factorului de compresie fără o degradare semnificativă a calităţii imaginii reproduse.

Standardizare

Comparaţia între performanţele diverselor metode de codare se face conform normelor CCITT (Comité Consultatif International Télégraphique et Téléphonique). Sistemele de comunicatie prin Fax respecta de asemenea recomandarile CCITT, astfel încât comunicaţia între două maşini Fax realizate de doi producători diferiţi să fie posibilă.

Standardul CCITT care cuprinde conversia digitală şi codarea documentelor Fax pentru transmisie este Standardul Grup 3 (sau G3). Standardul G3 conţine două tehnici de codare: prima exploatează pentru codare corelaţia orizontală din imaginea binară, iar a doua exploatează corelaţia bidimensională. Deoarece tehnicile din G3 sânt utilizare în transmisia datelor prin reţeaua telefonică, deci în condiţiile unei transmisii cu erori, schemele de codare prevăd introducerea de informaţie redundantă în scopul protecţiei la erori.

Documentele CCITT au densitatea de 3.86 pixeli / mm sau 7.7 pixeli / mm, dimensiunea standard A4 (210 x 298 mm) rezultând astfel 1728 x 2376 pixeli / document.

CUPRINS

BIBLIOGRAFIE

1. **Wikipedia**, enciclopedia liberă
2. **Cryptography**, Theory et Pratice, D. Stinton, Ed. Chapman & Hall/CRC 2002
3. **Criptografie cu chei publice**, A. Salomaa, Ed. Militară, 1994
4. **A Method for Obtaining Digital Signatures and Public-Key Cryptosystems**, Rivest, Shamir and Adleman, Communications of the ACM, 1978
5. **Digital Signatures using Reversible Public Key Cryptography for the Financial Services Industry**, American National Standards Institute, 1998