Universitatea Hyperion

Metode de criptografie publica si privata.

Conducator:

Prof. Ion Dumitru

Absolvent:

Horia Valentin Corcalciuc

Bucuresti – Mai 2006

Cpt. Janeway: It was downhill from there. You developed a feedback loop between your ethical and cognitive sub-routines. You were having the same thoughts over and over again. We couldn't stop it.

[...]

The Doctor: The more I think about it, the more I realise, there's nothing I could have done differently.

Star Trek Voyager, "Latent Image"

I am thnkful to Prof. Ion Dumitru for accepting to be the promotor of my work. Tthe discussions we had and his experience contributed much to the final form of the present work.

I wish to thank Prof. Freud Weierud for being so kind in taking his time and manifesting a great deal of patience when working with somebody so young. His immediate responses to my dilemas and the precision of those responses deserve many thanks.

I wish to thank my parents who are probably the only witnesses of my intelectual development. Probably also the only ones that can tell my troubles apart and most certainly those who ever did take the time to bring them to a resolution. This work in itself would not have been made true if it were not for their patience and understanding of the events revolving my interest.

Multumesc d-lui Prof. Ion Dumitru pentru ca a acceptat sa fie conducator al acestei lucrari. Discutiile si experienta sa au contribuit mult la forma finala a prezentei lucrari.

As vrea sa multumesc Prof. Freud Weierud pentru amabilitate si pentru rabdarea manifestata lucrand cu o persoana mult mai tanara. Ii sunt recunoscator pentru raspunsurile sale imediate la intrebarile mele si pentru precizia acestor raspunsuri.

Multumesc parintilor mei care sunt probabil singurii martori ai dezvoltarii mele intelectuale. Probabil deasemenea singurii care mi-au cunoscut dificultatile si s-au straduit sa ma ajute sa le rezolvam. Aceasta lucrare nu s-ar fi realizat fara rabdarea lor si intelegerea lucrurilor care m-au interesat.

Cuprins

1. Introducere

Lucrarea prezinta o trecere in revista a metodelor de criptografie de-a lungul istoriei cu accent pe metodele si procedurile moderne. Baza analizei este programul “Alph”, al carui unic autor sunt, pe care l-am lansat in prima sa varianta la 2 Sept. 2004 si poate fi gasit la adresa web http://freshmeat.net/projects/alph/ [1, 2] fiind scris in limbaj C. De la aceasta data am dezvoltat programul in mai multe etape, ultima dezvoltare (upgrade) datand din 27 decembrie 2005. Dezvoltarile successive au fost in principal dedicate introducerii de noi proceduri de criptografie dar si corectarii unor greseli semnalate de diversi utilizatori. La ora actuala, asa cum rezulta din consultarea site-ului mai sus mentionat, programul a fost vizualizat de peste 11100 de personae, downloadat de peste 3000 si sunt 24 de abonati care doresc sa primeasca automat orice noua versiune. Printre alti utilizatori, Universitatea din Hamburg a preluat programul ca material de studiu pentru studentii in informatica. Programul este facut pentru a fi utilizat in conjunctie cu programe externe care transfera date, rezultand intr-o encriptare si decriptare transparenta a informatiei. In acest fel poate fi utilizat ca filtru de mail, IRC filter, IM filter, etc. Ar mai fi de subliniat ca site-ul mai sus mentionat, primeste contributii de programe de la diversi autori din intreaga lume, dar nu le face publice decat dupa verificarea functionarii fara erori, conditiile de acceptare fiind legate de noutate, originalitate si un orizont de interes pentru potentialii utilizatori.

Cuvantul "criptologie" vine din limba greaca si este compus din "kryptos" care inseamna ascuns, si "logos" care inseamna cuvant. In felul acesta, "criptografia" se poate defini ca fiind arta de a scrie secrete si este la fel de veche ca scrisul insusi fiind folosita de-a lungul secolelor pentru a proteja comunicatiile diplomatice, militare sau private. In domeniul criptologiei se pot separa doua domenii distincte: *criptografia* si *analiza criptografica*. Criptografia cauta metode care sa asigure siguranta mesajelor pe cand analiza criptografica incearca sa descompuna mesajul prin dezasamblarea algoritmul folosit.

Preocuparea fundamentala a criptografiei este de a transforma sau a cifra un mesaj intr-o forma intermediara in care informatia originala este prezenta dar ascunsa. Astfel este posibil a transmite mesajul transformat (mesajul cifrat) fara a expune informatia pe care o

contine. Utilizand diferite transformari, putem crea diferite mesaje cifrate pentru acelasi mesaj original. Diferenta intre formele intermediare obtinute prin diverse transformari consta in interpretarea mesajului. Diferite algoritme vor produce diferite interpretari pentru acelasi mesaj cifrat. Nestiinta de a alege o interpretare anume pentru un mesaj cifrat in particular este exact rezultatul metodei prin care informatia este ascunsa.

Scopurile criptologiei se pot defini ca:

- Autentificarea datelor (integritatea datelor si autentificarea sursei acestora)
- Protejarea datelor fata de o parte terta.
- Confidentialitatea datelor.

Fiecare dintre acestea se pot defini ca servicii care pot fi implementate intr-un sistem cu o componenta criptografica. Acestea au la baza o metoda de autentificare care este o metoda complexa de partajare a datelor intre cele doua parti comunicante. Avem de-a face aici cu procese simple care se pot numi protocoale si care stabilesc metoda autentificarii.

Autentificarea datelor se face utilizand o metoda de autentificare a datelor la nivel de utilizator precum si printr-o metoda de verificare a integritatii mesajului.

Metoda de autentificare la nivel de utilizator este o metoda de a convinge sistemul de identitatea personala. O data realizat acest lucru, sistemul poate verifica daca respectiva identitate are anumite drepturi asupra sistemului. Astfel este necesar un procedeu pentru a determina identitatea persoanei. Acest proces se numeste autentificare la nivel de utilizator si exista diverse metode implementate (parola, pin, cartela magnetica, biometria etc.)

Autentificarea datelor are doua componente:

- verifica daca datele au fost modificate si
- daca este cunoscut emitatorul.

Integritatea datelor in sine nu are valoare criptologica, nu se poate depista daca datele primite au fost sau nu modificate. Acest lucru se poate determina doar daca se stie faptul ca mesajul a fost trimis de la sursa asteptata. Astfel aceasta metoda trebuie combinata cu procedura de

autentificare a datelor. Protejarea datelor fata de o parte terta, asigura confidentialitatea mesajului verificand daca mesajul original nu a fost deja interceptat si retransmis sau alterat. Fara aceasta componenta nu se putea spune daca mesajul provine de la emitatorul asteptat sau daca vine din alta parte, fiind modificat in prealabil.

Traditional un proces criptografic se poate exprima prin ecuatia simpla: un mesaj oarecare necodificat P(x) devine un mesaj codificat C(x) utilizand o transformare finita T iar pentru a obtine mesajul decodificat se aplica inversa transformarii T^{-1} mesajului codificat C(x).

$$C(x) = P(x) \circ T(x)$$
$$P(x) = C(x) \circ T^{-1}(x)$$

cu conditia:

x ¿ { a | a = mod N} unde N reprezinta numarul caracterelor din alfabetul utilizat in mesajul necodificat. (Unele cifruri ca de exemplu PLAYFAIR, nu utilizeaza functia inversa ci o bruiaza pentru a mari confuzia la decriptare si pentru a apropia mesajul de interpretarea umana.)

Se constata astfel ca o conditie necesara ca un cifru oarecare sa existe si sa aibe sens este ca tranformarea pe care o implica sa fie inversibila. Acest lucru ilustreaza conceptul criptografic de transparenta care impune ca indiferent de functia de transformare T si de inversa sa T^{-1} si procedurile pe care le implica acestea, mesajul sa poata sa fie adus la formatul sau original. Exista exceptii de la aceste reguli [ex. playfair], care presupun cresterea confuziei sau a difuziei insa acestea sunt caracteristicile unui cifru primitiv si regula de baza ramane neschimbata pentru metodele moderne de codificare. Se impune o singura restrictie asupra functiei de transformare: acesta trebuie sa se incadreze in clasele P sau NP de probleme. Adica este esential ca transformarea sa fie realizata in timp polinomial fie printr-un numar fix de pasi secventiali (masina Turing secventiala) fie printr-un numar de pasi paraleli (masina Turing nedeterminista). Aici se disting cele doua mari clase de cifuri: i) *cele care opereaza bit cu bit* si ii) *cele care opereaza pe blocuri de text*. In ecuatiile de mai sus am utilizat litera "x" pentru a indica un caracter sau o multime finita de caractere dintr-un

alfabet oarecare. Daca functia este valabila modulo N unde N reprezinta numarul de caractere atunci cifrul este universal si se poate aplica alfabetului.

Confuzia si *difuzia* pe care le-am mentionat mai sus sunt doua proprietati enuntate de catre Claude Elwood Shannon in 1949 intr-un articol intitulat "Communication Theory of Secrecy Systems" implicand operatiile unui cifru sigur. Definitia originala ne spune ca *difuzia* se refera la statisticile unui mesaj necodificat care sunt disipate in statisticile mesajului codificat iar *confuzia* se refera la complexitatea relatiei intre cheie si mesajul codificat. Astfel difuzia este asociata cu dependenta intre bitii de iesire si bitii de intrare. In exemplul dat - cifrul simetric PLAYFAIR - in anumite conditii bine definite de regula, unii biti de intrare in mesajul original sunt precedati de litera "x" sau in unele variante, litera "q". Trebuie astfel notat ca metoda Playfair nu satisface conditia de transparenta insa realizeaza o difuzie aproape perfecta; mesajul codificat si apoi decodificat nu mai rezulta in mesajul original ci in ceva complex, inteligibil doar mintii subiective umane. Intr-un caz ideal, adica un cifru perfect, pentru a obtine o buna difuzie fiecare bit de intrare modificat trebuie sa schimbe fiecare bit de iesire cu o probabilitate de 50%. Acest concept se mai numeste si Strict Avalanche Criterion (SAC) si este satisfacut de cateva cifruri in istoria criptografiei ca de exemplu ENIGMA si FEISTEL.

Cheia, in criptografia moderna, este o informatie care modifica aspectul mesajului codificat. Aceasta poate sa ia orice forma personalizata si este considerata un secret partajat intre partile comunicante. Cheia este o parte componenta a celor trei scopuri (autentificarea datelor, protejarea datelor fata de o parte terta si confidentialitatea datelor) si rezolva problema protejarii datelor. Protocolul Diffie-Hellman scoate in evidenta importanta cheii si este conceptul pe care se bazeaza criptografia care utlizeaza chei. Algoritmul se explica in exemplul urmator:

1) Alice si Bob cad de acord sa utilizeze un numar prim p = 23 si baza g = 5.
2) Alice alege un numar intreg secret a = 6 si il trimite lui Bob ($g^a \bmod p$):

   $5^6 \bmod 23 = 8$.
3) Bob alege si el un numar intreg b = 15 si il trimite lui Alice ($g^b \bmod p$):

   $5^{15} \bmod 23 = 19$.
4) Alice calculeaza $(g^b \bmod p)^a \bmod p$:

   $19^6 \bmod 23 = 2$.

5) Bob calculeaza $(g^a \bmod p)^b \bmod p$:

$8^{15} \bmod 23 = 2$.

Astfel Alice si Bob au ajuns la aceeasi valoare si numai a, b, $g^{ab} = g^{ba}$ sunt valori care sunt pastrate secret adica valori de cheie. Restul procesului se efectueaza in clar si nu este important daca restul informatiei este divulgat. Din moment ce Alice si Bob ajung la aceeasi valoare ei pot utiliza acea informatie ca o cheie pentru a-si codifica mesajele mai departe. In definitia generala se utilizeaza un grup ciclic G si un generator g din G; g este considerat o informatie care se poate divulga. Apoi se utilizeaza puteri a lui g pentru a ajunge la o valoare comuna. Protocolul este considerat sigur impotriva unei parti terte care intercepteaza transmisia.

Criptografia moderna se poate imparti in doua mari grupuri:

- *criptografie cu cheie simetrica*, din care fac patre cifrurile de bloc si cele de stiva, si
- *criptografia cu chei publice*.

Criptografia cu cheie simetrica utilizeaza aceeasi cheie pentru codificare si decodificare sau cheia pentru decodificare este dedusa usor din cheia utilizata pentru codificare. Cifrurile de stiva opereaza bit cu bit in contrast cu cifrurile bloc text care opereaza pe o grupare de biti cu o lungime finita intr-o singura transformare. Depinzand de modul de operare, cifrurile bloc pot fi implementate in mod CFB (adica cifruri cu autosincronizare). De asemeni, cifrurile de stiva pot fi concepute sa lucreze pe blocuri individuale de text pas cu pas. Printre cele mai cunoscute cifruri de bloc se afla FEISTEL, DES si IDEA, iar printre cifruri de stiva, RC4. Criptografia simetrica are un dezavantaj important si anume: doi oameni care vor sa comunice securizat trebuie sa cunoasca o cheie secreta care trebuie stabilita intr-un mod special si nu pe o cale pe care ar comunica in mod normal. Acest inconvenient este rezolvat de catre *criptografia cu cheie publica* (sau cheie asimetrica) care utlizeaza doua chei diferite. O cheie este publica si accesibila tuturor iar cealalta este tinuta secreta utlizandu-se cheia publica pentru codificare respectiv cheia secreta pentru decodificare. Algoritmele bazate pe cheie publica utilizeaza in general probleme de calcul

complexe (cum ar fi RSA care utilizeaza factorizarea). Pentru eficienta sistemelor de codificare hibride se procedeaza in felul urmator: o cheie este impartita utilizand un cifru cu cheie publica, iar restul comunicarii se face codificat utilizand un algoritm de stiva (care este mai rapid in timp real). Criptografia asimetrica mai asigura unul dintre scopurile criptografiei mai sus mentionate, anume confidentialitatea datelor, prin semnaturi digitale care reprezinta o metoda de a verifica pana la un punct (adica pana la punctul in care cheia secreta se presupune ca fiind necompromisa) ca mesajul primit a fost trimis de la emitatorul corespunzator. Exemple pentru acestea sunt semnaturile DSA si ElGamal. La fel, algoritmi hibrizi sunt utilizati in practica. In loc de a semna un intreg document, se semneaza doar o parte codificata intr-un hash[1] determinat al mesajului. Un program cunoscut care implementeaza un asemenea protocol este PGP (Pretty Good Privacy) sau varianta sa "open-source" GPG (GNU Pretty Good Privacy).

In principiu siguranta in toate algoritmele si schemele de codificiare ramane nedeterminata pentru ambele procedee fie ele simetrice sau asimetrice. Pentru cifrurile simetrice, siguranta este mai degraba anecdotica nefiind anuntata vreo decodificare cu succes pe vre-un algoritm de acest tip in mai multi ani. Un asemenea cifru ar putea avea ceva siguranta demonstrabila impotriva unui set limitat de incercari de decodificare. Pentru cifrurile asimetrice se obisnuieste sa se bazeze pe dificultatea problemei matematice asociate cu incercarea de decodificare dar si acest lucru nu este demonstrabil ca fiind sigur. Criptografia are un cifru cu o dovada puternica de siguranta: "one-time pad" (exemplu Vernam) dar acesta cere chei cel putin la fel de lungi ca textul necodificat fapt care este in principiu considerat foarte greu de realizat. Cand securitatea unui sistem esueaza, rareori acest lucru se datoreaza unei slabiciuni intr-un algoritm criptografic ci mai degraba constituie o eroare in implementare sau o eroare umana. Acesta este domeniul mentionat la inceput, si anume cel al analizei criptografice, care are ca scop sa gaseasca un defect intr-un sistem criptografic.

Exista o mare varietate de atacuri criptografice si ele pot sa fie clasificate dupa diversele metode utilizate. O distincie se refera la ceea ce stie analistul si ce poate sa faca sau sa afle ca sa ajunga la informatia secreta. Aici se pun mai multe intrebari. De exemplu: are

---

1[1]Hash - O functie de comprimare a datelor sau a unei stive de lungime inconvenabil de mare intr-o lungime mica.

analistul acces numai la mesajul cifrat? (exemplu alph – smash). Poate sa stie sau poate sa ghiceasca cateva litere? Sau chiar: poate sa aleaga mesaje necodificate arbitrare care sa fie codificate? Aceste scenarii corespund la incercarile de decodificare, respectiv: numai codificat, text codificat cunoscut sau text necodificat cunoscut. Pe cand analistii criptografici utilizeaza slabiciuni in insusi algoritmul de codificare alte atacuri se bazeaza pe implementare (side-channel attack). Daca un analist are acces la timpul pe care l-a consumat algoritmul pentru a codifica un mesaj el ar putea sa utilizeze un atac de tip contra-timp (timed attack) pentru a sparge un cifru care ar putea sa fie rezistent la analiza. De asemenea el ar putea sa analizeze tiparul si lungimea mesajului pentru a deriva informatia utiliza, metoda cunoscua ca analiza de trafic (traffic analysis). Daca sistemul criptografic a utilizat o cheie derivata de la o parola s-ar putea considera ca risc datorita dimensiunii sau a entropiei slabe. Acesta este un punct de slabiciune intalnit des in sistemele criptografice.

Metode cunoscute pentru a sparge cifruri bazate pe chei simetrice sunt:

- *Analiza criptografica lineara*
- *Analiza criptografica diferentiala*.

Exemplu tipic de *analiza criptografica lineara* consta in analiza frecventei literelor (exemplu alph – smash) in mesajul codificat (acest lucru presupunand bineintles ca este vorba despre un cifru de stiva) numita si "frequency-analysis". Aceasta analiza mai presupune o buna cunoastere a limbii de origine a mesajului. Analiza de frecventa se bazeaza pe faptul ca in orice limbaj scris anumite litere si combinatii de litere apar cu diferite frecvente. Mai mult, exista chiar o distribuite caracteristica de litere care apare in aproape orice exemplar de text dintr-o limba. In domeniu analizei criptografice primii de mentionat sunt Eli Biham si Adi Shamir care au publicat un articol in anii ‘80 cu o serie de atacuri posibile pe diferite cifruri si functii de "hashing" (implementat in Alph ca parametru SDBM, DJB2 etc...) cat si o slabiciune teoretica in DES (Data Encryption Standard). Acestia utilizau pentru DES (Data Encryption Standard) metoda diferentiala de atac si multe cifruri deveneau susceptibile de aceasta slabiciune imediat ce aticolul a fost publicat. Printre primele era cifrul bloc FEAL a carui versiune originala cedeaza dupa numai opt atacuri diferentiale pe un text necodificat ales.

Analiza criptografica diferentiala a cifrurilor consta in principal dintr-un atac bazat pe alegerea unor texte cunoscute insemnand ca atacatorul trebuie sa fie capabil sa obtina mesaje codificate pentru un set de mesaje necodificate. Metoda de baza utilizeaza perechi de mesaje necodificate care se diferentiaza printr-o constanta obtinuta de obicei printru-un SAU-exclusiv (XOR). Dupa aceea analistul calculeaza diferenta intre mesajele cifrate corespunzatoare sperand sa gaseasca tipare statistice in distributia lor. Pentru un cifru in particular, diferenta trebuie aleasa cu atentie si se procedeaza la o analiza a algoritmului: metoda standard este sa se traseaze o traiectorie de diferente foarte probabile in diversele etape ale codificarii, numita in final caracteristica diferentiala. Fiindca analiza diferentiala a devenit foarte cunoscuta, ea reprezinta o preocupare de baza a cercetatorilor de cifruri si noile implementari au chiar dovezi pertinente de rezistenta impotriva acestui tip de atac. Relatiile matematice formale pentru o analiza diferentiala sunt urmatoarele:

$$C[] = f_c(m[], k[])$$
$$\delta = f_c^{-1}(C[], m[])$$

unde C[] reprezinta mesajul cifrat, m[] mesajul de codficat, k[] cheia cifrului, f_c respectiv f_c^{-1} functia de codificare si decodificare respectiv iar delta exprima diferenta intre mesajele codificate. O analiza criptologica diferentiala presupune astfel conoasterea algoritmului cat si accesul la acesta pentru a putea genera mesaje codificate pe un text ales utilizand cheia secreta cat si alte chei. Singura necunoscuta ramane cheia care va trebui sa fie dedusa din analiza diferentiala.

Multe cifruri lineare cat si de bloc sunt susceptibile la acest tip de atac. Acesta lucreaza mult mai repede decat un atac fortat si ataca mai degraba algoritmul in sine decat mesajul sau alegerea cheii. Un exemplu simplu de cifru vulnerabil pe care putem sa-l enuntam este VIGENERE (exemplu VIGENERE in Alph) care utilizeaza chei transpuse repetate. Aplicand formulele de analiza diferentiala putem calcula cheia alegand mesajele corespunzatoare (am presupus in acest exemplu cheia secreta ca fiind "ana" si am ales un mesaj de codificat aleator "cal"):

c[] = f (m[], k[]) => c[] = VIGENERE ("cal", "ana") = "cnl"

rezultatul codificarii este astfel "cnl", un cuvant care nu are nici-un inteles. Pentru a afla cheia procedem la diferentiere.

$$d = f^{-1}(c[], m[]) \Rightarrow d = \text{VIGENERE}^{-1}(\text{"cnl"}, \text{"cal"}) = \text{"ana"}$$

Astfel dintr-o singura incercare am dedus cheia ca fiind "ana". Stiind cheia, putem sa deducem textul cifrat utilizand functia de decriptare VIGERNERE. O analiza mai consistenta decat acest exemplu presupune stabilirea unei statistici al functiei de analiza diferentiala. Se creeaza un tablou continand numarul de aparitii ale fiecarei litere din analiza diferentiala repetata iar apoi se determina cheia ca fiind o permutare intre cele mai frecvente litere din analiza. Pentru calcularea tuturor permutarilor se poate utiliza programul din anexa [ax Alph permute].

2. Internele programului Alph

Alph este un program de criptologie scris in totalitate in cod C. Manualul acestuia indica faptul ca acesta contine o versiune cat se poate de, dintr-un punct de vedere, ne-optimizata, a unor algoritme criptografice cat si subrutine de spargerea si analiza ale acestora. Am spus "dintr-un punct de vedere" fiindca fiecare cifru nu este optimizat sa ruleze perfect sub o anumita platforma ci s-a optat pentru o abordare mai generala: programul utilieaza subrutine "generale" pentru a putea fi compilat sub multe platforme. Acest lucru sporeste universalitatea programului contra vitezei sale de lucru. Intr-un cuvant, Alph urmareste compatibilitatea. Intrebuintand subrutine de programare de nivel jos al limbajului C, el poate fi rulat pe o gama larga de platforme cum ar fi Linux, BSD sau chiar DOS/Windows fara ca acestea sa impuna cerinte specifice asupra programului. Sub platforme care nu intrebuinteaza pipe[2]-uri, programul pur si simplu asteapta ca datele sa vina de la un anumit program sau sa fie introduse manual la linia de comanda.

2 Sub UNIX si alte sisteme asemanatoare, un pipe sau un pipeline este un set de procese care au iesirile / intrarile inlantuite astfel incat iesirea fiecarui proces ("stdout") constituie direct o intrare ("stdin") pentru un alt program. Conceptul este numit prin analogie cu o ‘teava’.

Trebuie inteles modul in care lucreaza acest program pentru a merge mai departe.
Alph este un program de filtrare. El nu produce nimic, decat in cazul utilizarii unei subrutine de cripto-analiza, ci pur si simplu leaga intrarea standard ("stdin") de iesirea standard ("stdout") lucrand intre acestea asemenea unui filtru.
Asa cum sugereaza imaginea din Fig. 1, programul Alph se afla intre intrare si iesire modificand datele care trec prin el. Datorita acestui concept de a leaga intrarea de iesire, programul nu incearca sa introduca date noi sau sa modifice formatul lor, ci pur si simplu

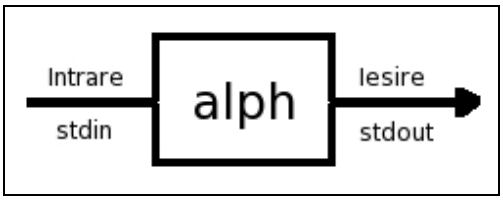

Fig. 1 Modul de functionare al programului Alph

converteste datele care au fost introduse de catre utilizator sau alt program cuplat la intrarea sa. In criptografia moderna fiecare cifru utilizeaza un convertor Base64 (cap. 3.2) pentru a reda datele finale. Alph utilizeaza aceasta metoda doar unde acest lucru este absolut necesar (cum ar fi cifrurile moderne). In rest el incearca sa nu modifice nici macar punctuatia datelor introduse.

2. 1 Structura programului Alph

Alph isi imparte modulele sau modurile de functionare in cinci mari categorii numite in ordine: 1) Classical, cuprinde cifrurile antice si utilizate in trecut, 2) Modern, cifrurile moderne utilizate in zilele noastre, 3) Hashes, categoria de hash-uri despre care vom discuta mai amanuntit mai tarziu, 4) Tools, cuprine utilitarele lui Alph permitand diferite functii de cripto-analiza si in sfarsit 5) Options, include optiunile care executa comenzile dorite. Pentru o vedere de ansamblu mai clara putem cere programului sa afiseze un scurt ajutor utilizand comanzile '-h', '--help' sau '/?'. Acesta va afisa un scurt sumar al tuturor algoritmelor implementate cat si modul in care acestea se pot utiliza. Pentru un ajutor mai amanuntit se poate face apel la manualul programului sub forma de man-page prin comanda man alph. Acesta va afisa o descriere mai amanuntita a programului si un scurt istoric al fiecarui

algoritm. Manualul este o parte componenta al acestui program si exista in varianta man si html pentru sisteme Unix respectiv Windows.

Fiecare versiune a programului Alph are un numar major si un numar minor. Varianta curenta (0.16) are numarul major 0 si numarul minor 16. Acesta terminologie tipica programelor open-source[3] ne spune cat de avansata este dezvoltarea programului. Numerele sunt inventate de catre producator. Numarul major (adica 0 in acest caz) se incrementeaza de-indata ce programul a trecut printr-o modificare majora iar numarul minor (adica 16 in acest caz) este numarul versiunii care aduce imbunatatiri la versiunea majora. Acesta din urma este traditional incrementat dupa fiecare publicare a unei versiuni noi ale programului. La ora actuala, programul este la versiunea majora 0 si la versiunea minora 16. Numarul major vorbeste in unele cazuri despre stabilitatea programului si este incermentat doar cand autorul sau autorii considera ca s-a atins un anumit nivel. Programele care au versiunea majora 0 sunt considerate in versiuni beta sau alfa si urmeaza sa fie supuse testelor si imbunatatirilor pana cand vor atinge o versiune stabila. Aceste doua numere sunt importante deoarece ele sunt traditional considerate puncte de plecare pentru actualizari sau teste speciale. Unele programe se despart uneori intr-o anumita etapa bifurcandu-se in doua: o dezvoltare continua pe aceasi linie urmand stabilitatea iar celalalta, considerata versiune de dezvoltare, merge inainte adaugand noi functii si ignorand in mare parte stabilitatea. Alph se afla la ora actuala la versiunea 0.16 care este considerata inca o versiune instabila dar cu mult peste testele de stabilitatea ale unei versiuni alfa si beta. Programul este presupus sa functioneze in cele mai multe cazuri insa isi rezerva dreptul, pana la versiunea majora 1, sa fie incomplet si instabil. Decizia este lasata utilizatorului daca sa foloseasca acest program intr-un mediu de lucru sau sa astepte varianta stabila. De cele mai multe ori, dezvoltarea pana la o versiune majora ia foarte mult timp, autorul asteptand reactii din partea utilizatorilor cat si testele sale asupra fiecarei parti a programului.

Fiecare distributie sau versiune de Alph se poate descarca de pe http://www.freshmeat.net/projects/alph sub forma de sursa in cazul sistemelor tip UNIX sau

3[3]Open source descrie practici generale in producerea si dezvoltare care promoveaza accesul la sursele produsului. Este privita ca o filozofie de catre unii si ca o metodologie pragmatica de catre altii. Dezvoltatorii si producatorii au adoptat aceasta terminologie devenita cunoscuta in 1998 si reprezentand un efort unificat al tuturor programatorilor pentru a dezvolta software de inalta calitate. S-a presupus ca accesul la codul sursa cat si, in unele variante, modificarea acestuia ar aduce imbunatatiri substantiale la produsele realizate sub aceasta sigla cat si la formarea unei comunitatii care dezvolta noi strategii si tehnici de programare cu rezonanta in informatica si educatie.

sub forma binara pentru platformele Windows. Trebuie notat ca pentru compilarea binarului COFF[4] (Common Object File Format), s-a utilizat un compilator "cross" de Linux creat dupa specificatiile lui Dj Delorie (http://www.delorie.org) peste un compilator standard GNU gcc. Modul in care s-a obtinut acest compilator este dincolo de subiectul acestei lucrari insa trebuie notat ca fiecare binar pentru platforma de Windows al programului Alph este compilat sub Linux. Inclusiv manualul este tradus din formatul man spre formatul html pentru a putea fi citit sub o platforma Windows cu usurinta.

Binarul vine ca atare, comprimat in prealabil utilizand zip, impreuna cu manualul programului in format html. Sursa insa este disponibila sub licenta GNU/GPL[5] si contine o suma de caracteristici care trebuiesc mentionate. In primul rand structura sursei este dispusa tipic unei surse cu auto-compilare autogen/autoconf. Ea vine impachetata in formatul bzip2 (algoritm imbunatatit zip) si tar (arhiva de banda) continand diferite script-uri rulabile care vor genera binarul automat. Arhiva contine cele 6 documente tipice distribuite cu fiecare versiune: AUTHORS, COPYING, ChangeLog, INSTALL, NEWS si README. Primul document descrie autorii programului, al doilea este o copie a licentei GNU, al treilea este un document care descrie schimbarile din toate versiunile, al patrulea este un document care explica modul si prerechizitele compilarii programului, al cincilea contine stirile despre noua versiune de program iar al saselea este un document ajutator care este deobicei citit primul inainte de compilare. Aceste documente sunt standard fiecarui soft Open-Source. Pentru compilare, se utilizeaza scripturile puse la dispozitie de catre program. In acest caz, este vorba despre un singur script "configure", ajutat de altele, care pregateste sursa pentru compilare. Acest script cand este rulat, are grija sa verifice daca parametrii sistemului sunt in ordine, cum ar fi prerechizitele sau formatul variabilelor, ca apoi sa creeze fisiere pentru utilitarul de compilare "make". In primul stadiu el cauta cateva functii esentiale rularii programului, verifica daca parametrii compilatorului si al linker-ului sunt corecti iar apoi in stadiul doi genereaza asa zisele "Makefiles" adica fisiere care indruma programul "make"

4 [4]COFF - Common Object File Format este un format de fisier introdus in UNIX System V Release 3, si preluat mai tarziu de cate Microsoft pentru Windows NT. A fost considerat invechit de catre formatul ELF introdus in System V Release 4 dar din 2005, COFF este inca utilizat pe Windows ca fiind un executabil portabil.

5 GNU - General Public License (GNU GPL sau GPL) este probabil cea mai cunoscuta licenta de software gratuit. Este scrisa de catre Richard Stallman pentru proiectul GNU si a devenit cea mai cunoscuta licenta, ultima versiune, versiunea 2, fiind publicata in 1991. GNU Lesser Public License (LGPL) este o versiune modificata a lui GPL si este scrisa pentru cateva librarii de soft.

prin sursa pentru a putea compila toate fisierle necesare programului cat si programul in sine. Daca unele componente nu sunt gasite de catre acest program, atunci el va anunta utilizatorul ca nu are sistemul potrivit pentru a compila programul si il indruma eventual cum ar putea reusi sa compileze programul. Acest script este partial scris de catre utilizator si partial generat automat de catre programul "autogen". Pentru a genera un script de acest tip, programatorul trebuie sa cunosasca limbajul de scriptare "automake/autogen", scriind in prealabil un fisier de configurare "configure.in" unde acesta specifica testele care trebuie efectuate de catre script cat si versiunea programului si alti parametrii. De asemenea programatorul trebuie sa cunoasca limbajul "automake" pentru a genera un prim "Makefile" care se afla la cel mai inalt nivel intr-un arbore de cod sursa. Acest fisier specifica directoarele unde programul de compilare "make" sau "gmake" va gasi dependentele programului de compilat. Aici se includ manualul, fisiere antet si librarii scrise de catre autor. Este esential ca aceasta combinatie de scripturi si fisiere sa fie in perfecta ordine deoarece datorita lor se va obtine un program corect compilat si functional.

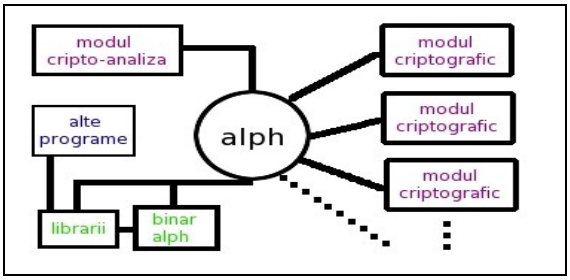

Fig. 2 Structura programului Alph.

Alph utilizeaza aceste trasaturi ale sistemului de compilare "automake/autoconf" extensiv data fiind structura sa aparte. Structura sursei corespunde in mare parte modului in care lucreaza programul. Directoarele relevante sunt "include", "modules" si "central" pe baza carora putem schita o schema formala a functionarii programului (Fig. 2).

Nucleul lui Alph este fisierul de cod "main.c" care uneste toate functiile din toate fisierele sursa la nivel operational. El descrie o interfata catre lumea exterioara combinand sursele fiecarui modul criptografic cat si modulul de analiza intr-o interfata universala. Nucleul se ocupa cu redarea si primirea datelor cat si coordonarea diverselor module pentru

diferite functii. Pe scurt, el defineste variabilele primare si interpreteaza linia de comanda. El se regaseste in sub-directoriul "central/" unde se gaseste si modulul de cripto-analiza alphan.c. In acesta se gasesc functiile de analiza criptografica oferite de catre Alph. In sub-directoriul "modules/" se gasesc fisiere pentru fiecare algoritm pe care Alph il abordeaza. Putem astfel gasi fisiere precum caesar.c, atbash.c, vigenere.c etc. S-a preferat aceasta structura deoarece se poate adauga un modul nou foarte simplu fara a mai modifica prea mult sursa originala a programului. Fiecare modul descrie o functie care ia cativa parametrii: sirul original de caractere pentru codificare si eventual cheile corespunzatoare algoritmului. Ulterior numele si parametrii acestei functii sunt adaugate unui fisier antet alphCRYPT.h sau alphan.h daca este vorba de un algoritm de codificare respectiv un modul de cripto-analiza. Mai departe, in directoriul "modules/" se regaseste un "Makefile" de forma:

```
/------------------------------ SNIP ------------------------------/
lib_LTLIBRARIES = libalphCRYPT.la

libalphCRYPT_includedir = -I$(top_srcdir)/include
libalphCRYPT_la_SOURCES = caesar.c \
atbash.c \
vigenere.c \
vernam.c \
playfair.c \
/--------------------------------SNIP--------------------------------/
```

Un dezvoltator de algoritme poate contribui usor urmand procedura pentru adaugarea unui cifru la programul Alph:

1.) Scrie algoritmul intr-un fisier fiind atent ca functia de criptare sa ia un parametru ca fiind sirul de caractere original de codificat si alti parametrii...

2.) Pune sursa sa in directoriul "modules/" si adauga numele fisierului la "libalphCRYPT_la_SOURCES" sub restul cifrurilor si codurilor abordate.

3.) Modifica fisierul "main.c" din "central/" pentru a prelua parametrii cat si pentru a afisa noua optiune si parametrii sai.

4.) Ca ultim pas el trebuie sa adauge functia in fisierul antet alphCRYPT.h pentru ca aceasta sa fie cunoscuta universal in tot programul.

Urmand acesti pasi, Alph va prelua automat modulul scris de programator si nu va cere schimbari. Pentru dezvoltatori de soft, Alph este disponibil cu un script de integrare scris in Bash ("regen.sh") care usureaza regenerarea fisierelor si scripturilor necesare compilarii programului.

Prin aceasta procedura se clarifica partea a doua din figura prezentata de mai sus, adica generarea programului. Atunci cand Alph este compilat, codul sursa creaza atat un binar "alph" cat si o librarie partajata libalphCRYPT. Aceasta din urma este o librarie universala care contine toate functiile de codificare/decodificare care, fiind partajata, este disponibila si altor programe. Mai amanuntit, codul sursa creaza un substitut pentru programul Alph care este de fapt un script bash care uneste la rulare librariile cat si programul. In acest fel el va functiona chiar daca libraria libalphCRYPT nu este instalata. Acesta este un punct puternic al programului Alph. El se orienteaza sa creeze o librarie cu toate algoritmele mai degraba decat un program monolit. O persoana care doreste sa utilizeze, sa zicem algoritmul FEISTEL, nu este obligata sa-l utilizeze doar prin intermediul programului Alph ci doar prin libraria sa. De exemplu: sa presupunem ca un programator vrea sa scrie un program de gestiune economica. La un moment dat el vrea ca o parte a datelor sa fie codificata utilizand algoritmul ENIGMA. Atunci el se uita in antetul alphCRYPT.h si regaseste functia de codificare/decodificare ENIGMA:

```
extern int enigma (char *org, char *rot, char *ref, char *ring);  /* ENIGMA */
```

Stiind parametrii, el poate folosi functia enigma() in programul sau fara probleme. Apoi, atunci cand isi compileaza programul de gestiune economica, adauga la lina de compilare libraria libalphCRYPT (sub sistemele UNIX acest lucru se face usor: gcc <cod_sursa.c> -lalphCRYPT). De asemenea, chiar sistemul mentionat mai sus "autogen/automake" are optiuni puternice pentru dezvoltatori care vor sa includa librarii scrise de o parte terta. Astfel

programul sau lucreaza transparent cu functia de codificare/decodificare enigma() fara ca acesta sa fie nevoit sa ruleze datele printr-un program (sau binar) extern cum ar fi alph. Aceasta trasatura este extraordinar de importanta si clarifica scopul programului Alph, adica acela de a prezenta o librarie unificata de algoritmi de criptografie iar ca scop secundar, acela de a fi un program usor de utilizat pentru codificarea sau decodificarea unor date utilizand pipe-uri sau date introduse manual de catre utilizator. Intr-un fel, din acest punct de vedere, programul se aseamana puternic librariei openssl care implementeaza o varietate de algoritme criptografice moderne numai ca Alph incearca sa le implementeze si pe cele vechi pe langa cele noi. Partea de cripto-analiza ramane insa specifica lui Alph si nu mai este prezenta in nici-o implementare cunoscuta. Ea este o trasatura interesanta a programului Alph si realizeaza unele functii prin utilitare care sunt cate odata esentiale unui criptolog. Acesta este cazul permutarii sau combinarii de simboluri sau a partii de analiza statistica de text care fac din Alph un instrument folositor.

2. 2 Setul de functii

Alph vine in doua versiuni: o versiune pentru, dar nu numai, sistemele de operare UNIX si hibride si una pentru Windows. Bineinteles ca programul trebuie sa fie universal in sensul ca acesta trebuie sa fie acelasi pe fiecare platforma. Nu se poate scrie un program cu parametrii diferiti pentru ambele platforme fiindca atunci programul nu ar mai fi consistent si in loc de un program am avea doua programe operational asemanatoare dar structural distincte. Din aceasta cauza, ambele versiuni trebuie unificate, ba chiar generate de acleasi metode de generare prezentate in subcapitolul anterior.

In scopul realizarii binarelor pentru ambele platforme: UNIX, format ELF 32-bits executabil LSB respectiv Windows/DOS, format COFF (MC-COFF) executabil pentru MS-DOS se intrebuinteaza un "cross-compiler". Adica un compilator, caruia fiindu-i dati anumiti parametri, poate sa genereze binare pe o platforma, care sunt destinate altor platforme. Intreaga dezvoltare a programului a fost realizata sub Linux (sistem hibrid UNIX) iar binarele pentru platforma Windows/DOS au fost generate de compilatorul "DJGPP go 32 cross-compiler". Ca rezultat, avem doua distributii ale programului unul in format sursa (care poate sa genereze si alte binare in afara de EFL sau COFF) si una in format binar pentru

Windows/DOS. Acesta din urma nu este distribuit sub forma sursa deoarece nu se obisnuieste sa se compileze surse in prealabilul executarii programului sub sistemele Windows/DOS. Daca un ultilizator de platforma anterioara doreste totusi sa compileze sursa, acesta poate sa utilizeze un program pentru mediul C deoarece intregul cod de program este distribuit o data cu sursa. Astfel diferentele intre distributii sunt minore iar programul este acelasi sub orice platforma care utilizeaza orice binar. Trebuie notat, asa cum am mai explicat, faptul ca se poate utiliza un "cross-compiler" pentru a genera binare si pentru platforme altele decat cele mai sus mentionate. Totul depinde de tipul compilatorului utilizat si ce binar genereaza acesta.

Alph faciliteaza introducerea de cod si extensii la setul de module criptografice, prezentand dezvoltatorului o interfata simpla de utilizat. Asa cum am explicat, nu sunt multi pasi necesari pentru a aduga extensii la program. Exista totusi cateva impedimente sau, mai bine spus, precautii care trebuiesc luate in considerare atunci cand se adauga cod. Cea mai importanta este se doreste pastrarea universalitatii programului. Adica se doreste ca programul sa fie usor de compilat in orice mediu fara probleme. Pentru a realiza aceasta conditie, trebuie ca modulele lui Alph sa nu includa cod care utilizeaza setul de functii extensiv pentru o platforma data. Mai precis o functie anume, care este suportata pe o platforma unde este scris noul modul, poate sa nu fie suportata pe alte platforme. Din aceasta cauza, Alph trebuie sa se apropie cat mai tare,stric chiar, de standardul ANSI C. Acest lucru influenteaza codul cat si viteza de operare a programului: codul se mareste deoarece functii complexe trebuie rescrise iar viteza de operare scade deoarece numarul de variabile creste. Pentru a evita aceste impedimente, Alph recurge la o serie de simplificari care incearca sa compenseze dificultatile. Un bun exemplu este faptul ca programul este scris in totalitate in C evitand functiile si intructiunile de C++. Pentru un program care lucreaza cu variabile mici si functii aritmetice este preferabila evitarea unui limbaj bazat pe obiecte. Alph nu are blocuri de date mari care trebuie manipulate ci incearca in toate cazurile sa auto-genereze datele de care are nevoie (ex. patratul polybus).

Datorita subsistemul "man" al sistemului de operare Linux fiecare functie utilizata in programare este insotita de documentatia respectiva. Aceasta se afla deobicei in sectiunea "man 3" al sistemului "man", sectiune destinata programarii sub sistemul de operare Linux. O trasatura interesanta, si relevanta in contextul subcapitolului, al acestei documentatii, este

sectiunea "CONFORMING TO". In aceasta sectieune se specifica sistemele de operare sub care aceasta functie este valida. De exemplu pentru functia strlen(), destinata calculului de lungime al unui sir, functie utilizata extensiv in modulele lui Alph, aceasta se gaseste in antetul string.h si este valabila pe sistemele: SVID 3, POSIX, BSD 4.3 si ISO 9899. Programul urmareste compatibilitatea cu standardul din urma, adica ISO 9899, care este standardul oficial reprezentand limbajul de programare ANSI C. Setul de instructiuni este astfel strict si se limiteaza la o serie restransa de functii permise. Functiile sunt prezente in antete generale precum stdlib.h, string.h sau unistd.h etc... Acest lucru garanteaza ca programul (codul sursa) poate fi compilat sub orice mediu de dezvoltare C si sub orice platforma.

Daca o functie nu este destul de generala atunci compilatorul va afisa un mesaj de avertizare insa programul va reusi totusi sa se compileze. Acest lucru functioneaza foarte simplu si este legat de secventa prin care un program trece atunci cand acesta este compilat: mai inati codul este procesat de catre "parser", codul este converit intr-un obiect si in final, in functie de simbolurile din obiectele generate, acestea sunt "linkate" cu librariile respective. Aici intervine o problema. Parserul si generatorul de obiecte nu sesizeaza daca o functie anume nu este regasita printre functiile dintr-o librarie. Daca un cod sursa a generat un simbol, o functie, iar acesta nu se gaseste printre cele din setul de functii din libraria specificata la "linkare" atunci "linker-ul" generaza un cod de eroare. In cazul unui compilator de gen "cross-compiler" mentionat mai sus, acesta este dotat cu un set de librarii de compatibilitate intre librariile platformei de origine si librariile platformei de destinatie. Pentru a evita o compilare aparent posibila dar de fapt strict imposibila, sau pentru a oferi compatibilitate in ambele sensuri, "cross-compiler"-ul foloseste un sistem de "stub"-uri pentru a inlocui functiile care nu pot fi traduse. Ce se intampla este ca, in cazul unei functii care nu poate fi tradusa, acesta este inlocuita cu un "stub". Adica functia este inlocuita de catre o functie care nu face nimic efectiv. In acest caz, programul nu va mai functiona, toate aparitiile acelei functii ne-indeplinind nimic. S-ar putea face o comparatie cu mediul de programare asambler si comanda NOP: abstract aceste functii sunt inlocuite cu NOP-uri. Din fericire, "linker"-ul afiseaza aceste stub-uri si avertizeaza programatorul daca functiile nu au cum sa fie traduse.

In concluzie, Alph impune ca functiile implicate in realizarea sa sa fie functii care sunt cat mai universale. Acesta este un motiv pentru care codul este plasat de catre alti programatori, printre altele, si sub sectia de programe artistice. Adaugarea unui modul aditional la Alph impune un anumit stil de programare care este destul de lent in practica insa foarte detaliat.

3. Bazele alfa-numerice

Alfabetul se poate defini ca un set standardizat de litere, simboluri de baza scrise - fiecare reprezentand un fonem al unei limbi vorbite, fie cum exista acum sau cum a fost in trecut. Exista si alte forme de scriere anume logogramele, in care fiecare simbol reprezinta un morfem, sau cuvant, si silabe, in care fiecare simbol reprezinta o silaba.

Cuvantul "alfabet" vine din compunerea a doua simboluri "alpha" (de unde si denumirea programului care face obiectul acestei lucrari) si "beta" denumind primele doua litere din alfabetul grecesc. Exista mai multe tipuri de alfabete, majoritatea fiind lineare, aceasta insemanand ca sunt compuse din linii. Exceptiile notabile sunt alfabetul Braille, codul Morse (implementat in Alph ca parametru MORSE) si scrierea cuneiforma din orasul antic Ugarit. Pentru ca alfabetul latin, urmat de caracterele de control si de expresie adaugate de literatura, sa poata fi implementat intr-un sistem informatic s-a recurs la o standardizare numita ASCII (American Standard Code for Information Interchange) deasemenea cunoscut international drept standardul ISO646-US.

3. 1 Codul ASCII

Codul ASCII reprezinta text in calculatoare, echipamente de telecomunicatii si alte dispozitive care intrebuinteaza text. Majoritatea codarilor noi au ca baza istorica acest standard. ASCII a fost publicat ca standard in 1967 si a fost imbunatatit in 1986. La ora actuala defineste coduri pentru 33 de caractere ne-imprimabile, majoritatea fiind caractere de control de linie care afecteaza modul in care textul este procesat, plus 95 de caractere

imprimabile incluzand spatiu, caracterele alfabetului in scris mic si mare si restul semnelor de punctuatie si de algebra simpla. Ca alte reprezentari de caractere, ASCII definieste o corespondenta intre tipare de biti si simbolurile unei limbi scrise permitand astfel dispozitivelor sa comunice intre ele si sa proceseze, stocheze si sa comunice informatii pe baza de caractere. Acest cod a fost extins dupa necesitati ca sa includa si caractere care apartin altor limbi. Notabila este a treia extensie (Unicode, Universal Character Set (UCS)) de la ASCII care defineste caracterele speciale ale alfabetului roman. Strict, ASCII este un cod bazat pe sapte biti, insemnand ca utilizeaza tipare de biti reprezentate prin sapte numere binare (de la 0 la 127 in zecimal) ca sa reprezinte informatia de caractere. Pe vremea cand ASCII a fost introdus, multe calculatoare lucrau deja cu grupuri de opt biti (bytes sau octeti); al optulea bit era considerat bit de paritate pentru verificari de eroare pe diferite linii de comunicatii sau alte functii specifice. ASCII rezerva primii 32 de octeti (0-31 zecimal) pentru controlul de caractere: coduri care nu sunt destinate sa poarte vreo informatie de caractere ci doar sa controleze dispozitive (cum ar fi imprimantele) care utlizeaza ASCII. De exemplu, caracterul 10 reprezinta functia "line feed" (care comunica imprimantei comanda de a avansa pagina), iar caracterul 27 reprezinta "escape" care este identic tastei "Esc" gasit in coltul de stanga sus pe majoritatea tastaturilor. Codul 127 (toti bitii setati), alt caracter special, echivaleaza cu "delete" care a fost utilizat prin anii 80 pentru a gauri o linie completa de hartie de calculator stergand astfel toate datele de pe acea sectiune. Primii utilizatori ai codului ASCII au creat cateva coduri pentru a reprezenta meta-informatii, cum ar fi sfarsit de line etc. Acestea provoaca des un conflict atunci cand datele sunt transferate. De exemplu, sfarsitul de linie (end-of-line) in fisiere de text si date variaza de la un sistem de operare la altul. Atunci cand datele sunt mutate de la un sistem la altul, procesul trebuie sa recunoasca aceste caractere si sa le transforme corespunzator.

Alph utilizeaza aceste caractere speciale pentru a gasi terminarea datelor din sirul de caractere. Aceasta este o particularitate proprie doar programelor care asteapta caracterul de sfarsit de transmisie (End of Transmission: EOT - ASCII 0x04/004) sau caracterul de sfarsit de fisier (End of File: EOF - avand valoarea 0 sau 1 in functie de sistemul utilizat) ca sa identifice sfarsitul introducerii datelor. In cazul in care nici-unul dintre aceste doua caractere speciale nu a fost gasit in sirul de caractere, Alph procedeaza pana la o limita "BUFSIZ"

definita ca 1024 de bytes de catre standardul POSIX[6] in fisierul de intrare-iesire standard (stdio.h). Daca aceasta limita este atinsa si mai exista date de primit, programul realoca in stiva sa interna inca o data 1024 de bytes, adica inca o data limita "BUFSIZ". Acest proces continua pana s-a epuizat memoria (caz in care programul returneaza un mesaj de eroare) sau pana cand toate datele utilizatorului au fost receptate de catre program. In principiu trebuie notat ca exista doua cazuri in care datele au fost introduse cu succes: programul gaseste caracterul de sfarsit de fisier sau utilizatorul a intrerupt sirul de date prin caracterul EOT (ctr+d in sistemele Unix si ctrl+z in sistemele Windows). Aici se poate vedea diferenta in interpretare a celor doua sisteme Windows si Linux. Diferenta este doar aparenta, fiindca indiferent de combinatia de taste, caracterul de sfarsit de transmisie ramane la fel si este interpretat de catre program in mod identic. Programul asteapta una dintre cele doua variante ca sa poata sesiza sfarsitul introducerii datelor de catre utilizator.

Alph este conceput sa utilizeze codul ASCII pentru sistemele de criptare antice bazandu-se pe o proprietate interesanta a limbajului de programare C. Acest limbaj face o corelare intre un caracter si valoarea sa in cod ASCII. Anume, orice caracter unic are ca interpretare algebrica (valoarea sa) in codul ASCII. De exemplu: caracterul "a" are valoarea zecimala "97" in codul ASCII. Astfel, intr-un context algebric (operatii aritmetice asupra caracterului), "a" va lua valoarea de "97" dar, intr-un context caracter (adaugarea sa intr-un sir de caractere), "a" are pur si simplu valoarea de caracter "a". Daca am scrie o operatie simpla:

```
int valoare = 'a' + 'b';
```

unde am definit variabila "valoare" ca fiind de tip intreg, atunci ea va lua valoarea de 195; adica valoarea algebrica lui "a" (97) plus valoarea algebrica a lui "b" (98).

Alph foloseste aceasta proprietate pentru functiile de criptare care utilizeaza metoda criptografica de substituire. Programul mai apeleaza la aceasta proprietate pentru a obtine circularitatea alfabetului ('z'+1 = 'a') si pentru a fixa caracterele in limitele alfabetului latin.

---

6 "Portable Operating System Interface" - un set de standarde software introdus de catre "IEEE POSIX Working Group" ca sa permita aplicatiilor scrise pe un singur sistem si care pot sa ruleze neschimbate pe o varietate de sisteme.

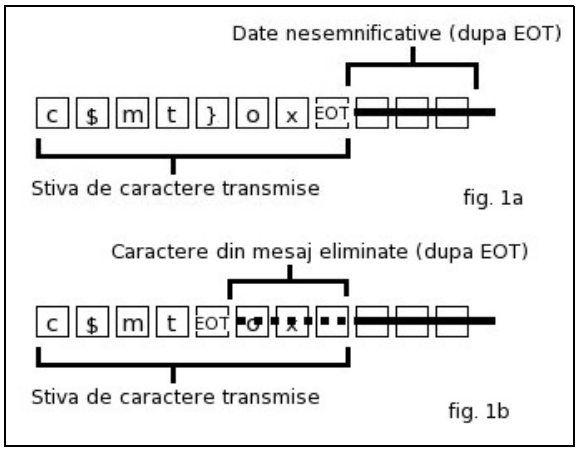

Fig. 3 Problema transmisiilor de caractere, 1a) Transmisie fara caractere pierdute, 1b) Transmisie in care se pierd caracterele dupa un “fals” EOT.

In primul caz (Fig. 3a, 1a) transmisia de date se desfasoara in mod normal, sirul de caractere generat de functia de transformare fiind transmis complet, incheiandu-se prin caracterul de sfarsit de transmisie. In al doilea caz (Fig. 3b, 1b), transmisia se incheie intr-un mod neasteptat, functia de transformare generand ea insesi un caracter de sfarsit de transmise (caracterul ASCII 0x04 - EOT) eliminand cateva caractere (caracterul "o" si "x") din mesajul complet care ar fi trebuit sa fie receptat. In final, pentru cifruri avansate care opereaza asupra blocurilor sau metode binare complexe de cifrare, Alph intrebuinteaza tot tabelul ASCII (255 de caractere) pentru a reprezenta datele intermediare. Aici se intalneste o problema de interpretare de caractere la nivel de transmisie sau de comunicare a datelor intre emitator si destinatar. Asa cum este definit setul de caractere ASCII acesta contine caractere de control care se interpreteaza univoc de catre toate sistemele. Daca datele obtinute printr-un proces de cifrare complex contin valori de control, atunci, destinatarul le va interpreta ca atare chiar daca ele nu fac parte din mesajul original. Acest lucru poate influenta receptia datelor in continuare si poate chiar sa intrerupa transmisia inainte ca mesajul complet sa fi fost transmis. Din punct de vedere al formalismului matematic: un caracter oarecare din alfabetul latin "x" este cifrat printr-o functie a unui cifru complex "f" intr-un caracter oarecare din codul ASCII, "c". Ecuatia formala se poate scrie matematic in modul urmator: c = f(x); unde

f: X->C, unde X reprezinta multimea de introducere a datelor, in cazul nostru, alfabetul latin, si C reprezinta multimea de valori, in cazul functiei nostre, intregul cod ASCII.

In momentul in care sirul de caractere (mesajul) este transmis, receptorul primeste fiecare caracter in parte si data fiind natura functiilor de programare de citire a sirurilor de caractere, acesta le interpreteaza. Daca receptorul primeste un caracter de tablulare, atunci cand va reciti mesajul, acesta va aparea in stiva. Daca receptorul primeste un caracter de sfarsit de linie, acesta nu are cum sa verifice daca mesajul a fost transmis in intregime decat in cazul in care emitatorul specifica in prealabil dimensiunea stivei ce va fi transmisa. Chiar daca este cunoscuta dimensiunea mesajului care trebuie sa fie primit, receptorul va stoca aceste caractere de control si la o urmatoare citire le va interpreta la fel. Astfel putem observa ca functia de transformare poate sa genereze caractere in cifrare care sa poata perturba transmisia completa a mesajului sau sa stocheze datele intr-o stiva de memorie si la o recitire sa le interpreteze din nou gresit. Putem sa dam un exemplu concret in care transmisia va continua insa la o recitire a mesajului de catre receptor aceasta sa decurga gresit. Un sir de caractere generat de functia de transformare ar putea sa contina caracterul special "\0" (ASCII - Null 0x00) care nu va intrerupe transmisia insa daca va fi stocat de receptor intr-o stiva, acesta va impune la recitire ca orice caracter dupa "\0" sa fie ignorat deoarece "\0" defineste caracterul special de sfarsit de stiva.

3. 2 Base64

Este asadar necesar de a gasi o baza alfa-numerica pentru a transmite un mesaj binar sau cifrat fara ca aceste caractere speciale de control sa perturbe transmisia sau continutul datelor. De aceea s-a introdus o metoda numita Base64 care este un sistem de numarare positional utilizand baza 64. Un numar in Base64 este cea mai mare putere a lui doi care poate sa fie reprezentata utilizand numai caractere ASCII afisabile. Acest lucru a dus la cifrarea transferului emailului de exemplu. Toate variantele cunoscute ale acestei baze alfanumerice utilizeaza caracterele A-Z, a-z si 0-9 in aceasta ordine pentru primele 62 de numere insa simbolurile alese pentru ultimele doua variaza considerabil intre diferitele sisteme care utilizeaza Base64. Protocoalele care utilizeaza aceasta baza de numerotare a caracterelor sunt:

1) e-mail – MIME

In formatul de e-mail MIME, Base64 este o schema de cifrare binara - text in care o secventa arbitrara de bytes este convertita intr-o secventa de caractere ASCII afisabile. Este definit ca o cifrare MIME de continut pentru a fi utilizata in internet e-mail. Singurele caractere utilizate sunt caracterele mici si mari ale alfabetului roman (A-Z, a-z), numeralele (0-9) si simboluile "+" si "/" cu simbolul special "=" utilizat ca sufix special.

2) *e-mail - UTF-7*

UTF-7 a introdus un nou sistem modificat al lui Base64. Aceasta schema de cifrare este utilizata ca sa cifreze UTF-16 utilizat ca format intermediar in UTF-7 in caractere ASCII afisabile. Este o varianta a lui Base64 utilizat in MIME. UTF-7 intentioneaza sa permita utilizarea "unicode"-ului in e-mail fara sa utilizeze un cifru de transfer separat. Principala diferenta fata de versiunea MIME a lui Base64 este ca nu utilizeaza simbolul "=" pentru sufix fiindca acesta tinde sa creeze complicatii. In schimb, pentru sufix, adauga "0" la sfarsitul fiecarui octet.

3) *IRC (Internet Relay Chat) – IRCu*

In protocolul P10 server-server utilizat de catre "daemon"-ul IRC-IRCu si software-ul compatibil, o versiune de Base64 este utilizata pentru cifrarea numerica si binara ale IP-urilor clientilor. Numericele clientilor cat si ale serverelor au o dimensiune fixa care se potriveste cu un numar exact de cifre Base64 fara nevoie de sufix. IP-urile binare au ca prefix zero-uri ca sa incapa in dimensiunile impuse. Setul de simboluri este putin diferit de MIME utilizand "[" si "]" in loc de "+" si "/".

4) *HTTP (Hyper Text Transfer Protocol)*

Base64 este foarte util pentru protocolul HTTP atunci cand acesta are nevoie sa trimita siruri de identificare cu dimensiuni mari. ID-uri mari (de marimea a 128-biti) trebuie sa fie transmise intre aplicatia de identificare si clientii respectivi prin metode conventionale HTTP cum ar fi GET/POST. De asemenea, multe aplicatii trebuie sa cifreze date binare pentru ca acestea sa fie incluse in URL-uri sau pentru generarea de "field"-uri ascunse. Utilizarea unui cifru Base64 standard ar da multe probleme atunci cand ar trebui transmise caracterele "/" si "+" care sunt de obicei transmise in URL-uri in secvente de numere hexazecimale ("%XX - unde XX este codul caracterului ASCII in hexazecimal). Cand acesta este ulterior utilizat in baze de date sau transferat intre medii de lucru eterogene vor aparea probleme de compatibiliate asupra caracterului "%" (deoarece acest caracter mai este utilizat in ANSI SQL ca "wildcard"). Din acest motiv, se utilizeaza o varianta de Base64 modificata care nu utilizeaza sufixul "=", iar in loc de caracterele "+" si "/" se utilizeaza prin conventie "*" si "-" respectiv. Alta varianta utilizeaza "!-" in loc de "*-" pentru a inlocui caracterele din Base64 deoarece "+" si "*" ar putea fi rezervate pentru expresii.

In principiu pentru a converti datele in Base64 se utilizeaza algoritmul urmator: primul byte este plasat in cel mai semnificativ opt biti dintr-o stiva de 24 de biti, urmatorul in cei opt din mijloc si ultimul in cel mai putini semnificativi opt biti. Daca exista mai putin de trei bytes de cifrat atunci bitii stivei vor fi zero. Stiva este apoi utilizata sase cu sase biti, cel mai semnificativ primul, ca indice in sirul:

"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

si caracterul de iesire indicat. Daca erau numai unul sau doi bytes de intrare atunci numai primele doua sau trei caractere din iesire erau utilizate si sufixate respectiv cu unul sau doua caractere "=". Acest lucru previne ca biti suplimentari sa fie adaugati la reconstruirea datelor. Algoritmul continua pe restul datelor de intrare.

Trebuie notat ca un sir de caractere convertit in Base64 isi mareste lungima in forma codata de circa trei ori. Acest lucru este firesc deoarece se trece de la un set de caractere cu 255 de simboluri la un set cu 64 de caractere. Astfel fiecare caracter din setul ASCII de 255

de caractere este exprimat in medie prin trei caractere din Base64, lucru evident din faptul ca: (nr. caractere ASCII)/(nr. caractere Base64) = 255/64.

Base64 face parte din metodele de "character - mapping", adica metode prin care unele caractere sunt asociate cu alte caractere din alt set. Acest lucru influenteaza transmisia prin faptul ca mesajul transmis are o lungime mai mare decat mesajul necodificat insa in acelasi timp mareste gradul de confuzie si poate sa perturbe un ascultator acesta asteptandu-se la alta marime de mesaj. Orice e-mail transmis foloseste MIME, o varianta bine cunoscuta de Base64, pentru a trimite fisiere atasate. Acest lucru mareste dimensiunea fisierului atasat de aproximativ trei ori si este adesea o problema deoarece marimea permisa de catre serverul de e-mail a unui fisier atasat nu este egala cu marimea fisierului ci cu marimea fisierului inmultita cu trei. Trebuie mentionat ca aceasta codificare in base64 se face direct de la client si nu lucreaza impreuna cu serverul. Astfel serverul nu va sti foarte bine ce se afla dincolo de literele si cifrele din Base64 ci mai degraba le va lua ca atare daca sunt precedate de antetul corespunzator mentionat in standardul rfc 2821. Nu se poate determina daca mesajul original a fost in forma binara sau in forma text si la convertire programul pentru decodificare trebuie sa stie forma originala. De obicei acest lucru se rezolva prin contextul in care a fost transmis mesajul. Formatul unui e-mail specifica daca urmatorul bloc defineste un fisier atasat prin sintaxa standard a protocolul de e-mail (rfc 2821). Cele mai cunoscut convertoare din si spre Base64 sunt doua mici programe "uuencode"/"uudecode" aparute pe platforma BSD 4.0. Din pacate multe distributii de Linux cat si unele variante mai simplificate de BSD nu au pastrat acest program in arhiva de baza "bin" (continand toate utilitarele esentiale utilizate la linia de comanda a unui terminal). Alte variante de distributii de UNIX includ aceste doua comenzi in pachete secundare care pot sa nu fie instalate pe sistem.

Pentru a programa un codificator Base64 exista doua mari optiuni in programare. Prima utilizeaza map-uri (liste definind trecerea de la un set de caractere la altul) iar a doua face apel la mutari de biti utilizand doar putin map-urile. Alph implementeaza varianta strandard al codului Base64 printr-un modul utilizand un algoritm prin operatii asupra bitilor asa cum este descris mai sus in acest capitol. S-a preferat in Alph acest algoritm deoarece este considerat a fi modern. El poate fi apelat de la linia de comanda:

./alph --BASE64

dupa care acesta asteapta, asa cum am mentionat, datele de la utilizator postfixate de un caracter de sfarsit de transmisie (EOT). Decriptarea se face simetric adaugand la apelarea programului optiunea '-d':

./alph --BASE64 -d

acesta asteptand din nou, dupa introducerea datelor, un caracter de sfarsit de transmisie. La fel se pot utiliza pipe-urile sub un sistem care le suporta. Putem lua un exemplu simplu:

echo "Universitatea Hyperion Bucuresti" | ./alph --BASE64

prin care sirul de caractere "Universitatea Hyperion Bucuresti" este trecut printr-un pipe catre programul Alph prin intermediul comenzii echo. Rezultatul este un sir de caractere in Base64:

VW5pdmVyc2l0YXRlYSBIeXBlcmlvbiBCdWN1cmVzdGkKkVVx

Pentru a verifica, introducem acest sir de caractere cu optiunea decodificare:

echo "VW5pdmVyc2l0YXRlYSBIeXBlcmlvbiBCdWN1cmVzdGkKkVVx" | ./alph --BASE64 -d

si obtinem mesajul original plus un caracter de noua linie (newline):

Universitatea Hyperion Bucuresti

Faptul ca am obtinut un caracter in plus in mesajul decodificat este o curiozitate care se explica relativ simplu. Mesajul de mai sus sub forma Base64 nu este exact codificarea

mesajului nostru original "Universitatea Hyperion Bucuresti" ci este mai degraba "Universitatea Hyperion Bucuresti\n". Acest lucru se datoreaza faptului ca am utilizat comanda "echo" pentru a transmite printr-o ‘teava’ (pipe) sirul de caractere. Descierea acestei comenzi ne arata faptul ca:

"`echo' writes each given STRING to standard output, with a space
between each and a newline after the last one."

aceasta adauga la sfarsitul sirului un caracter de noua linie. Pentru a afisa corect in Base64 sirul nostru original utilizam comanda echo cu optiunea '-n' care nu adauga un caracter de noua line la sfarsitul sirului. Astfel, si corect, sirul nostru "Universitatea Hyperion Bucuresti" se traduce in Base64 ca fiind urmatorul sir de caractere:

"VW5pdmVyc2l0YXRlYSBIeXBlcmlvbiBCdWN1cmVzdGk="

Amintim ca programul Alph trebuie sa fie transparent si sa lucreze ca filtru intre ceea ce introduce utilizatorul si iesirea pe care o asteapta. El nu trebui sa faca nici-un fel de moficiare atunci cand cifrul insusi lucreaza transparent. Exista expceptii insa pe care acesta nu le poate evita (implementat in Alph ca parametru PLAYFAIR). In acel caz el incearca sa deocdeze pana la momentul in care interpretarea umana asupra rezultatului devine obligatorie.

Se mai poate mentiona ca programul Alph lucreaza cu setul de caractere impus de ASCII si nu are vre-un alfabet pre-definit in cod. Majoritatea algoritmelor nu utilizeaza "character-mapping" pentru a codifica sau a decodifica. Exista exceptii cum ar if ENIGMA insa in principiu el se bazeaza pe operatii asupra caracterelor si relatiile lor in limbajul de programare folosit. In cazul in care se foloseste alt set de caractere (alt aflabet posibil, chirilic etc...) este posibil ca acele cifuri care nu utilizeaza "character-mapping" sa functioneze corect deoarece programul se bazeaza pe aritmetica asupra setului de caractere. Atunci cand este vorba despre un cifru specific adaptat unei limbi anume de exemplu ENIGMA, sau atunci cand se foloseste un set international de simboluri (implementat in Alph ca parametru

MORSE) pentru comunicare, Alph se asteapta la caractere din limbajul respectiv. In acelasi timp, in multe cazuri, pentru dezvoltarea unui sistem criptografic relativ modern, s-a renuntat la alfabetul specific al limbii si s-a pastrat alfabetul englez compus din 26 de caractere pentru a usura prelucrarea datelor si pentru a evita erori.

4. Tipuri de cifruri

In acest capitol facem o trecere in revista cronologica a cifrurilor prezente in Alph impartite in diverse categorii. Vom analiza in principiu cifrurile bazate pe criptografia simetrica (fara cheie publica) si asimetrica (avand cheie publica) punand accentul pe cifrurile suportate de Alph. Ne vom concentra mai ales pe criptografia simetrica deoarece, asa cum vom vedea, ea este adesea “motorul” cifurilor asimetrice. Multe metode moderne de criptografie, mai exact protocoale criptografice, utilizeaza un cifru simplu pentru a cifra datele.

4. 1 Categorii

O prima impartire in categorii este facuta de Alph si reprezinta o triere cronologica a cifrurilor implementate. Alph isi imparte functiile criptografice in "Classic", "Modern” si "Hashes" care reprezinta cifrurile clasice utilizate in criptografia traditionala, cifrurile moderne si de actualitate care mai sunt utilizate si la aceasta data si functiile de hash-ing respectiv. Sub fiecare categorie se pot gasi cifrurile si hash-urile impreuna cu parametrii de utilizare. Aceasta impartire este mai degraba operationala si referitoare la program decat o impartire standard al cifrurilor. Mai mult decat atat, sectia "Hashes" include hash-uri care sunt vechi dar si pe cele noi. Pentru a defini tipurile de cifruri avem nevoie de o impartire mai amanuntita.

Pentru a intelege mai bine trebuie mai intai stabilite primitivele criptografice. Toate serviciile criptografice se pot realiza prin mai mult primitive criptografice: distingem intre primitive de criptare: primitive de autentificare si protocoale criptografice. Primitivele de criptare pot sa fie utilizate sa ofere confidentialitate, primitivele de autentificare pot fi

utilizate ca sa ofere autentificarea de date iar protocoalele ofera autentificare la nivel de utilizator si servicii de gerare de chei.

Primele dintre acestea, adica primitivele de criptare si mai concis criptarea, ne permite sa transformam un text normal intr-un text cifrat. Ca sa ajungem la textul original, aplicam inversul transformarii, numita decriptare. Aceste transformari sunt publice si permit analiza algoritmelor lor si posibilitatea de a dezvolta implementari eficente. Un singur parametru ramane insa secret: cheia care este cunoscuta doar de catre emitator si/sau de receptor. Aceasta cheie este singurul lucru care trebuie stiut ca sa fie posibila criptarea si decriptarea mesajului. Astfel este important de a gestiona cheile personale care trebuie sa fie pastrate secrete atunci cand acestea sunt necesare. Discutam doua tipuri de primitive de criptare: *simetrice* sau cifruri conventionale si *cifruri de tip cheie publica*.

In principiu exita doua tipuri de scheme de criptare:

Cifrurile simetrice sunt cele mai vechi si cele mai folosite pana in prezent. In aceste scheme, cheia utilizata pentru a decifra textul cifrat este identica atat pentru emitator cat si pentru receptor. Cel mai cunoscut cifru din aceasta categorie este "Data Encryption Standard" (DES), care a fost preluat in 1977 de catre "American NBS" (National Bureau of Standards) sub numele de FIPS 46. De atunci a fost folosit peste tot in lume si pana la aceasta data nu au fost semnalate erori conceptuale majore. DES utilizeaza o cheie de 56 de biti care este din pacate destul de slaba. S-a determinat ca printr-o cautare exhaustiva (brute-force) pentru toate valorile posibile ale aceste chei, intr-o singura zi costurile investitiei ar fi in jur de 200 000$ (care necesita doar cateva perechi de text necifrat si textul corespunzator sub forma cifrata). In ultimii ani, E. Biham si A. Shamir si mai tarziu M. Matsui au publicat atacuri aspura cifrului DES care reusesc sa-l sparga intr-un sens academic, adica necesita mult mai putine operatii, insa acest lucru nu constituie o amenintare practica la DES fiindca necesita o suma majora de texte cunoscute cifrate si decifrate. O siguranta mai buna poate fi obtinuta utilizand DES-triplu (3DES). In acest caz, obtinem o cheie eficienta de 112 biti. In acelasi timp, acesta ofera o protectie la atacurile academice ale lui DES. Nu este suficient de a alege un cifru sigur; trebuie sa specificam o cale sigura de operare. Depinzand de natura canalului de comunicatie sau spatiului de stocare, trebuie sa alegem intre Cipher-Block-Chain (CBC), Cipher-Feedback (CFB) si Output-Feedback (OFB) asa cum este specificat in FIPS 81. Criptarea bloc cu bloc (sau Electronic Code Book (ECB)) este utilizata doar pentru criptarea

cheilor. Aceste patru sunt modurile de operare specifice pentru un cifru care opereaza asupra unui bloc de text. Ultimele doua, adica Cipher-Feedback (CFB) si Output-Feedback (OFB), sunt modurile de operare standard pentru un cifru de bloc. [3]

In mod CFB, blocul de text cifrat anterior este criptat si rezultatul este combinat cu blocul de text necifrat utilizand un OR-exclusiv pentru a produce blocul de text cifrat curent. Este posibil de a defini modul CFB astfel incat sa utilizeze un feedback care este mai mic decat un bloc de text intreg. Un vector de initializare sau o valoare "c0" este utilizata ca "seed" pentru procedura. Un exemplu de implementare este prezentat in Fig. 4.

Modul de operare OFB este similar cu modul CFB cu exceptia cantitatii de informate care este trecuta prin OR-ul exclusiv. Aceasta cantitate este generata independent de textul cifrat si textul in forma normala. Metoda se bazeaza pe un vector "s0" care este utilizat ca

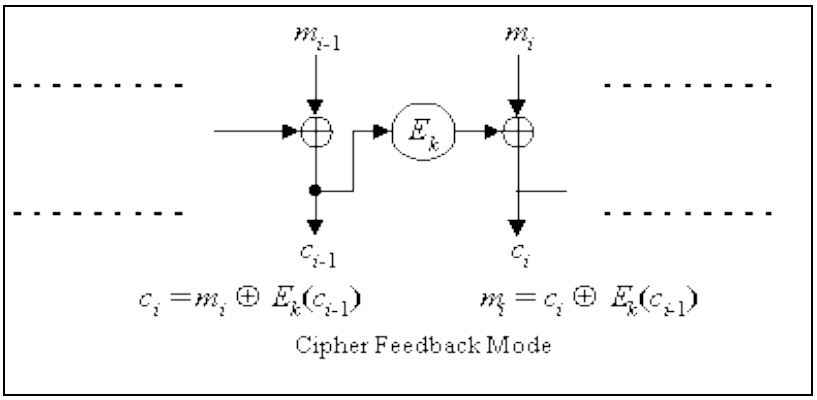

Fig. 4 Modul de functionare a cifrului DES in mod CFB. O parte din mesajul cifrat anterior este utilizata pentru a obtine urmatorul bloc de text cifrat.

"seed" pentru secventa de blocuri de date.

Putem remarca faptul ca toate aceste metode prezinta un mod de a trata datele intr-un cifru si este strict legata de formatul textului. Modurile de operare, desi produc texte cifrate distincte, nu au nici o legatura cu algoritmul cifrului. Secventa de cifrare ramane complet aceeasi si functioneaza pe aceeasi principii. Daca se doreste, se poate compara un algoritm al unui cifru cu modul de operare al programului Alph. Acesta ia o cantitate de date de la intrare, ii aplica functia de transformare a cifrului si scoate la iesire datele cifrate. Indiferent daca intrarea este chiar textul necifrat sau o portiune sau, asa cum este prezentat mai sus, un amestec intre textul cifrat si textul necifrat, algoritmul este acelasi si, logic vorbind, nu face

decat sa aplice o functie de transformare. Mai multe cifruri opteaza sa faca un amestec intre textul cifrat si textul necifrat intr-o combinatie prestabilita insa acest lucru nu rezulta intr-un nou cifru ci este, asa cum s-a precizat punctual, doar un mod de operare. In practica, in multe implementari de cifruri (si la fel pentru DES), codul indica faptul ca algoritmul de cifrare este separat de restul implementarii. Pentru a obtine operarea prin CFB, OFB sau altele, implementarea pre- sau post- proceseaza textul necifrat sau textul cifrat inainte de a aplica functia transformata de cifrare. Bineinteles ca exista o cantitate binecunoscuta de moduri de operare insa, pe de alta parte, nimeni nu este obligat sa utilizeze un mod de cifrare standard. Acest lucru implica faptul ca multe implementari sunt destul de incompatibile. In paragraful anterior am spus ca o cifrare bloc-cu-bloc este utilizata doar pentru a cifra cheia. Din pacate, alta implementare care nu considera aceasta metoda sigura sau viabila dintr-un motiv anume, poate sa opteze sa nu faca la fel. Rezultatul este mai mult sau mai putin dezastros si se rezuma in cateva cuvinte: intre diferite tipuri de implementari al unui alogritm de cifrare anume, este posibil sa constatam ca textul cifrat rezultant este diferit. Din aceasta cauza, la decriptare, se poate ca aplicand functia inversa (care aici implica si modul de operare), sa nu reusim sa decriptam un text cu o implementare anume desi algoritmul de cifrare este identic in toate implementarile. Mai clar, un text cifrat cu o implementare anume poate sa fie diferit de un text cifrat cu alta implementare chiar daca ambele implementari aplica acelasi algoritm de cifrare. Acest lucru se datoreaza modului de operare care poate sa difere intre implementari. Din fericire, cum am mai precizat, aceste moduri de operare mai sus discutate sunt oarecum standardizate si multe implementari le respecta facand ca implementarile sa fie compatibile. Trebuie insa sa recunoastem ca o regula de genul "modul de operare bloc-cu-bloc este utilizat doar pentru a cifra o cheie" nu are nici o baza criptologica. Aceasta regula nu este fondata pe faptul ca ea este mai "sigura" sau mai "eficienta" din punct de vedere al algoritmului. Chiar in acest caz (adica al operarii bloc-cu-bloc doar asupra cheiii) putem sa spunem, luandu-ne rezerva ca analizam din punct de vede strict criptologic, ca este chiar o metoda "nesigura" si ca alt mod de operare ar putea sa cifreze cheia intr-un mod mai "sigur". Acest lucru este evident chiar din punctul de vedere al complexitatii fiecarui mod de operare asupra cheii si al mesajului.

Nu le vom analiza complexitatea amanuntit insa, daca spunem ca fiecare implementare nu este mai mult decat o functie oarecare (asa cum am precizat programul

Alph nu face, in general, decat sa aplice o functie de transformare unui text dat), putem sa determinam acest grad de complexitate aplicand operatorul de complexitate a lui Landau fiecarei functii. Aceast mod de analiza se obisnuieste sa fie aplicat in practica pentru a genera statistici asupra cifrului. Din definitia teoriei complexitatii, dezvoltata de catre fizicianul Lev Landau, deducem ca acest grad de complexitate (pe care Landau l-a notat cu "O") al unei functii este determinat de puterea dominanta in functie. Acesta teorie, de natura pur matematica, este foarte utila pentru analiza complexitatii diferitor functii si ofera un instrument puternic de masura pentru diverse algoritme. Daca ne uitam la graficul de mai sus (Fig. 4) putem deduce ca o cheie va fi criptata dupa o procedura bloc-cu-bloc, adica $k_i = E_n(k_i)$ unde am notat cu E_n functia de criptare la un moment "n" si k_i portinuea de cheie curenta. Dupa cum vedem acesta este modul de operare bloc-cu-bloc descris mai sus. Acesta este o procedura lineara care are puterea: $O(k_i)=O(E_n)$ deoarece este vorba despre o functie lineara si puterea dominanta depinde strict de algorimul de criptare. Aplicam aceasi teorie si mesajului cifrat. Acesta este reprezentat in graficul de mai sus ca fiind: $c_i = xor(m_i, En(c_{i-1}))$ care este si ea o functie lineara datorita proprietatilor aditive ale functiei XOR (OR exclusiv). La fel, se poate spune ca puterea acestui mod de operare este: $O(c_i) = O(E_n)$ deoarece puterea dominanta in aceasta functie este determinata strict de algoritmul de criptare. Astfel constatam ca puterea modului de operare CFB este identica modului de operare bloc-cu-bloc.

Putem trage concluzia ca aceste moduri de operare, asa cum am mai mentionat, nu influenteaza complexitatea cifrului din punct de vedere strict algoritmic. Ele sunt mai degraba moduri de implementare specifice cu scopul de a formata textul. Desi aceasta concluzie reduce modurile de operare la niste simple optiuni de formatare, trebuie sa subliniam ca tocmai aceste metode de operare fac tratarea textului posibila. Ele sunt foarte importante deoarece impun un standard puternic de formatare care poate sa elimine erori care apar adesea in timpul transmisiei.

Este dincolo de scopul programului anexat sa implementeze aceste moduri de operare. Trebuie sa ne amintim ca programul Alph nu este intocmai program ci mai degraba o librarie care ofera algoritme criptografice. Din punct de vedere al algorimelor criptografice, modul de operare este complet irelevant. Utilizatorul librariei poate sa formateze sau sa compacteze textul asa cum doreste in programul sau, utilizand libraria Alph doar pentru a realiza functia de criptare. Acesta posibilitate ofera flexibilitate si nu forteaza programatorul

sa utilizeze vre-un mod anume. Acest lucru este util deoarece nu intotdeauna este dorita formatarea textului pentru o transmisie. Poate sa fie de exemplu o simpla criptare pentru o parola sau pentru o metoda de autentificare oarecare.

Categoria cifrurilor asimetrice sau cifrurile cheie-publica reprezinta al doilea tip de categorie generala. Ele sunt cele mai recente metode de criptare. Spre deosebire de sistemele simetrice, cheia utilizata la criptare este diferita de cheia utilizata pentru decriptare. Fiecare parte (emitator si receptor) are astfel doua chei. Fiecare are o cheie secreta si una pe care o face publica. Daca A vrea sa trimeata un mesaj lui B, atunci acesta cripteaza mesajul cu cheia publica a lui B. Fiindca B este singurul care are acces la cheia secreta, B este singurul care poate sa decifreze mesajul si sa-l citeasca.

Primul cifru de acest gen este sistemul RSA (abreviere pentru Rivest, Shamir si Adleman care sunt numele celor trei inventatori). Siguranta acestui sistem este legata de problema matematica de factorizare: este usor de a genera doua numere prime si a le multiplica, dar daca un numar foarte mare, care este produsul a doua numere prime, este dat atunci este foarte greu de a determina factorii primi. Bineinteles ca aici mai intervine timpul de viata al cheii. Trebuie notat ca fiecare cheie secreta sau publica are un timp de viata impus de catre creator. Un fapt destul de interesant este ca exista proiecte publice pe internet (unul pe care putem sa-l mentionam GIMPS) care utilizeaza calculatoarele celor "interesati" pentru a cauta si a factoriza numere prime. In mod asemanator functioneaza si proiectul SETI pentru gasirea vietii inteligente extraterestre. Un utilizator, daca doreste, poate sa ia un mic program de pe Internet care va rula in timpii morti ai procesorului pentru o cautare de numere prime si factorii lor. Intrebarea este daca acest utilizator, care participa in acest calcul distribuit, are dreptul sa vada numerele prime la care a lucrat si calculatorul sau. Raspunsul este nu. Justificarea acestui proiect este ca aflarea numerelor prime cat si factorii lor reprezinta de fapt extinderea cunostintelor omenirii. Lucru destul de controversat deoarece in 1997 cel mai mare numar factorizat gasit avea 430 de biti si poate fi utilizat pentru a ataca numere de 512 biti. Astfel, este de recomandat celor care vor sa utilizeze un sistem asimetric sa utilizeze o chie de minim 640 biti daca nu de 768 sau chiar 1024 pentru o valabilitate de cateva luni!

Probabil cel mai cunoscut cifru asimetric este PGP (Pretty Good Privacy). Acesta este o suita completa de criptografie pentru orice utilizator. Sistemul utilizeaza diferite algoritme printre care RSA, DES si El-Gamal, pentru a oferi posibilitate oricarui utilizator de a-si securiza comunicatiile sau de a genera semnaturi digitale si pentru a-si confirma identitatea.

Cea mai mare problema a sistemelor asimetrice este performanta relativ slaba comparata cu cea a cifrurilor simetrice. De exemplu, o implmentare DES pe un calculator personal 586 ar putea sa ajunga la o rata de criptare de 15Mbit/s pe cand o implementare RSA pe acelasi calculator ar ajunge doar la 6 Kbit/s. DES este astfel in medie de o mie de ori mai rapid decat un sistem RSA. In schimb, sistemele cheie-publica ofera beneficii atunci cand este vorba de administrarea cheilor: daca fiecare utilizator isi genereaza propria cheie, numai un canal autentic este necesar, eliminand canalele secrete (curieri) care sunt adesea costisitoare si nesigure.

Intr-un sistem fara un server central de incredere sau sigur, numarul de chei poate sa fie redus. Intr-adevar, sa presupunem ca avem o retea de n utilizatori care doresc sa comunice cu toti ceilalti. Fiindca fiecare comunicare cere o cheie secreta, numarul total de chei este de n(n-1)/2. In sistemul de cheie-publica fiecare utilizator are nevoie de numai o pereche de cheie personale si publice rezultand in numai 2n chei. Daca n este, sa zicem, 1000 atunci acest lucru ar insemna 500000 comparat cu 2000 in cazul unuia simetric. In sisteme cu un server central asigurat care gestioneaza cheile, ambele implementari au nevoie de acelasi numar de chei.

Astfel in practica intalnim des sisteme hibride in care se utilizeaza un sistem de cheie publica pentru distribuirea de chei secrete si un cifru simetric pentru criptarea datelor.

Functiile de hash-ing sau functiile "one-way" (pe o directie) reprezinta a treia mare categorie de cifruri. O functie one-way este definita ca o functie f astfel incat pentru fiecare x in domeniul lui f, f(x) este usor de calculat; dar pentru aproape toate y in domeniu lui f, este computational imposibil sa gasim un x astfel incat y = f(x). O conditie suplimentara este ca este greu de a gasi o a doua pre-imagine: dat fiind x si valoarea corespunzatoare a lui f(x), ar trebuie sa fie greu de gasit un x' diferit de x care are aceasi imagine sub f.

Functiile one-way sunt utilizate pentru a proteja parole: a stoca o imagine one-way a unei parole intr-un calculator decat parola insesi. Apoi aplicam aceasta functie one-way intrarii utilizatorului si verficiam daca este identica valorii stocate. Mai exact, verificam doar parolele deja trecute prin functia one-way.

O functie de hash-ing este o functie care schimba o intrare de lungime variabila intr-un numar fix de biti de iesire. Ca sa fie utila aplicatiilor criptografice, o functie de hash-ing trebuie sa satisfaca unele cerinte. Putem sa distingem intre doua tipuri de Hash-uri. Un MAC (Message Authentication Code) care utilizaeza o cheie secreta si un MDC (Manipulation Dectection Code) care functioneaza fara cheie. Pentru un MAC avem nevoie sa fie imposibil de a calcula MAC-ul fara cunostinta cheii secrete iar pentru un MDC este nevoie sa fie o functie one-way si, in multe cazuri, sa fie rezistenta la coliziuni, ceea ce inseamna ca ar trebuie sa fie greu sa se gaseasca doua argumente care se hash-eaza la aceasi rezultat.

Functiile de hash-are pot fi utlizate pentru a proteja autenticitatea unei cantitati mari de date cu o cheie secreta scurta (MAC), sau de a proteja auntenticitatea unui sir scurt de date (MDC). Cateodata un MDC este utilizat in combinatie cu cifrarea, care rezulta in protectia confidentialitatii cat si autenticitatii.

Exista cateva scheme care au fost propuse sa fie utilizate ca functii de hash-are. Cea mai utilizata schema pentru un MAC este chiar modul CBC al lui DES (cu transformari aditionale ale iesirii), asa cum este specificat in ISO-9797. Cateva MDC-uri au fost construite bazandu-se pe DES. Altele sunt bazate pe SHA (Secure Hash Algorithm pentru FIPS 180), si RIPE-MD 160. Functiile de hash-ing ajung la rate foarte mari de procesare de date. Ele sunt considerate foarte eficiente din acest punct de vedere.

Pentru a intelege mai bine cum sunt plasate aceste categorii si pentru a continua mai departe sumarizam plecand de la "arta de a scrie secrete" si alcatuim un tablou (Fig. 5) pe care il vom comenta.

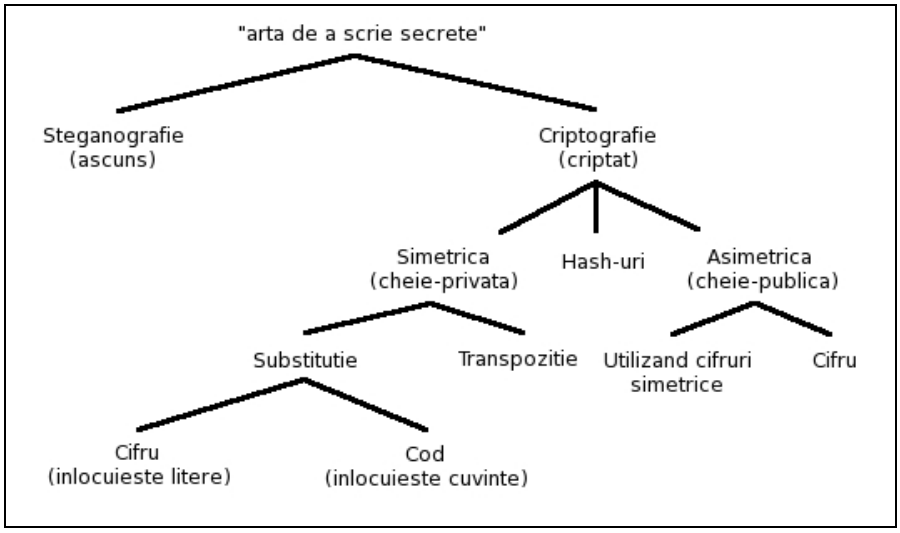

Fig. 5 Categoriile ‘artei de a scrie secrete’ reprezentata printr-un arbore.

Asa cum putem vedea (Fig. 5), la sfarsitul arborelui avem trei mari forme de a scrie secreta: Steganografia, Coduri, Cifruri si Hash-uri. Acestea sunt formele finale la care recurge un program sau o persoana pentru a aplica o transformare sau transformarea inversa respectiva pentru a cripta si/sau decripta un text. Recunoastem aici Hash-urile pe care le-am discutat mai sus in acest capitol. Ele intra intr-o categorie aparte si se diferentiaza de toate celalalte categorii prin faptul ca aceasta metoda implica un calcul matematic asupra unui intreg text pentru a-l micsora iar mesajul rezultant nu contine informatii care reflecta mesajul origial. De fapt, aceasta este o caracteristica importanta al hash-urilor. Ele nu contin deloc informatia completa a mesajului care a fost hash-at. Parte din informatie este continuta intr-un hash insa nu in totalitate. Cu alte cuvinte, asa cum am discutat, nu exista o metoda de a inversa un hash printr-o functie inversa. Din aceasta cauza, Hash-urile ocupa un loc aparte in criptografie si le categorizam ca fiind o categorie proprie.

Steganografia este o alta categorie, care nu are nici o legatura cu toate celalalte si despre care nu am mai discutat. Comunicarea secreta realizata prin ascunderea unui mesaj este cunoscuta sub numele de steganografie. Cuvantul este compus din doua cuvinte grecesti: steganos, care inseamna ascuns si graphein, care inseamna a scrie. Cateva dintre cele mai vechi referiri la scrierile secrete provin de la Herodot, "parintele istoriei", potrivit filozofului si omului de stat roman, Cicero. In "Istorii", Herodot prezinta in ordine cronologica disputele

dintre Grecia si Persia din secolul al V-lea i.Cr., pe care le-a vazut ca o infruntare intre libertate si sclavie, intre statele independente ale Greciei si persii asupritori. Potrivit lui Herodot, arta de a ascunde secrete a fost cea care a permis grecillor sa nu fie cuceriti de Xerxes, Regele Regilor, conducatorul persilor. Dusmania indelungata dintre Grecia si Persia a ajuns intr-un punct critic atunci cand Xerxes a inceput sa construiasca Persepolis, noua capitala a regatului sau. In cinstea acestei capitale, Xerxes a primit daruri si tributuri din toate colturile imperiului si al statelor vecine cu exceptia Greciei si a Spartei. Acesta a hotarat sa pedepseasca indrazneala acestor doua state de a nu-si prezenta omagiile corespunzatoare si in 480 i.C.r. a adunat cea mai mare forta militara cunoscuta vreodata in istorie.

Din nefericire pentru el, la pregatirile armatei, a fost prezent si Demaratos, un grec exilat care traia in orasul persan Susa. Acesta se simtea inca dator patriei sale natale Grecia si s-a hotarat sa trimeata un mesaj de avertizare dezvaluind complotul lui Xerxes. Problema era, cum sa poata sa trimeata mesajul fara ca acesta sa poata fi interceptat pe drum. Herodot scrie:

"Dar cum n-avea nici un alt mijloc sa le aduca la cunostinta vestile din Persia - caci era primejdie mare sa fie prins - iata ce a pus la cale. El lua o tablita dubla, ii rase ceara si apoi scrijeli pe lemn hotararea regelui. Dupa ce facut toate acestea, intinse iarasi ceara peste cele scrise, asa ca purtatorul, ducand o tablita curata, sa n-aiba nici o neplacere din partea pazitorilor drumurilor. Cand tablita ajunse in Lacedemona, lacedemonienii nu stiau ce sa creada, pana cand, dupa cate am aflat si eu, fiica regelui a dat sfatul ca ceara sa fie topita. Urmandu-i povata, lacedemonienii au gasit instiintarea, au citit-o si apoi au trimis-o la toti ceilalti eleni." [4] Deoarece avertizarea a fost primita la timp, grecii au luat masuri si au construit o flota de vreo 200 de nave urmand ca pe 23 Septembrie 480 i.C.r. sa fie luati persii prin surprindere si sa fie complet macelariti in golful Salamina de langa Atena.

Aceasta nu a fost singura data cand s-a utilizat o forma rudimentara de steganografie. Herodot mai face referire la un caz in care Histaios a vrut sa-l convinga pe Aristagoras din Milet sa se revolte impotrvia regelui persan. Pentru a-i trimete instructiunile lui Aristagoras, Histaios a ras in cap un mesager si i-a scris mesajul pe scalp. Apoi a asteptat ca sa-i creasca parul acestuia si l-a trimes pe drum. Bineinteles ca nu a trezit nici-o suspriciune si ajungand la destinatie s-a ras din nou in cap si i-a aratat lui Aristagoras mesajul. Era evident o perioada

in care lumea nu se prea grabea sa schimbe informatii rapid. Grecii nu sunt singurii care au apelat la metode de steganografie. Notabil mai este procedura aplicata in China veche conform careia se scria un mesaj pe o panza de matase fina care era apoi trecuta in ceara fierbinte si rulata intr-o bila. Mesagerul urma sa inghita aceasta bila pentru a o purta la destinatie.

In secolul al XVI-lea, savantul italian Giovanni Porta a descris cum se poate ascunde un mesaj intr-un ou fiert tare, utilizand o cerneala din amestecul a treizeci de grame de alaun cu o jumatate de litru de otet, cu care se scrie apoi pe coaja. Oul absoarbe cerneala prin porii cojii si mesajul se imprima pe albus iar apoi pentru a citi mesajul se poate decoji oul si se citeste mesajul imprimat pe albus. Steganografia include de asemenea si practica scrierii utilizand cerneala invizibila. Acest lucru este mentionat de Pliniu cel Batran care specifica faptul ca seva unei anumite plante (thitymallus) poate fi utilizata ca o cerneala invizibila. In forma uscata, seva dispare insa incalzita putin, acesta se carbonizeaza repede si apare din nou. Multe lichide au aceasta proprietate deoarece ele contin foarte mult carbon si se carbonizeaza rapid atunci cand le este aplicata o sursa calda.

O metoda foarte interesanta de steganografie este o metoda germana cunoscuta sub numele de micropunct. Agentii germani din America Latina ajungeau sa micsoreze o pagina A4 de text intr-un punct mai mic de un milimetru diametru utilizand o tehnica simpla de lentile si timpi mari de expunere al unui film obisnuit. Din pacate pentru interceptori, tehnica era dubla, iar mesajul steganografiat era si criptat in prealabil. Aici ajung sa se uneasca steganografia si criptografia.

In graficul nostru (Fig. 5) am impartit criptografia simetrica in doua ramuri: *transpozitia* si *substitutia*. Transpozitia se limiteaza sa rearanjeze pur si simplu literele dintr-un mesaj realizand un fel de anagrama. In cazul mesajelor foarte scurte aceasta metoda este destul de banala deoarece exista foarte putine posibilitati. General vorbind, se poate analiza un mesaj care utilizeaza doar transpozitia printr-o functie de permutare. Programul, Alph, implementeaza o procedura de a permuta si a gasi toate permutarile unice dintr-un mesaj prin sectia lui de utilitare (Tools) cu optinuea '-p' (permute). Aceste utilitare de analiza, pe care le vom discuta mai tarziu, sunt incluse in a doua librarie oferita de alph, alphan sau libalphan.

Pe masura ce lungimea textului creste, posibilitatile de permutare cresc exponential. Bineinteles numarul de posibilitati de permutare este egal cu permutari de numarul de litere intr-un mesaj. Exista doua posbilitati de transpozitie. S-ar putea face o transpozitie aleatoare prin aranjarea aleatoare a literelor sau s-ar putea implementa un algoritm specific pentru rearanjarea lor. In primul caz, adica daca am rearanja aleator literele dintr-un mesaj, am obtine un cifru extrem de "solid" insa decodificarea lui ar fi foarte dificila. De aceea se opteaza adesea pentru o transpozitie controlata. Un exemplu este transpozitia in Zig-Zag (utilizata in alph – ZIGZAG), care presupune rearanjarea intermitenta a literelor pe doua coloane si citirea lor secventiala pentru a obtine mesajul cifrat. Pentru a decifra, se aplica pur si simplu inversul procedurii. Acest algoritm de criptare este prezent in Alph incepand cu versiunea 0.19 a programului. Observam insa ca acest cifru, pe care l-am numit, ZIGZAG este vulnerabil total la un atac exhaustiv de cautare permutativa. Functia de analiza prin permutari este insa prezenta in Alph incepand cu verisunea 0.15. Acest lucru arata cat de slab este cifrul ZIGZAG. Pe de alta parte, daca mesajul este destul de lung, este posibil ca spargerea sa nu poate fi obtinuta in timpi computationali rezonabili (aceste detalii vor fi discutate in capitolul de analiza criptografica) deoarece numarul de permutari este prea mare.

O alta forma de transpozitie este scitalul Spartan provenind din secolul V i.C.r. Scitalul este un baston de lemn in jurul caruia se infasura o fasie de piele sau pergament. Expeditorul scrie apoi mesajul in lungul bastonul si desfasoara fasia. Pentru a decripta acest mesaj este nevoie de un scital de exact aceleasi dimensiuni ca acela utilizat la criptare. In anul 404, i.C.r., la Lysandros din Sparta s-a infatisat un mesager, plin de sange si cu vesmintele zdrentuite, unul dintre cei cinci supravietuitori ai teribilei calatorii din Persia. Mesagerul i-a inmanat cureaua lui Lysandros, care a infasurat-o in jurul scitalului sau ca sa afle ca Pharnabazos din Persia planuia sa-l atace. Multumita scitalului, Lysandros a fost pregatit si a respins atacul.

Alternativa la transpunere este *substitutia*. Una dintre cele mai vechi descrieri ale criptarii prin substitutie apare in Kama-sutra, text scris in secolul al IV-lea d. Cr. de invatatul brahman Vatsyayana, bazat insa pe manuscrise datand din secolul al IV-lea i. Cr. Kama-sutra recomanda ca femeile sa studieze 64 de arte, precum gatitul, invesmantarea, masajul si prepararea parfumurilor. Lista cuprinde de asemeni si alte arte, mai putin obisnuite, cum ar fi invocarea spiritelor, sahul, legatul cartilor si tamplaria. La numarul 45 de pe lista se afla

mlecchita-vikalpa, arta scrierii secrete, destinata sa le ajute pe femei sa-si ascunda legaturile amoroase. Una dintre tehnicile recomandate este gruparea literelor alfabetului in perechi alese la intamplare iar apoi substituirea fiecarei litere din mesajul original cu perechea ei. Aici, facem referire la capitolul 3 (Bazele Alfanumerice), si precizam ca este normal ca acesta tehnica sa fie aplicata unui alfabet par. Cel latin este par si contine 26 de litere permitand criptarea utilizand metoda din Kama-sutra.

Acesta forma de scriere secreta se numeste cifru prin substitutie, fiindca fiecare literea din textul in clar este inlocuita cu alta litera, actionand astfel pe o cale complementara in raport cu transpozitia. In transpozitie, fiecare litera isi pastreaza identitatea, schimbandu-si insa pozitia, in timp ce in substitutie fiecare litera isi schimba identitatea insa isi pastreaza pozitia.

4. 2 Atbash-ul ebraic si criptografia romana

Prima folosire a unui cifru prin substitutie in scopuri militare apare in Razboiul galic dus de Iulius Cezar. Cezar povesteste cum i-a trimes un mesaj lui Cicero, care era asediat si pe punctul de a capitula. Prin procedeul substitutiei, a inlocuit literele romane cu litere grecesti, facand mesajul neinteligibil pentru inamic. Cezar relateaza transmiterea in conditii dramatice a mesajului. In caz ca trimisul gal nu va putea patrunde pana la Cicero, il sfatuieste sa lege scrisoarea de cureaua sulitei sale si s-o arunce inauntrul fortificatiilor. In scrisoare il anunta pe Cicero ca a plecat cu legiunile si ca va sosi acolo in scurt timp; il indeamna sa-si pastreze curajul. Galul, temandu-se de primejdie, arunca sulita, asa cum fusese sfatuit. Din intamplare, sulita s-a infipt pe un turn si timp de doua zile a ramas neobservata. A treia zi, un soldat o vede, o smulge si o duce lui Cicero. Aceasta citeste mai intai el singur scrisoarea, apoi o reciteste in adunarea soldatilor, care sunt curpinsi de cea mai mare bucurie [4].

Scrierea secreta a fost utilizata pe larg de catre fortele militare romane. Mesajele care erau transmise intre cele mai departate puncte ale imperiului roman erau mai intai cifrate si apoi purtate de catre un mesager. De notat ca mesajele cifrate nu aveau ca unic scop sa fie ascunse de fortele inamice. Ele mai trebuiau sa fie ascunse de ochii mesagerului care, purtand mesajul la el, ar fi putut sa isi schimbe alianta si sa livreze mesajul unui alt destinatar. Un

astfel de caz este mentionat in istorie, ani mai tarziu, in timpul vietii reginei Maria a Scotiei care fusese tradata de catre un mesager. Acesta, in loc sa livreze scrisorile intre rebeli si regina, a "interceptat" fiecare mesaj si l-a pus in mana celui mai mare spargator de cifruri al timpului. Cifrul nu reprezenta mai mult decat un simplu amestec de cod si substitutie, foarte vulerabil chiar la vremea aceea, care a fost compromis de catre autoritati si a dovedit conspiratia impotriva reginei Eliabeta a Angliei. In era romana insa Cezar nu se temea de interceptari fiindca era singura natiune la vremea aceea care avea cunostinte amanuntite despre scrierea secreta. Se poate spune ca mai era si singura natiune dotata cu o gramatica si un limbaj dezvoltat capabil sa sustina bazele unei cifrari.

Cezar a folosit scrierea secreta atat de des incat Valerius Probus a scris un tratat despre cifrurile sale, care din pacate nu s-a pastrat. Totusi in "Vietile celor doisprezece Cezari", de Suetonius, din secolul al II-lea d.Cr., avem o descriere amanuntita a unuia dintre tipurile de cifru prin substitutie folosite de Iulius Cezar. La baza cifrului statea o simpla substitutie in care fiecare litera din mesajul clar era incrementata cu trei in baza alfabetului. Pur si simplu fiecare litera a mesajului era inlocuita cu litera aflata trei pozitii mai departe in alfabet. Datorita metodei de cifrare, acest cifru mai este cunoscut si prin numele de cifru decalat. Desi Suetonius mentioneaza doar o decalare cu trei pozitii in cifrul Cezar, e limpede ca, folosind orice schimbare intre pozitiile 1 si 25, e posibila generarea a 25 de cifruri distincte.

Atbash-ul este un cifru prin substitutie pentru alfabetul Ebraic. El consta in substituirea "aleph" (prima litera) cu "tav" (ultima), "beth" (a doua) cu shin (penultima) si asa mai departe reversand alfabetul. Cateva cuvinte in Cartea lui Jeremiah, carte care face parte din biblia Ebraica care a devenit mai tarziu o parte din Vechiul Testament, sunt criptate prin Atbash. Acest cifru a fost asociat cu metodologiile esoterice al misticismului evreu pentru interpretarea textelor religioase ebraice cum ar fi Kabbalah. Spre deosebire de cifrul lui Cezar, Atbash este complet transparent si mesajul cifrat este decifrat prin exact metoda de cifrare al algoritmului. Daca un text cifrat in Atbash ii este aplicat algoritmul de cifrare acesta rezulta intr-un mesaj clar. Aceasta particularitate nu este prezenta la Cezar. Pentru a decifra un ‘text cezar’ trebuie mai intai sa inversam functia de criptare pentru a obtine functia de decriptare. Pentru un mesaj care a fost cifrat prin decalarea cu trei poztitii inainte, functia de decriptare este decalarea cu trei pozitii inapoi.

Fiecare mesaj poate sa fie privit din perspectiva unei metode de cifrare, numita algoritm si a unei chei care precizeaza detaliile acestei metode de cifrare. In acest caz, algoritmul determina inlocuirea fiecarei litere din mesajul original cu alta litera din alfabetul cifrat iar cheia determina alfabetul cifrat. Asa cum am precizat mai devreme, cheia poate sa fie simetrica adica ea este folosita atat in criptare cat si in decriptare sau asimetrica atunci cand aceasta difera la criptare si decriptare. Un analist care ar studia un mesaj interceptat ar putea determina care este algoritmul de cifrare, sau pur si simplu algoritmul de cifrare ar putea sa fie cunoscut, dar nu ar putea sa determine care alfabet cifrat, determinat de cheie, a fost utilizat. Secole mai tarziu in 1883 in cartea "Criptografia Militara" al unui lingvist danez August Kerckhoffs von Niewenhof se precizeaza ca: "Securitatea unui sistem criptografic nu trebuie sa se bazeze pe ascunderea cripto-algoritmului. Siguranta lui depinde doar de ascunderea cheii."

In cazul decalarii Cezar, algoritmul permite 25 de chei, adica 25 de alfabete cifrate si 25 de pozitii posibile de decalare. In cazul cifrului Atbash, algoritmul permite doar o singura cheie care este de fapt oglindirea literelor din alfabet. Algoritmul Cezar este foarte vulnerabil la un atac prin forta bruta deoarece orice mesaj cifrat prin acest algoritm nu poate avea decat 25 de chei posibile si pentru a-l sparge este suficient doar 25 de incercari. Algoritmul Atbash insa nu se bazeaza pe o cheie anume si daca este cunoscut faptul ca un text este cifrat in Atbash, o simpla re-cifrare a mesajului cifrat va rezulta in texul clar.

Atbash si alte asemenea cifruri biblice erau probabil destinate mai curand sa sporeasca misterul decat sa ascunda intelesuri, insa au fost de ajuns ca sa trezeasca interesul pentru criptografia adevarata. Calugarii europeni au inceput sa redescopere vechile cifruri prin substitutie, au inventat unele noi si, in scurt timp, au contribuit la reintroducerea criptografiei in civilizatia apuseana. Prima carte europeana despre folosirea criptografiei a fost scrisa in secolul al XIII-lea de calugarul franciscan englez Roger Bacon, “Epistola despre operele secrete si desartaciunea magiei" curpindea sapte metode pentru a pastra mesajele secrete si avertiza: "E nebun cel ce scrie un mesaj in alt fel decat in acela care sa-l ascunda de plebe." Pentru Atbash, aceasta nu este ultima utilizare a sa. Secole mai tarziui cifrul Atbash inca isi spune cuvantul si este chiar folosit la descifrarea unor texte din documentele de la Marea Moarta care au fost recent descoperite la Qumran. Lucrand la textele de la Marea Moarta, Schonfield a utilizat cifrul ca sa traduca niste cuvinte care erau

neidentificate de care cercetatori. De exemplu, aplicand Atbash cuvantului "hagu", a obtinut cuvantul Ebraic, "tsaraph", care inseamna, "test." Pasajele care contin "habu" sunt importante deoarece se referea la "Profesorul Dreptatii", care este crezut a fi o referinta la Isus. Mai tarziu, Schonfield a devenit foarte interesat de acuzatiile de erezie impotriva Cavalerilor Templieri si in particular etimologia cuvantului Baphomet. Era convins ca acest cuvant ascunde alt inteles si aplicand cifrul Atbash a dezvaluit un nou inteles. Daca scriem cuvantul Baphomet in Ebraica si tinem cont de modul de citire la da dreapta la stanga si aplicam Atbash, rezultatul este prezentat mai jos (Fig. 6):

Fig. 6 O transformare simpla Atbash.

Desi scris in Ebraica, acest cuvant "Sofia" este cuvantul grecesc pentru intelepciune. Dar mai exista o conotatie pentru cuvantul Sophia care era o zeitate considerata sa fie mireasa lui Dumnezeu. Multe persoane au sustinut ca Templarii au fost urmasii zeitatii si ca ei au incercat sa restabileasca aspectul feminin al divinitatii exercitate de biserica. Trebuie reamintit ca patronul acestora, St. Bernard de Clairvaux a avut o obsesie absoluta cu Maria si a fost responsabil pentru numele ei de Regina Cerurilor si Mama lui Dumnezeu. Aceste texte interpretate cu ajutorul lui Atbash de catre Hugh Sconfield in noua lor forma au produs o serie de controverse care au resuit sa scuture fundatiile Crestinitatii si raman confidentiale pana la aceasta data.

Se poate spune in concluzie ca ambele cifruri au stabilit bazele pentru evolutia criptografiei prin substitutie. Aceste cifruri si extensiile lor au dat nastere unei intregi avalanse de noi cifruri, mai perfectionate, care domina si acum criptologia. Ambele cifruri sunt prezentate ca primele in lista de cifuri disponibile in programul Alph si sunt

implementate printr-o serie de algoritme. De remarcat este o particularitate a cifrului Atbash. Alfabetul ebraic poseda un numar impar de litere, 27 la numar, ceea ce implica un alfabet cu pivot. Dat fiind natura de oglindire al cifrului Atbash, litera din mijloc, anume Mem, este practic oglindita spre ea insesi. Astfel aplicat unui alfabet cu numar impar, Atbash-ul nu va modifica litera din mijloc care va ramane neschimbata. Acest lucru nu se aplica insa alfabetului roman care nu are pivot central si toate literele sunt schimbate prin oglindire, litera "m" corespunzand literei sale invecinate "n". Neschimbarea unei litere intr-o substitutie ar putea fi un avantaj pentru raspandirea confuziei insa in cazuri practice s-a dovedit a fi o greseala fatala atunci cand cifrul este atacat.

4. 3 Cifruri prin subsitutie

De la caderea Romei pana la al doilea razboi mondial se impun cateva cifuri care, bazate pe substitutie, ajung sa domine scena criptografiei. Este vorba aici de substitutii polialfabetice si monoalfabetice la nivel de litere bazate pe chei. Notiunea de cheie a fost introdusa prima data de catre Blaise de Vigenere, un diplomat francez nascut in 1523 care, familiarizandu-se cu scrierile altor oameni care s-au dedicat criptografiei, la varsta de douazeci si sase de ani, a fost trimis la Roma pentru doi ani intr-o misiune diplomatica. La inceput interesul sau pentru criptografie a fost strict profesional si a utilizat cifrurile existente pentru a-si securiza comunicatia. Mai tarziu, la vreo treizeci si doi de ani, Vigenere considera ca a acumulat destula avere pentru a-si abandona cariera si se dedica numai criptografiei avand ca unic scop elaborarea unui cifru perfect.

Cifrurile din acea vreme erau bazate pe substitutii monoalfabetice (adica utilizau un singur alfabet cifrat) insa Vigenere era convins ca aceasta metoda nu va putea sa ofere siguranta in mana unor profesionisti. De aceea el creeaza pentru prima data in istorie un sistem de criptografie polialfabetic utilizand chiar o cheie pentru substitutiile succesive. In primul rand el repartizeaza alfabetul intr-un patrat specific punand alfabetul de la A la Z in prima linie apoi alfabetul de la B la A prin Z pe a doua si tot asa alcatuind un patrat care contine 26 de linii de alfabete incrementate cu unu de fiecare data. El obtine astfel 26 de alfabete cifrate care pot sa fie utilizate drept substitute pentru textul in clar. Daca s-ar folosi doar o line de alfabet din cele 26, transformarea nu ar fi cu nimic mai elaborata decat o

simpla substitutie Cezar. Pentru a complica substitutia el inventeaza notiunea de cheie si spune ca se va cauta care alfabet din acest patrat incepe cu prima litera a cheii astfel incat acesta sa fie utilizat pentru cifrarea primei litere din mesajul in clar. Pentru a doua litera din cheie se cauta linia din patratul de alfabete care incepe cu aceasta si se cauta pozitia ei in acest alfabet ca sa fie utilizata pentru substitutie. Pentru a treia se repeta:

Cheie: h o r i a h o r

Mesaj in clar: h y p e r i o n

Mesaj cifrat: o m g m r p c e

Vigenere a lungit cheia prin repetitie astfel incat aceasta sa se potriveasca lungimii mesajului in clar. Acest lucru este necesar deoarece lungimea cheii poate diferi de la o cifrare

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y

Fig. 7 Exemplu de patrat Vigenere.

la alta si in caz ca aceasta este mai scurta decat mesajul original, cifrarea nu se poate face deoarece lipseste indicatia asupra alfabetului. O cheie mai lunga trece procesul de criptare prin mai multe alfabete distincte si complica mesajul cifrat rezultant.

In tabelul de mai sus (Fig. 7) am prezentat un patrat Vigenere alcatuit din 26 de alfabete. Cautarea se face dupa cheie pe linii si mesajul este cifrat pe coloane dupa alfabetul

de sus. Aceasta forma de cifrare este deosebita deoarece ea contine mai multe elemente care fortifica siguranta cifrului. In primul rand se poate observa ca o litera care apare de mai multe ori in textul cifrat poate reprezenta de fiecare data o litera diferita in textul clar. La fel de mult induce in eroare si faptul ca o litera care apare de mai multe ori in textul clar poate fi reprezentata prin litere diferite in textul cifrat. Cifrul VIGENERE admite si un numar imens de chei. Expeditorul si destinatarul pot sa cada de acord asupra oricarui cuvant din dictionar, asupra oricarei combinatii de cuvinte sau cuvinte inventate pentru a alcatui cheia. Un atac prin forta bruta de a cauta toate cheile posibile nu are o solutie in timp computational nici pe sistemele informatizate din zilele noastre.

Munca lui Vigenere s-a concretizat in "Traitte des Chiffres", publicat in 1586 insa acesta a fost neglijat pentru doua secole cand va aparea din nou in 1854 re-inventat de catre John Hall Brock Thwaites, un dentist din Bristol.

Alph implementeaza cifrul Vigenere din versiunea intiala data fiind importanta acestui cifru. Procedura este absolut identica celei descrise mai sus cu exceptia faptului ca Alph respecta punctuatia si spatiile, inserand pur si simplu caracterul specific in mesajul crifrat atunci cand mesajul original il contine.

Mai tarziu Charles Babbage impreuna cu John Hall Brock Thwaites au dezvoltat o metoda pentru a sparge cifrul Vigenere utilizand functii rudimentare de criptanaliza pe care le vom acoperi in alt capitol. Aceasta descoperire a facut cifrul VIGENERE destul de inutilizabil si in secolul al XIX-lea practic nu mai exista un cifru destul de solid pentru a putea fi utilizat in aplicatiile serioase.

Pe la mijlocul acestui secol insa, s-a publicat un document descriind un cifru semnat de catre Wheatstone pe data de 26 Martie 1854. Schema a ajuns sa fie cunoscuta dupa numele unui prieten de-al sau, Lord Playfair care l-a popularizat. La inceput a fost rejectat de catre biroul de afaceri straine datorita complexitatii sale. Atunci cand Wheatstone s-a oferit sa demonstreze ca trei din patru copii intr-o scoala ar putea sa il invete in cinsprezece minute, secretarul biroului de afaceri straine britanic a raspuns: "Acest lucru este foarte posibil insa nu a-ti putea sa faceti diplomatii sa-l inteleaga." El a fost utilizat in primul razboi mondial de catre britanici iar al doilea razboi mondial de catre australieni.

Cifrul PLAYFAIR sau patratul Playfair este o cifrare manual simetrica si una dintre primele substitutii digrafice. Tehnica cifreaza perechi de litere (digrafe) in loc de o litera

singura ca intr-un simplu cifru de substitutie. Playfair este astfel semnificativ de greu de spart deoarece metodele de analiza criptografica traditionale esueaza. De asemenea, el este primul cifru care face apel la un asa zis patrat "polybus". Acesta este un tabel generat de catre o cheie de 5 pe 5 care contine 25 de litere din alfabet o singura data. Pentru a genera un tabel dupa cheie, ar trebui mai intai sa umplem spatiile din tabel cu litere din cheie luand doar o data in considerare literele duble din cheie. Apoi s-ar umple spatiile care au ramas cu restul literelor din alfabet in ordine. Pentru a cifra un mesaj, am sparge mesajul in clar in grupe de doua litere si le-am deduce din tabel utilizand patru reguli:

- Daca ambele litere sunt aceleasi (sau daca a ramas doar o singura litera), adauga un "X" dupa prima litera. Cifreaza noua pereche si continua.
- Daca literele apar pe acleasi rand din tabel, inlocuiestele cu literele din dreapta lor trecand la celalta latura daca ele sunt la sfarsitul patratului.
- Daca literele apar pe aceasi coloana din tabel, inlocuiestele cu literele imediat dedesubt trecand la celalta latura daca ele sunt la sfarsitul patratului.
-Daca literele nu sunt pe aceasi linie sau coloana, inlocuiestele cu literele pe acelasi rand respectiv dar la celealta pereche de colturi a unui dreptunghi definit de perechea initiala. Ordinea este importanta - prima literea cifrata este perechea care se afla pe acelasi rand ca prima litera din mesajul in clar.

Pentru a decrifra mesajul se utilizeaza inversul acestor reguli. Particularitatea lui PLAYFAIR este ca acest cifru nu este complet reversibil datorita adaugarii lui "X" in cifru. Transformarea inversa la decriptare rezulta intr-un text aproximativ egal cu textul in clar. Acest lucru deruteaza bineinteles orice fel de analiza asupra textului cifrat facand orice tentativa de spargere a cifrului aproape imposibila. Pe de alta parte, cifrul nu reuseste sa indeplineasca conditia de transparenta iar acest lucru ar putea induce destinatarul in eroare atunci cand acesta va decifra mesajul. Indiferent de acest lucru si de capabilitatea destinatarului de a decifra textul rezultant din transformarea inversa, PLAYFAIR reuseste sa obtina un grad foarte ridicat de confuzie in textul sub forma cifrata.

Alph implementeaza acest cifru urmand cele patru reguli mentionate mai sus si respectand semnele de punctuatie. Algoritmul genereaza mai intai un patrat polybus din cheie

si aplica cele patru reguli asupra textului in clar. La descifrare el aplica inversul lor si obtine mesajul in clar. De multe ori acesta poate sa apara distorsionat data fiind natura cifrului si lasa utilizatorul sa interpreteze mesajul in clar.

Spre sfarsitul secolului al XIX-lea, Wheatstone descifreaza un anunt in "The Times" al unui student la Oxford care ii sugera iubitei lui sa fuga impreuna. Cateva zile mai tarziu, Wheatstone a introdus propriul sau mesaj criptat cu acelasi cifru, sfatuind cuplul sa nu comita acest act necugetat. La scurt timp, a aparut un al treilea mesaj, de data asta necriptat si provenind de la doamna in cauza: "Draga Charlie, nu-mi mai scrie. Cifrul nostru a fost descoperit." Curand apar o gramada de texte cifrate in ziare. Criptografii au inceput sa lanseze calupuri de texte cifrate doar sa-si provoace colegii. Altii introduceau mesaje criptate critice la adresa unor politicieni si curand intreaga literatura si presa erau implicate in domeniul criptografiei. Putem aminti povestirile "Calatorie spre centrul Pamantului" de Jules Verne in care intreaga povestire pleaca de la descifrarea unui pergament sau "Aventura barbatului care danseaza" in care protagonistul, Sherlock Holmes, prezinta o metoda de descifrare al unui text bazat pe o substitutie monoalfabetica prin smboluri. In America, literatura era dominata de catre Edgar Allan Poe care scriind pentru ziarul "Alexander Weedy Messenger" din Philadelphia, lanseaza o provocare cititorilor pretinzand ca poate descifra orice mesaj cifrat prin substitutie monoalfabetica. Sute de cititori au trimes mesajele lor prin posta iar el le-a descifrat cu succes pe toate. Aceste evenimente sunt ilustrate intr-o povestire scurta "Scarabeul de aur", scrisa de Poe in care se prezinta un exemplu de steganografie si criptografie.

In jurul anului 1885, este publicat un articol numit "The Beale Papers" de 23 de pagini continand un text cifrat despre locatia secreta a unei comori valorind vreo 20 milioane de dolari. Acest cifru a fascinat generatii intregi de persoane interesate de criptografie prin natura sa distincta. Cifrul BEALE, numit dupa autorul sau, este un cifru care utilizeaza numere pentru a substitui caractere din mesajul in clar. Orice criptolog s-ar gandi imediat la o analiza de frecventa care ar da rezultatele in litere demascand astfel misterul numerelor. Din pacate, cifrul BEALE transforma literele in numere intr-un mod aleator fiind posibil ca o litera sa nu apara ascunsa sub acelasi numar de mai multe ori. Acest lucru indica un cifru cu substitutie polialfabetica. Totusi analiza de frecventa nu merge deoarece se pare ca numerele

par in ordine aleatoare iar diferitele alfabete cifrate nu sunt utilizate intr-o succesiune prestabilita asa cum este in cazul cifrului VIGENERE.

Algoritmul BEALE introduce o serie de concepte noi care nu au mai fost utilizate in alte cifruri. In primul rand cheia utilizata pentru cifrare trebuie sa contina toate literele din textul in clar cel putin o data. Este indicat ca literele din cheie sa contina de mai multe ori literele din mesajul in clar. BEALE utilizeaza literele din cheie ca "pozitii numerice" pentru mesajul in clar. De exemplu, daca folosim Alph pentru a cifra textul "Hyperion", entropia cheii ar putea sa fie ceva de genul "YipYipHeron". Cheia contine astfel de mai multe ori litera "y", "i" si "p" continute in mesajul original. BEALE transforma cheia intr-o serie de pozitii. Astfel, cheia s-ar putea scrie:

Y	=	1,4
i	=	2, 5
p	=	3, 6
H	=	7
e	=	8
r	=	9
o	=	10
n	=	11

Utilizand aceasta cheie, mesajul original este transformat in: "7-1-6-8-9-5-10-11", adica pozitiile cheii in loc de mesajul original. Daca mai rulam o data insa Alph pentru acelasi mesaj si acelasi cheie, am putea sa obtinem: "7-1-6-8-9-2-10-11" care este de asemenea un raspuns valabil. Notam ca diferenta intre primul si al doilea rezultat este cifra "2" in loc de "5". Acest lucru este complet valabil fiindca cifra "2" reprezinta pozitia literei "i" in cheie care se alfa atat pe pozitia "2" cat si pe pozitia "5".

Pentru o cheie foarte lunga (eventual "running-key" ca in alph - VERNAM) obtinem rezultate complet diferite la fiecare rulare cat si in timpul rularii pentru aceasi cheie si acelasi mesaj de cifrat. Siguranta acestui cifru creste exponential cu lungimea cheii. Pentru o cheie foarte mare avem text cifrat complet diferit pentru aceasi cheie. O analiza a mesajului cifrat

ne va duce in eroare fiindca mesajul nu-si pastreaza o ordine bine definita pana la sfarsitul sau.

Alph implementeaza BEALE in totalitate respectand semnele de punctuatie. Singura informatie irecuperabila la decrifrarea unui mesaj cifrat BEALE este capitalizarea literelor. Lucru care este firesc deoarece aceasta informatie se pierde atunci cand mesajul clar este transformat in numere. Numerele nu au cum sa fie purtatoare de informatie asupra capitalizarii literelor si aceasta se pierde. Acest lucru nu schimba cu nimic procedura BEALE. Chiar in versiunea originala, din articolul Beale, decifrarea nu da nici-un indiciu asupra capitalizarii literelor. Ceea ce este diferit in implmentarea Alph este pastrarea semnelor de punctuatie, dealtfel prezenta in toate cifrurile implementate de Alph, pentru indeplinirea functiei de modul transparent. Daca nu se doreste pastrarea semnelor de punctuatie atunci acestea se pot elimina din mesajul original.

Misterul Beale continua pana si in zilele noastre si nu este cunoscuta oficial locatia acelei comori de 20 de milioane de dolari. Unele specultii sustin faptul ca el a fost spart de catre Agentia de Siguranta Nationala al Americii insa nu a fost nimic declarat oficial. In prefata publicatiei din 1885, Beale include o nota de autor care consta intr-un sfat oamenilor care ar incerca sa dezlege misterul si sa afle locatia acelei comori:

"Inainte de a face publice documentele, as vrea sa spun o vorba celor care ar putea fi interesati de ele, dandu-le cateva sfaturi pe care le-am dobandit in urma experientei mele amare. Asadar: dedica acestei sarcini numai atata timp cat iti permite munca ta, iar daca n-ai deloc timp, las-o balta... Repet, sa nu sacrifici niciodata interesele tale si ale familiei pentru ceva care se poate dovedi o iluzie, asa cum am facut eu; insa, asa cum am mai spus, cand ai incheiat ziua de lucru si stai comod la gura sobei, putin timp dedicat acestui subiect nu poate face rau nimenui, iar eforturile ti-ar putea fi rasplatite."

4. 4 Coduri

Din puct de vedere tehnic, un cod este o substitutie la nivelul cuvintelor sau al expresiilor pe cand un cifru este o substitutie la nivelul literelor din alfabet. Prin urmare, codare inseamna substituirea cuvintelor sau al expresiilor pe cand cifrare inseamna subsitutia literelor. Contrar definitiilor destul de permisive ale acestor cuvinte, termenul de "cod" si

"cifru" sunt foarte distincte in domeniul criptografiei. Este adevarat ca ambele implica acelasi resultat, adica ascunderea intelesului unui mesaj, insa acestea opereaza in moduri complet diferite.

La prima vedere codurile par a oferi mai multa siguranta decat o simpla substitutie asupra literelor. In ultimul caz, pentru a descifra un cifru monoalfabetic trebuie doar descifrate valorile reale ale alfabetului pe cand intr-un cod trebuie identificat un numar nedefinit de cuvinte sau expresii. In practica insa, codurile prezinta doua mari neajunsuri. In primul rand pentru a stabili un protocol intre emitator si receptor trebuie alcatuita o carte a codurilor care va fi baza substitutiilor intre cuvintele sau expresiile din mesajul original si rezultatul codat. O astfel de carte ar putea cuprinde o suma nedefinita de cuvinte de cod care ar substituii mesajul original. Acest lucru este foarte daunator vitezei si flexibilitatii metodei de ascundere al mesajului. Mai mult decat atat, trebuie sa ne inchipuim ca aceste cuvinte ar trebui sa fie invatate pe dinafara de catre ambele parti deoarece tiparirea unei asemenea carti de cod ar putea sa coste mai mult decat intregul efort de comunicare. Din alta perspectiva am putea spune ca deindata ce o asemenea carte a codurilor ar cadea in mainile inamice, intreaga comunicare pe baza acesteia ar fi compromisa si chiar daca s-ar afla aceasta compromitere de catre utilizatorii acestei carti de cod (ceea ce nu este deloc un fapt sigur), intreaga comunicare ar fi blocata pana cand s-ar alacatui o noua carte a codurilor.

Din aceasta cauza se utilizeaza mai degraba o combinatie intre un cifru si un cod pe baza unui "nomenclator". Un "nomenclator" este un sistem de criptare care are la baza un alfabet cifrat, folosit pentru cea mai mare parte a unui mesaj, si o lista de cuvinte de cod. La cifrare se utilizeaza ambele, cifrul si codul intr-o ordine dorita. Acest lucru nu este cu mult mai sigur decat un cod sau cifrul alaturat insa complexitatea metodei creste si, in primul rand, cartea codului (mai precis aici "nomenclatorul") devine mai mica si mai simpla de memorat. Un lucru important insa este faptul ca un cod poate fi utilizat pentru un timp limitat. Atunci cand mesajele de telegraf au fost la moda in comunicatiile de lunga distanta, s-a dezvoltat un numar mare de coduri comerciale care codau fraze complete in cuvinte simple (deobicei grupe de cate cinci litere). Astfel telegrafii au devenit familiari cu aceste cuvinte cum ar fi AYYLU ("Nu a fost codat clar, va rugam repetati."). Selectia acestor cuvinte a fost facuta pentru mai multe motive: lungime, pronuntabilitate etc. Intelesurile au fost alese ca sa se potriveasca nevoilor cum ar fi negocierilor comerciale, termeni militare, termeni diplomatica

si toate la un loc pentru spionaj. Cartile de cod au fost implementate in primul rand ca sa diminueze costurile de cabluri. Utilizarea codarii datelor pentru compresia datelor precede era calculatoarelor. Un simplu exemplu este codul de telegraf Morse unde caractere frecvent folosite au reprezentari mai scurte. Tehnici precum codarea Huffman sunt acum utilizate in algoritmele interene ale calculatoarelor pentru a comprima fisiere mari de date intr-o forma mai compacta pentru stocare si transmisie. O prima codare in era calculatoarelor a fost codarea in binar iar apoi, cum am discutat intr-un capitol anterior, codarea ASCII.

4. 4. 1 Morse

Inceputurile codurilor nu sunt precis definite. Cea mai veche notatie gasita vorbeste de Pingala, un invatat care a trait in India veche undeva intre anii 400 i.Cr. si 200 i.Cr., care a utilizat codul binar de silabe lungi si scurte, foarte similar cu Morse, cu o silaba lunga egala in lungime cu doua silabe scurte, pentru a reprezenta simboluri muzicale. Incepand cu anii 1830, Samuel Morse si Alfred Vail au dezvoltat un telegraf electronic utilizand curentul electric pentru a controla un electromagnet care a fost localizat atat la nodul de transmisiei cat si la cel de receptie. Limitele tehnologice la timpul acela facea imposibila afisarea caracterelor individuale si invetatorii au trebuit sa implementeze o metoda alternativa de comunicare. Incepand cu 1837, William Cooke si Charles Wheatstone operau telegrafele electrice in Anglia, care deasemenea controlau electromagneti in receptori insa, sistemele lor, roteau ace ca sa indice caracterele transmise. In contrast, sistemul de telegraf a lui Morse si Vail, care au inceput sa opereze in 1844, marca o hartie atunci cand un curent electric era transmis. Electromagnetul receptorului rotea o armatura astfel incat incepea sa zgarie o hartie miscatoare, si atunci cand curentul era suprimat, receptorul retracta armatura astfel incat o portie de hartie ramanea nemarcata.

Codul Morse a fost dezvoltat astfel incat operatorii puteau sa traduca indentatiile marcate pe hartie intr-un mesaj text. Initial, Morse a planuit sa transmita numai numere, si sa utilizaeze un dictionar pentru a transforma numerele in cuvinte. Mai tarziu, codul a fost largit ca sa includa litere si caractere speciale ca sa poata sa fie utilizat pentru mesaje mai complete. Marcarile mai mici erau numite puncte si cele mai mari, linii si literele cele mai frecvent utilizata in limba engleza au fost corelate celor mai scurte secvente.

In telegrafele Morse originale, armatura receptorului producea un zgomot atunci cand se misca. Operatorii au invatat direct sa citeasca zgomotul la inceputul si sfarsitul secventelor de puncte si linii iar utilizarea hartiei a devenit redundanta.

Cand codul Morse a fost adoptat de catre transmisiile radio, punctele si liniile erau in mod normal transmise ca tonuri lungi si scurte. S-a observat ulterior ca multe persoane au devenit mai eficiente ascultand Morse decat urmarindu-l in scris. Ca sa oglindeasca sunetul Morse, operatorii vocalizau o linie ca "dah" si un punct ca "dit". Atunci cand dit nu este un element final al unui caracter, sunetul este prescutat la "di-" ca sa mentina un ritm vocal imbunatatit.

Mesajele in cod Morse au fost in general transmise de catre operatori manuali, si astfel au fost introduse variatii depinzand de emitator si receptor. In general orice cod reprezentand simboluri scrie la lungimi variabile poate sa fie numit un cod Morse dar termenul este utilizat specific pentru doua tipuri de Morse utilizat pentru alfabetul englez si simbolurile asociate.

Ulterior codul Morse a fost scos din uz in 1999 atunci cand a fost subtituit cu Systemul Maritim Global de Siguranta. Cand flota franceza a incetat sa utilizeze codul morse in 1997, mesajul final transmis a fost "Catre toti. Acesta este ultimul apel inainte de linistea eterna."

Mai recent au fost organizate competitii de viteza intre expertii codurilor Morse si expertii de mesagerie rapida ale telefoanelor celulare (SMS). Codul Morse a castigat de departe influentand producatorii de telefonie mobila sa implementeze din nou acest cod pentru transmisii. Chiar la ora aceasta unele companii, precum Nokia, au produs mobile capabile sa transmita mesajul primit sub forma de SMS prin vibratii.

Codul international Morse a fost inventat de catre Friedrich Clemens Gerke in 1848 si a fost utilizat pentru telegrafie intre Hamburg si Cuxhaven in Germania. Dupa cateva mici schimbari in 1865 a fost standardizat la congresul international de telegrafie de la Paris (1865), si mai tarziu normat de catre ITU (International Telecommunications Union) ca fiind codul Morse international. Codul international Morse mai este utilizat si astazi, desi a devenit aproape exclusiv domeniul radioamatorilor. Pana in 2003, ITU a mandatat eficienta codului Morse ca parte din licentierea radio amatorilor din lume. In multe tari, anumite parti de banda ale radioamatorilor mai sunt inca rezervate pentru transmisia de coduri Morse. Fiindca Morse

se bazeaza pe un sistem de intreruperi, el cere un echipamant mai putin complex si poate sa fie utilizat in medii de zgomot inalt si mic. De asemenea, el cere o banda de date foarte mica comparata cu transmisia de voce, tipic intre 100-150Hz, comparat cu o banda de 4000Hz pentru o banda de voce. Mai mult decat atat, codul Morse intre radio-amatori este combinat cu un cod numit Q-cod sau Z-cod [7] care evita dificultatile de comunicare intre radioamatori care nu vorbesc aceasi limba.

Exista doua simboluri pentru a reprezenta litere, numite puncte si linii sau mai précis, dit si dah. Lungimea unui dit determina viteza la care mesajul este trimes si este utilizata pentru o referinta de timp. De asemenea codul Morse include simboluri pentru semnele de punctuatie incluzand spatiile. Intr-un mesaj Morse, un dah este conventional de trei ori lungimea unui dit. Spatierea intre dit si dah intr-un caracter este de lungimea unui dit. Spatierea intre litere intr-un cuvant este de lungimea unui dah (3 dit). Spatierea intre cuvinte este de sapte dit. Remarcabil in codul Morse este existenta caracterelor speciale asemanatoare codului ASCII despre care am discutat. Codul Morse include secvente speciale pentru EOT de exemplu si altele precum EOC (End of contact) sau Clear (Terminarea comunicarii) sau SOS (Save our Souls).

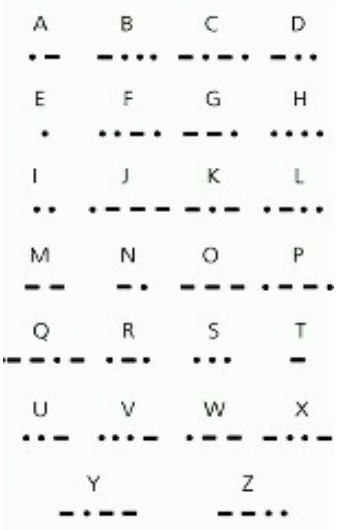

Fig. 8 Codul Morse.

7 Coduri specifice radioamatorilor permitand comunicarea prin acronime desemnand diferite mesaje universale. Acestea sunt speciale si datorita lor se poate face o comunicare de baza intre operatori care nu vorbesc aceeasi limba.

Alph implementeaza codul Morse, incepand cu versiunea 0.16, intr-unul din modulele sale. El utilizeaza conventia dit si dah pentru a transforma mesajul necodat intr-o secventa de puncte si linii. De asemenea include simbolurile pentru semnele de punctuatie cat si simbolurile pentru numerale. Ceea ce nu face este sa includa simbolurile speciale deoarece acestea pot fi implementate manual si nu sunt la fel de standardizate precum restul de cod Morse. El opereaza asupra decriptarii unui mesaj in mod asemanator si transforma un mesaj codat in Morse intr-un text normal respectand toate simbolurile codului Morse. Cum am mai precizat, modulul este separabil de Alph si poate fi utilizat ca o librarie pentru a cifra un mesaj in alt program decat executabilul Alph.

4. 4. 2 Navajo

Al doilea razboi mondial reprezinta un eveniment esential pentru dezvoltarea criptografiei. In acest timp apar diverse metode de a securiza un mesaj datorita nevoii de a comunica securizat si pentru nu a fi depistat de catre inamici. Apar diverse masini de criptografie si diverse cifruri de mana pentru a securiza orice transmisie. Se poate spune ca aproape fiecare tara care a participat la acest razboi si-a dezvoltat o metoda de securizare si pe langa aceasta si un centru de cercetarea transmisiilor interceptate. Bineinteles ca aceste metode erau foarte complexe insa foarte repede s-a constatat ca in toiul bataliei pe un camp de lupta, utilizarea unui cod complex pentru cifrare si decifrare se dovedeste destul de lenta si nu poate sa poarte informatia de la un grup de oameni la altul in timp necesar. Un corespondent de razboi american vorbeste despre acest neajuns: "Cand campul de batalie se reducea la o zona mica, totul se hotara de la o clipa la alta. Nu era timp pentru cifrari si descifrari. In asemenea clipe, engleza neoasa devenea ultimul sprijin - cu cat mai colorata, cu atat mai bine." Din nefericire pentru soldatii americani, multi japonezi absolvisera deja colegii americane si vorbeau fluent engleza si astfel informatii valoroase privind strategia si tacticile americane cadeau in mainile inamicului.

Unul dintre primii oameni care au reactionat la acest neajuns a fost Philip Johnston, un inginer stabilit la Los Angeles, prea batran ca sa lupte, dar dorind sa participe la eforturile de razboi [5]. La inceputul lui 1942 a inceput sa creeze un sistem de criptare inspirandu-se din cunostintele dobandite in copilarie. Fiu al unui misionar prostestant, Johnston crescuse in

rezervatiile Navajo din Arizona si cunostea bine cultura Navajo. El era unul dintre putinii oameni care putea sa vorbeasca fluent limba acestora, drept care a fost folosit ca interpret in discutiile dintre cei din tribul Navajo si functionarii de stat. Foarte repede a devnit traducator la Casa Alba pentru reprezentanti din tribul Navajo care cereau presedintelui Roosevelt drepturi pentru comunitatea lor. Dandu-si seama cat de complexa era limba acestora, Johnston s-a gandit ca ea ar putea servi drept cod practic imposibil de spart pentru persoane care nu si-au trait copilaria impreuna cu oamenii din tribul Navajo. Mai tarziu, i-a impartasit ideea locotenent-colonelului James E. Jones, ofiterului de transmisiuni de la Camp Elliot, in apropiere de San Diego. A fost suficient sa-i arunce cateva fraze in Navajo ofiterului uluit, pentru ca Johnston sa-l convinga ca ideea merita toata atentia. Doua saptamani mai tarziu, s-a intors cu doi indieni Navajo, gata sa faca o demonstratie in fata ofiterilor superiori din marina. Cei doi Navajo au fost izolati, iar unuia dintre ei i s-au dat sase mesaje tipice in engleza, pe care acesta le-a tradus in Navajo si le-a transmis colegului sau prin radio. Destinatarul Navajo a tradus mesajele inapoi in engleza, le-a notat si le-a inmanat ofiterilor, care le-au comparat cu originalele. Jocul susotelii in Navajo s-a dovedit imbatabil, iar ofiterii din marina au autorizat un proiect pilot si au ordonat ca recurutarile sa inceapa imediat.

Problema care se punea, era recrutarea unor indieni dintr-un trib cu un grad mare de alfabetizare. Johnston a ales tribul Navajo ca demonstratie datorita legaturilor sale personale cu acesta. Ofiterii superiori cautau insa oameni dintr-un trib destul de mare care sa poata vorbi fluent engleza si sa stie limba lor nativa. Au fost considerate si alte triburi in alegere insa Johnston a hotarat ca Navajo este tribul ideal spunand:

"Navajo este singurul trib din Statele Unite care nu a fost infestat cu studenti germani in timpul ultimilor douazeci de ani. Acestia au studiat diversele dialecte tribale pretinzand ca sunt studenti la litere sau antropologi etc. si au dobandit fara indoiala o cunoastere temeinica a tuturor dialectelor tribale in afara de Navajo. Din acest motiv, Navajo este singurul trib care poate oferi securitate deplina in actiunea noastra. De asemenea, tinem sa precizam ca dialectul tribului Navajo este complet neinteligibil pentru toate celalalte triburi si pentru toti ceilalti oameni eventual cu exeptia a 28 de americani care l-au studiat. Acest dialect este echivalentul unui cod secret pentru inamic, si e perfect adaptat unei comunicari rapide si sigure." [5].

S-a alcatuit un dictionar, o carte a codurilor, de 274 de cuvinte. Ramanea problema traducerii cuvintelor mai putin previzibile si numerele de persoane si locuri. Solutia a fost inventarea unui alfabet fonetic cifrat pentru ortografierea cuvintelor dificile. De exemplu cuvantul "Pacific" era scris ca "pig", "ant", "cat", "ice", "fox", "ice", "cat" (porc, furnica, pisica, gheata, vulpe, gheata pisica), cuvinte care erau apoi traduse in Navajo: bi-sodih, wol-la-chee, moasi, tkin, ma-e, tkin, moasi. In opt saptamani cursantii au invatat tot dictionarul si alfabetul, facand inutile cartiele de coduri care ar fi putut cadea in mainile inamicului. Pentru cei din tribul Navajo, memorarea era un lucru banal, cultura lor necunoscand scrisul, asa incat se obisnuisera sa memoreze povestile populare si istoriile familiilor. Cum spunea William McCabe, unul dintre cursanti, "in Navajo, totul se afla in memorie - cantece, rugaciuni, absolut totul. Asa am fost crescuti".

La sfarsitul perioadei de pregatire, indienii Navajo au fost testati. Expeditorii au tradus o serie de mesaje din engleza in navajo, le-au transmis, iar apoi destinatarii au tradus mesajele inapoi in engleza, folosind, dupa caz lexiconul si alfabetul pe care le invatasera pe de rost. Cursantii s-au descurcat perfect. Pentru a verifica eficienta sistemului, o inregistrare a transmisiunilor a fost inaintata Serviciului de Informatii Secrete al Marinei, echipa care sparsese o gramada de coduri printre care si Purple, cel mai puternic cifru japonez. Dupa trei saptamani de criptanaliza intensa, mesajele le dadeau inca dureri de cap spargatorilor de coduri. Acestia au numit limba Navajo o "succesiune stranie de sunete guturale si nazale, de ti se impleticeste limba in gura... nici macar nu putem s-o transcriem, daraminte s-o descifram”. Codul Navajo a fost considerat un succes. Doi soldati navajo, John Benally si Johnny Manuelito, au ramas sa antreneze urmatorul grup de recruti iar ceilalti 27 de vorbitori de cod Navajo au fost repartizati la patru regimente si trmisi in Pacific.

Impenetrabilitatea codului Navajo se datora faptului ca navajo apartine familiei limbilor na-dene care nu se inrudesc limbilor asiatice si europene. Un verb in navajo e conjugat nu numai in functie de subiect ci si in functie de complement. Terminatia verbului depinde de categoria careia ii apartine complementul: lung (de ex., pipa, creion), subtire si flexibil (sarpe, curea), granulat (zahar sare), sub forma de snopi (fan), vascos (noroi), si multe altele. Verbul incorporeaza de asemenea adverbe, care arata daca vorbitorul a trait ceea ce spune sau povesteste din auzite. In consecinta un singur verb poate echivala cu o intreaga propozitie, decifrarea intelesului ei fiind practic imposibila pentru straini.

Unele din cuvinte care nu se aflau in cartea de coduri cuprinzand cele 274 de cuvinte autorizate trebuiau insa transcrise litera cu litera. Astfel, a devenit de uz comun transmisia litera cu litera a cuvintelor. Solutia a fost sa se adauge cuvinte care sa serveasca drept substitute suplimentare (homofone) pentru litere folosite in mod curent. Doua cuvinte suplimentare au fost introduse ca alternativa pentru fiecare dintre cele sase litere cele mai frecvente (e, t, a, o, i, n), iar un cuvant suplimentar pentru urmatoarele sase (s, h, r, d, l, u).

In total au existat 420 de vorbitori de cod navajo. Desi curajul lor ca luptatori a fost recunoscut, rolul special pe care l-au jucat in securitatea comunicatiilor a fost considerat informatie secreta. In cele din urma, in 1968, secreatul codului Navajo a fost dezvaluit iar in anul urmator, vorbitorii de cod si-au organizat prima lor reuniune. Apoi, in 1982, au fost decorati si guvernul American a proclamat ziua de 14 August "Ziua nationala a vorbitorilor de cod Navajo". Cea mai mare recompensa pe care au primit-o a fost probabil spusele generalului Sizo Arisue, seful spionajului japonez care a recunoscut ca desi sparsesera codul Fortelor Aeriene Americane, n-au reusit nici macar sa se apropie de codul Navajo.

Alph implementeaza codul NAVAJO, incepand cu versiunea 0.18, utilizand transmisia litera cu litera si omitand cartea de coduri pentru cuvinte speciale deoarece aceasta intervine doar cand este nevoie de cuvinte speciale specifice actiunilor militare. Se utilizeaza separatorul original pentru cuvintele specifice fiecarei litere si se face distinctie intre literele mari si mici cat si semnele de punctuatie. Decriptarea se face simetric si transparent asa cum este regula in Alph. De asemenei, codul NAVAJO este prezent in libraria lui Alph si da programatorului posibilitatea de a utiliza codarea in Navajo utilizand functia specifica acestuia.

Codurile pot fi folosite cu mai multe scopuri. Ele au avut la redescoperirea lor, rolul tehnic de a simplifica intelesul unui mesaj pentru ca acesta sa fie transmisibil in cel mai scurt timp posibil. Un mesaj complex continand o multime de caractere bine plasate, poate fi transformat intr-un numar simplu de impulsuri (on/off in cazul codului Morse) care nu ingreuneaza transmisia de semnale. La fel, acest lucru este valabil si pentru codarea in binar care si ea este o forma de a reprezenta literele recunoscute uman intr-un sir de "impulsuri" usor de prelucrat si de transmis. Deseori codarea se imbina cu o cifrare pentru a compune un mesaj ascuns si pentru a-l transmite. Pe de alta parte, trebuie sa tinem cont ca notinea de cod a fost doar dezvoltata de-a lungul secolelor si ca ea exista si a aparut de foarta multa vreme

probabil, ducandu-ne inapoi in timp pana la scrierile antice. In cazul unora, de exemplu semagramelor, nu exista o transformare asupra alfabetului ci asupra intelesului si se pot lua in seama, conform definitiei codurilor, ca fiind coduri. Pe de alta parte, in cazul scrierilor egiptene, avem mai degraba o cifrare decat un cod. Fiecare litera are cel putin un singur simbol asociat. Mai mult decat atat, scrierile vechi egiptene sunt homofone si permit asocierea unui sunet cu mai multe simboluri.

4. 5 Criptografia Mecanica

In 1894, Guglielmo Marconi a inceput sa faca experimente legate de o proprietate a circuitelor electrice. In anumite conditii, daca printr-un circuit trece un curent electric, acesta poate induce un curent intr-un alt circuit izolat, aflat la distanta de primul. Printr-o proiectare mai buna a circuitelor, sporind puterea si adaugand antene, Marconi a resuit in scurt timp sa transmita si sa receptioneze pulsuri de informatii la distante de pana la 2,5 km. Inventase radio-ul. Telegraful era deja creat de o jumatate de secol, insa avea nevoie de un cablu care sa transporte mesajul intre expeditor si destinatar. Sistemul lui Marconi avea marele avantaj ca se dispensa de cabluri - semnalul calatorea prin aer. In 1896, cautand sprijin financiar pentru ideea sa, Marconi a emigrat in Marea Britanie, unde si-a inregistgrat primul brevet. Primul sau mare succes a fost in 1899 cand a echipat doua vase cu statii radio, astfel incat ziaristii care comentau Cupa Americii, cea mai mare cursa de iahturi din lume, sa poata transmite stiri la distanta pentru ziarele de a doua zi.

Acesta descoperire a pus pe jar cadrele militare care o priveau cu speranta si totodata cu frica. Era clar ca metodele clasice de ascunderea mesajelor devenisera inutile si atunci trebuiau inventate noi cifruri care sa permita unei comunicatii transmise la distanta sa fie sigura. Creatorii de coduri si cifruri au inventat cateva cifruri noi care au fost insa sparte unele dupa altele.

Unul dintre cele mai famioase cifruri de razboi a fost cifrul german ADFGVX, introdus pe 5 martie 1918, chiar inaintea marii ofensive germane care a inceput la 21 martie. Ca in orice atac, trupele germane vroiau sa profite de elementul de surpriza, iar un comitet de criptografi a ales cifrul ADFGVX dintr-o multime de candidati, crezand ca ofera cea mai mare siguranta. Erau de fapt convinsi ca este indescifrabil. La inceputul lui iunie 1918,

artileria germana se afla numai la 100 km de Paris si se pregatea pentru asaltul final. Singura speranta pentru aliati era sa sparga cifrul ADFGVX pentru a afla exact unde planuiau germanii sa strapunga liniile de aparare. In cele din urma, in noaptea de 2 inuie, un criptolog pe nume Georges Painvin, a dezlegat un mesaj ADFGVX. Pierzand elementul de surpiza, armata germana a fost infranta, dupa aproape cinci zile de macel pe campul de lupta.

Particularitatea acestui cifru este ca acesta elimina toate literele din mesajul original si le substituie cu o serie de alte litere: a, d, f, g, v sau x. De aici si numele cifrului. Rezistenta cifrului la atacuri este destul de puternica deoarece la o prima vedere textul cifrat pare ca o insirare de litere repetitive. Este greu sa aplicam o analiza de frecventa sau orice alta metoda deoarece nu se poate identifica precis care simbol se repeta in sirul cifrat de cele mai multe ori. Secventa de litere pare astfel complet aleatoare si fara algoritmul precis pentru decriptare nu putem a ne dam seama ce se ascunde sub aceste litere. ADFGVX foloseste si el un patrat Polybus insa nu necesita o cheie specifica pentru a-l genera. El functioneaza pe un principiu asemanator unui hash, constand in ce am numit ca "map" pentru a transforma un domeniu de valori intr-altul.

Alph implementeaza pe deplin acest cifru dar nu respecta semnele de punctuatie datorita naturii cifrului. ADFGVX este un cifru care "lungeste" mesajul; textul cifrat se mareste aproximativ de trei ori fata de mesajul original. Nu ar avea sens sa introducem semne de punctuatie sau spatieri deoarece nu putem delimita in clar unde incepe un cuvant si unde se termina in mesajul cifrat. Aceasta este si metoda originala de transformare pentru o cifrare ADFGVX.

Cea mai veche masina criptografica este discul de cifrare (Fig. 9), inventat in secolul al XV-lea de arhitectul italian Leon Alberti, unul dintre parintii cifrului polialfabetic. El a luat doua discuri din cupru, unul putin mai mare decat celalalt si a inscriptionat alfabetul pe marginea fiecaruia. Asezand discul mai mic deasupra celui mai mare si fixandu-le cu un ac servind drept ax, a construit o simpla masina de cifrare polialfabetica aceasta fiind cunoscuta astazi sub numele de masina Alberti. Cele doua discuri puteau fi miscate independent si alfabetele puteau avea pozitii relative pentru cifrarea unui text.

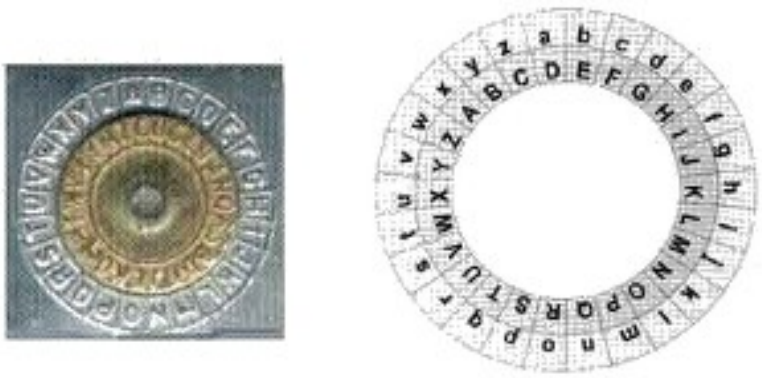

Fig. 9 Imagine a discurilor Alberti

Este cea mai veche forma cunoscuta de cifrare mecanica si implementeaza un algoritm pe care Alph il suporta. Procedura sau siguranta acestui cifru nu este prea diferita de alte algoritme polialfabetice despre care am mai discutat. Intr-un fel, ea este o simpla substitutie VIGENERE utilizand o cheie pentru a selecta alfabetele. Ceea ce este remarcabil insa, este ca masinaria lui Leon Alberti este portabila si nu necesita prea mari cunostinte pentru a fi folosita fata de predecesorii ei. Ea este portabila si poate sa fie dusa oriunde... fie pe un camp de lupta, fie distribuita unor operatori de comunicatii.

Un dispozitiv similar, bazandu-se pe aceasta masinarie, este un disc folosit in razboiul civil american. Discul este aproape identic si foloseste o decalare numerica pentru cheie. O decalare cu unu inseamna ca discul interior se va roti cu o pozitie construind un nou alfabet de cifrare atunci cand se iau literele pornind de la discul mare, pe discul mic. Singura diferenta este ca alfabetul de cifrare nu se schimba in timpul procedurii. Astfel, discul acesta este o masina care implementeaza cifrul CAEZAR permitand pana la 26 de pozitii sau alfabete pentru cifrare. El a fost inmanat tuturor persoanelor destinate sa primeasca sau sa trimeata un mesaj cifrat stabilindu-se dinainte pozitia pe care se va face comunicarea. Alph implementeaza acest cifru sub numele de USASS (literele care sunt inscriptionate pe suprafata discului interior) respecand semnele de punctuatie si luand ca parametru pozitia pe care se cifreaza sau descifreaza.

4. 5. 1 Enigma

Cu aproape jumatate de secol inainte, in 1918, inventatorul german Arthur Scherbius si prietenul sau Richard Ritter au infiintat o companie Scherbius & Ritter, o intreprindere inovatoare care producea de toate, de la turbine la perne electrice. Scerbius se ocupa de cercetare si dezvoltare. Unul dintre proiectele la care tinea era sa elimine sistemele inadecvate de criptografie folosite in primul razboi mondial, inlocuind cifrurile create cu creionul si hartia cu o forma de criptare care exploata tehnologia secolului XX. Dupa ce a studiat electrotehnica la Hanovra si Munchen, a construit o masina criptografica, de fapt o versiune electrica a discului de cifrare al lui Alberti numita Enigma. Inventia lui Scherbius avea sa devina cel mai de temut sistem de criptare din istorie.

Enigma, a fost utilizata in mediul comercial in anii 1920 si a fost adoptata de serviciile militare si guvernamentale ale unor natiuni - cea mai notabila fiind Germania inainte si dupa al doilea razboi mondial. Modelul german militar, Wehrmacht Enigma, este versiunea cea mai discutata. Masina a capatat o mare popularitate dat fiind faptul ca aliatii au fost capabili

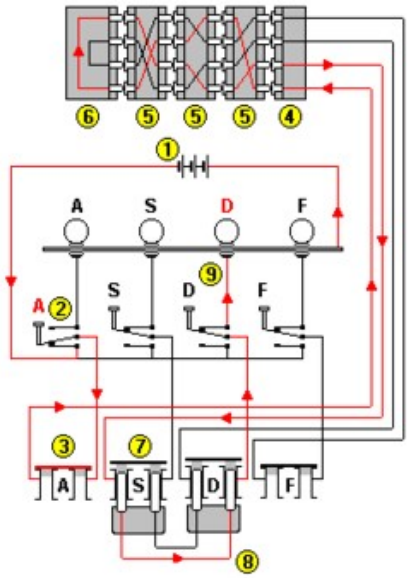

Fig. 10 Diagrama a unei masini Enigma. Cand tasta "A" este apasata, semnalul este trimis catre lampa "D", la fel, tasta "D" rezulta in "A" dar "A" nu rezulta niciodata in "A".

sa decripteze un numar mare de mesaje care fusesera cifrate utilizand masinaria. Experienta castigata prin aceasta sursa, numita ULTRA, a fost de un ajutor semnificativ pentru eforturile de razboi aliate. Influenta exacta a lui ULTRA este in dezbatere, insa se stie clar ca razboiul european a fost scurtat doi ani datorita descifrarii cifrului Enigma german.

Desi cifrul Enimga are cateva slabiciuni criptografice, in practica, numai combinatia cu alti factori semnificativi a permis spargatorilor de coduri sa descifreze mesaje: greseli ale operatorilor, defecte in procedura si capturarea ocazionala al unei masinarii sau carte de coduri.

Ca alte masinarii, Enimga este o masina care are o combinatie de sisteme mecanice si electrice. Mecanismul mecanic consta intr-o tastatura; un set de discuri rotitoare numite rotoare aranjate adiacent in jurul unui ax si un mecanism de pas pentru a invarti unu sau mai multe rotoare la fiecare apasare pe tastatura. Mecanismul exact variaza, insa cea mai comuna metoda este ca rotorul din dreapta sa paseasca o data cu fiecare apasare pe tastatura, si ocazional pasirea unui rotor invecinat. Miscarea continua a rotoarelor rezulta in diferite transformari criptografice dupa fiecare apasare.

Partile mecanice actioneaza astfel incat sa formeze un circuit electric variat - cifrarea este insa facuta electric. Atunci cand o tasta este apasata, circuitul este completat; curentul curge prin cateva componente si aprinde o lumina pe o anumita lampa, indicand litera rezultanta. Operatorul ar apasa pe tasta "A" si s-ar aprinde litera "Z"; "Z" ar fi atunci prima litera al mesajului cifrat. Operatorul ar face la fel pentru fiecare litera din mesajul in clar.

Pentru a explica modul de functionare al lui Enimga, utilizam diagrama din Fig. 10. Pentru a simpifica exemplul, am utilizat doar patru componente din fiecare. In realitate sunt 26 de lampi, chei, stechere si fire in rotoare. Curentul curge de la baterie (1) prin intrerupatorul bi-directional (2) la stecherele (3). Stechele permiteau "map"-area si rearanjarea conexiunilor dintre tastatura (2) si roata fixata de intrare (4). Apoi, curentul merge prin stecherul, in cazul acesta, neutilizat (3) prin roata de intrare (4) si prin trei sau patru rotoare notate (5). Apoi reflectorul (6) intoarce curentul, prin alta directie, inapoi prin rotoare (5) si roata de intrare (4), apoi merge prin stecherul 'S' conectat la cablul (8) pana in stecherul 'D', si alt intrerupator bi-directional (9) ca sa aprinda lampa.

Rotoarele (roti sau tambure - Walzen in germana) formeaza inima masinariei Enimga si sunt cele care implementeaza cifrele de substitutie. Aproximativ 10 cm in diametru, fiecare rotor este un disc facut dintr-un cauciuc avand o serie de arcuri de alama pe o fata aranjate intr-un cerc; pe cealalta parte au o serie de contacte electrice corespondente. Pinii si contactele reprezinta un alfabet (tipic alfabetul de 26 de litere A-Z). Cand sunt plasate alaturat, pinii unui rotor stau pe contactele celuilalt formand astfel o conexiune electrica. In tr-un rotor, un set de 26 de fire conecteaza fiecare pin pe o parte la alt contact iar in cealalta parte, intr-un tipar complex. Modul de conectare difera pentru fiecare rotor.

De sine statator, un rotor formeaza numai o substitutie simpla. De exemplu pinul corespunzator literei "E" ar putea sa fie conectat la litera "T" pe partea opusa. Complexitatea vine, de fapt, de la insirarea rotoarele - de obicei trei sau patru - si miscarea continua a acestora. Cand ele sunt plasate in masina, un rotor poate sa aibe una din cele 26 de pozitii posibile. Ele pot fi rotite de mana, utilizand un mic sunt, de catre operator. Astfel, fiecare rotor are o litera care defineste un alfabet si care este vizibila printr-un mic ecran. Modul in care acestea sunt setate, din cele 3 x 26 pozitii, in cazul a trei rotoare, definesc setarile de "ring". In prototipurile de inceput, modelele Enigma aveau aceste setari fixe; o complicatie introdusa in verisuni ulterioare este posibilitatea de a ajusta ring-urile relativ la conexiunile

din rotor. Pozitia ring-ului sau setarile ring-ului sunt cunoscute sub numele de "Ringstellung" (setare de ring-uri).

Fiecare rotor contin un stutz (cateodata mai multe), utilizate pentru a controla pasul rotorilor. In versiunile militare, stutz-urile sunt plasate pe ring-ul de alfabet.

Atunci cand un curent intra printr-un rotor, el era trimes pe o cale "diferita" spre al doilea rotor. Acesta facea acelasi lucru si trimetea semnalul pe alta cale spre al treilea care repeta. Dupa aceea semnalul atingea reflectorul care trimetea inapoi curentul prin cele trei rotoare care iarasi deflectau semnalul pe cai diferite asa cum este ilustrat in imaginea de mai jos (Fig. 11).

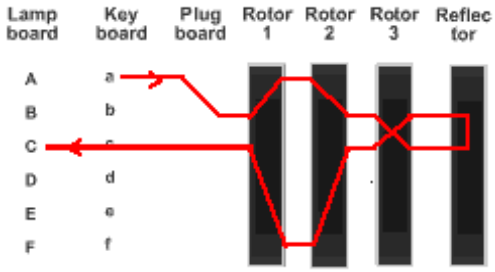

Fig. 11 Exemplu de functionare a masinii Enigma

Numarul rotoarelor diferea. Enigmele utilizate pentru armata si fortele aeriene veneau echipate cu mai multe rotoare; atunci cand Enigma a aparut erau in total trei. Pe 15 decembrie 1938, acest numar a fost schimbat la cinci, din care trei erau alese pentru a fi folosite pentru cifrare. Acestea erau numerotate cu numerale romane pentru a le distinge. Forta navala germana a avut la dispozitie mai multe rotoare decat celalalte servicii: la inceput cinci, dupa care sapte si in final opt rotoare. Enimgele cu patru rotoare permiteau ca un rotor aditional sa fie adaugat pe langa cele trei. Acest lucru era facut prin inlocuirea reflectorului original cu un reflector mai subtire si adaugand un rotor mai special, al patrulea. Al patrulea rotor era de doua tipuri Beta sau Gama. Acest rotor nu pasea, insa era posibil sa fie plasat in oricare din cele 26 de pozitii valide.

Pentru a nu implementa un cifru prin subsitutrie simpla, unele roatoare se invart la apasarea unei chei. Acest lucru asigura ca transformarea criptografica este diferita la fiecare

pozitie producand un cifru prin substitutie polialfabetic complex. Primul rotor se schimba o data cu fiecare apasare, al doilea rotor avanseaza cu o pozitie dupa cele 26 de pozitii ale primului rotor iar al treilea avanseaza si el o data dupa cele 26 de pozitii ale celui de al doilea rotor. Al doilea rotor insa, avanseaza in acelasi timp cu al treilea rotor, insemnand ca al doilea rotor poate sa paseasca de doua ori o data cu fiecare apasare de tasta rezultand intr-o perioada redusa.

Aceasta pasire dubla face rotoarele sa devieze de la un odometru normal. O pasire dubla se produce astfel: primul rotor paseste si ia al doilea rotor un pas inainte dupa el. Daca al doilea rotor s-a miscat prin acest pas in pozitia in care trebuie sa roteasca dublu, la urmatoarea mutare el impinge al treilea rotor si il misca inainte dar se va misca si el astfel pasind de doua ori.

Cu trei rotoare, masina are o perioada de 26*26*26 = 17576. Istoric, mesajele erau limitate doar la cateva sute de litere si nu exista riscul de a repeta o pozitie intr-un singur mesaj.

Cu exceptia catorva prototipuri de Engima, ultimul rotor era urmarit de un reflector (Umkehrwalze), si aceasta este o trasatura patentata si distinsa printre alte masinarii inventate in acea perioada. Reflectorul conecteaza iesirile ultimului rotor in perechi, redirectionand curentul inapoi prin rotoare pe o cale diferita. Reflectorul asigura ca Enigma sa fie reciproca astfel facand ca o cifrare sa fie la fel ca o descifrare. Mai mult de atat, rotorul face ca nici o litera nu poate sa fie cifrata in ea insesi, adica fiecare litera din alfabet va avea complet alt corespondent in textul cifrat. Desi aparent o trasatura benefica, acest lucru era o greaseala criptografica si a fost utilizata pentru a sparge cifrul ENIGMA. In modelul comercial, reflectorul poate sa fie inserat in doua pozitii diferite. In Enigma pentru Abwehr, reflectorul pasea in timpul cifrarii similar celorlalte rotoare. Pentru Wehrmacht, reflectorul este fixat si nu se rotea si a aparut in patru versiuni.

Stecherul (Steckerbrett) este o configurare variabila care poate a fie configurata de catre operator si vizibila pe panoul Enigmei. A fost introdus de catre armata germana in 1930 si a fost curand preluat de catre forta navala. Stecherul contribuie foarte mult la puterea cifrarii mai mult decat ar contribui un rotor in plus. Un cablu plasat pe stecher conecteaza

litere in perechi, de exemplu, E si Q ar putea sa fie o perece. Efectul este de a schimba alfabetul de intrare si de iesire printr-o substituire in plus.

In concluzie, transformarea (E) pentru o cifrare se poate descrie prin:

$$E=PRMLUL^{-1}M^{-1}R^{-1}P^{-1}$$

unde P este stecherul, R este rotorul din dreapta, M este rotorul din mijloc, L este rotorul din stanga si U este reflectorul.

Comunicatiile militare germane au fost impartite intr-un numar de retele diferite, fiecare utilizand diferite setari pentru masinile Enigma. Aceste retele erau numite chei la Bletchley Park si au avut nume de cod precum Red, Chaffinch sau Shark. Fiecare unitate care opera pe o retea ii era atribuita o lista de setari specificand Enigma pentru o perioada de timp. Pentru ca un mesaj sa fie cifrat corect si decifrat, atat emitaroul cat si receptorul trebuiau sa-si seteze masina Enimga in acelasi mod; rotoarele, selectia lor, setarile de ring si de stecher trebuiau sa fie identice. Pentru acest lucru s-a alcatuit si distribuit o carte de coduri.

In forma initiala, Enigma avea patru chei principale [6]:

- Pozitia rotoarelor (Walzenlage) si ordinea in care sunt utilizate.
- Pozitia initiale ale rotoarele - aleasa de catre operator, diferit pentru fiecare mesaj.
- Setarile de ring (Ringstellung) - pozitiile de inceput al fiecarui rotor din punct de vedere al alfabetului
- Setarile de stecher (Steckerverbindungen) - conexiunile dintre setarile de stecher si tastatura.

Enigma a fost conceputa sa fie sigura chiar daca se stiau setarile dintr-un rotor. Cu o configuratie secreta, numarul total de configuratii calculate e in jur de 10^{114} (aproximativ 380 de biti); cu o configurare cunoscuta si cateva constrangeri, acest numar este redus la 10^{23} (76 biti). Utilizatorii lui Enimga erau asigurati de siguranta ei prin numarul mare de posibilitati; era de neconceput un atac prin forta bruta.

Alph implementeaza cifrul ENIGMA utilizand trei rotoare, un reflector si setarile de ring, toate la alegere. Cheia este complexa si programul indeamna utilizatorul sa citeasca manualul in legatura cu setarile corecte pentru metoda de cifrare. Manualul care vine cu Alph indica rotoarele posibile si reflectoarele disponibile [6]. Toate acestea au fost luate din surse relativ sigure de la centrul de criptografie din Georgia U.S.A. Aditional, incepand cu versiunea 0.17, Alph implementeaza o metoda de a include o substitutie de stecher disponibila prin parametrul --plug=<A-Z>. Cifrarea si decifrarea se face transparent, asa cum este cazul masineriei originale Engima, setarile acesteia trebuind sa fie identice in ambele cazuri.

Datorita complexitatii sale si rezistentei ei la orice forma de atac criptologic, masinaria Engima a devenit punctul cheie al unor scriitori de fictiune. S-au scris carti si s-au facut chiar filme ("Engima" -2001, "All the Queen's Men"-2002) care au toate ca subiect furtul unei Engima din dotarea armatei germane. Alitatii nu au reusit sa decripteze cifrul complex al fortelor germane si in concluzie au incercat sa fure o masinarie ca sa alfe cum functioneaza. Bineinteles si aceasta tentativa a fost inutila deoarece nu puteau sa afle decat cateva din multele rotoare si oricum setarile de ring erau diferite de fiecare data - nemaivorbind de setarile de stecher sau secventa rotoarelor.

Inspirati de aceasta masinarie, desi sursele oficiale nu sunt clare, japonezii vin cu o noua inventie care operaza asemanator cu Engima. Aceasta va fi cunoscuta sub numele de cod, PURPLE.

4. 5. 2 Purple

Pe 6 decembrie 1941, Guvernul Japonez trimete un mesaj, care a fost impartit pentru transmisie in 14 pati (mesajul de 14-parti), la ambasada din Washington. Un mesaj de acompaniament sugereaza ca mesajul de 14-parti sa fie livrat in Statele Unite pe Decembrie 7 la ora 13, ceea ce corespunde la 7:30 la Pearl Harbor, Hawaii. Mesajul care a rupt negocierile intre Japonia si Statele Unite a fost cifrat utilizand masina PURPLE.

97-shiki-obun In-ji-ki (Masina de Cifrare 97), numita si Angooki Taipu B de catre japonezi era cunoscuta de catre forurile americane sub numele de cod PURPLE. In functie de cheia utilizata de catre mesajele cifrate, PURPLE a primit subdesignatii gen JAA-1 si JAA-2.

Masina PURPLE este o masina prin substitutie care inlocuieste fiecate litera din textul in clar in text cifrat (sau vice versa) utilizand un alfabet permutat sau de cifrare. Alfabetul de permutare se schimba dupa ce fiecare litera este cifrata sau decifrata producand un sistem de substitutie poliafabetica. Intrarea si iesirea la masina PURPLE consta in doua masini clasice de scris electrice. Mesajele care trebuie sa fie cifrate sau decifrate pe o masinarie PURPLE au fost introduse pe tastatura masinii de scris si textul rezultant se tiparea pe masina de scris de la iesire.

Cel mai important element criptografic al masinariei PURPLE este un intrerupator cu pas de 25 de pozitii (cunoscut si ca un releu cu pas sau un uniselector). Acest dispozitiv conecteaza intrarea unui terminal la una dintre cele 25 de iesiri. Un electromagnet, atasat la intrerupator, avanseaza intrerupatorul la urmatoarea pozitie de exemplu de la iesirea 1 la iesirea 2, raspunzand la un puls electric. Cand magnetul este conectat la circuitul electric al masinariei de scris, intrerupatorul cu pas avanseaza la urmatoarea pozitie consecutiva de fiecare data cand o cheie este apasata pe masina de scris. Dupa ce ajunge in pozitia 25, urmatorul puls va intoarce intrerupatorul la pozitia 1.

Interesant este faptul ca acest intrerupator a foast dezvolatat in Statele Unite, intr-un patent (No. 447,918) acordat lui Almon B. Strongwer pe 10 martie 1891, datorita nevoii de a automatiza reteaua de telefoane. Acest dispozitiv, cu o intrarea conectata la un telefon, selecta unul dintre cele 10 nivele de intrerupatoare si dupa aia unul dintre cele zece contacte pe acel nivel. Acest intrerupator putea astfel sa redirijeze un apel telefonic la una dintre cele100 de posibile destinatii. In timp, aceste intrerupatoare au fost imbunatatite si permiteau mai multe nivele si mai multe contacte.

Armata americana, mai precis SIS (Signal Inteligence Service), nu au resusit niciodata sa obtina o masinarie PURPLE ci au incercat sa o reproduca utilizand dispozitive proprii pentru a o analiza. Chiar si dupa razboiul al doilea mondiat, numai cateva parti ale masinariei au fost recuperate.

Masinaria PURPLE [7] imparte literele alfabetului (A-Z) intr-un grup de sase si alt grup de douazeci de caractere. Aceste doua grupe sunt designate grupul de "sase" (sixes) si

grupul de "douazezi" (twenties) reprezentand, respectiv, vocalele (AEIOUY) si consoanele (BCDFGHJKLMNPQRSTVWXZ). O diagrama amanuntita este prezentata mai jos (Fig. 12).

Masinaria PURPLE, introduce fiecare litera intr-un tablou de stechere (input plugboard). Daca litera introdusa a fost o vocala atunci aceasta intra in intrerupatorul de sase (sixes switch), urmand o permutare a ei, si iese la masina de scris (output type writer) trecand din nou prin acelasi panou de stechere (output plugboard). Ambele panouri sunt de fapt identice insa pentru claritate acesta a fost impartit in doua. Daca litera introdusa a fost o consoana, ea iese prin panoul de stechere

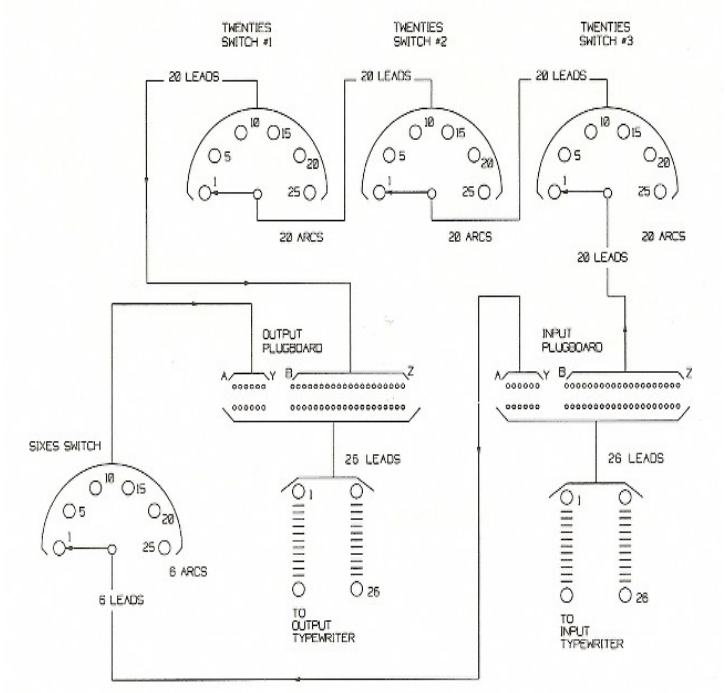

Fig. 12 Imagine schematica a modului de lucru al masinii PURPLE.

consoana, ea iese prin panoul de stechere si intra succesiv in cele trei intrerupatoare urmand sa faca trei permutari (twenties #3, #2 si #1) ca dupa aceea sa iasa la masina de scris prin panoul de stechere (output plugboard).

Cele trei intrerupatoare ale douazecilor (twenties switch #2, #2 si #1) au de fapt 20 de nivele. Intrerupatorul saselor (sixes switch) a fost conceput sa produca 25 de alfabete

necorelate. Iesirile douazecilor sunt legata impreuna (asa cum era cazul la masinaria ENIGMA) si impreuna ajung sa faca o permutare de 25*25*25 adica 15,625 de posibilitati.

Fiecare intrerupator al douazecilor era construit din patru intrerupatoare a cate sase nivele fiindca nu erau disponibile intrerupatoare de 25 de pozitii. Toate cele patru sectii se miscau ca una. Intrerupatorul saselor includea o sectiune de sase nivele care se misca in paralel. A doua sectie a saselor controla miscarea douazecilor si lampile care indicau pozitiile intrerupatoarelor.

Este important de remarcat ca in timpul cifrarii sau decifrarii orice litera a saselor era inlocuita numai prin ea insesi sau alta litera a saselor. Similar, literele douazecilor erau substituite numai intre ele. Acest lucru a fost bineinteles o mare greseala si a fost principalul motiv pentru care PURPLE [7] a cedat in mainile criptologilor.

Configuratia originala japoneza a tabloului de stechere este necunoscuta fiindca nu a fost recuperata. In figura de mai sus, masina de scris de la intrare este conectata la intrerupatorul #3 pe cand masina de scris de la iesire este conectata la intrerupatorul numarul #1. Acesta configuratie este corecta pentru modul de decifrare al masinariei PURPLE. Intrarea si iesirea trebuiau sa fie schimbate intre ele atunci cand se cifra un mesaj. Notam ca acest lucru este diferit de masinaria ENIGMA care implementa o cifrare transparenta si satisfacea conditia de reciprocitate. In cazul masinariei ENIGMA orice mesaj cifrat cu o configuratie, era decifrat cu aceeasi configuatie. In cazul masinariei PURPLE insa, secventa trebuia sa fie inversata pentru decifrare sau cifrare. Era insa posibil de cifrat in mod "invers" (de exemplu cifrare in mod decifrare si vice-versa).

Ceea ce nu se arata in figura de mai sus este cablarea intrerupatoarelor, care necesita aproape 2000 de conexiuni cablate. Intrerupatorul saselor are sase nivele cu 25 de contacte de iesire pe nivel. Cu sase nivele si 25 de contacte e nevoie in total de 150 de fire.

Figura de mai jos (Fig. 13) arata cablarea pe nivelul 5 al intrerupatorul saselor si da o imagine a complexitatii cablarii. Intrerupatorul douazecilor are 20 de intrari a cate 25 de pozitii, ceea ce implica 500 de fire pentru fiecare dintre cele trei intrerupatoare.

Pentru a intelege operarea masinariei PURPLE consideram circuitul prezentat in Fig. 13. Cand o tasta pe masina de scris de la intrare este apasata, de exemplu "K", un current electric este aplicat la intrarea "K" a tabloului de stechere. Depinzand de substitutia panoului

de stechere, "K" ar putea sa fie schimbat intr-un sase, adica o vocala cum ar fi "U". Fiindca "U" este a cincea vocala, curentul din masina de scris va fi aplicat la nivelul 5 al saselor.

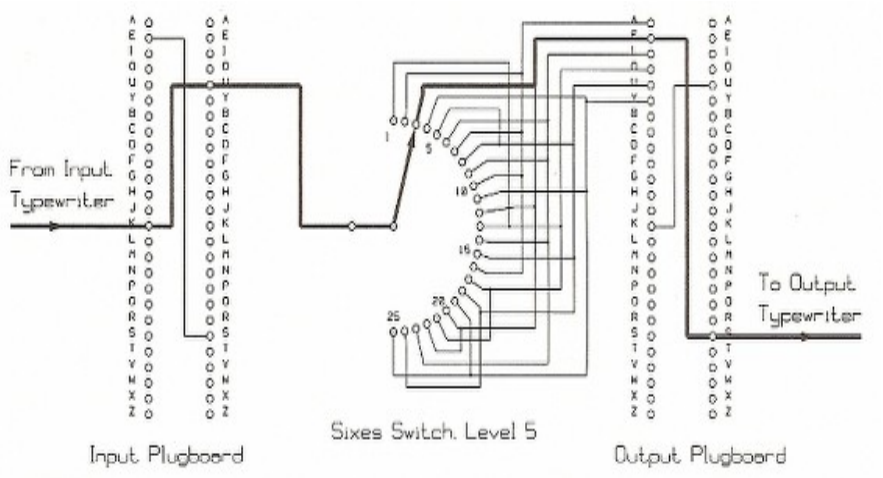

Fig. 13 Exemplu de cablare pentru circuitele masinii PURPLE

Daca intrerupatorul saselor este in pozitia 3 atunci, dat fiind cablarea interna al intrerupatorul saselor, el va face contact cu iesirea 2 ceea ce rezulta in "E". Urmatorul panou de stechere conecteaza "E" la alta litera, poate "S" si in final curentul produce un "S" la masina de scris de la iesire.

Aici am facut special ca prin substitutia stecherului sa intram in intrerupatorul saselor. Daca insa tabloul de stechere produce o consoana, aceasta trebuie sa intre in intrerupatorul douazecilor. Curentul ar trebui sa parcurga intrerupatoarele #3, #2 si apoi #1 inainte sa intre in tabloul de stechere de la iesire.

Dupa ce fiecare litera este cifrata sau decifrata, intrerupatorul saselor si unul dintre intrerupatoarele douazecilor o sa se miste la urmatoarea pozitie si o sa creeze o noua permutare. Rezultatul este un sistem polialfabetic avand o perioada de 25 pentru sase litere si 15,625 pentru urmatoarele douazeci.

Drumul prin masinaria PURPLE este astfel dependent de permutarea intrerupatoarelor, miscarea lor si alfabetul din substitutia tabloului de stechere.

Miscarea intrerupatorului saselor este simpla avansand o pozitie dupa ce fiecare litera este cifrata. Cand intrerupatorul saselor atinge pozitia 25, urmatoarea litera cifrata il va reinitializa la pozitia 1. Miscarea saselor insa determina miscarea douazecilor.

Pentru douazeci, un intrerupator se misca dupa ce fiecare litera este cifrata. Un intrerupator "rapid" avanseaza o pozitie dupa ce fiecare litera este cifrata, doar daca intrerupatorul saselor nu a atins pozitia 25. In acest ultima caz, intrerupatorul "mijlociu" avanseaza o pozitie si dupa aia intrerupatorul "incet" avanseaza o pozitie cand sasele atinge pozitia 24 urmat (la urmatoarea cifrare) de rotorul mijlociu. Mai jos luam un exemplu extrem conform pozitiilor celor patru intrerupatoare si ilustram pozitiile:

Pozitiile Intrerupatoarelor

Sasele	Douazeci #1	Douazeci #2	Douazeci #3
21	1	25	5
22	2	25	5
23	3	25	5
24	4	25	5
25	4	25	6
1	4	1	6
2	5	1	6

Impartirea alfabetului in vocale si consoane era preluata de la predecesorul lui PURPLE, Angooki Taipu A sau masinaria RED. Cand a fost prima data introdusa, masinaria RED cifra vocalele (A, E, I, O, U si Y) ca vocale si consoanele prin consoane. Textul cifrat produs era astfel pronuntabil sub regulile telegrafice la timul acela, dar asta a fost important fiindca telegramele continand text aleator erau mai scumpe de trimis decat o telegrama in limbaj "pronuntabil". De exemplu, Boris Hagelin a patentat (U. S. Patent 1,484,477) o masinarie (CRYPTOCODE-TYPER, Modelul A 4), care putea sa converteasca grupe de coduri numerice (de exemplu "343859834") in grupe de litere "pronuntabile" (cum ar fi "olavuxutip"). Japonezii sigur au cunoscut acest dispozitiv inainte sa creeze PURPLE fiindca au cumparat alte dispozitive criptografice de la Hagelin.

Masinile RED si PURPLE erau complet diferite in operarea lor criptografica. RED utiliza un rotor injumatatit (Damm half-rotor) cu 26 de contacte prin inelele de pe o axa. Aceste contacte erau impartite in doua grupe de 20 si respective 6 contacte. Contactele erau conectate la un rotor de 60 de pozitii (cel mai mic multiplu comun al lui 20 si 6) la capatul axului. O roata contrala miscarea rotoarelor, avansand una, doua sau trei pozitii pentru fiecare litera cifrata cu o perioada variabila de 41 sau 43 de pozitii.

Fiindca principiile de operare a lui RED si PURPLE erau complet diferite este dificil de speculat dece a fost pastrata o divizie a literelor din alfabet. Cea mai simpla explicatie pare a fi faptul ca proiectantul sau proiectantii nu au considerat aceasta diviziune o slabiciune semnificativa. In special cand vocalele sunt cifrate ca vocale, asa cum s-a intamplat cand RED a fost introdus, efectul este remarcabil. Distributia vocalelor in Japoneza Romanizata include multe bigrame, cum ar fi OO, UU, AI si IA, sau chiar trigrame cum ar fi YUU sau YOO. Aceste combinatii sunt foarte slabe atunci cand ele nu sunt substituite cumva si cedeaza la cea mai simpla incercare de analiza criptografica.

Japonezii au abandonat diviziunea intre vocale si consonane dupa cateva luni dupa ce au introdus masinaria RED si dupa aceea oricare dintre cele 26 de litere putea sa intre via tabloul de stechere in intrerupatorul saselor. Pare logic de presupus ca japonezii si-au dat seama ca aceasta corespondenta vocala-vocala reduce considerabil puterea cifrului. Din pacate pentru ei, criptologii americani obtinusera deja mesaje si le utilizasera pentru a descrie principiile de operare ale masinariei. Schimbarea metodei de cheie a insemnat pentru ei doar o mica schimbare de abordare. Trebuiau sa distinga analiza asupra unei vocale sau a unei consoane. RED a fost utlizat foarte mult de catre britanici, germani si chiar rusi pana cand in 1941 a fost abandonat fiindca Statele Unite au reusit sa sparga cifrul.

Similar cu masinaria ENIGMA, PURPLE foloseste un sistem de chei care determina cifrarea sau decifrarea unui mesaj. Se utilizeaza in principiu 5 chei care determina cifrarea unui mesaj: pozitia initiala a saselor, pozitiile douazecilor pentru intrerupatoarele 1-3 si secventa de miscare ale intrerupatoarelor (care poate sa fie in orice ordine 123, 321, 231 etc...). La decifrare se vor inversa intrerupatoarele douazecilor pastrandu-se secventa de miscare pentru a obtine mesajul in clar.

Masinaria PURPLE insa, fata de ENGIMA, prezenta o serie de avantaje pe care aceasta din urma nu putea sa le satisfaca. In primul rand, si o observatie destul de importanta,

este faptul ca masinaria PURPLE cifra direct un mesaj. O data selectate pozitiile intrerupatoarelor si secventa de miscare, un operator trebuia doar sa bata la masina mesajul care urma sa fie cifrat. Mesajul cifrat aparea direct pe o hartie din celalata masina de scris cuplata la iesire. In cazul masinariei ENIGMA, aceasta aprindea doar niste lumini care determinau litera de iesire. Ea trebuia sa fie scrisa de mana sau noata de catre operator dupa care se proceda la urmatoarea litera. Acest avantaj i-a dat cifrului PURPLE o viteza considerabila eliminand nevoia unui operator instruit. Din alt punct de vedere, ENIGMA era foarte versatila si rotoarele nu trebuiau sa fie rearanjate pentru cifrare. ENGIMA mai era si, cum am descris-o, modulara. Ea permitea reconfigurarea ei in cateva secunde, in care se puteau selecta diverse roatoare si chiar reflector pentru a cifra un mesaj. PURPLE era o masina statica. Singurele chei erau pozitiile intrerupatoarelor si miscarea lor. PURPLE nu avea intrerupatoare alternative care puteau sa fie alese.

Aceasta diferenta s-a vazut cu claritate in timpul celui de al doilea razboi mondial. In timp ce o masina PURPLE nici macar nu a fost gasita si spargerea ei s-a facut prin proceduri simple criptanalitice, ENGIMA era in centrul atentiei pe doua continente, pana cand ea a fost declarata de nedescifrat si s-au organizat misiuni pentru furtul ei.

4. 6 Cifrul ideal

Gilbert Sandford Vernam (1890-7 februarie 1960) lucand la AT&T Bell Labs ca inginer, a inventat in 1917 un cifru care avea sa se transforme mai tarziu in asa zisul "one-time pad cipher". Vernam a propus un cifru de telegrafie cu o cheie preparata in prealabil pe hartie si care se combina caracter dupa caracter cu texul mesajului in clar pentru a produce mesajul cifrat. Pentru a decifra textul cifrat, aceasi cheie ar fi folosita caracter dupa caracter rezultand in mesajul in clar.

El si-a patentat inventia (U. S. Patent 1,310,719) pe 22 iulie 1919 si reprezinta operatia XOR aplicata impulsurilor individuale sau biti utilizati pentru a coda caractere in codul Baudot pentru telegrafie. Vernam nu a utilizat termenul de "XOR" in patent, dar a implementat aceasta functie utilizand logica de relee. In expemplul pe care l-a dat Vernam, litera in clar "A" este codata ca "++--" in codul Baudot, si cheia "B" este codata ca "+--++".

Rezultatul combinarii este "-+-++", care coda un "G". Combinand acest "G" si caracterul cheii "B" la receptor se produce "++---" adica primul caracter in clar, "A". Agentia Nationala de Siguranta (NSA) a declarat acest pantent ca fiind "una dintre cele mai importante inventii in istoria criptografiei".

Putin timp dupa aceea, Joseph Mauborgne, la vremea respectiva capitan al US Army Signal Corps a propus, in aditie la VERNAM (implementat in Alph ca parametru VERNAM), ca toate caracterele din cheie sa fie aleatoare. Aceste doua idei combinate produc un cifru zis "one-time pad" desi nici-unul dintre Vernam sau Mauborgne nu l-a definit ca atare.

Claude Shannon, lucrand si el la Bell Labs, a dovedit ca un cifru "one-time pad" nu se poate sparge. Este prima si unica metoa pentru care exista aceasta dovada. Shannon a mai dovedit ca toate cifrurile care se vor dovedi a fi de nespart, ar fi variante de cifru "one-time pad".

In terminologie moderna, cifrul VERNAM, este un cifru de stiva in care textul in clar este XOR-at cu o alta stiva de caractere aleatoare sau pseudoaleatoare (generate prin procese deterministe) de aceasi lungime pentru a produce un text cifrat. Daca stiva de cheie este intr-adevar aleatoare si utilizata doar o data atunci cifrul este un "one-time pad". RC4 este un exemplu de un cifru Vernam care a fost utilizat pana in 2004 cand a fost inlocuit de un cifru imbunatatit RC5 (implementat in Alph ca parametru RC5). Acestea din urma definesc, in mod interesant, ceea ce am mai discutat: hash-urile care sunt prin definitie unidirectionale (nu se pot "sparge").

Alph implementeaza cifrul VERNAM utilizand ca parametru un fisier care contine caractere. Daca textul cheie, fisierul, nu contine destule caractere atunci Alph (libraria libalphCRYPT) intoarce utilizatorului o eroare de entropie.

5. Criptologia moderna

Toate conceptele care se gasesc in istorie privind criptografia si mai pe larg criptologia, se pastreaza si astazi si au un rol foarte important in tot ceea ce apare de la o zi la alta. De la Vigenere, care a introdus substitutia polialfabetica si pana la Vernam care a

introdus cheia infinita, toate conceptele se concretizeaza astazi sub forma unor metode distincte bazate doar pe imbunatatiri la nivel de performanta ale algoritmelor traditionale. Pe cand acum jumatate de secol un operator ENIGMA trebuia sa noteze fiecare caracter de la iesire, astazi un simplu utilizator utilizeaza programul Alph pentru a cifra blocuri mari de text. Pe cand nemtii trimeteau semnale ADFGX prin radioul lui Marconi la 1918, astazi transmisia este foarte simpla utilizand facilitatile moderne precum protocoale de email sau telefonie. Un lucru este clar insa: puterea de calcul care a crescut este doar o singura componenta si nu este mai mult decat un parametru pentru un criptolog versat. Influenta dezvoltarii puterii de calcul impinge doar vag criptologii sa creeze cifruri mai complexe si mai sigure. De acest lucru ne dam seama daca utilizam puterea bruta pentru a descifra un text cifrat utilizand un cifru mai vechi inainte de era calculatoarelor. Rezultatul este clar, forta bruta nu ne va ajuta mai deloc sa spargem ceea ce este dovedit stiintific nedescifrabil. Zis pe scurt, forta bruta, oricat de mare sau de bruta nu va reusi niciodata sa sparga niste limitari stiintifice. Chiar daca unim toate calculatoarele din lume pentru un calcul paralel, un "one-time" pad bine facut nu va ceda niciodata sub presiunea fortelor. Chiar si asa, daca luam o singura transpozitie aleatoare a unui text relativ mic:

"Horia, Universitatea Hyperion Bucuresti"

care contine doar 35 de caractere atunci exista mai mult de 5×10^{31} de combinatii posibile. Daca o persoana ar putea verifica o combinatie pe secunda, si daca toti oamenii din lume ar lucra zi si noapte, tot ar dura mai mult de o mie de ori viata universului ca sa se verifice toate combinatiile posibile. Un calculator, chiar si astazi dupa toate tehnologiile de OCR (Optimal Character Recognition) si de dictionare, este complet agramat si necesita un operator sa verfice daca a procedat corect.

Criptografia moderna se orienteaza acum spre modul in care este interpretat un caracter (cap.3 Bazele alfanumerice) si incearca sa lucreze la acest nivel. Mai nou, se utilizeaza blocuri de date in loc de caractere individuale la nivel de memorie pentru a cifra. Desigur, mai tarziu se vor gasi (unele s-au gasit deja) slabiciuni in aceasta metoda de operare. Este clar insa ca era informatiei a determinat un nou standard al criptografiei. Nu datorita tehnologiei avansate ci datorita noilor metode de comunicatie.

5. 1 Retele Feistel

‘Tatal’ cifrurilor de bloc este Horst Feistel (30 ianuarie 1915 - 14 Noiembrie 1990), nascut in Berlin, un cripgraf care a lucrat la proiectarea cifrelor la IBM, intitiind cercetarea care a culminat cu dezvoltarea lui DES (Data Encryption Standard) in 1970.

Un cifru FEISTEL este un cifru de bloc cu o structura particulara, numita dupa criptograful Horst Feistel; el mai este cunoscut si sub numele de retea Feistel. O mare proportie de cifruri de bloc utilizeaza aceasta schema, incluzand DES. O structura Feistel are avantajul ca cifrarea si decifrarea sunt foarte similare sau chiar identice in unele cazuri (ceea ce ne aminteste de ENGIMA), cerand doar o reversie a cheii. Astfel, dimensiunea codului sau circuitului necesar pentru a implementa un astfel de cifru este practic injumatatit.

Retelele Feistel si constructii similare combina mai multe "runde" de operatii repetate cum ar fi:

- *Amestecarea de biti* (numita permutari de cutii P)
- *Functii simple ne-lineare* (numite si substitutii prin cutii S)
- *Amestecul linear* (in sensul algebrei modulare) utilizand XOR.

pentru a produce o functie care contine cantitati mari de date, numite de Claude Shannon (cap. 4 sub. 6 Cifrul Ideal) "confuzie si difuzie". Amestecarea de biti creaza difuzie iar substitutia, confuzia. In criptografie confuzia se refera la a face o relatie intre cheie si textul cifrat cat de complex si adanc posibil iar difuzia este definita ca proprietatea ca redundanta in statisticele textului in clar este disipata in statisticele textului cifrat. Difuzia este asociata cu dependenta bitilor de la iesire de bitii de la intrare. Intr-un cifru cu o difuzie perfecta, doar schimbarea unui bit de la intrare ar schimba intregul text ceea ce se mai numeste si SAC (Strict Avalanche Criterion). FEISTEL utilizeaza cutiile P si amestecul linear de biti pentru a atinge o difuzie aproape perfecta si se poate spune ca indeplineste conditiile SAC.

Retele Feistel au fost introduce pentru prima data, in domeniu comercial, in cifrul LUCIFER de la IBM care a fost conceput de insusi Feistel si Don Coppersmith. Retele

Feistel au castigat respect atunci cand guvernul SUA a adoptat DES. Ca si alte componente a lui DES, exista natura iterativa al contructiei Feistel care face implementarile criptosystemului in electronica foarte usoara.

Multe cifruri de bloc simetric se bazeaza pe retele Feistel si pe structura si proprietatiile cifrului Feistel care a fost intens explorat de cate criptologi. In special Michael Luby si Charles Rackoff care au analizat cifrul si au demonstrat ca daca functia "round-robin"[8] este un generator criptografic sigur de numere pseudoaleatoare, cu Ki utilizat ca seed, atunci 3 runde sunt suficiente pentru a face cifrul de bloc sigur, pe cand 4 runde sunt suficiente sa faca cifrul "puternic" sigur, insemanad ca este sigur pentru un atac prin text cifrat ales (cap. 1 Introducere). Din cauza acestui rezultat, cifrurile Feistel au fost cateodata numita cifruri de bloc.

Modul de operare al cifrului FEISTEL este urmatorul:

1.) Imparte textul in clar in doua blocuri egale (L_0, R_0)

2.) Pentru fiecare runda i=1,2,...,n, calculeaza:

$L_i=R_{i-1}$

$R_i=L_{i-1}+f(R_{i-1},K_i)$

unde f este functia, de obicei tot XOR, si K_i este sub-cheia.

Atunci textul cifrat va fi (L_0,R_0)

3.) Repeta.

Indiferent de natura functiei f, decifrarea se face prin:

$R_{i-1}=L_i$

$L_{i-1}=R_i+f(L_i,K_i)$

Un avantaj al acestui model este ca functia utilizata nu trebuie neaparat sa fie inversibila si poate sa fie foarte complexa. Am spus ca functia f este de obicei XOR. Acest lucru nu este complet adevarat. Functia poate sa fie orice insa pentru ilustrare se foloseste o functie simpla si relativ sigura cum ar fi XOR.

---

8 Un round-robin este un aranjament prin care se aleg in ponderi egale toate elementele intr-un grup intr-o ordina rationala, deobicei de sus pana jos a unei liste si dupa aia pornind de la inceput.

Diagrama de mai sus (Fig.14) a acestui model arata cum trece textul in clar in text cifrat. Este de notat reversia sub-cheii pentru decriptare; aceasta fiind singura diferenta intre cifrare si descifrare.

Mai exista insa un tip de cifru Feistel, numit Feistel debalansat care utilizeaza o structura modificata in care L_0 si R_0 nu sunt egale in lungime. Un exemplu de asemenea cifru este SKIPJACK.

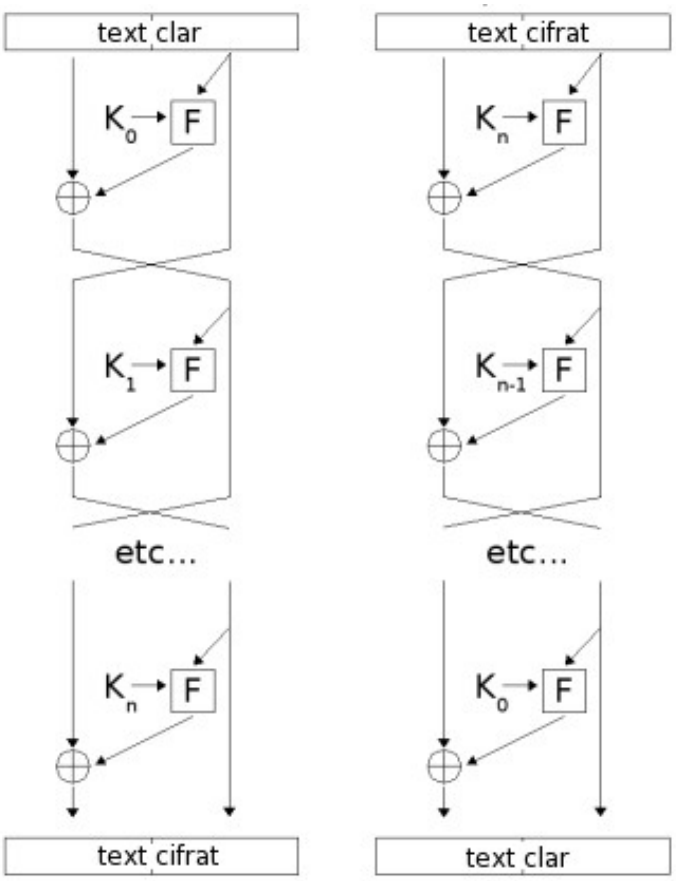

Fig.14 Diagrama de cifrare pentru cifrul Feistel

Constructia Feistel este utilizata si in algoritme care nu sunt cifruri de bloc. De exemplu, Optimal Asymmetric Encryption Padding (OAEP) utilizeaza o simpla retea Feistel pentru a randomiza textele cifrate in unele scheme de cifrare asimetrica.

Data fiind natura cifrului FEISTEL, textul cifrat sparge orice conventie de caractere si produce numere care corespund la caractere care nu exista. De fapt orice tip de cifru care opereaza pe blocuri de text sau pe biti individuali nu avea cum sa respecte standardul de

caractere. Din aceasta cauza se utilizeaza la iesire un codor, mai precis un codor BASE64 (cap. 3.2), pentru a reda textului cifrat proprietatea de lizibilitate si implicit pentru a putea fi transmis prin sistemele informatice.

Alph respecta intru-totul acesta conventie si libraria utilizeaza functia de BASE64 din functiile disponibile pentru a coda iesirea dintr-o criptare prin FEISTEL. De asemenea, inainte de a intra in functia FEISTEL, se utilizeaza codorul BASE64 pentru a reda numerele aleatoare produse la cifrare. In concluzie, Alph poate fi utilizat pentru a cifra si descifra texte utilizand cifrul FEISTEL. Alegerea XOR pentru functia f este pur si simplu demonstrativa si nu este posibila definirea unei alte functii: alph si libraria libalphCRYPT este un utilitar care implementeaza cifruri si nu un utilitar pentru a crea noi cifruri.

5. 2 Criptografia cuantica

Criptografia cuantica este un mod de abordare a securizarii comunicatiilor bazat pe anumite fenomene din mecanica cuantica. Spre deosebire de criptografia traditionala, care utilizeaza mai multe tehnici matematice pentru a restrictiona informatiile fata de o parte terta, criptografia cuantica este bazata pe fizica informatiei. Procesul de a trimite si de a stoca infromatie este intotdeauna implementat printr-o metoda fizica, de exemplu fotoni in fibrele optice sau electroni in curentul electric. Ce poate o parte terta sa masoare, depinde exclusiv de legile fizicii. Utilizand fenomene cuantice cum ar fi superpozitii cuantice sau imbinarea cuantica, s-ar putea proiecta si implementa un sistem de comunicare capabil sa determine daca informatia are o "scurgere" catre o parte terta. Acest lucru se datoreaza faptului ca masuratorile pe purtatorul cunantic provoaca perturbari si lasa urme.

Marea problema a criptografiei asimetrice este distribuirea cheii. O solutie care se bazeaza pe matematica este criptografia cheii publice. O alta, se bazeaza pe fizica: criptografia cuantica. Pe cand criptografia cu cheie publica se bazeaza pe dificultatea de a calcula diferite probleme matematice (cum ar fi factorizarea numerelor cap 4.1 Categorii), criptografia cuantica se bazeaza pe legile mecanicii cuantice. Dispozitivele de criptografie cuantica utilizeaza fotoni individuali si se bazeaza pe principiul de nedeterminare al lui Heisenberg (nu se pot determina simultan cu precizie oricat de buna, valori pentru anumite

perechi de variabile observabile, ca de exemplu pozitia si impulsul unei particule, energia si timpul de viata al unei stari cuantice etc.) sau pe fenomenul de corelatie ‘imbinare’ cuantica (fenomenul de a descire starile cuantice a doua sau mai multe obiecte prin referinta la ele insesi chiar daca ele sunt separabile in spatiu).

Bazandu-se pe cele doua principii (nedeterminarea si imbinarea cuantica), doua tipuri de protocoale criptografice au fost inventate. Ambele se bazeaza pe faptul ca sistemele cuantice sunt perturbate prin masuratori asupra lor. Primul tip utilizeaza o polarizare de fotoni pentru a coda bitii de informatie si se bazeaza pe randomizarea cuantica. Al doilea tip utilizeaza imbinarea starilor de fotoni pentru a coda bitii si se bazeaza pe faptul ca infromatia definind cheia apare numai dupa masuratorile lui Alice si Bob (cap. 1 Introducere).

Exista doua protocoale de criptografie cuantica in functiune. Cum am definit in capitolul 1, Alice si Bob vor sa faca o comunicatie iar Eve, al treilea element pe care inca nu l-am definit, este o persoana care vrea sa afle ce au vorbit cei doi.

Prima metoda [8], utilizeaza pulsuri de lumina polarizata, cu un foton per puls. Consideram doua tipuri de polarizari, lineare si circulare. Polarizarea lineara poate sa fie verticala sau orizontala si polarizarea circulara poate sa fie de stanga sau de dreapta. Orice tip de polarizarea a unui singur foton poate sa codeze un bit de informatie, de exemplu, o polarizare verticala pentru "0" si una orizontala pentru "1" sau o polarizare circulara de stanga pentru "0" si o polarizare circulara de mana dreapta pentru "1". Pentru a genera o cheie aleatoare, Alice trebuie sa trimeata sau o polarizare orizontala sau o polarizare verticala cu probabilitati egale. Pentru a preveni ca Eve sa intercepteze mesajul, Alice utilizeaza aleator polarizarea circulara si alege aleator intre polarizarea de stanga sau de dreapta a fotonilor. Siguranta acestei scheme este bazata pe faptul ca Eve nu stie daca un puls anume codeaza 0 sau 1 utilizand polarizarea lineara sau cea circulara. Daca Eve incearca sa masoare starea si ghiceste gresit, ea o va perturba si Alice si Bob pot sa monitorizeze asemenea perturbari ba chiar ar putea sa afle cat din cheie a fost aflata. Nici Bob nu stie care polarizari au fost utilizate pentru un puls dat dar el poate sa ghiceasca si jumatate din timp el va ghici corect. Dupa ce fotonii sunt toti primiti, Alice ii poate spune unde a ghicit corect si unde a gresit.

A doua metoda [9], utilizeaza perechi de fotoni corelati. Acestia pot fi creati de catre Alice, de catre Bob sau de alta sursa inclunzand-o pe Eve. In orice caz, fotonii sunt distribuiti astfel incat Alice si Bob obtin un foton din fiecare pereche.

Schema se bazeaza pe trei proprietati ale imbinarii:

Prima, putem produce stari perfect corelate in sensul in care daca Alice si Bob testeaza daca particulele lor au polarizari verticale sau orizontale atunci obtin intotdeanuna raspunsuri opuse. Acelasi lucru este adevarat daca ambii masoara orice alta pereche de polarizari complementare. Dar, rezultatele lor sunt complet aleatoare: este imposibil pentru Alice sa prezica daca ea va obtine o polarizare verticala sau orizontala.

In al doilea rand, aceste stari au o proprietate numita ne-localitate cuantica care nu are nici un analog in fizica traditionala sau experienta de zi cu zi. Daca Adlice si Bob fac masurari de polarizare atunci raspunsurile lor nu vor fi perfect corelate dar vor fi oarecum corelate. Ceea ce inseamna ca exista o probabilitate de peste 50% ca Alice poate, prin masuratorile ei si deduca oarecum corect masuratorile lui Bob si vice versa.

In al treilea rand, orice incercare de a intercepta mesajul de catre Eve, va slabi aceste corelari intr-un mod in care Alice si Bob pot sa-si dea seama.

Pentru a asigura aceste chei, fiindca ar fi dezastros daca Eve ar obtine chiar si putine informatii despre cheie, exista asa zisa amplificare a informatiei private (privacy amplification) si consta intr-o metoda de a reduce treptat cheia intre Alice si Bob despre care Eve nu are habar.

Din cauza principiului lui Heisenberg, un atac "man-in-the-middle" este imposibil. Daca Eve incearca sa intercepteze mesajul, ea il va modifca substantial daca utilizeaza un detector incorect. Ea nu poate sa transmita fotoni lui Bob fiindca va introduce erori inaceptabile in comnicare. Daca Alice si Bob utilizeaza un sistem de corelare de fotoni, atunci este aproape imposibil de a intercepta mesajul deoarece creerea a trei fotoni corelati ar decreste corelarea individuala a fiecarui foton si incercarea de atac ar deveni detectabila. Re-transmisia ambilor fotoni, odata ce au fost seaparati, devine imposibila. Practic, la momentul de fata, nu exista o metoda pentru a "sparge" un asemenea protocol. Tot ce este posibil, este intreruperea comunicatiei sau inducerea de erori de linie insa nu este posibila refacerea datelor de catre o parte terta.

Mai trebuie spus ca, la momentul actual domeniul este la inceput si metoda criptografiei cuantice este limitata la distante de transmisie de cel mult 100 km, deoarece transmisia semnalului luminos prin orice mijloc s-ar face (de obicei prin ghiduri de lumina ca de exemplu fibrele optice) este afectata de interactia cu mediul (de obicei aerul) ceea ce induce perturbarea starilor cuantice ale semnalului transmis: fie pierderea (atenuarea) polarizarii, fie pierderea corelatiilor initiale a doi sau mai multi fotoni din fascicolul transmis. Se fac studii in spatiul cosmic, unde vidul avansat reduce pierderile in codarea semnalului, fie ca aceasta codare s-a facut prin polarizari sau corelatii.

5. 3 Analiza criptografica

Sursele primelor mentiuni de criptanaliza dateaza din jurul anului 750, la inceputul califatului abasid (sau dinastia abasida) care a vestit epoca de aur al civilizatiei islamice. Artele si stiintele au inflorit toate in egala masura beneficiind de un mediu propice pentru aceasta dezvoltate culturala. Civilizatia islamica s-a extins pana ce jumatate din lumea cunoscuta era ocupata. Bogatia acestei culturi a fost in mare masura rezultatul unei societati in care domneau bunastarea si pacea. Califii abasizi au fost mai putin preocupati de cuceriri decat predecesorii lor, ocupandu-se in principal de faurirea unei culturi intr-o societate organizata si imbelsugata.

Administratia la timpul acela, pentru a pastra secrete, utiliza o metoda cunoscuta de cifrare prin substitutie polialfabetica asemanatoare decalarii CAESAR. Pe baza acestei metode, apare un manuscris semnat de Al Kindi care ilustreaza o metoda de analiza criptografica eficienta care in zilele noastre este cunoscuta sub numele de "analiza de frecventa".

Al Kindi s-a bazeaza pe faptul ca in orice limba scrisa, unele litere apar mai frecvent decat altele. De asemenea, acest lucru se pot aplica mai nou si digramelor sau trigramelor, adica perechi (tripleti) de litere care apar intr-o limba scrisa. Al Kindi rezuma tehnica in doua paragrafe scurte:

"Un mod de a descifra un mesaj criptat, daca stim in ce limba e, este sa gasim un alt text in clar in aceeasi limba, suficient de lung incat sa umple aproximativ o pagina, iar apoi sa numaram aparitiile fiecarei litere. Vom numi litera care apare cel mai des "prima",

urmatoarea litera care apare cel mai des "a doua", urmatoarea litera care apare cel mai des "a treia" si asa mai departe, pana cand trecem in revista toate literele din textul esantion.

Apoi cercetam textul cifrat pe care vrem sa-l dezlegam si ii clasificam si lui simbolurile. Gasim simbolul care apare cel mai des si il inlocuim cu "rima" litera din textul esantion in clar, urmatorul simbol care apare cel mai des este inlocuit cu "a doua" litera, urmatorul simbol care apare cel mai des este inlocuit cu "a treia" litera si asa mai departe pana cand trecem in revista toate simbolurile criptogramei pe care vrem sa o dscifram."

Puterea metodei lui Al Kindi consta in faptul ca nu este necesara cunosterea algoritmului (atat timp cat acesta este o substitutie polialfabetica) si nici cunoasterea simbolurilor din textul cifrat in cazul in care acestea sunt glyfe sau semne. Este necesara doar cunoasterea statisticii limbii in care este scris textul.

Astfel, pe baza manuscrisului lui Al Kindi, s-au alcatuit statistici pentru fiecare limba. Se stie acum care litera este cea mai frecventa intr-o limba sau chiar, dupa cercetari ulterioare, se stiu chiar ce secvente de litere apar cele mai frecvent. Daca luam de exemplu limba engleza, se stie ca litera cea mai frecventa este 'e' urmata de 't' si asa mai departe... De asemenea, se cunosc digramele, adica secventele de doua litere care apar cel mai frecvent, care in cazul limbii engleze sunt 'th', 'he' si asa mai departe. Sau trigramele: 'the', 'ing', 'con' si asa mai departe...

Daca luam un text cifrat, acesta contine litere sau simboluri, care in cazul substitutiei simple polialfabetice se repeta cu o anumita perioada. Daca substitutia noastra a transformat litera 'e' in litera 'z', atunci, daca a fost vorba de un text scris in limba engleza, in textul cifrat, litera 'z' va fi cea care are probabilitatea sa apara de cele mai multe ori. Astfel, putem presupune corect, din punct de vedere statistic, si inlocui in textul cifrat litera 'z' cu litera 'e'. Pentru a doua litera care apare cel mai frecvent in textul cifrat, putem presupune ca este vorba de litera 't' in cazul in care este vorba despre un text scris in limba engleza. Procedura se repeta astfel si pentru alte litere individuale, bigrame, trigrame. Daca entropia este suficient de mare atunci se poate stipula ca aceasta metoda va descifra orice text cifrat prin orice algoritm de substitutie polialfabetica.

Analiza prin frecventa este una dintre cele mai solide solutii de cercetare a unui text cifrat. Este probabil metoda de la care se asteapta cele mai corecte rezultate si consta intr-o clasa aparte printre celelalte metode de analiza criptografica. Foarte des, acesta metoda este

un ultim pas dupa ce textul a fost prelucrat prin alte metode. Ea se utilizeaza cel mai frecvent in combinatie cu alte metode de analiza care reusesc sa ‘curete’ pur si simplu textul. Analiza de frecventa a fost si ea utilizata la descifrarea Enigmei, dupa ce s-au aplicat alte algoritme pentru "limpezirea" textului. Ceea ce este remarcabil in cazul analizei de frecventa este ca ea nu necesita absolut nici o cunostinta asupra alogitmului de cifrare sau a textului in clar decat, si cel mai important lucru, limba in care acesta a fost scris. Alte metode sunt specifice unui anumit tip de algoritm sau chiar necesita, in cazul "know plaintext attack", accesul la algoritmul folosit. De fapt, fara a sti algoritmul folosit la cifrare, singura posibilitate ramane cea descrisa de Al Kindi acum mai bine de 1500 de ani: analiza de frecventa.

Posibilitatile de contrare ale acestei metode sunt relativ putine. Nu prea exista un mod bine definit pentru a contra analiza de frecvente deoarece ea se bazeaza mai degraba pe limbaj si pe universalitatea mesajelor decat pe metoda algoritmica de cifrare sau chiar cheia. De fapt, cheia, in contextul metodei lui Al Kindi, nu mai are absolut nici o relevanta. Metoda permite ocolirea cheii pe cand celalte metode se bazeaza mai degraba pe recuperarea ei. Astfel, pentru a contra analiza de frecvente, cea mai bun alegere ar fi o metoda prin care s-ar uniformiza frecventele lngvistice intr-un mesaj. Mai precis, daca metoda de cifrare utilizeaza un algoritm care rupe legatura intre caracterele, sau literele, care apar cel mai frecvent intr-un anumit limbaj si mesajul in forma cifrate atunci cifrul ar fi capabil sa reziste incercarilor de analiza de frecventa. Aceasta metoda apare la inceput la M Del Vayo (implementat in Alph ca parametru MDELVAYO) care a pus la punct o forma de normalizare al distributiilor de frecvente intr-un mesaj utilizand ceea ce se cunoaste acum sub numele de "tabla de sah pentru uniformizare" (straddling checkerboard). Asa cum implica si numele, metoda contine o tabla de sah care are rolul de a re-substitui cele mai frecvente caractere din limbaj prin alte caractere. Metoda nu este complexa si s-ar putea spune ca ea face de fapt doar o simpla substituie asa cum ar face tabloul de stechere al masinariilor Enigma sau Purple. Singura diferenta este ca substitutia este destinata intr-u totul caracterelor care apar cu mare frecventa intr-un limbaj.

Aceasta metoda este prezenta intr-un cifru destul de obscut numit Vic. Se presupune ca acet cifru, dezvoltat de catre NKVD mai este inca in actiune si nu s-a descoperit o metoda clara pentru a decifra un asemenea mesaj desi se cunoaste metoda de cifrare. Cauza principala este ca acest cifru utilizeaza un numar mare de chei care intervin la fiecare pas in

procedura de cifrare. Indiferent de acest lucru, un anumit pas din cifrarea in Vic, este utilizarea unei table de sah pentru a uniformiza frecventele literelor din mesajul cifrat. Adevarul este ca rezultatul lui Vic sunt secvente de numere insa intr-un pas anterior mesajul se afla inca sub forma literala. In forma originala, tabla de sah a lui Vic a fost conceputa si adaptata limbei ruse si substituie cele mai frecvente litere din limba rusa. Metoda insa ii apartine lui M Del Vayo care a construit o intreaga metoda de cifrare bazata pe o tabla de sah pentru normalizarea frecventelor.

Alph implementeaza analiza de frecventa aducand ceva nou in aceasta tehnica. El face o subsitutie peste alfabetul cifrat insa are posibilitatea sa aplice substitutia de mai multe ori. Acest lucru Alph il numeste ca fiind "sharpness" adica nivelul de ascutime al "filtrului" de analiza de frecventa. Celalalte implementari cunoscute, nu au aceasta posibilitate si Alph isi rezerva drepturile de autor asupra acestei metode.

Pentru orice text, Alph poate sa aplice filtrul de analiza de frecventa si da rezultatul asa cum l-ar da orice metoda de cifrare incorporata in program. Textul rezultant este afisat pe stdout si da posibilitatea utilizatorului sau programatorului sa-l interpreteze mai departe. Daca se doreste un nivel mai adanc de ascutime ("sharpness") se da un parametru numeric lui Alph prin care acesta va sti de cate ori sa aplice analiza de frecventa inainte sa afiseze textul rezultant.

O metoda de analiza cunoscuta care incorporeaza trasaturi din diverse alte proceduri de analiza criptografica, este metoda “hill-climbing”. Aceasta are la baza o metoda de analiza diferentiala a algoritmelor de cifrare cu o cheie privata.

Se propune gasirea textului in clar plecand de la un text cifrat necunoscut si o cheie necunoscuta. Abordarea cea mai cunoscuta in acest caz este metoda "brute-force" in care se incearca decifrarea textului utilizand toate cheile posibile. Aceasta metoda exhaustiva da rezultate numai in cazul in care problema este rezolvabila in timp computational. Deseori insa, nu se stie daca problema are aceasta natura si se procedeaza la rezolvarea ei sperand ca acesta va da rezultate. De cele mai multe ori, acest lucru nu se adevereste. Masina pe care se ruleaza cautarea exhaustiva va cauta interminabil cheia pe care a folosit-o utilizatorul pentru a cifra mesajul.

Metoda “hill-climbing” are la baza o abordare putin mai subtila. Ea implica un algoritm aleator de alegere de chei si o analiza de frecventa comparativa. In primul pas, se alege o cheie din spatiul de chei. *Spatiul de chei* este definit ca toate combinatiile alfabetice de litere de acelasi lungime de la A la Z. De exemplu, daca se presupune o cheie de o lungime de trei litere atunci spatiul de chei este: AAA, AAB, ... ZZY, ZZZ. Cheia se alege la aproximativ o patrime din spatiul total de chei, pentru exemplul nostru, aceasta ar fi FFF, si se aplica algoritmul in mod descifrare a textului cifrat. Acest lucru va rezulta intr-un text, care poate sa fie chiar textul in clar daca FFF a fost chiar cheia utilizata la cifrare, caruia i se va aplica o analiza de frecventa. Se urmareste mai precis apropierea textului de indexul de coincidenta. *Indexul de coincidenta* este definit ca suma patratelor tuturor probabilitatilor de aparitie a literelor individuale intr-o limba (in cazul nostru, limba engleza). Rezultatul se noteaza, si se aplica aceeasi procedura pentru cheia +1 si cheia -1, adica in exemplul nostru FFE si FFG, care de asemenea se vor nota separat. Apoi, cele trei rezultate se vor compara si se ia ca noua cheie de referinta, cheia care a dat cel mai apropiat rezultat de indexul de coincidenta.

Evolutia teoretica a unui astfel de algoritm poate fi reprezentata ca in figura 15.

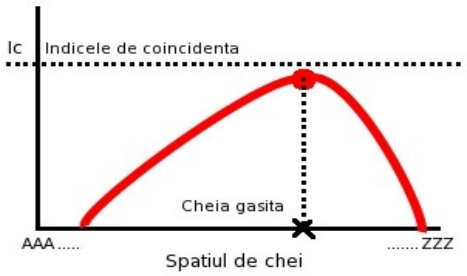

Fig. 15 Evolutia teoretica (ideala) a algoritmului de analiza criptografica prin metoda “hill-climbing”

Cautarea se face de la prima cheie din spatiu de chei, adica AAA, pana la ultima cheie ZZZ. Atunci cand rezultatul este mai mare intr-o parte, adica in exemplul de fata, cand se adauga la cheie, cifrul in modul de decriptare produce un text mai apropiat de indexul de coincidenta si algoritmul "urca". La un moment dat, cand algoritmul mai adauga la cheie si produce textul decifrat, acesta din urma da un index de coincidenta mai mai mic decat cel

precedent. In acest caz, algoritmul se intoarce inapoi la "stanga" scazand din cheie si ajungand la maximul curbei (Fig. 15). Maximul graficului se obtine la pozitia abscisei reprezintata de cheia care a fost folosita la cifrarea mesajului in clar. Cunoscand algoritmul, si mai nou cheia, se poate obtine textul in clar.

Acesta procedura functioneaza foarte bine in teorie insa in practica este posibil sa avem mai multe maxime ale evolutiei curbei din Fig. 15. Acest lucru este normal deoarece nu avem preciza perfecta a indicelui de coincidenta si nu avem un text infinit care sa satisfaca o buna statistica. La fel, datorita difuziei si confuziei anumitor algoritme de criptare, pot aparea probleme atunci cand se incerca determinarea cheii perfecte.

Ceea ce se poate intampla este prezentat in figura 16.

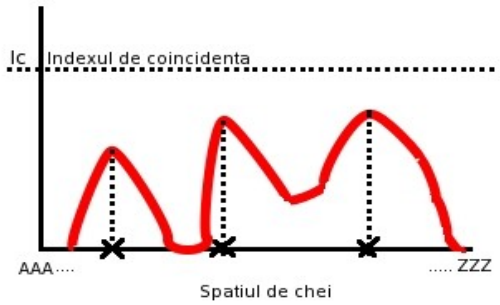

Fig. 16 Evolutia practica (posibila) a algoritmului de analiza criptografica prin metoda “hill-climbing”

Astfel avem practic mai multe puncte de maxim. Acestea nu ar prezenta o problema deoarece ar fi pur si simplu mai multe chei posibile. Chiar daca acestea sunt multe, una dintre ele ar fi cheia care a fost folosita la cifrarea textului. Din punctul de vedere al algoritmului, acest lucru prezinta o problema deoarece el se blocheaza in primul punct pe care il gaseste. Logica lui ii spune sa pastreze cheia cea mai apropiata de indexul de coincidenta. Astfel, el nu poate sa progreseze si sa gaseasca urmatoarele maxime. Pentru a rezolva aceasta dificultate, algoritmul implementat in Alph foloseste un salt Monte-Carlo pentru a "sari" la urmatoarea cheie. Atunci cand acesta constata ca nu mai poate progresa nici in stanga nici in dreapta, el genereaza o noua cheie aleatoare in spatiul de chei si sare pe acesta continuand

procedura de "alunecare" in stanga si in dreapta. Numarul de salturi este determinat, in versiunea curenta a lui Alph, de spatiul de chei. Pentru un spatiu de chei ca in exemplul nostru, adica de trei litere, el poate sa faca un maxim de 26*26*26 adica 17576 de salturi posibile. Utilizatorul insa, ii este prezentata posibilitatea de a modifca numarul de salturi pe care acesta le va face utilizand parametrii transmisi lui Alph.

Alph ia pentru acesta metoda de analiza, un parametru care ii specifica algoritmul, un parametru care ii specifica o cheie de inceput si un numar care reprezinta numarul de salturi pe care acesta le va face si implicit numarul de chei pe care acesta le va gasi eventual. El va furniza utilizatorulu, dupa ce a terminat numarul de salturi, numarul de chei posibile si o lista a acestor chei care corespund maximelor posibile, asa cum este ilustrat in Fig. 16.

Trebuie notat ca modul in care opereaza Alph, si chiar algoritmul utilizat sunt complet originale si nu au mai fost implementate in acest fel. De aceea Alph rezerva drepturile complete de autor sub licenta GNU/GPL mentionata in capitolul introductiv.

In afara de analiza de frecvente, analiza diferentiala sau forta bruta, celalalte metode, inclusiv metoda prezentata mai sus “hill-climbing”, sunt metode hibride care combina diverse idei de baza. Cum putem vedea, metoda “hill-climbing” are in ea caracteristici de analiza de frecventa, de forta bruta si de analiza diferentiala. Analiza de frecvente este singurul concept care ramane in picioare atunci cand nu se cunoaste cheia, sau chiar daca exista intr-adevar o cheie, sau algoritmul de cifrare. La ora actuala, exista programe care incearca sa "sparga" cifruri si, in cel mai bun caz, acestea se bazeaza pe un atac prin forta bruta. Practic, nu exista program care sa aibe versatilitatea unei metode inteligente cum ar fi analiza de frecvente sau “hill-climbing”. Spargerea cifrurilor ramane astfel inca la nivelul "invechit" al hartiei si al creionului [10].

6. Concluzii

De secole intregi regine, regi si generali s-au bazat pe cifruri pentru a distribui informatii si pentru a pastra secretele guvernelor pe care le sustineau. In acelasi timp, toti au inteles implicatiile drastice atunci cand mesajele lor ajungeau in mainile rivalilor, dezvaluind secretele natiunilor lor. Interceptarea cifrurilor a fost motorul care a antrenat dezvoltarea cifrurilor si al codurilor. Intr-adevar, in trecut sau in viitor, state si civilizatii au existat sau se vor naste si se vor dizolva influentate de scurgerea de informatii catre partile rivale. Avem ca exemplu intreaga disputa dintre Grecia antica si Pesria sau exemple mai noi cum ar fi primul si al doilea raboi mondial in care informatiile au fost cateodata pierdute si rezultatele acestor greseli au fost devastatoare.

Dorinta de pastrare a secretului de catre state, si chiar indivizi, a facut posibila infiintarea institutelor de dezvoltare a codurilor si a cifrurilor care au elaborat si elaboreaza in continuare cifruri si coduri din ce in ce mai solide. In acelasi timp, s-au infiintat alte instutitii care au avut rolul de a sparge cifrurile si de a fura secrete.

Criptologii si cei pasionati de stiinta secretelor au fost intodeauna considerati experti lingvistici. Foarte rar problema unui cifru este de natura matematica. In majoritatea cazurilor criptologii profesionisti aveau mai degraba o pregatire de natura umanista decat studii amanuntite de matematica. Chiar si in al doilea razboi mondial, la departamentul de securitate nationala al Statelor Unite se lucra in stransa legatura cu ziarele pentru a recruta noi "experti" in criptografie. In aceste ziare, sectia de "jocuri" in care se gaseau (si se gasesc si azi) cuvinte incrucisate era monitorizata de guvern. Se organizau "concursuri" nationale de cuvinte incrucisate. "Jucatorii" erau stimulati sa rezolve cat mai multe integrame si sa le trimeata prin posta la redactie. Guvernul, organizator secret al acestor concursuri, a recrutat in timpul de al doilea razboi mondial o mare parte din castigatorii jocurilor de integrame. De fapt, in spargerea Enigmei, nu au fost implicate prea multe modele matematice. A fost o lupta a lingvistilor.

Atat in scrierea lucrarii cat si in intreaga dezvoltare a programului Alph, s-a avut in vedere gasirea surselor codurilor. S-a cautat autorul intitial al codurilor prezentate si s-a preferat metoda acestuia fata de ultimele dezvoltari moderne pe aceasi idee. Chiar manualul programului indica in mod foarte clar ca este vorba de un studiu si ca programul intra in clasa

de programe educationale care incearca sa unifice toate codurile la un loc si sa le faca disponibile utilizatorilor sau programatorilor interesati de acest domeniu. Desi se recunoaste ca fiecare modul a lui Alph ar putea sa fie scris, din punct de vedere al programarii, printr-o metoda mult mai eficienta s-a pastrat totusi metoda initiala a autorului cifrului in ceea ce priveste cifrarea si descifrarea. Un exemplu pe care il putem da este cifrul PLAYFAIR care este programat exact dupa descrierile lui William Playfair. Acest lucru "dauneaza" intr-un fel programului deoarece reconstituirea la decifrare este inexacta. O alta interpretare, o alta alternativa la programul Alph ar fi sa se ia metoda si sa se construiasca un algoritm solid care nu respecta intr-u totul toata definitia cifrului. Avantajul ar fi un cifru optimizat insa rezultatul ar fi pierderea originalitatii cifrului si creerea unui nou cifru care nu prea are legatura cu metoda orginala.

La momentul actual (versiunea 0.20) programul Alph contine urmatoarele cifruri si coduri:

La categoria *cifruri clasice*: CAESAR, ATBASH, ZIGZAG, VIGENERE, MDELVAYO, BEALE, VERNAM, NIHILIST, PLAYFAIR, BIFID, USASS, PURPLE, ENIGMA, NAVAJO, TWOSQUARE, LEWIS, ALBERTI, ADFGX.

La categoria *cifruri moderne*: MORSE, ROT13, FEISTEL, BASE64.

La categoria *hash-urilor*: UNIXELF, WEINBERGER, SDBM, DJB2, XOR, MD5.

Pe tot parcursul dezvoltarii lui Alph, s-a constatat faptul ca toate cifrurile sunt de fapt foarte simple si ca o singura persoana ar putea sa creeze o noua metoda care sa dea de gandit celor mai mari experti in domeniu. Dificultatea de a combina literele din alfabet prin diverse metode sau functii reversibile este minora in comparatie cu munca unui spargator de cifruri pentru a dezlega creatia unei singuri minti.

Alph a necesitat mult timp ca sa fie adus la stadiul in care se afla. Trebuie notat ca marea cantitate de munca investita in acest program a fost impartita intre programarea propriu zisa pe de o parte si cercetare “arheologica” pe de alta parte. Din informatia foarte limitata prezentata pe Internet cat si din surse obscure si greu clarificatoare au luat nastere o librarie si un program care tind sa faca o ordine anume. De asemenea, au fost folosite foarte putine surse, sau daca s-au folosit surse, acestea contin individual prea putine informatii ca sa

constituie o referinta solida de la care s-ar putea deduce originiea cifrului sau a librariei care il contine. Pentru cateva cifruri s-a preferat chiar o abordare personala si s-a lucrat cu oamenii din domeniu pentru a dobandi informatiile necesare reconstituirii cifrului. Se poate lua spre exemplu cifrul PURPLE care nu are nici o sursa disponibila de informatii pe Internet sau chiar in cartile de specialitate dat fiind faptul ca acest cifru este considerat pierdut asa cum se precizeaza in capitolul dedicat criptografiei mecanice. A intervenit colaborarea si rabdarea unor oameni carora le sunt indatorat si carora le platesc respectele in capitolul de multumiri.

Pentru a inchide acest studiu asupra criptografiei publice si private, as dori sa amintesc vorbele unui om care si-a dedicat timpul si avera studiului acesteia. Simon Singh deschide, asa cum prezenta lucrare nu inseamna punctul final al dezvoltarii programului Alph, cartea sa numita Cartea Codurilor [5] prin urmatoarea propozitie: "Singurii capabili sa-mi indice greselile sunt exact cei care nu au libertatea sa o faca."

Asa cum am mentionat in introducere, programul Alph, versiunea prezenta 0.20 este publicat pe Internet si poate fi descarcat de la adresa http://freshmeat.net/projects/alph/ in varianta sursa. El poate fi compilat si rulat pe o mare diversitate de sisteme asa cum am mentionat in text. Din cauza dimensiunilor mari (programul are peste 45.000 de linii de cod, incluzand sistemul de autocompilare) este imposibila atasarea sa intr-o anexa la prezenta lucrare.

7. Referinte

[1] http://freshmeat.net/projects/alph/ (**2004**)

[2] http://packetstormsecurity.org/ (**2006**)

[3] Modern Cryptography: Theory and Practice, Prentice Hall PTR, (**2003**)

[4] Istorii, Herodot, vol. II, Cartea a VII-a, traducere de A. Piatkowski, Editura Stiintifica, Bucuresti (**1998)**.

[5] The Code Book, Simon Singh, Fourth Estate, London (**1999**).

[6] The Enigma Cipher Machine: Codes and Ciphers, Tony Sale, http://www.codesandciphers.org.uk/ (**1995**)

[7] "Purple Revealed: Simulation and Computer-Aided Cryptanalysis of Angooki Taipu B" - Wes Freeman, Geoff Sullivan si Frode Weierud, *Cryptologia* (**2003**) 1

[8] A Bibliography of Quantum Cryptography, Gilles Brassard, Universite de Montreal Press (**1984**)

[9] Quantum Cryptography Based on Bell's Theorem, Artur Ekert, *Physisical Review Letters*, 67 (**1991**) 661

[10] Practical Cryptography, Niels Ferguson and Bruce Schneir, Wiley N. Y. (**2003**)