“Kompüter təhlükəsizlik alqoritmlərinin proqramlaşdırılması”

Fənninin

MÜHAZİRƏLƏRİ

“Kompüter sistemləri və şəbəkələri”

kafedrasının dosenti

Əsgər – zadə B.Ə.

Simmetrik şifrələmə alqoritmləri ilkin mətnin emalı üsulları ilə fərqlənirlər. (Bloklu şifrələmə və axınlı şifrələmə üsulları mövcuddur).

Mətnin blokuna, müsbət tam ədəd və ya bir neçə müsbət tam ədədlər kimi baxılır. Blokun uzunluğu ikinin qüvvətinə bərabər seçilir. Bloklu şifrələmə alqoritmlərinin çoxunda aşağıdakı növ əməliyyatlar istifadə edilir:

- cədvəlli əvəzetmə, bitlərin bir qrupu başqa qrup bitlərlə əvəz edilir. Bu S-box adlanır;

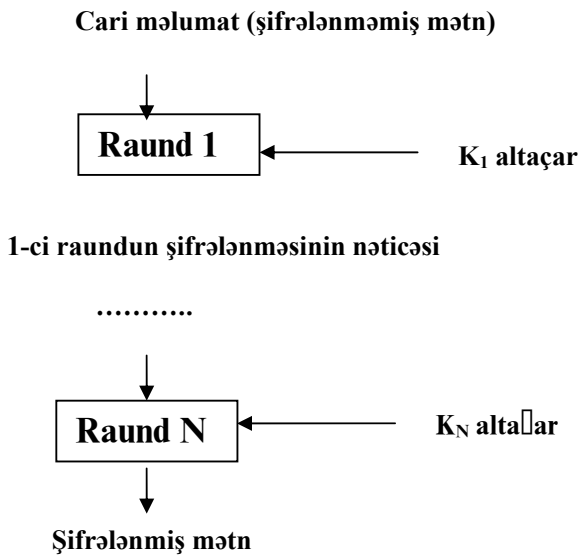
- yerdəyişmə ilə, məlumatın bitlərinin yerləri dəyişdirilir;

- 2 moduluna görə cəmləmə əməliyyatı. XOR və ya $\oplus$ işarələnir;

- 2^{32} və ya 2^{16} modula görə cəmləmə əməliyyatı;

- bir neçə bit dövrü sürüşdürmə.

Bu əməliyyatlar alqoritmdə dövrü olaraq təkrarlanır və raund adlanır. Hər bir raundun girişini ondan əvvəlki raundun çıxışı və xüsusi alqoritm əsasında K şifrələmə açarından alınmış açar təşkil edir. Raundun açarı altaçar adlanır. Şifrələmə alqoritminin strukturu belədir:



Standart şifrələmə alqoritmi bir çox əlavələrdə istifadə oluna bilməlidir:

- verilənlərin şifrələnməsində. Alqoritm verilənlər fayllarının və böyük verilənlər axınının şifrələnməsi üçün effektiv olmalıdır;

- təsadüfi ədədlərin yaradılması. Təsadüfi ədədlərin yaradılması üçün alqoritm effektiv olmalıdır;

- Xeş – funksiya üçün effektiv olmalıdır.

Platformalar

Standart şifrələmə alqoritmi cürbəcür platformalarda realizə olunmalıdır.

- şifrələmə/deşifrələmə üçün olan xüsusi aparuturada alqoritm effektiv realizə olunmalıdır;

- böyük prosessorlarda. 32 bitlik prosessorlarda effektiv realizə olunmalıdır;

- orta ölçülü prosessorlarda. Mikrokontrollerlərdə və başqa orta ölçülü prosessorlarda alqoritm işlənməli;

- kiçik prosessorlarda. Smart-kartlarda belə realizə imkanı olmalıdır.

Feystel şəbəkəsi

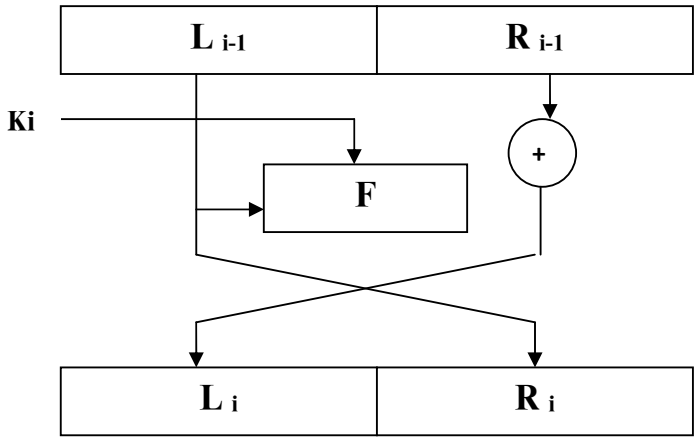
Bloklu alqoritm şifrələnməmiş mətnin n-bitli blokunu şifrələnmiş mətnin n-bitlt blokuna çevirir. N-uzunluqlu blokların sayı 2^n-ə bərabərdir. Əks çevrilmənin mümkün olması üçün belə blokların hər biri şifrələnmiş mətnin öz unikal blokuna çevrilməlidir. Blok uzunluğunun kiçik olması, belə bir əvəzetmə zamanı şifrələnməmiş mətnin statistik xüsusiyyətlərini pis gizlədir. Əgər blok uzunluğu 64 bit olarsa, o zaman şifrələnməmiş mətnin statistik xüsusiyyətləri bilinmir. Belə halda mətnin çevrilməsi üçün açar kimi özü çevrilmə olduğundan onun apparat və proqram realizasiyasının effektivliyi mümkün olmur.

Bir tərəfdən simmetrik şifrələmə alqoritmlərinin bütün tələblərinə cavab verdiyi üçün, digər tərəfdən isə sadə və kompakt olduğu üçün Feyştel şəbəkələri çox geniş yayılmışdır.

Feyştel şəbəkəsinin strukturu (quruluşu)

Giriş bloku bir neçə eyni uzunluqlu altbloklara bölünür. Altbloklar – budaqlar adlanır. Əgər blokun uzunluğu 64 bitdirsə onda hər biri 32 bit olan iki budaq istifadə olunur. Hər budaq, bir-birindən asılı olmadan emal olunur. Bundan sonra budaqların tsiklik olaraq sola sürüşdürülməsi yerinə yetrilir. Belə çevrilmələr bir neçə dövr və ya raund təkrarlanır.

İki budaqlı blok üçün hər bir raundun strukturu belə olur:



F funksiyası yaradan adlanır. Hər bir raund üçün F funksiyası bir budaq üçün hesablanır və nəticə o biri budağın bitlər ardıcıllığı ilə XOR əməliyyatı yerinə yetirilir. Sonra budaqların yeri dəyişdirilir. Raundların optimal sayı 8-32 olmaladır. Raundların sayının artırılması alqoritmin kriptodavamlılığını artırır. Görünür bu xüsusiyyət feyştel şəbəkəsinin geniş yayılmasına səbəb olub. Alqoritmin kriptodavamlılığını artırmaq üçün, alqoritmin özünü dəyişmədən, raundların sayının artırılması kifayətdir. Son zamanlar raundların sayı fiksə olunmur, lakin mümkün olan sərhəd göstərilir.

Feyştel şəbəkəsi, hətta F-in F^{-1} funksiyası olmasa belə, əksçevriləndir. Çünki, aydınlaşdırmaq üçün F^{-1} funksiyasının hesablanması lazım gəlmir. Şifrin açılması üçün eyni alqoritm istifadə olunur. Bu dəfə girişə şifrələnmiş mətn verilir, açarlar isə əks ardıcıllıqla istifadə edilir.

Müasir dövrdə daha tez-tez feyştel şəbkəsinin 4 budaqdan ibarət olan 128 bitlik blokundan istifadə edilir. Budaqların uzunluğunun deyil, sayının artması 32 mərtəbəli prosessorların populyarlığı ilə əlaqəlidir. Deməli, 64 mərtəbəlilərə nisbətən, 32 mərtəbəli sözlərlə əməliyyatların aparılması daha effektivdir.

Feyştel şəbəkəsi əsasında qurulan alqoritmləri F funksiyası xarakterizə edir. Həmçinin başlanğıc və son çevrilmələrdə variantları fərqləndirir. Belə çevrilmələr ağartmaq (Whitening) adlanır və giriş mətnin başlanğıc randomayzını yerinə yetirir.

Kriptoanaliz

P,K və ya hər ikisini öyrənmək istənilməsi prosesi **kriptoanaliz** adlanır. Şifrələmə alqoritmlərinə mümkün olan hücumlardan – güc hücumudur, yəni sadə yolla seçilib ayırmaq. Açarlar çoxluğu çox böyük olduğu halda açarı seçib ayırmaq real deyil. Açarın uzunluğu n-ə bərabər olduğu halda, mümkün olan açarların sayı 2^n-ə bərabərdir. Deməli, açar nə qədər uzun olsa alqoritm o qədər güc hücumuna davamlı olacaq.

Rəqibə bir neçə şifrələnmiş - şifrələnməmiş məlumatlar cütlüyü məlum olma halına əsaslanaraq, müxtəlif cür hücumlar tipi mövcuddur. Şifrələnmiş mətnin analizi zamanı düşmən çox zaman mətnin analizinin statistik üsullarından istifadə edir. Bu halda, onun mətnin tipi haqqında ümumi məlumatı olur, məsələn, ingilis və ya rus dilində mətn, konkret əməliyyat sistemi, faylı, ilkin mətn konkret proqramlaşdırma dilindədir və s. Bir çox hallarda kriptoanalitikin ilkin mətn haqqında kifayət qədər çox informasiyası olur. Kriptoanalitikin bir neçə şifrələnmiş və şifrələnməmiş məlumatları əldə etmək imkanı var və ya

kriptoanalitik məlumatın əsas formatını və ya əsas xarakteristikalarını (xüsusiyyətlərini) bilə bilər. Əgər şifrələnmiş mətndə ilkin mətn haqqında heç bir məlumat yoxdursa, kriptoqrafik sxem tamamilə təhlükəli sayılır. Kripoqrafik sxemin hesablanması aşağıdakı hallarda təhlükəsizdir:

1. Aydınlaşdırmanın qiyməti məlumatın qiymətindən baha olarsa;

2. Aydınlaşdırma vaxtı məlumatın mənalı vaxtından çox olarsa.

Xətti və differensial kriptoanaliz

Belə güman edilir ki, xətti və differensial kriptoanaliz üçün kifayət qədər cütlüklər məlumdur (şifrələnmiş, şifrələnməmiş mətn).

Differensial kriptoanaliz anlayışı 1990-cı ildə Edi Bikam və Adi Şamir tərəfindən qəbul olunub. Onun son vəzifəsi – alqoritmin xüsusiyyətlərindən, əsasən S-box xüsusiyyətindən istifadə edərək, raundun alt açarının təyin edilməsidir. Konkret kripoanaliz üsulu şifrələmə alqoritmindən asılıdır.

Əgər alqoritmin əsasını Feyştel şəbəkəsi təşkil edirsə, onda hesab etmək olar ki, m bloku iki hissədən – m_0 və m_1-dən ibarətdir. Differensial kriptoanaliz bu iki hissədə şifrələmədə alınan fərqlərə baxır (DES üçün XOR əməliyyatının köməkliyi ilə təyin olunurlar).

Bir-birindən fiksə olunmuş fərqli qoşa şifrələnmiş mətn seçilir. Sonra şifrələnmə alqoritminin bir raundundan sonra alınan fərqlərin analizi edilir və açarların ehtimalı təyin olunur. Əgər eyni x fərqi olan çox saylı giriş cütlükləri üçün eyni altaçardan istifadə etdikdə, çıxış qiymətlərində U alınan fərqlərdə uyğun olaraq eyni olur, onda demək olar ki, X- U-nu təyin edilmiş ehtimalla cəlb edir. Əgər ehtimal 1-ə yaxındırsa, onda hesab edilir ki, raundun alt açarı verilən ehtimalla tapılıb. Alqoritmin raundlarının sərbəstliyi - hər raundun altaçarının tapılma ehtimalına vurulmasıdır.

Raundların optimal sayının hesablanması üçün differensial kriptoanalizin nəticələrindən istifadə olunur.

Xətti kriptoanaliz üsulu isə imkan verir ki, kifayət qədər çox sayda cütlüklər (şifrələnmiş mətn, şifrələnməmiş mətn) olduqda, açar tapılsın. Bu üsulun əsas prinsipi:

İşarə edək: P[1], ….. , P[n] – şifrələnməmiş məlumat bloku;

C[1], ….., C[n] – şifrələnmiş məlumat bloku;

K[1], ….. , K[n] – açar.

$$A[i, j, \ldots, k] = A[i] \oplus A[j] \oplus \ldots\ldots \oplus A[k]$$

Kriptoanalizin məqsədi $P \neq 0.5$ ehtimalı ilə yerinə yetirilən aşağıdakı xətti tənliyin tapılmasıdır:

$$P[\alpha_1, \alpha_2, \ldots, \alpha_a] \oplus C[\beta_1, \beta_2, \ldots, \beta_v] = K[\gamma_1 \ldots\ldots, \gamma_c], \quad (1)$$

Burada α_i , β_i , γ_i - məlumat bloklarında və açarda fiksə olunmuş mövqelərdir. P nə qədər çox 0,5 fərqlənirsə, tənlik o qədər çox yararlı hesab edilir.

(1) tənliyinin mənası: əgər şifrələnməmiş məlumatın və şifrələnmiş məlumatın bəzi bitləri üzərində XOR əməliyyatını apardıqda alınan bit, açarın bəzi bitlərinin XOR əməliyyatını göstərir. Bu xətti yaxınlaşma adlanır və P ehtimalı ilə yerinə yetirilə bilər.

Tənliklərin alınması: sol tərəflər çox saylı cütlüklər olan uyğun olaraq şifrələnməmiş və şifrələnmiş blok fraqmentləri üçün hesablanırlar. Əgər yarıdan çox hallar üçün 0-alınarsı, onda $K[\gamma_1 \ldots\ldots, \gamma_c] = 0$ hesab edilir. Əgər əksər hallarda 1-alınarsa, onda $K[\gamma_1 \ldots\ldots, \gamma_c] = 1$ olar.

Alqoritmin işlənməsində istifadə olunan kriteriyalar

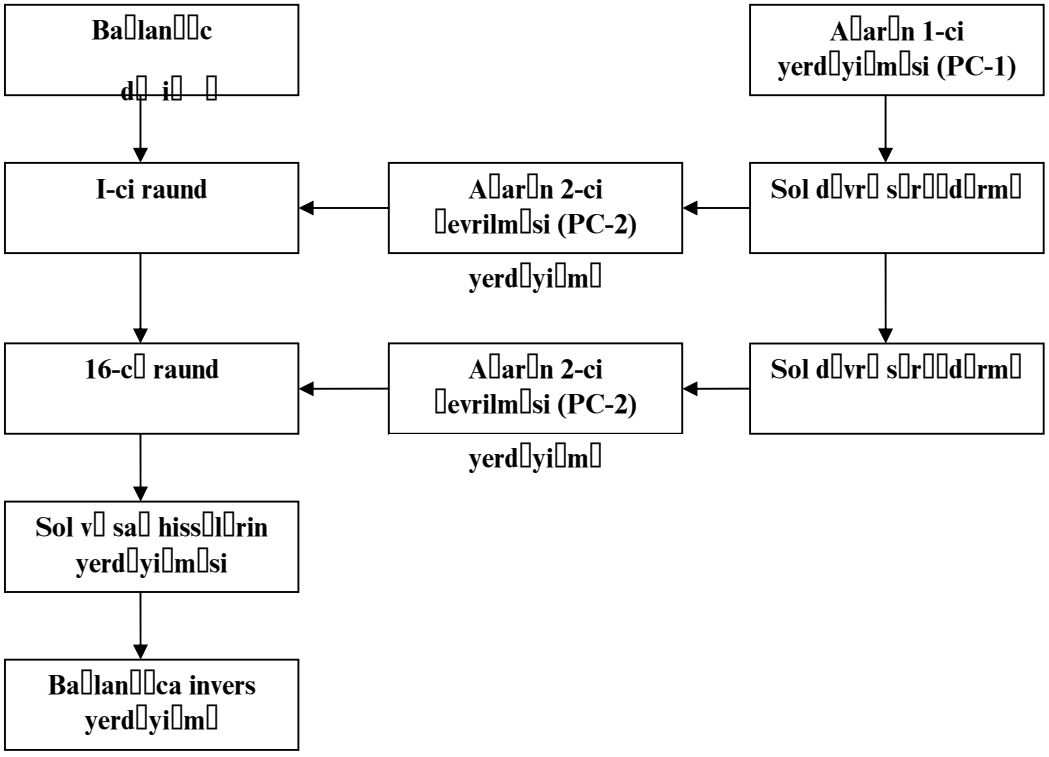
DES alqoritmi

Əsas prinsipləri

Ən geniş yayılmış və məlum olan simmetrik şifrələmə alqoritmlərindən biri DES-dir (Data Encryption Standard). Bu alqoritm 1977-ci ildə işlənib, 1980-cı ildə isə standart kimi qəbul olunub (FIPS PUB 46).

DES – iki budaqlı klassik Feştel şəbəkəsidir. Verilənlər 56 bitli açardan istifadə ilə, 64-bitli bloklarla şifrələnir. Aloritm bir neçə raund nəticəsində 64-bitli girişi 64-bitli çıxışa çevirir. Açarın uzunluğu 56 bitə bərabərdir. Şifrələmə prosesi 4 etapdan ibarətdir. Birincidə 64 bit uzunlulu ilkin mətnin başlanğıc yerdəyişməsi (IP) yerinə yetirilir. Yerdəyişmə zamanı bitlərin yerləri standart cədvələ uyğun olaraq dəyişir. Növbəti etap eyni funksiyanın 16 raundundan ibarət olur. Bu funksiya sürüşdürmə və əvəzetmə əməliyyatlarından istifadə edir. 3-cü etapda sonuncu (16-cı) iterasiyanın çıxışında sol və sağ hissələrin yerləri dəyişdirilir. Nəhayət, 4-cü etapda nəticənin (3-cü etapda alınan nəticə) yerdəyişməsi IP^{-1} yerinə yetirilir.

IP^{-1} yerdəyişmə başlanğıc yerdəyişmənin IP inversiyasıdır.



56 bitli açarın istifadə üsulu sağ tərəfdə göstərilir. Əvvəl açar yerdəyişmə funksiyalarının girişinə ötürülür. Sonra hər raund üçün k_i altaçarı, sol dövrü sürüşdürülmə və yerdəyişmənin kombinasiyası kimi hesablanır. Yerdəyişmə funksiyası hər bir raund üçün eyni olur. Amma k_i açarları – fərqli olur. Çünki hər raund üçün açarın bitlərinin sürüşdürülməsi nəticəsində

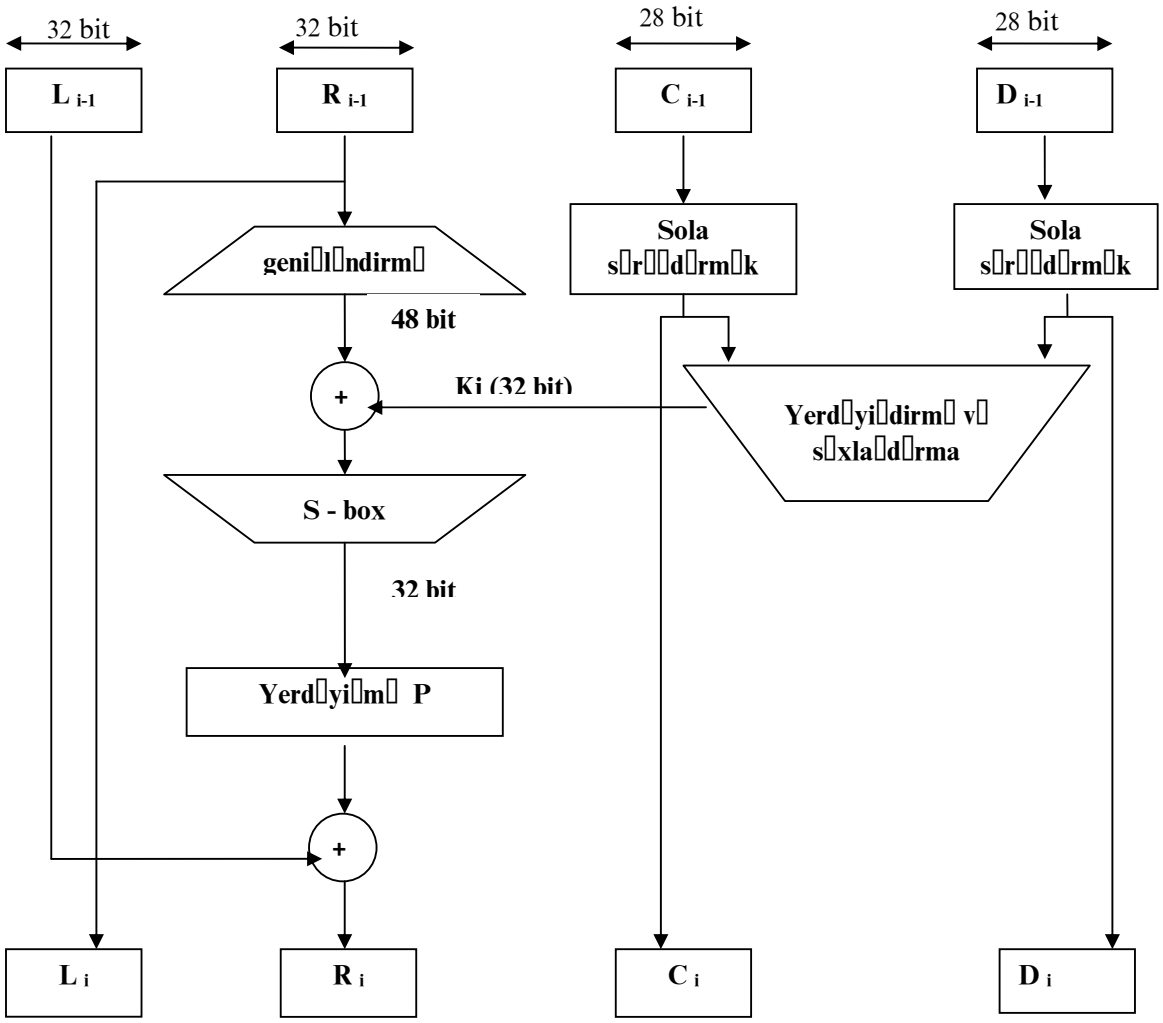
Şifrələmə

Başlanğıc yerdəyişmə

Başlanğıc yerdəyişmə və onun inversiyası standart cədvəldən məlum olur. əgər buna əks funksiyanı ttbiq edəriksə $Y=IY^{-1}(X) = IY^{-1}(IP(M))$, onda bitləri başlanğıc ardıcıllığını almış olarıq M.

Raundun təşki olunma ardıcıllığı

Hər bir raundun təşkil olunma ardıcıllığına baxaq.



DES-in i-ci raundu

64 bit olan giriş bloku 16 raunddan keçir və hər raunddan sonra aralıq 64 bit olan nəticə alınır. Aralıq nəticələrin sol və sağ hissələri 32 bit olur və L və R kimi işarələnir. Hər bir iterasiyanı aşağıdakı kimi təsvir etmək olar:

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$, burada $\oplus$ - XOR əməliyyatıdır.

Beləliklə, sol L_i hissənin çıxışı sağ hissənin girişinə bərabərdir. Sağ R_i hissənin çıxışı isə, XOR əməliyyatının L_{i-1} və (R_{i-1} –dən və K_i -dən) asılı olan F funksiyasına tətbiqinin nəticəsi olur.

F funksiyasını nəzərdən keçirək.

F funksiyasının girişinə ötürülən R_{i-1} uzunluğu 32 bitdir. Əvvəlcə R_{i-1} 48 bitə kimi genişlənir. Bu genişləndirmə belə olur: 32 bit 4 bitdən ibarət olan qruplara bölünür. Hər qrup 2 qonşu qrupun sərhəd bitlərini əlavə etməklə, 6 bitə kimi genişlənir. Məsələn: əgər məlumatın girişinin bir hissəsidir:

… e f g h i j k l m n o p …

genişləndirmənin nəticəsində alınan məlumat

… d e f g h i h i j k l m l m n o p q …

Bundan sonra isə, alınan 48 bitlik məlumatın və 48 bit altaçarın K_i, XOR əməliyyatı yerinə yetirilir. Nəticə (48 bit olan) əvəz etmə funksiyasının girişinə ötürülərək 32 bitlik nəticə alınır.

Əvəzetmə S-box-dan ibarət olur. Hər birinin girişində 6 bit, çıxışında isə 4 bit yaradılır. Bunlar xüsusi cədvəl ilə təşkil olunur. S-box-un birinci və sonuncu giriş bitləri cədvəldə sətrin nömrəsini təyin edir. Ortadakı 4 bit sütun nömrəsini təyin edir. Sətr və sütun kəsişməsini 4 çıxış biti təyin edir. Məs: əgər giriş -011011 olarsa, onda sətrin nömrəsi 01 (1-ci sətr) və sütun nömrəsi 1101-ə bərabər olar(sütun 13). 1-ci sətrin və 13-cü sütunun kəsişməsi 5-ə bərabərdir, deməli çıxış – 0101 olur.

Sonra alınan 32 bitlik nəticə yerdəyişmə P ilə emal edilir. Bunun məqsədi maksimum bitlərin yerdəyişməsidir ki, növbəti şifrələmə raundunda hər bir bitin ayrı S-box ilə emalı ehtimalını artırsın.

Alt açarların yaradılması

Hər bir raundun K_i açarı 48 bitdən ibarətdir. K_i açarı belə bir alqoritm əsasında yaradılır. Alqoritmin girişində 56 bitlik açar üçün əvvəldə, Permuted Choice – 1 (PC-1) cədvəlinə uyğun olaraq yerdəyişmə yerinə yetirilir. Nəticədə alınan 56 bitli açar 2 hissəyə bölünərək 28 bitdən ibarət olan, uyğun olaraq C_0 və D_0 işarə olunur. Hər bir raunda C_i və D_i bir-birindən asılı olmayaraq 1 və ya 2 bit sola sürüşdürülür. Alınan nəticələr növbəti raundun girişi olur. Onlarda Permuted Choice – 2 (PC-2)-un girişi olub çıxışında 48 bitlik nəticə yaradır. Bu nəticə $F(R_{i-1}, Ki)$ funksiyasının girişi olur.

Aydınlaşdırma

Şifrin açılması şifrələmə prosesi ilə analojidir. Alqoritmin girişində şifrələnmiş mətn, istifadə edilir, amma Ki açarlar isə əks ardıcıllıqla istifadə edilir. K_{16} – birinci raunda, K_1 isə sonuncu raunda istifadə olunur. Deyək ki, şifrələmənin i-ci raundunun çıxışı Li // Ri-dir. Onda şifrin açılışının uyğun (16-i)-ci raundunun girişi Ri // Li olacaq.

Aydınlaşdırmanın, son IP-1 yerdəyişməsində girişdə R16 // L16 olmaması üçün, sonuncu raundda sol və sağ hissələrin yerləri dəyişdirilir. Bu mərhələnin çıxışı şifrələnməmiş mətn olur.

Aydınlaşdırma prosesinə diqqət yetirək. Şifrələnmiş mətni və açarı götürüb onlardan alqoritmin girişi kimi istifadə edək. 1-ci addımda başlanğıc yerdəyişməni IP yerinə yetirdikdə, 64 bitlik nəticə alırıq $L_0^d // R_0^d$. Bilirik ki IP və IP^{-1} əks məna daşıyır.

Deməli

$L_0^d // R_0^d = IP$ (şifrələnmiş mətn)

Şifrələnmiş mətn = $IP^{-1}(R_{16} // L_{16})$

$L_0^d // R_0^d = IP(IP^{-1}(R_{16} // L_{16})) = R_{16} // L_{16}$

Beləliklə, aydınlaşdırma prosesinin birinci raundunun girişi şifrələmə prosesinin 16-cı raundunun çıxışının 32-bitlik çıxışına(hansının ki, sol və sağ hissələri əksinə yazılıb) ekvivalentdir.

İndi isə göstərməliyik ki, aydınlaşdırmanın birinci raundunun çıxışı şifrələmə prosesinin 16-cı raundunun 32-bitlik girişinə ekvivalentdir.

Bunun üçün birinci növbədə şifrələmə prosesinə baxaq.

$$L_{16} = R_{15}$$
$$R_{16} = L_{15} \oplus F(R_{16}, K_{15}).$$

Aydınlaşdırma zamanı:

$$L_1^d = R_0^d = L_{16} = R_{15}$$
$$R_1^d = L_0^d \oplus F(R_0^d, K_{16}) = R_{16} \oplus F(R_0^d, K_{16}) = (L_{15} \oplus F(R_{15}, K_{16})) \oplus F(R_{15}, K_{16})$$

XOR aşağıdakı xüsusiyyətlərinin:

$$(A \oplus B) \oplus C = A \oplus (B \oplus C);$$
$$D \oplus D = 0;$$
$$E \oplus 0 = E;$$

Əsasında: $L_1^d = R_{15}$ və $R_1^d = L_{15}$.

Deməli, aydınlaşdırmanın birinci raundunun çıxışı L_{15} // R_{15}-ə bərabərdir. Bu da şifrələmənin 16-cı raundun girişinin yerdəyişməsinə bərabərdir. Bu əməliyyatların 16-cı raund üçün yerinə yetirilməsini göstərmək olar.

Bu prosesi ümumi terminlərdə şifrələmə alqoritminin i-ci raundu üçün:

$$L_i = R_i$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i).$$

Bu bərabərlikləri belə də yazmaq olar:

$$R_{i-1} = L_i$$
$$L_{i-1} = R_i \oplus F(R_{i-1}, K_i) = R_i \oplus F(L_i, K_i)$$

Beləliklə, i-ci raundun girişlərini çıxış funksiyası kimi təsvir edək.

Aydınlaşdırma prosesinin sonuncu çıxışı L_0 // R_0 –dır. Əks yerdəyişmə IP^{-1}-nin girişi L_0 // R_0 olması üçün sağ və sol budaqların yeri dəyişməlidir. Amma:

$$IP^{-1}(L_0 // R_0) = IP^{-1}(IP\ (\text{şifrələnməmiş mətn}))=(\text{şifrələnməmiş mətn})$$

Deməli, şifrələnməmiş mətn alınır, və bu DES aydınlaşdırılmısının mümkünlüyünü göstərir.

DES-in problemləri

Açarın uzunluğu 56 bit olduğuna görə 2^{56} – açar mümkündür. Müasir dövrdə açarın belə uzunluğu kifayət deyil.

DES alqoritminə alternativ -3DES, IDEA Rijhdeal hesab olunurlar.

DES alqoritminin əsasını 8 əvəzedici cədvəl və ya hər iterasiyada istifadə olunan S-boxes təşkil edir.

Burdan belə bir fikir irəli sürmək olar ki, S-boxesin zəif yerlərini bilən üçün kriptoanaliz mümkündür. Buna baxmayaraq uzun illər ərzində aparılan tədqiqatlar S-boxes-in zəif yerlərini aşkar etməyib.

3 DES alqoritmi

DES-in çatışmayan cəhəti –açarın uzunluğunun kiçik olması olduğundan, bu alqoritmə alternativ olan variantların axtarışı aparılır. Bir tərəfdən yeni alqoritmin işlənilməsi, məs: IDEA, digər tərəfdən isə, bir neçə açarlardan istifadə etməklə DES-in təkrar şifrələnməsi.

2DES-in çatışmayan cəhətləri

Açarın uzunluğunun artırılmasının ən sadə üsulu DES-in 2 açardan təkrar istifadəsidir.

Şifrələnmiş C mətnini, P şifrələnməmiş məlumatdan K_1 və K_2 açarı ilə, aşağıdakı şəkildə almaq olar:

$$C = E_{K_2}\left[E_{K_1}[p]\right]$$

Aydınlaşdırmaq üçün iki açar əks ardıcıllıq ilə tətbiq olunmalıdır:

$$P = D_{K_2}\left[D_{K_1}[C]\right]$$

Bu zaman açarın uzunluğu 56*2=112 bitə bərabərdir.

«Aralıq qarşılanma» ilə hücum

Iki DES üçün «aralıq qarşılanma» hücumu mövcuddur. Bu hücum alqoritmin xüsusiyyətinə əsaslanır. Bilirik ki:

$C = E_{K_2}\left[E_{K_1}[P]\right]$ şifrələnmiş mətndir.

Onda $X = E_{K_1}[P] = D_{K_2}[C]$

Baxaq hücum nədən ibarətdir. Tələb olunur ki, hücum edənə: şifrələnməmiş və ona uyğun olan şifrələnmiş mətn cütlüyü məlum olsun: (P,C). Belə halda, birincisi, 2^{56} K_1 açarı üçün P şifrələnir. Bu nəticə cədvəldə yadda saxlanır, sonra cədvəl X-a görə nizamlanır. Növbəti addım, C-nin aydınlaşdırılması üçün K_2 açarın 2^{56} qiymətindən istifadəsindən ibarətdir. Hər bir aydınlaşdırmanın ona bərabər olan qiyməti birinci cədvəldə axtarılır. Uyğün qiymət tapıldıqda, hesab olunur ki, bu açarlar düzgündür və onlar növbəti cütlük üçün (şifrələnmiş mətn-şifrələnməmiş mətn)yoxlanır.

Əgər bir cütlük (şifrələnmiş-şifrələnməmiş mətn) məlum olarsa, onda kifayət qədər çoxlu səhv açarlar alına bilər.

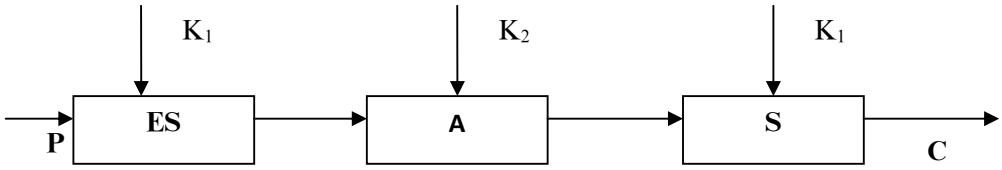
Əgər cinayətkar iki cütlük əldə etmək imkanına malikdirsə, onda 2 DES-in açılışının mümkünlüyü faktiki olaraq sadə DES ilə eyni olur, yəni 2^{56} olur.

Iki açarlı 3 DES

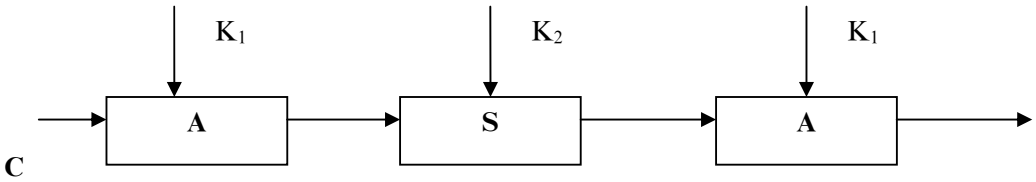
«Aralıq qarşılaşma» hücumuna əks təsir kimi, üç açarlı şifrələmənin üçüncü mərhələsindən istifadədir. Bu, bu gün üçün praktiki imkanlardan yüksəkdir və hücum qiyməti 2^{168} –ə bərabərdir. Açarın uzunluğu 56*3=168 olur. Bu isə çox böyükdür.

Bu üsula alternativ olaraq iki açarlı üç pilləli şifrələmə üsulundan istifadə təklif olunur. Belədə şifrələmə-aydınlaşdırma –şifrələmə (ŞAŞ) ardıcıllığı istifadə edilir.

$$C = E_{K_1}\left[D_{K_2}\left[E_{K_1}[P]\right]\right]$$



3 DES il ifrlm



3 DES il aydnlama

Ikinci mərhələdə şifrələmənin və ya aydınlaşdırmanın olması çox da böyük əhəmiyyət kəsb etmir. Aydınlaşdırmanın olması 3DES-i sadə DES ilə əvəz edir, $K_1=K_2$ kimi istifadə edərək:

$$C = E_{K_1}\left[D_{K_2}\left[E_{K_1}[P]\right]\right] = E_{K_1}[P]$$

3DES-ə məlum olan hücumlar mövcud deyil, 3DES-in açarın qiyməti 2^{112}-yə bərabərdir.

Blowfish – alqoritmi

Blowfish – iterasiyaların sayı 16-ya bərabər olan Feyştel şəbəkəsidir. Blok uzunluğu – 64 bit, açarın uzunluğu – 448 bitə kimi ola bilər. Hər bir şifrələmənin başlanğıcında olduğu kimi, burada da mürəkkəb inisializasiya fazasına baxmayaraq, verilənlərin şifrələnməsi olduqca tez yerinə yetirilir.

Alqoritm, əsasən, tez-tez dəyişilməyən açarı olan əlavələr üçün istifadə olunur. Alqoritmin başlanğıc əl sıxma fazası mövcuddur ki, bu zaman tərəflərin autentifikasiyası və ümumi parametrlərin və məxfiliklərin uzlaşdırılması danışılır.

Alqoritm iki hissədən ibarətdir: açarların genişləndirilməsi və verilənləri şifrələnməsi. Açarın genişləndirilməsi, 448 bit uzunluqlu açarı – 418 bayt ümumi uzunluğu olan bir neçə altaçarlar massivinə çevirir.

Alqoritmin əsasını 16 iterasiyalı Feyştel şəbəkəsi təşkil edir. Hər bir iterasiya, açarlardan asılı olan yerdəyişmədən və açar və verilənlərdən asılı olan, əvəzetmədən ibarətdir. XOR və 32 bitlik sözləri cəmlənməsi- əsas əməliyyatlardır.

Blowfish çoxsaylı altaçarlardan istifadə edilir. Bu açarlar şifrələmədən və aydınlaşdırmadan əvvəl hesablanmalıdır. Alqoritmin əsas elementləri aşağıdakılardır:

1. 18-32 bitlik altaçarlardan ibarət olan K massivi; K_1, K_2, K_3,….. K_{18}.

2. Hər birinin 256 girişi olan 4-32 bitlik S-boxes. Birinci indeks S-boxes-un, ikinci isə-cirişin nömrəsini göstərir.

$S_{1,0}$, $S_{1,1}$, $S_{1,2}$,…… $S_{1,255}$;

$S_{2,0}$, $S_{2,1}$, $S_{2,2}$,…….$S_{2,255}$;

$S_{3,0}$, $S_{3,1}$, $S_{3,2}$,…….$S_{1,255}$;

$S_{4,0}$, $S_{4,1}$, $S_{4,2}$,…….$S_{4,255}$;

Altaçarların hesablama metodlarına sonra baxarıq.

Şifrələmə

Giriş 64-bitlik verilənlər blokunu təşkil edir. 0, iki 32-bitlik hissədən ibarətdir: L_i və R_i.

$L_i=L_{i-1}$ XOR K_i;

$R_i=F(L_{i-1})$ XOR R_{i-1};

Swap L_i and R_i.

F funksiyası

L_i dörd 8-bitlik elementə A,B,C,D bölünür.

$F(L_i)=((S_{1,A}+S_{2,A} \bmod 2^{32})$ XOR $S_{3,C})+S_{4,D} \bmod 2^{32}$

Aydınlaşdırma şifrələmədən ancaq K_i əks ardıcıllıqla istifadə olunması ilə fərqlənir.

Altaçarların generasiyası

Altaçarların hesablanması üçün Blowfish alqoritminin özündən istifadə edilir.

1. Birinci açarlar massivinin və dörd S-boxes fiksəedilmiş sətr ilə inisiallaşdırılır;

2. XOR əməliyyatının K_1 açarının birinci 32-biti ilə yerinə yetirilməsi, XOR əməliyyətının K_2 açarının ikinci 32-biti ilə yerinə yetirilir və s. Dövrün təkrarlanması bütün açarlar massivinin bitləri ilə cəmlənənə kimi davam etməlidir;

Qısa açarlar üçün onların özləri ilə konkatenasiyası yerinə yetirilir.

3. (1) və (2) punktlarında göstərildiyi kimi alınan altaçarlardan istifadə edərək, sıfırıncı sətrin Blowfish alqoritmi ilə şifrələnməsi;

4. K_1 və K_2-nin (3)-cü addımda alınan çıxış ilə əvəz edilməsi;

5. Blowfish alqoritmində altaçarların fiksasiyasından istifadə edərək (3) addımının çıxışının şifrələnməsi;

6. K_3 və K_4-nin (5)-ci addımda alınan çıxış ilə əvəz edilməsi;

7. prosesin davamı açarlar massivinin elementlərinin əvəz edilməsi, sora isə S-boxes-in dördündə Blowfish alqoritminin çıxışları ilə uyğun olaraq modifikasiyası ilə əvəz edilməsi.

Bütün altaçarların yaradılmasına 521 iterasiya tələb olunur.

IDEA alqoritmi

Isveçin federal texnoloji institutunun əməkdaşları tərəfindən (Syudza Lbi və Ceyms Massey) işlənib hazırlanmış IDEA (International Data Encrytion Algorithm) alqoritmi simmetrik bloklu şifrələmə alqoritmidir. Birinci versiyası 1990-cı ildə çap olunub, 1991-ci ildə kriptoqrafik hücumlardan güclü müdafiə versiyası, 1992-ci ildə isə onun dəqiq izahı verilib.

IDEA alqoritmi – DES alqoritminin əvəz edilməsi üçün nəzərdə tutulan alqoritmlərdən biridir. IDEA alqoritminin iş prinsipi blok alqoritminə əsaslanır, 128-bit uzunluqlu açardan istifadə edərək 64-bit verilənlər bloku şifrələnir.

IDEA-nın məqsədi yerinə yetirilməsi sadə olan, kriptoqrafik cəhətdən daha möhkəm (davamlı) alqoritmin yaradılmasıdır.

Kriptoqrafik möhkəmlik

Aşağıdakı keyfiyyətlər IDEA-nın kriptoqrafik möhkəmliyini xarakterizə edir:

1. Blok uzunluğu: blok uzunluğu kifayət qədər olmalıdır ki, məlumatın statistik keyfiyyətlərini gizlətsin. 90-cı illərdə 64 bitlik blok ölçüsü olduqca güclü idi.

2. Açar uzunluğu: açar uzunluğu kifayət qədər uzun olmalıdır ki, sadə açarların baxılması ilə tapılmasın. 128-bitlik açar ilə IDEA kifayət qədər təhlükəsiz hesab olunur.

3. Konfuziya: şifrələnmiş mətnin açardan asılılığı mürəkkəb və qarşılıqlı olmalıdır.

4. Diffuziya: şifrələnmiş mətnin hər bir biti, şifrələnməmiş mətnin hər bitinə təsir etməlidir. Bir şifrələnməmiş bitin çoxlu sayda şəfrələnmiş bitlərə yayılması şifrələnməmiş mətnin statistik strukturunu gizlədir. Şifrələnmiş mətnin statistik xüsusiyyətlərinin şifrələnməmiş mətnin statistik xarakteristikalarından asılılığı sadə olmalıdr. Bu baxımdan IDEA alqoritmi çox effektivdir.

IDEA-da 2 sonuncu punkt 3 əməliyyatla yerinə yetirilir. Bu onu DES-dən fərqləndirir (DES-də hər şey XOR əməliyyatı və kiçik xətti olmayan S-boxes əsasında qurulub).

Hər əməliyyat 16-bitlik iki giriş üzərində yerinə yetirilir və 16-bitlik çıxış yaradır. Əməliyyatlar bunlardır:

1. Bitlərlə XOR əməliyyatı. Bu əməliyyat $\oplus$ işarə olunur.

2. 2^{16} (65536 moduluna görə) modula görə tam ədədlərin cəmi, bu halda girişlər və çıxışlar işarəsiz 16-bitlik tam ədələr kimi izah olunur. Bu əməliyyatı $\oplus$ işarə olunur.

3. $2^{16}+1$ modula görə (65537) tam ədələrin hasili, bu halda girişlər və çıxışlar işarəsiz 16-bitlik tam ədələr kimi izah olunur. Ancaq sıfırlardan ibarət olan blok 2^{16} olur. Bu əməliyyatı • kimi işarə edək.

Aşağıdakı səbəblərə görə bu üç əməliyyat bir-birinə uyğun gəlmir:

1. Üç əməliyyatdan ibarət olan elə bir cütlük yoxdur ki, distributiv qanunu ödəsin, məs:

$a \bullet (a + b) <> (a \bullet b) + (a \bullet c)$

2. Üç əməliyyatdan ibarət olan elə bir cütlük yoxdur ki, assosiativ qanunu ödəsin, məs:

$a + (b \oplus c) <> (a + b) \oplus c$

XOR funksiyasında əsaslanan DES alqoritmindən fərqli olaraq üç əməliyyatın kombinasiyasına əsalanan IDEA alqoritmi kriptoanalizi daha da çətinləşdirir.

Konfuziya(luq) – şifrələnmiş mətn açar arasındakı statistik bağlılığı məhv edilməsi.

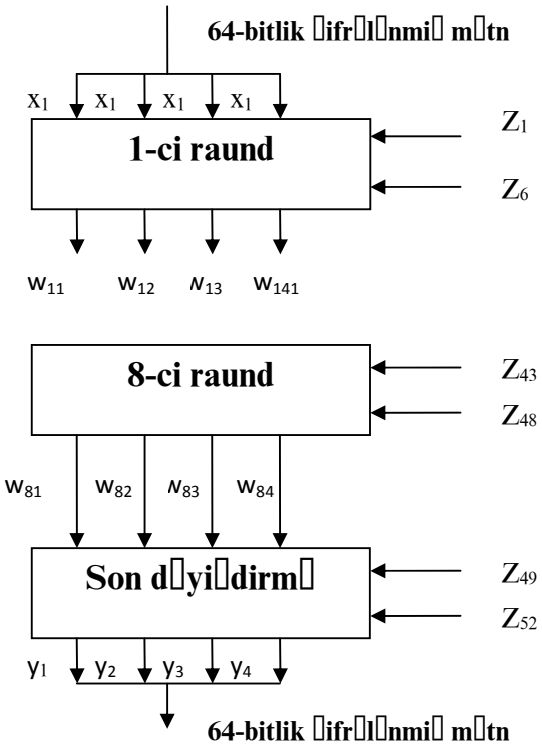
Diffuziya(luq – şifrələnməmiş statistik xüsusiyətlərinin və qaydalarının şifrələnmiş mətn geniş statistik xüsusiyyətlərində və qanunları diapazonunda yayılması

Şifrələmə

Hər alqoritmdə olduğu kimi bu alqoritmin də iki girişi var: şifrələnməmiş blok və açar. IDEA-nın şifrələnməmiş blokunun uzunluğu 64 bit, açar uzunluğu isə 128 bit təşkil edir.

IDEA 8 raunddan ibarətdir. Raundların ardınca son dəyişdirmə yerinə yetirilir. Hər blok dörd 16-bitlik altbloklardan ibarət olur. Hər raund girişdə dörd 16-bitlik altbloklara alır və çıxışda isə dörd 16-bitlik altbloklar yaradır. Son dəyişdirmədə girişdə dörd 16-bitlik altbloklar alınır və çıxışda dörd 16-bitlik altbloklar yaradır. Hər raund altı dənə 16-bitlik altaçardan istifadə edir. Son dəyişdirmədə isə dörd 16-bitlik altaçardan istifadə edlir, yəni alqoritmdə 52 altaçardan istifadə olunur.

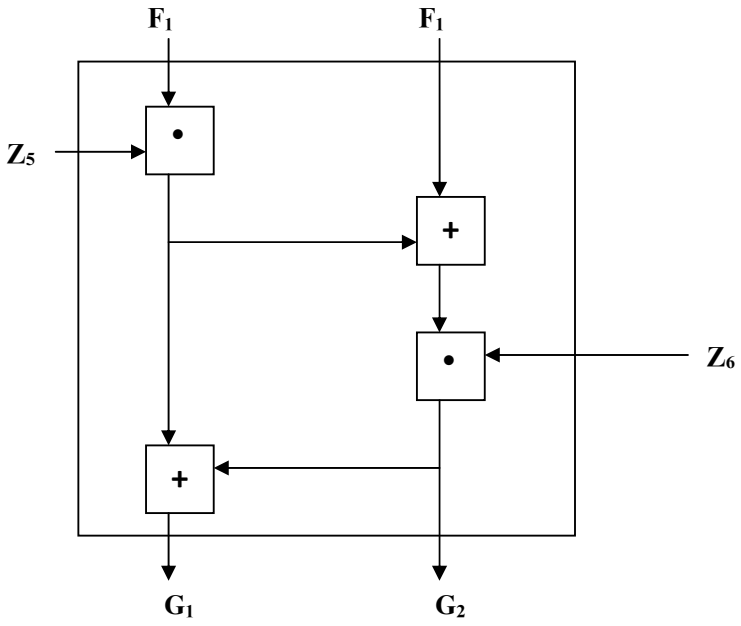
IDEA alqoritmi aşağıdakı kimi təsvir olunur.



Ayrıca bir raundun əməliyyatlar ardıcıllığı

Alqoritmin əsas elementlərindən biri, diffuziyanı əldə edən VC(MA)

(vurma /cəmləmə) adlanan strukturdur.

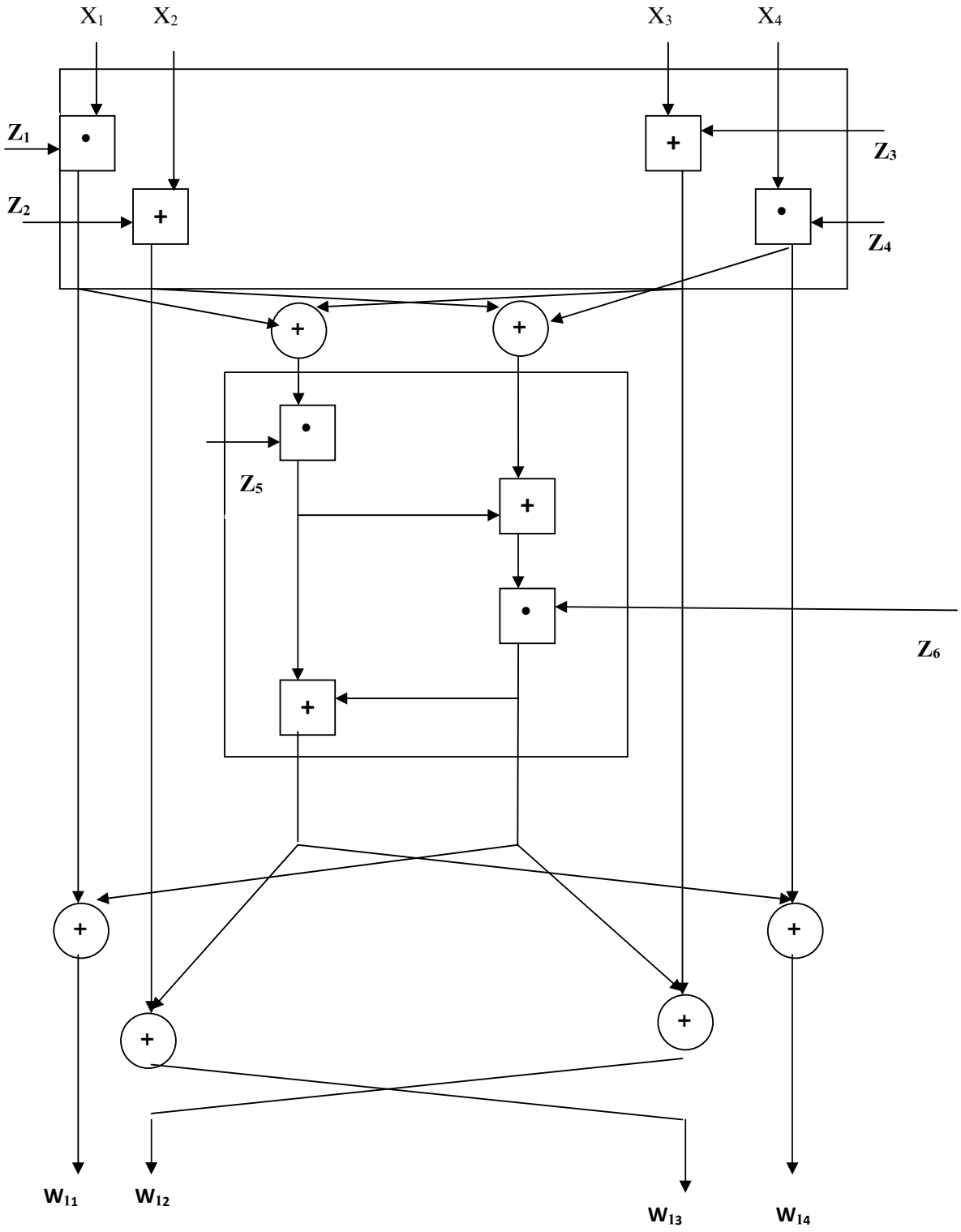


VC (vurma/c□ml□m□) (MA) strukturu.

Bu strukturun girişinə iki 16-bitlik verilən və iki 16-bitlik altaçara verilir. Çıxışda isə iki 16-bitlik nəticə yaradılır. Yoxlama göstərir ki, bu strukturun (quruluşun) çıxışının hər biti girişin şifrələnməmiş blokunun və altaçarların hər bir bitindən asılıdır. Bu strukturun alqoritmdə 8 dəfə təkrarlanması yüksək diffuzluğu təmin edir.

Hər bir raund, cəmləmə və vurma əməliyyatlarından istifadə etməklə, dörd giriş altbloklarının dörd altaçar ilə kombinasiyasının yaradılması ilə başlanır. Bu əməliyyatın nəticəsi olan dörd çıxış blokunun cütlüyü üçün, XOR əməliyyatı yerinə yetirilir və iki 16-bitlik blok alınır. Bu bloklar VC strukturuna giriş olur (şəkildən göründüyü kimi IDEA-nın 1-ci raundu). VC struturunun girişinə bu iki 16-bitlik bloklardan başqa iki altaçar da daxil olur və bunlar 16-bitlik iki çıxış yaradır.

Nəticədə, birinci əməliyyatın dörd çıxış altblokunun VC strukturunun iki çıxış altbloku ilə kombinasiyası, XOR əməliyyatından istifadə edərək, verilən iterasiyanın dörd çıxış altblokunun yaradılması üçün istifadə edilir.



IDEA-nın 1-ci raundu

Qeyd etmək lazımdır ki, ikinci və üçüncü giriş ilə (X_2 və X_3)yaradılan iki çıxış, ikinci və üçüncü çıxışların yaradılması üçün (W_{12} və W_{13}) yerlərini dəyişirlər.

Son dəyişdirilmə kimi qeyd olunan alqoritmin 9-cu raundunu nəzərdən keçirək. Burada eyni struktur təkrar olunur. Yeganə fərq ikinci və üçüncü girişlərin yerlərinin dəyişdirilməsindədir. Bunun səbəbi, aydınlaşdırmanın strukturunun (quruluşunun) şifrələmənin quruluşu ilə eyni olması üçündür. Qeyd edək ki, 9-cu mərhələdə, birinci 8-mərhələdən fərqli olaraq, onların hər biri üçün altı giriş altaçarı lazımdır, dörd giriş altaçarı tələb edilir (bax sxemə-son dəyişdirmə).

Z_1=Z [1……..16] Z_9=Z [26…...41] Z_{17}=Z [51…...66]

Z_2=Z [17…...32]

Z_3=Z [33……48]

Z_4=Z [49……64]

Z_5=Z [65…...80]

Z_6=Z [81…...96]

Z_7=Z [97….112]

Z_8=Z [113….128]

Z_{10}=Z [42…..57]

Z_{11}=Z [58…..73]

Z_{12}=Z [74…..89]

Z_{13}=Z [90….105]

Z_{14}=Z [106….121]

Z_{15}=Z [122….09]

Z_{16}=Z [10…...25]

Z_{18}=Z [67..….82]

Z_{19}=Z [83……98]

Z_{20}=Z [99….114]

Z_{21}=Z [115..….2]

Z_{22}=Z [3…….18]

Z_{23}=Z [19..….34]

Z_{24}=Z [35…...50]

Z_{25}=Z [76……..91]

Z_{26}=Z [92…...107]

Z_{27}=Z [108……123]

Z_{28}=Z [124……11]

Z_{29}=Z [12…...27]

Z_{30}=Z [28…...43]

Z_{31}=Z [44….59]

Z_{32}=Z [60….75]

Z_{33}=Z [101…116]

Z_{34}=Z [117….4]

Z_{35}=Z [5…..20]

Z_{36}=Z [21…..36]

Z_{37}=Z [37…52]

Z_{38}=Z [53….68]

Z_{39}=Z [69….84]

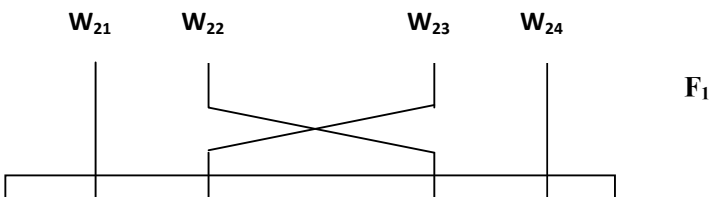
Z_{40}=Z [85…...100]

Z_{41}=Z [126…...13]

Z_{42}=Z [14..….29]

Z_{43}=Z [30……45]

Z_{44}=



Altaçarların yaradılması

128 bit olan şifrələmə açarından əlli iki (52) 16-bitli açtaçarların yaradılması belə yerinə yetirilir: Z_1, Z_2,……,Z_8 kimi işarələnən birinci 8 altaçarlar, açarın özündən alınır. Z_1 – birinci 16 bitə bərabər götürülür, Z_2 – növbəti 16bit və s. Sonra 25 bit sola dövrü sürüşdürmədən sonra, növbəti 8 altaçar yaradılır. Bu prosedur 52 altaçarı tərtib edənə kimi davam etdirilir.

Qeyd edək ki, hər raundun birinci açarı özünün altçoxluq bitlərindən alınır. Əgər açarı bütövlükdə Z[1…..128] qeyd etsək, onda 8 raundun birinci açarları:

Z_1=Z[1……..16] Z_{25}=Z[76…..91]

Z_7=Z[97…..112] Z_{31}=Z[44…..59]

Z_{13}=Z[90…..105] Z_{37}=Z[37…..52]

Z_{19}=Z[83…....98] Z_{43}=Z[30…..45]

Hər raundda birinci və 8-cidən başqa, altaçarın 96-bitinin istifadəsinə baxmayaraq açarın bitlər çoxluğu hər bir iterasiyada kəsişmir. Bunun səbəbi də raundda 6 altaçarın istifadə olunması ilə izah olunur, baxmayaraq ki, açarın hər hər rotasiyaı zamanı altı altaçar alınır.

Aydınlaşdırma

Aydınlaşdırma şifrələmə prosesi ilə analojidir. Aydınlaşdrma şifrələnmiş mətnin IDEA strukturunun giriş kimi istifadəsindən ibarətdir, lakin açarlar yığımı başqadır. Aydınlaşdıran şifrin açılması üçün istifadə olunan açarlar U_1,….U_{52} şifrələyən açarlardan alınır:

1. Aydınlaşdırmanın i-ci raundunun birinci dörd altaçarı şifrələmənin

(10-i) - ci raundunun birinci dörd altaçarından alınır. Burada son dəyişdirilmə 9-cu raund hesab olunur. Aydınlaşdırmanın birinci və dördüncü açarları, şifrələmənin uyğun olaraq 1-ci və 4-cü altaçarlarının 2^{16} +1 modulunda multiplikativ inversiyasına ekvivalent olurlar. 2-ci raunddan 8-ci raunda kimi aydınlaşdırmanın ikinci və üçüncü altaçarları, şifrələmənin uyğun olaraq üçüncü və ikinci altaçarlarının (2^{16}) moduluna görə additiv inversiyasına ekvivalent olurlar. 1-ci və 9-cü raundlar üçün aydınlaşdırmanın ikinci və üçüncü altaçarlarının (2^{16}) moduluna görə additiv inversiyasına ekvivalent olurlar.

2. Birinci 8 raund üçün aydınlaşdırmanın i-ci raundunun sonuncu iki altaçarı şifrələmənin (9-i) raundunun son iki altaçarına ekvivalentdir.

Multiplikativ inversiya üçün notasiyalar (işarələr) istifadə olunur. Z_j^{-1}, yəni:

$$Z_j \cdot Z_j^{-1} = 1 \bmod (2^{16}+1)$$

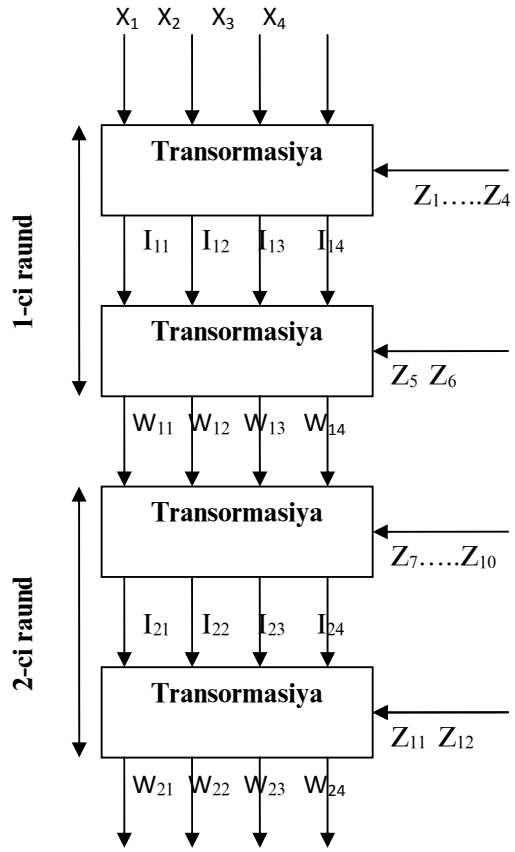
(2^{16} +1) sadə ədəd olduğuna görə, hər bir sıfırdan fərqli olan tam ədədin

$Z_j \leq 2^{16}$ unikal multiplikativ inversiyası $(2^{16}+1)$ moduluna görə var.

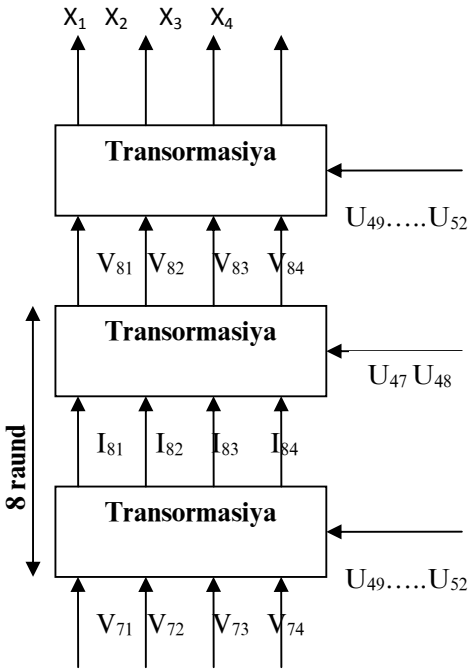
Additiv inversiya üçün belə notasiyadan $(-Z_j)$ istifadə etdikdə, alırıq:

$$-Z_j + Z_j = 0 \bmod(2^{16})$$

Aydınlaşdırma alqoritminin uyğun altaçarlardan istifadə edilməsinin korrekt nəticəsini sübut etmək üçün, eyni zamanda şifrələmə və aydınlaşdırma prosesinə baxaq. 8 raundun hər biri iki dəyişdirilmə mərhələsinə bölünür. Birincisi – transformasiya, ikincisi – şifrələmə adlanır.



IDEA-nın şifrələnməsi



IDEA-nın aydınlaşdırılması

Düzbucaqlılarda yerinə yetirilən dəyişdirilmələrə baxaq. Şifrələmə zamanı transformasiyanın çıxışında aşağıdakı əlaqələr dəstəklənir:

$$y_1 = W_{81} \bullet Z_{49} \qquad y_3 = W_{82} \bullet Z_{51}$$

$$y_2 = W_{83} \bullet Z_{50} \qquad y_4 = W_{84} \bullet Z_{42} \qquad (1)$$

Aydınlaşdırmanın birinci raundunun birinci mərhələsində isə növbəti əlaqələr dəstəklənir:

$$J_{11} = y_1 \bullet U_1 \qquad J_{13} = y_3 \bullet U_3$$

$$J_{12} = y_2 \bullet U_2 \qquad J_{14} = y_4 \bullet U_4 \qquad (2)$$

(2) =dəkiləri (1) əvəz etdikdə

$$j_{11} = y_1 \bullet z_{49}^{-1} = W_{81} \bullet Z_{49} \bullet z_{49}^{-1} = W_{81};$$

$$j_{12} = y_2 + -Z_{50} = W_{83} + Z_{50} + -Z_{50} = W_{83};$$

$$j_{13} = y_3 + -Z_{51} = W_{82} + Z_{51} + -Z_{51} = W_{82};$$

$$j_{14} = y_4 \bullet z_{52}^{-1} = W_{84} \bullet Z_{52} \bullet z_{52}^{-1} = W_{84};$$

Beləliklə, aydınlaşdırmanın birinci mərhəlsinin çıxışı, şifrələmə prosesinin sonuncu mərhələsinin girişinə, ikinci və üçüncü blokların ardıcılığı istisna olmaqla, ekvivalentdir. Növbəti münasibətlərə baxaq:

$$W_{81} = I_{81} \oplus MA_R(I_{81} \oplus I_{83}, I_{82} \oplus I_{84})$$

$$W_{82} = I_{82} \oplus MA_R(I_{81} \oplus I_{83}, I_{82} \oplus I_{84})$$

$$W_{83} = I_{83} \oplus MA_L(I_{81} \oplus I_{83}, I_{82} \oplus I_{84})$$

$$W_{84} = I_{84} \oplus MA_L(I_{81} \oplus I_{83}, I_{82} \oplus I_{84})$$

Burada $MA_R(x,u)$ – MA strukturunun sağ çıxışıdır, x və u girişlər ilə və $MA_L(x,u)$ – MA struturunun sol çıxışıdır. Onda x və u girişləri ilə aşağıdakı alınır:

$V_{11}=J_{11}\oplus MA_R(J_{11}\oplus J_{13}, J_{12} \oplus J_{14})=W_{81}\oplus MA_R[(W_{81}\oplus W_{83},W_{82} \oplus W_{84})=$

$I_{81} \oplus MA_R(I_{81}\oplus I_{83}, I_{82} \oplus I_{84}) \oplus MA_R(I_{81}\oplus I_{83}, I_{82} \oplus I_{84}) \oplus$

$\oplus I_{83} \oplus MA_R(I_{81}\oplus I_{83}, I_{82} \oplus I_{84}),$

$I_{82} \oplus MA_L(I_{81}\oplus I_{83}, I_{82} \oplus I_{84}) \oplus I_{84} \oplus MA_L(I_{81}\oplus I_{83}, I_{82} \oplus I_{84})]=$

$I_{81} \oplus MA_R(I_{81}\oplus I_{83}, I_{82} \oplus I_{84}) \oplus MA_R(I_{81}\oplus I_{83}, I_{82} \oplus I_{84})= I_{81};$

Buna uyğun olaraq alınır:

$V_{12}= I_{83}$

$V_{13}= I_{82}$

$V_{14}= I_{84}$

Beləliklə, aydınlaşdırmanın ikinci mərhələsinin çıxışı şifrələmə prosesinin axırıncıdan əvvəlki mərhələsinin girişinə, ikinci və üçüncü altblokların ardıcıllığını istisna etməklə, ekvivalentdir. Analoji olaraq, göstərmək olar ki:

$V_{81}= I_{11}$

$V_{82}= I_{12}$

$V_{83}= I_{13}$

$V_{84}= I_{14}$

Nəhayət, aydınlaşdırma prosesinin transformasiyasının çıxışının şifrələmə prosesinin birinci mərhələsinə, ikinci və üçüncü altbloklarının ardıcıllığını istisna olmaqla ekvivalent olması nəticəsində, aydınlaşdırma prosesinin çıxışının şifrələmənin girişinə ekvivalentliyi alınır.

Alqoritm QOST-28147

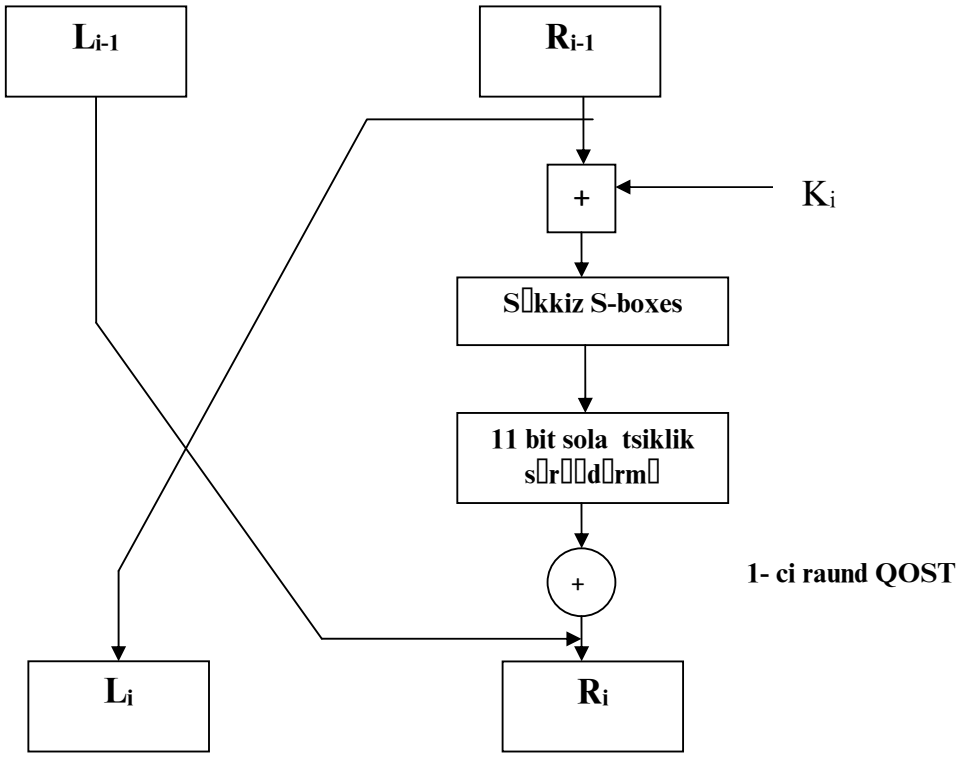
Bu alqoritm 1989-cu ildə işlənib və simmetrik şifrələmənin bloklu tipidir. Blokun uzunluğu 256-bit, açar 256-bayt, raundların sayı 32-yə bərabərdir. Feyştel şəbəkəsinin klassik formasıdır.

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

F funksiyası sadədir. Əvvəlcə sağ bölüm və i-ci altaçar 2^{32} moduluna görə cəmlənir. Sonradan nəticə 8 dörd bitlik qiymətlərə bölünür və hər biri S-box-ın girişinə ötürülür. Alqoritm S-box-dan səkkiz cür istifadə edir, hər birinin 4-bitlik çıxışı var. Bütün S-box-ların çıxışı 32-bitlik sözlə birləşirlər, bu da sonradan 11-bit sola sürüşür. Nəhayət XOR köməyi ilə nəticə sol hissə ilə birləşir və sonda yeni sağ hissə alınır.

Şəkildə QOST 28147 alqoritminin 1-ci raundu göstərilir.



Açarların generasiyası sadədir. 256-bitlik açar 8 ədəd 32-bitlik altaçarlara bölünür. Alqoritm 32 raunddan ibarət olduğuna görə, aşağıdakı cədvəldə göstərildiyi kimi, hər bir altaçar 4 raundda istifadə olunur:

raund	1	2	3	4	5	6	7	8
altaçar	1	2	3	4	5	6	7	8
raund	9	10	11	12	13	14	15	16
altaçar	1	2	3	4	5	6	7	8
raund	17	18	19	20	21	22	23	24
altaçar	1	2	3	4	5	6	7	8
raund	25	26	27	28	29	30	31	32
altaçar	8	7	6	5	4	3	2	1

Altaçarların generasiyası cədvəli.

Hesab edilir ki, QOST alqoritminin möhkəmliyi onun S-boxes strukturundadır. Müasir dövrdə RF Mərkəzi Bankında istifadə edilən S-boxes-lər məlumdur və kifayət qədər dayanıqlı hesab olunur. S-box- un giriş və çıxışı 4-bitli ədədlərdən ibarət olduğundan, hər bir S-box bəzi ardıcıllıqla düzülmüş 0 ÷ 15 kimi ədədlər sətrindən ibarət olur. Onda ədədin ardıcıllıqda S-box-un giri şinin qiyməti, ədəd özü isə – S-box-un çıxışının qiyməti olur.

DES və QOST 28147 əsas fərqləri:

1. DES-də altaçarların yaradılması prosesi mürəkkəbdir.

2. DES- də 56-bitlik açardan, QOST-da isə 256-bitlik açardan istifadə edilir. Güclü S-box seçdikdə, QOST çox güclü hesab edilir.

3. Des-in S-boxes 6-bitlik giriş və 4-bitlik çıxışı, QOST-un isə S-box, 4-bitlik girih və çıxışıvar. Alqoritmlərin hər ikisi S-box-dan istifadə edir, amma S-box QOST-un ölçüsü xeyli kiçikdir.

4. DES-də qeyri-müntəzəm yerdəyişmə P, QOST-da isə 11-bitli tsiklik sola sürüşdürmə istifadə edilir. QOST-a bir giriş bitinin dəyişməsi, bir raundun S-box-na təsir edir, bu isə növbəti raundun iki S-box-ə , bu isə sonrakı raundun 3 S-box-na təsir edir və s. QOST-a 1 giriş bitinin dəyişməsinin hər çıxış bitinin nəticəsinə təsirinin almaq üçün 8 raund tələb olunur; DES-də isə 5 raund tələb olunur.

5. Des-in 16 raundu var, QOST-un isə 32 raundu var və bu onu differensial və xətti kriptoanalizə daha dayanıqlı edir.

Simmetrik şifrələmə alqoritmlərinin iş üsulları.

Simmetrik şifrələmənin bloklu alqoritmlərinin yerinə yetirilməsi 4 üsuldan ibarətdir: ECB, CBC, CFB, OFB.

ECB – Electronic Codebook – şifrələnməmiş mətnin 64-bitdən ibarət olan hər bir bloku qalan bloklardan asılı olmayaraq, eyni şifrələmə açarından istifadə etməklə şifrələnir Tipik əlavələri: tək qiymətlərin təhlükəsiz ötürülməsi.(məs, kriptoqrafik açarın)

CBC - Chiper Block Chaiming – kriptoqrafik alqoritmin girişi, şəfrələnməmiş mətnin növbəti blokuna və şifrələnmiş mətnin əvvəlki blokuna tətbiq olunmuş XOR əməliyyatının nəticəsi olur. Tipik əlavələri: ümumi bloklu istiqamətləndirmə ilə ötürülmə, autentifikasiya.

CFB – Chipher Feedback – alqoritm hər dəfə çağrıldığı zaman girişdə j-dənə bir qiyməti emal edilir. Əvvəlki şifrələnmiş blok alqoritmə giriş kimi istifadə olunur; alqoritmin j –bitdən ibarət olan çıxışına və növbəti şifrələnməmiş j-bitdən ibarət blokuna XOR əməliyyatını tətbiq etsək, alınan nəticə j bitdən ibarət olan

şifrələnməmiş növbəti blok olur. Tipik əlavələri: axınla istiqamətləndirilmiş ötürülmə, autentifikasiya.

OFB – Output Feevback – yalnız növbəti blokun şifrələnməsi zamanı, alqoritmin girişinə əvvəlki blokun şifrələnməsinin nəticəsinin ötürülməsi müstəsna olmaqla, CFB-yə analojidir. Ancaq bundan sonra şifrələnməmiş mətnin növbəti j-bitinin üzərində XOR əməliyyatı aparılır. Tipik əlavələr: səs-küylü kanal ilə axınlı istiqamətləndirilmənin ötürülməsi (məs, peyk ötürməsi).

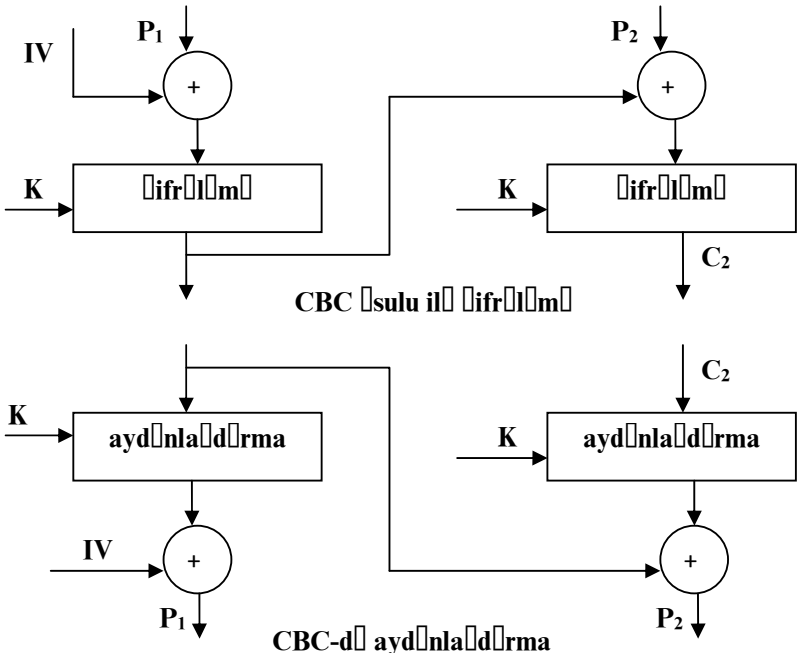
ECB üsulul

Ən sadə üsul hesab olunur – şifrələnməmiş mətn ardıcıl olaraq, blok ardınca blok emal olunur. Eyni açardan istifadə edərək hər blok şifrələnir. Əgər məlumatın uzunluğu alqoritmin blokundan uzun olarsa, onu uyğun uzunluqlu bloklara bölürlər, belə ki, lazım gələrsə, sonuncu blok fiksə olunmuş qiymətlərlə doldurulur (əlavə olunur). Bu üsuldan istifadə etdikdə eyni şifrələnməmiş bloklar eyni şifrələnmiş bloklar ilə dəyişdirilir.

ECB rejimi kiçik verilənlər üçün əlverişlidir. ECB-nin çatışmayan cəhəti ondan ibarətdir ki, məlumatda bir dəfədən çox rast gəlinən şifrələnməmiş mətnin eyni bloku şifrələnmişi həmişə eyni cür olur. Bu baxımdan ECB üsülü böyük məlumat üçün təhlükəlidir. Əgər məlumatın çox saylı eyni blokları olarsa, onda bu xüsusiyyətdən kriptoanalizdə istifadə edilə bilər.

CBC üsulu

CBC üsulunun çatışmayan cəhətini aradan qaldırmaq üçün, elə bir üsul əldə etmək lazımdır ki, eyni şifrələnməmiş blokları fərqli şifrələnmiş bloklara çevirsin. Bunun üçün alqoritmin girişində cari şifrələnməmiş bloka və əvvəlki şifrələnmiş bloka tətbiq olunmuş XOR əməliyyətının nəticəsi istifadə olunur.



Şifrələnmiş məlumatın birinci blokunun alınması üçün inisiallaşdırılmış vektordan (IV) istifadə edilir, hansı üçün XOR əməliyyatı, şifrələnməmiş məlumatın birinci bloku ilə yerinə yetirilir. Aydınlaşdırma zamanı IV üçün aydınlaşdırma alqoritminin çıxışı ilə XOR əməliyyatı yerinə yetirilir, şifrələnməmiş mətnin birinci blokunun alınması üçün, IV həm göndərənə, həm də alana məlum olmalıdır. Maksimal təhlükəsizlik üçün IV açar kimi müdafiə olunmalıdır.

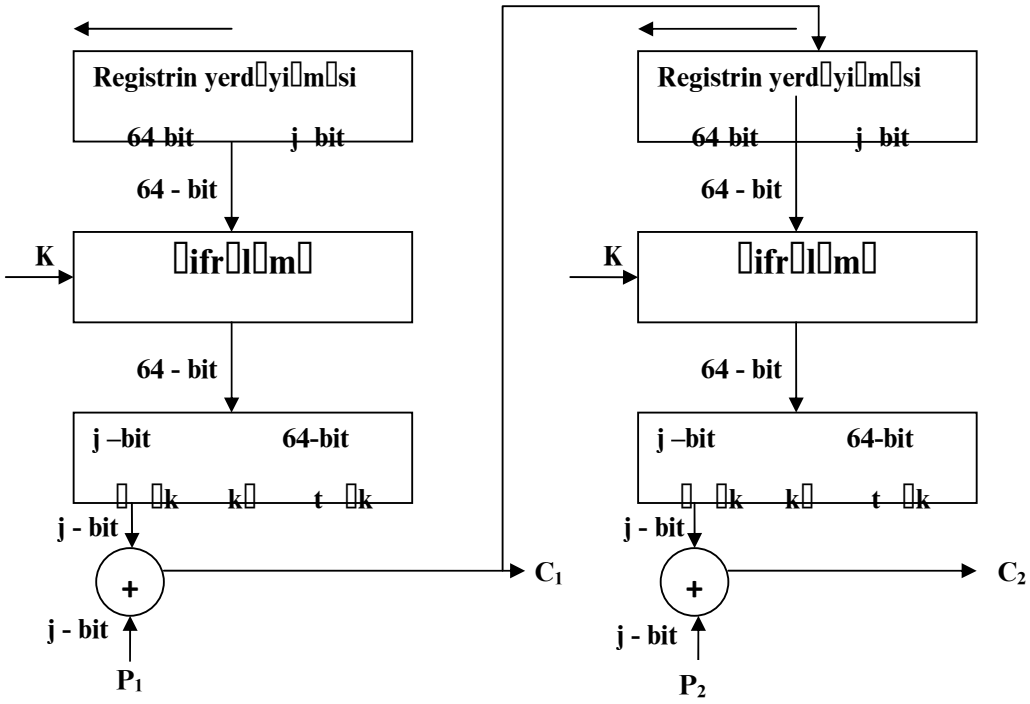
CFB üsulu

Bu bloklu alqoritm müəyyən uzunluqlu blokların şifrələnməsi üçünnəzərdə tutulub. Lakin, son iki üsuldan istifadə etməklə, bloklu alqoritmi axınlı şifrələmə alqoritminə çevirmək olar. Axınlı şifrələmə alqoritmi məlumatın, uzunluğu kifayət qədər böyük olan tam saylı bloklara bölünməsinin vacibliyini ləğv edir. Beləliklə, əgər simvollar axını ötürülürsə, hər simvol bloklu şifrələmə alqoritminin simvolu istiqamətləndirilmə üsulundan istifadə etməklə, şifrələnib həmən ötürülə bilər,

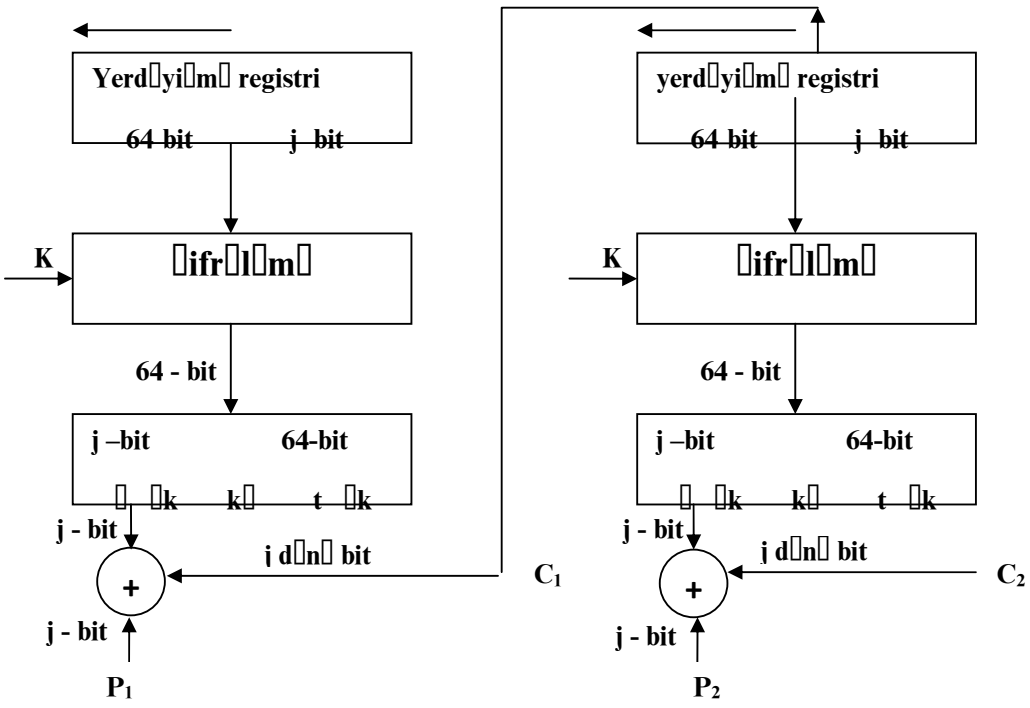
Bloklu şifrələmə alqoritminin bu üsulunun üstün cəhətlərindən biri, şifrələnmiş və verilən mətnin uzunluğunun eyni olmasıdır.

Tutaq ki, ötürülmə üçün istifadə olunan verilənlər bloku j-bitdən ibarətdir, adətən j =8. CBC üsulunda olduğu kimi, burada şifrələnmiş mətnin əvvəlki bloku üçün və şifrələnməmiş mətnin növbəti bloku üçün XOR əməliyyatı istifadə edilir. Beləliklə şifrələnmiş mətnin istənilən bloku, ondan əvvəldə yerləşən şifrələnməmiş mətndən olan funksiyadır.

Şifrələnməyə baxaq. Şifrələmə funksiyasının girişi kimi, yerdəyişmə registri olur, hansı Ki, əvvəlcədən inisiallaşdırma vektoruna IV yönəldilir. Alqoritmin çıxışının sol j-bitləri üçün, şifrələnmiş mətnin C_1 birinci blokunun alınması üçün, şifrələnmiş mətnin P_1 birinci blokunun, j-bitləri ilə XOR əməliyyatı yerinə yetirilir. Bundan başqa, registr j-bit qədər sola sürüşdürülür və C_1 bu registrin sağ j-bitinə yerləşdirilir. Bu proses bütün mətn şifrələnənə qədər davam edir.



CFB şifrələmə



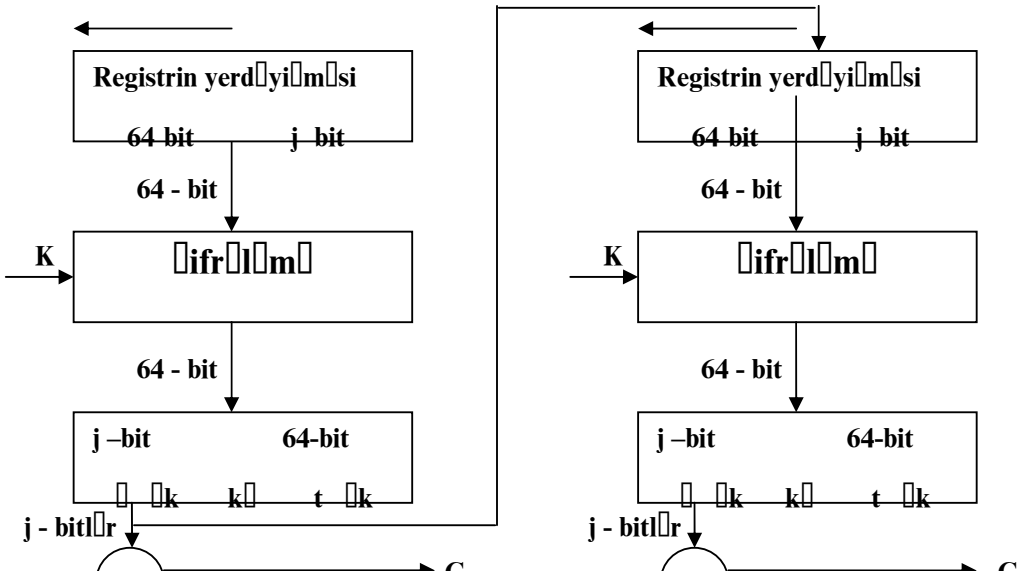
CFB şifrinin aydınlaşdırma sxemi

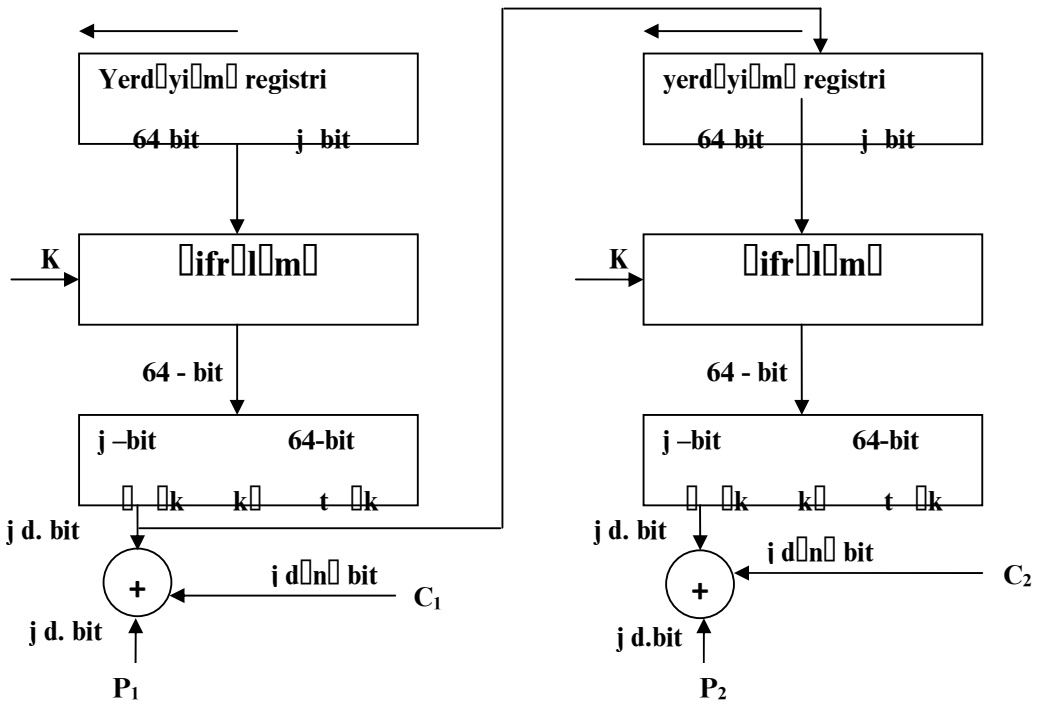
Aydınlaşdırma üçün eyni sxemdən istifadə edilir. Fərqi ondadır ki, şifrələnmiş mətni olan blok üçün alqoritmin çıxışı ilə XOR əməliyyatı yerinə yetirilir ki, nəticədə şifrələnməmiş blok alınsın.

OFB üsulu

Bu üsul CFB üsulu ilə eynidir. Fərqi ondadır ki, OFB üsulunda alqoritmin çıxışı yenidən registrə ötürülür, CFB üsulunda isə registrə ötürülən, üzərində XOR əməliyyatı yerinə yetirilən şifrələnməmiş blok və alqoritmin nəticəsi olur.

OFB üsulunun üstün cəhəti ondan ibarətdir ki, əgər ötürülmə zamanı səhvə yol verilibsə, onun növbəti şifrələnən bloklara təsiri olmur və bununla növbəti blokların aydınlaşdırılması imkanı qalır. Məs., əgər C_i-də səhf bit yaranarsa, o yalnız bu blokun aydınlaşdırılmasını və P_i alınmasına imkan vermir. Sonra gələn bloklar ardıcıllığı aydınlaşdırıla bilər. CFB üsulundan istifadə etdikdə isə, C_i giriş kimi registrə ötürülür və deməli sonra gələn axının səhfinə səbəb olur. OFB üsulunun CFB üsulu ilə müqayisədə çatışmayan cəhəti ondadır ki, məlumatın axınının modifikasiyasına hücum daha təhlükəlidir.





OFB □sulu il□ ayd□nla□d□rma

Təsadüfi ədələrin yaradılması

Şəbəkələrin təhlükəsizliyinin təmini üçün istifadə olunan əlavələrdə kriptoqrafiyadan istifadə olunmasında təsadüfi ədələrdən istifadə çox vacib rol oynayır.

Təsadüfi ədələrə olan tələblərə, onların yaradılması üsullarına baxaq.

Təsadüfi ədələrə olan tələblər

Təsadüfi ədələr ardıcıllığına olan əsas iki tələb mövcuddur: təsadüfilik və naməlumluq

Təsadüfilik – psevdotəsadüfi ədələr ardıcıllığının yaradılması zamanı güman edilir ki, verilən ədədlər ardıcıllığı müəyyən statistik mənada təsadüfi olmalıdır. Aşağıdakı iki kriteriyadan istifadə edərək sübut etmək olar ki, ədələr ardıcıllığı təsadüfidir:

1. Eyni paylaşdırılmalı: ədələr ardıcıllığının paylaşdırılması eyni olmalıdır, yəni hər bir ədədə rast gəlmə tezliyi eyni olmalıdır.

2. Qeyri-asılılıq: ardıcıllığın ədələri bir-birindən asılı olmamalıdır.

Eyniliyin sübuta yetirən testlər var amma asılılığı sübuta yetirən yoxdur.

Şifrələmənin üsulları

İnformasiya sistemlərində kompüter kriptoqrafiyasının yeri

Kompüter sistemlərinin IT-də olan təhlükəsizlik roluna diqqət yetirək.

Heç olmazsa, 3 təhlükəsizlik servisinin realizə olunması üçün kriptoqrafiya əhəmiyyət kəsb edir:

- şifrələmə;

- bütövlülüyün yoxlanması;

- autentifikasiya;

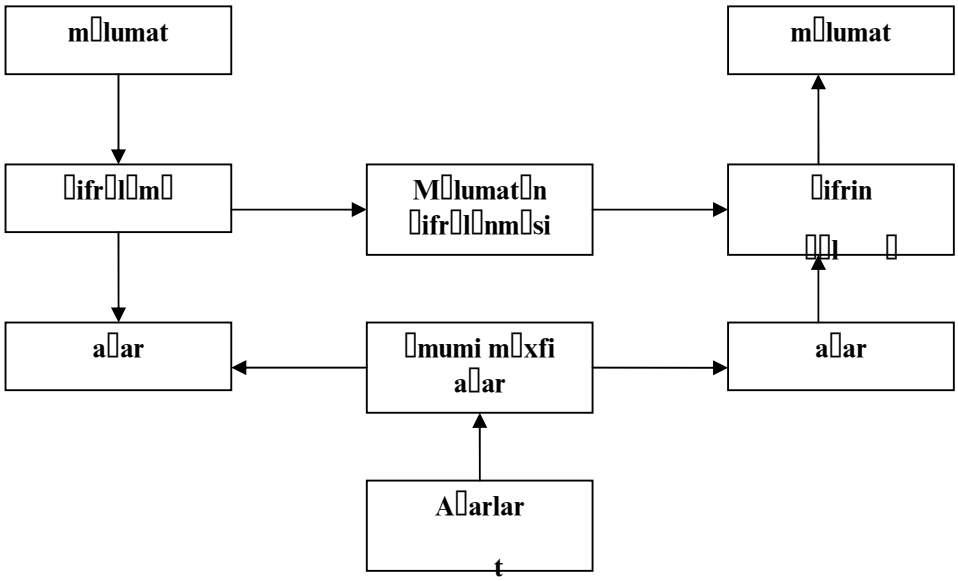
İnformasiyanın məxfiliyini təmin edən ən güclü üsul şifrələmədir.

Məs., portativ kompüterlərin oğurlanması zamanı, yalnız şifrələmə orda olan verilənlərin məxfiliyinə imkan yaradır.

Şəbəkə protokollarının steklərinin şəbəkə və nəqlolunma səviyyəsində şifrələmə və bütövlüyünün yoxlanılmasında təhlükəsizlik servislərinin tipik yeri var.

Şifrələmənin 2 əsas üsulu mövcuddur: simmetrik və asimmetrik.

I. Simmetrik



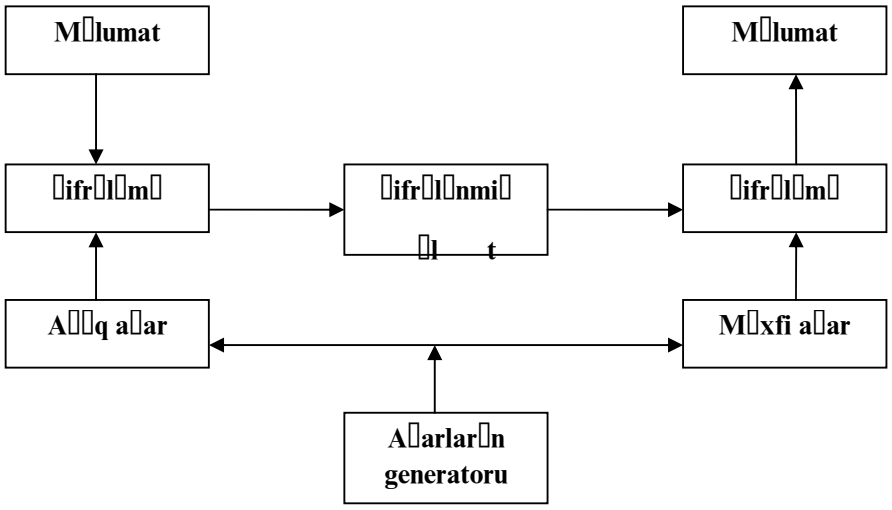
Simmetrik □ifr□l□m□ □sulundan istifad□

Simmetrik şifrələmə üsulunun çatışmayan cəhətlərindən əsası odur ki, məxfi açar, informasiyanı göndərənə (ötürənə) və informasiyanı alana məlum olmalıdır.

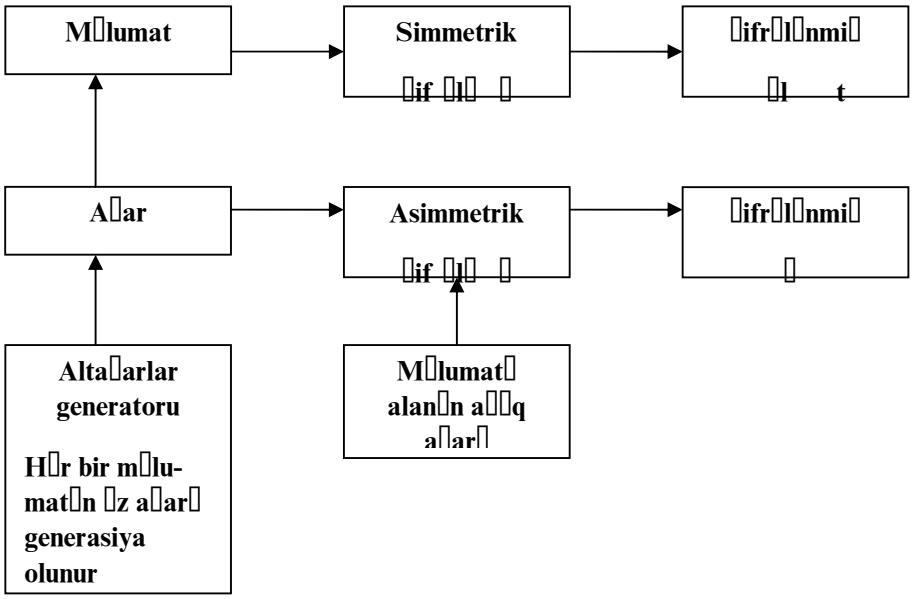
Bir tərəfdən açarın ötürülməsi problemi, digər tərəfdən informasiyanı alan (şifrələnmiş və ya şifrələnməmiş), informasiyanı konkret olaraq göndərəndən alınmasını təsdiq edə bilmir, çünki belə bir məlumatı o özü generasiya edə bilərdi.

II. Asimmetrik

Şifrələmənin asimmetrik üsulunda 2 açardan istifadə olunur. Bu açarlardan biri məxfi deyil, şifrələmə üçün istifadə edilir (və başqa məlumutlarla birgə elan edilə bilər) və ikinci (məxfi açarın) açar şifrin açılması üçün istifadə edilir. Məxfi açarın şifri ancaq informasiyanı alana məlum olur, onun köməyi ilə informasiyanı açsın. Asimmetrik üsulların ən populyarı - RCA (Rayverst, Samir, Adleman) , çox saylı sadə (100) rəqəmləri və onların hasili ilə əməliyyatlara əsaslanır.

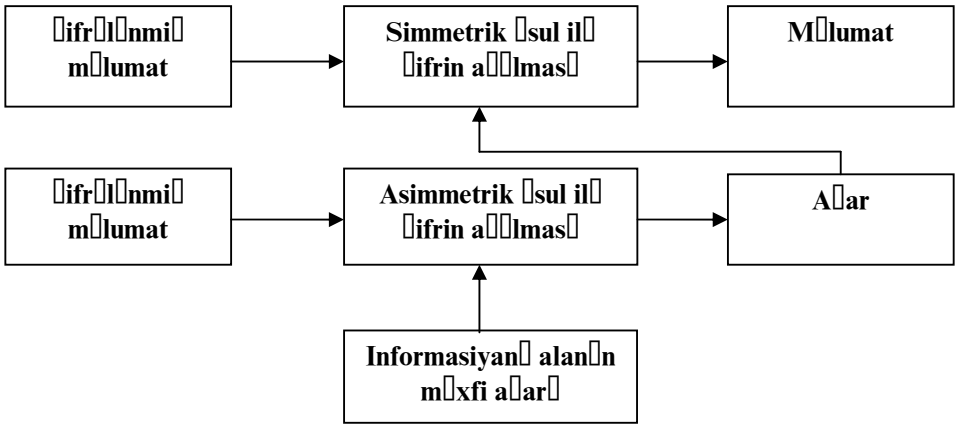


Asimmetrik üsulun çatışmayan cəhəti – tezliyin aşağı olmasıdır(3-4 dəfə). Bu səbəbdən bu iki metodun birləşməsi tövsiyyə edilir, bu baxımdan effektivliyin artırılması üçün məlumat əvvəlcədən təsadüfi şifrələnir, soradan məlumatı alanın açıq asimmetrik açarı ilə bu açar şifrələnir, bundan sonra məlumat və açar şəbəkə ilə ötürülür.



Simmetrik v□ asimmetrik □sullar□n birg□ realiz□ olunmas□ il□ m□lumat□n effektiv □ifr□l□nm□si sxemi

Aşağıda effektiv şifrələmənin şifrinin açılması göstərilir.



Şifrələnmiş məlumatın effektiv açılması

Assimetrik üsul – alqoritm Diffi-xalnan

Bütövlüyün yoxlanılması

Bütövlüyün yoxlanılmasının kriptoqrafik əsasını 2 anlayış təşkil edir:

- xeş funksiya;

- elektron-rəqəm imza (ERI).

Xeş funksiya – çətinliklə verilənlərin əks dönməsi mümkün olan funksiyadır. Blokların simmetrik şifrələnməsi ilə realizə olunur. Sonuncu nəticə (bir-birindən asılı olan) sonuncu blokun (əvvəlkilərdən asılıdır) şifrələnməsi ilə alınır və xeş funksiya adlanır.

Verilənlərin bütövlüyünü yoxlamaq üçün (xeş funksiyanın verilənlərə tətbiqi – daycest adlanır) xeş funksiyanı – h, verilənləri – E, yoxlanılan verilənləri isə – T^1 işarə edək . verilənlərin bütövlüyü üçün: $h(T^1) = h(T)$ bərabərliyi ödənilməlidir. Əgər bu bərabərlik ödənilirsə , deməli $T^1 = T$.

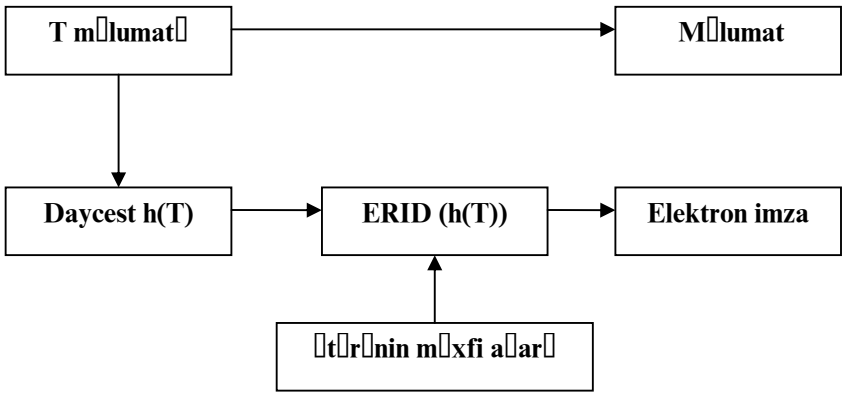
İndi isə ERI əldə edilməsi və yoxlanılması üçün asimmetrik şifrələmədən istifadə edək. Açıq açar ilə şifrələnmiş T mətnini E(T), məxfi açar ilə şifri açılmış T mətni D(T) işarə edək.

ERI üçün asimmetrik üsullardan istifadə edilməsi üçün:

$E(D(T)) = D(E(T)) = T$ bərabərlik ödənilməlidir.

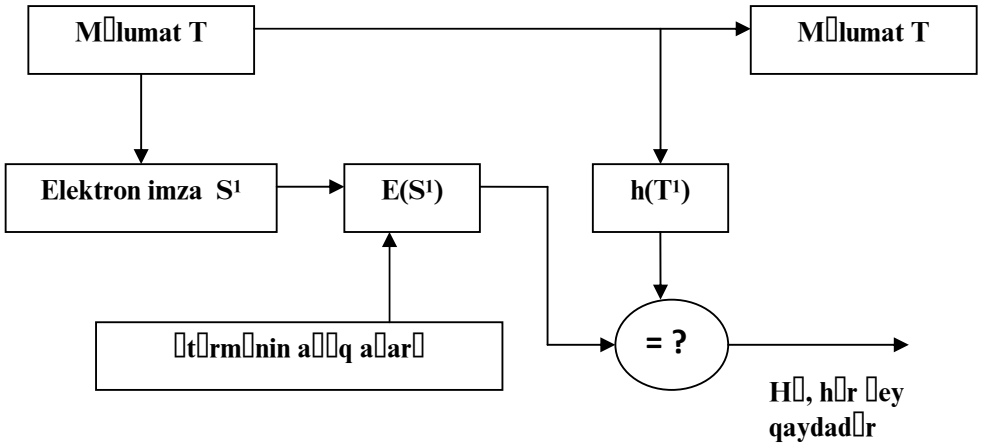
E əməliyyatı ilə - açıq açar ilə şifrələmə,

D əməliyyatı ilə – məxfi açar ilə şifrin açılması



Elektron imzan□n □ld□ edilm□si

Elektron imzanın əldə edilməsi daycestin h(T) D əməliyyatı ilə şifrələnməsi nəticəsində alınır.



Elektron imzan□n yoxlan□lmas□

Bu bərabərlikdən: $E(S^1) = h(T^1)$, $S^1 = D(h(T^1))$ alınır. (buna D tətbiq edirik).

Beləliklə elektron rəqəmli imza məlumatın bütövlüyünü müdafiə edir və məlumatı göndərəni təsdiq edir.

İdentifikasiya və autentifikasiya

Əsas anlayışlar

I və A təhlükəsizlik servislərinin ən əsasıdır, müdafiə xəttinin ən birincisidir (keçid xəttidir).

İdentifikasiya – subyektə (istifadəçiyə, prosesə, apparat-proqram komponentinə) özünü (adını) təqdim eməyə icazə deməkdir.

Autentifikasiyadan istifadə etməklə ikinci tərəfi – subyektin özü olmasında arxayın edir. Autentifikasiya – bir tərəfli və iki tərəfli ola bilər.

Şəbəkə daxilində bu servislərin 2 əsas aspekti mövcuddur:

1. Autentifikasiya(yəni subyektin təsdiqlənməsi) üçün nədən istifadə olunur;
2. Necə təşkil olunur (müdafiə olunur)

Verilənlərin identifikasiya/autentifikasiyası

Subyektin təsdiqlənməsi üçün aşağıdakılar vacibdir:

- onun hər hansı bir bildiyi şey (parol, şəxsi identifikasiya nömrəsi, kriptoqrafiya açarı);

- onda olan hər hansı əşya(şəxsi kartı və ya s.)

- onun özünün bir hissəsini təşkil edən əlamət(səsi, barmaq izləri, yəni biometrik xarakteristikaları).

Açıq şəbəkə daxilində autentifikasiya/identifikasiya tərəfləri arasında informasiyanın mühafizəsi lazım gəlir: şəbəkənin aktiv və passiv qulaq asılmasından, protokolların açıq ötürülməsindən, hətta protokolların şifrələnməsində kömək etmir. Autentifikasiyanın daha mürəkkəb protokolu tələb olunur.

Şəbəkə hücumlarından başqa, identifikasiyanı və autentifikasiyanı çətinləşdirən bir sıra səbəb var. Bunlardan:

1. hər şey oğurlana, dəyişdirilə bilər;

2. autentifikasiyanın keyfiyyətliliyi – istifadəçi və sistem administratoru üçün narahatlıq doğurar;

3. müdafiənin yüksək olması onun bahalanmasına gətirib çıxarır.

Qeyd etmək lazımdır ki, autentifikasiya/identifikasiya servisi əldə edilmənin hücum obyektinə çevrilə bilər.

Əgər sistem elə qurulsa ki, bir neçə uğursuz müraciətdən sonra, identifikasiya informasiyasını daxil edən qurğu blokirovka olsun, ziyan verən bundan istifadə edərək, leqal istifadəçinin işini, klaviaturanın düymələrini bir neçə dəfə sıxmaqla dayandıra bilər.

Parollu autentifikasiya

Parollu autentifikasiyanın əsas etibarı (xüsusiyyəti) – onun sadə və adət olunmasıdır. Əməliyyat sistemləri və başqa servislərdə parol sistemi quraşdırılıb. Parollardan düzgün istifadə – bir çox təşkilatlar üçün təhlükəsizliyin təmin

edilməsi deməkdir. Buna baxmayaraq, ümumi xarakteristikalara görə, bu üsul daxil olmanın yoxlanılmasının ən zəifi hesab olunur.

Parolun yadda qalan olması üçün (yolaşın adı, sevimli komandanın adı və s.) onu sadələşdirirlər. Bu, parolun açılışını sadələşdirir. Məs., klassik bir misal: Rixard Zorge (sovet kəşfiyyatçısı), obyektin hər bir neçə sözdən sonra «Karamba» sözündən istifadə etməsi əsasında, yüksək məxfi seyfin açılmasına nail olub.

Bəzi hallarda parollar məxfi saxlanılmır, standart olaraq sənədlərdə göstərilir və sistem quraşdırılmasından xeyli vaxt keçdikd belə dəyişdirilmir.

Parollara daxil olunan zaman gizli baxmaq mümkündür (bunun üçün hətta optik qurğulardan da istifadə olunur) (parolun yoldaşa deyilməsi və s. .)

Buna baxmayaraq, aşağıdakı tədbirlərin görülməsi, parollu müdafiənin etibarını yüksəldir:

1) texniki məhdudiyyətlərin qoyulması (parol qısa olmamalıdır, hərf, rəqəm və işarədən ibarət olmalıdır və s.);

2) periodik olaraq dəyişilməlidir(parolların istifadə müddəti idarə olunmalıdır və s.);

3) parollar faylına daxil olma məhdudlaşdırılmalıdır;

4) sistemə müvəffəqiyyətsiz daxilolmaların sayının azaldılması;

5) istifadəçilərin öyrənilməsi;

6) proqramm-parol generatorlarında istifadə olunması.

Parollardan başqa, digər autentifikasiya üsullarından istifadə olunmasına baxmayaraq, bu tədbirlərin görülməsi həmişə lazımdır.

Birdəfəli parollar

Parolların daha qüvvətlisi – brdəfəli parollardır. Belə parolların ən qüvvətlisi «Bellcore» kampaniyasının S/KEY sistemi sayılır. Sistemin ideyası aşağıdakılardan ibarətdir:

f – bircinsli funksiya verilib (yəni qısa müddətdə əks hesablanması mümkün deyil). Bu funksiya istifadəçiyə və autentifikasiya serverinə də məlumdur.

K – yalnız istifadəçiyə məlum olan açardır. Ilk etapda f funksiyası K açarına n-dəfə tətbiq olunur, nəticəsi isə serverdə yadda saxlanır.

Bundan sonra istifadəçinin təsdiqlənməsinin yoxlanılması proseduru belə olur:

1. Istifadəçinin sisteminə server (n-1) yollayır;

2. istifadəçi f funksiyasını K açarına (n-1) dəfə tətbiq etdikdə, nəticəni şəbəkə ilə autkntifikasiya serverinə ötürür;

3. server f funksiyasını istifadəçidən alınan nəticəyə tətbiq etdikdən sonra, onu serverdə əvvəlcədən yadda saxlanmış ölçü ilə müqayisə edir. Eyni olduqda istifadəçinin özü olduğu təsdiqlənir, server istifadəçinin yeni göndərdiyini yadda yadda saxlayır və sayqacı (n) vahid qədər azaldır.

f – funksiyasının əks hesablanması mümkün olmadığından, parolun əldə edilməsi və autentifikasiya serverinə müdaxilə, K məxfi açarın və növbəti birdəfəli parolun əldə edilməsi mümkün deyil.

S/KEY sistemi – internet – standart (RES 1938) sğatusu əldə edib.

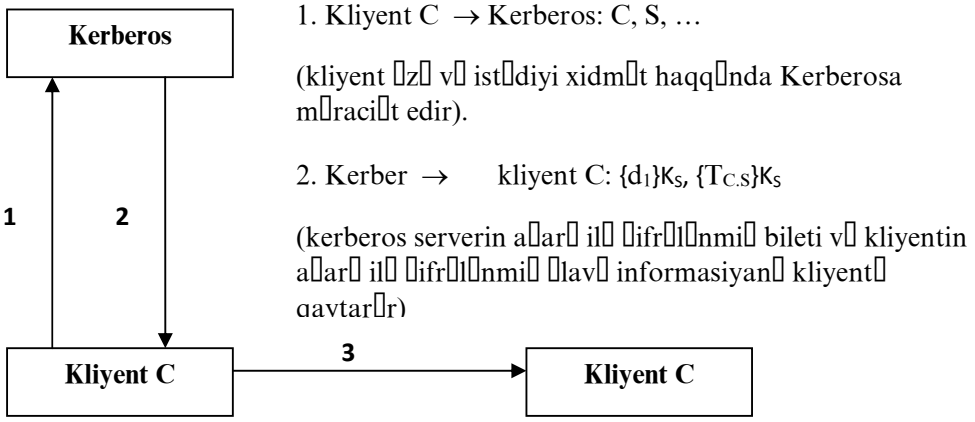
«Kerberos» autentifikasiya serveri

Kerberos proqrammı 1980-ci illərin ortalarında Masaçuset texnoloжi institutunda işlənib hazırlanıb və o zamandan çox prinsipial dəyişikliklərə məruz qalıb. Muasir dövrün əməliyyat sistemlərinin çoxunda Kerberos koiyent komponentlərinin yeri var.

Kerberos aşağıdakı məsələlərin həlli üçün yararlıdır. Açıq şəbəkənin (müdafiə olunmayan) düyünlərində subyektlər-istifadəçilər, həmçinin kliyent-server proqramm sistemləri yerləşdirilib. Hər subyektin məxfi açarı var. Hər bir C-subyekt özünün doğruluğunu S-subyektə təsdiqləmək üçün (bu olmasa S subyektiS subyektinə xidmət göstərməyəcək), o nəinki özünü açıqlamalıdır, həmçinin məxfi açarın bilməsini göstərməlidir. C məxfi açarı S-ə elə-belə göndərə bilməz, çünki: birincisi şəbəkə açıqdır, ikinci s-nin məxfi açarını bilmir(heç bilməməlidir də). Məxfi açarın bilməsinin göstərmək üçün gizli üsul tələb olunur. Kerberos sistemi inanılan (etibarlı) 3-cü tərəf kimi çıxış edir. Məxfi açarın sahibkarı olaraq subyektləri idarə edir və onlara özlərini bir-birinə təsdiq etməkdə kömək edir.

Kerberosun köməkliyi ilə S serveri əldə etmək üçün, C(kliyent) Kerberosa mürasiət edib, özü haqda və istədiyi xidmət haqqında sorğu göndərir. Buna cavab olaraq, Kerberos, serverin məxfi açarı ilə şifrələnmiş biletdə olan informasiyanın bir hissəsinin surətini (kopiyasını), kliyent verilənlərin ikinci hissəsinin şifrini açıb, bilet ilə bir yerdə serverə ötürməlidir. Server, biletin şifrini açıb, onun məzmununu

kliyentin göndərdiyi əlavə informasiyanın məzmunu ilə müqayisə etməlidir. Onların eyniliyi göstərir ki, kliyent verilənlərin şifrini açа bildi (server və Kerberos-dan başqa onlar heç kimə məlum deyildi), məxfi açarın bilməsini göstərdi. Qeyd etmək lazımdır ki, məxfi açar yoxlama zamanı şəbəkədə ötürülmür (hətta şifrələnmiş vəziyyətdə) – yalnız şifrələmə üçün istifadə olunmalıdır.



S – serverin – kliyentin özlülüyünün yoxlanılması sxemi

Burada C və S kliyent və server varəsində məlumatdır (məs. ad);

d_1 və d_2 – əlavə əlavə informasiyadır(biletə görə);

$T_{C.S}$ – S serverdən istifadə üçün C kliyentin biletidir;

K_C və K_S – uyğun olaraq kliyent və serverin məxfi açarları;

{info}K – K məxfi açar ilə şifrələnmiş info informasiyası;

Bu prosedurun sadələşdirilmiş sxemidir.

AÇIQ AÇARLI KRIPTOQRAFİYA.

Assimetrik şifrələmə alqoritmlərinə olan əsas tələblər.

Assimetrik şifrələmə alqoritmlərin yaradılması kriptoqrafiyanın tarixində yeganə və vacib nailiyyətlərindən biridir.

Açıq açarlı şifrləmə alqoritmlərinin yaradılması, çətin məsələlərin həlli zamanı, simmetrik şifrələmədən istifadə edərkən, iki çətin məsələnin həlli üçündür.

1-ci məsələ - açarın paylanmasıdır. Simmetrik şifrələmədə iki tərəfə ümumi açarın qabaqcadan hansı bir üsulla olursa olsun məlum olması tələb olunur. Açıq açarlı şifrələmənin əsasını qoyan Diffi, bu tələbin kriptoqrafiyanın əsasına, yəni kommunikasiyaların məxfiliyinə zidd olduğunu qeyd edir.

2-ci məsələ - rəqəmli imzanın lazım olduğunu, yəni elə bir mexanizmin yaradılması lazımdır ki, heç bir iştirakçının əvəz edilməsi mümkün olmasın. Elə bir üsulun yaradılması vacibdir ki, ondan istifadə edərkən bütün iştirakçılar arxayın olmalıdır ki, elektron məlumat konkret iştirakçı tərəfindən ötürülüb. Bu tələb, autentifikasiyadan fərqli olaraq daha güclü tələbdir.

Diffi və Kelman iki məsələnin həllində yüksək nəticələrə nail oldular.

Əvvəlcədən, açıq açarlı şifrələmənin ümumi xüsusiyyətlərinə və bu alqoritmlərinə olan tələblərə baxaq.

Bir açarı şifrləmək üçün o birini – aydınlaşdırma üçün istifadə edən alqoritmin tələblərini açıqlayaq və qeyd edək ki, şifrləmə alqoritmi və şifrləmə açarı məlum olduqda belə, aydınlaşdırma açarını hesablamaq mümkün deyil.

Qeyd etmək lazımdır ki, bəzi alqoritmlərdə, məs. RSA, hər iki açardan biri həm şifrələmə, həm də aydınlaşdırma üçün istifadə oluna bilər.

Əvvəlcə iki xüsusiyyətin ikisinə də malik olan alqoritmlərə baxaq, sonra isə ikinci xüsusiyyətə malik olmayan açıq açarlı alqoritmlərə keçərik.

Bəzi terminlərlə tanış olaq.

Simmetrik şifrələmədə istifadə olunan açarı –**məxfi açar** adlandıraq. Açıq açarlı şifrələmədə istifadə olunan iki açarı ÷açıq açar və bağlı açar adlandıraq. Simmetrik şifrələmədə istifadə olunan açar ilə qarışdırmamaq üçün, məxfi saxlanan açarı –məxfi açar yox, bağlı açar adlandırırıq. Bağlı açar- KR, açıq açar isə – KV kimi işarə edək.

Fərz edək Ki, açıq açar iştirakçılar üçün əlçatandır, amma bağlı açar hər iştirakçı tərəfindən lokal yaradılır və paylanan deyil.

Istənilən vaxt iştirakçı öz bağlı açarını dəyişdirə bilər və ona uyğun açıq açarı elan edə bilər, yəni onunla köhnə açarı əvəz edə bilər.

Diffi və Xelman açıq açarlı şifrələmə alqoritminin hansı tələbləri ödəməli olduğunu göstərir.

1.Açarların hesablanması sadə olmalıdır (açıq açar KV, bağlı açar KR).

2.Açıq açar və şifrələnməmiş məlumat olduqda, şifrələnmiş mətnin yaradılması asan olmalıdır:

$$C=E_{KV}[M]$$

3.Bağlı açar olduqda, məlumatın aydınlaşdırılması sadə olmalıdır:

$$M=D_{KR}[C] = D_{KR}[E_{KU}[M]]$$

4.Açıq açarı KU bildikdə, bağlı açarın KR tapılması mümkün olmamalıdır.

5.Açıq açarı KU və şifrlənmiş məlumatı C bilərək, məlumatın M əldə edilməsi mümkün olmamalıdır.

Altıncı tələb, ancaq bəzi açıq açarlı alqoritmlər üçün yerinə yetirilir:

6.Şifrləmə və aydınlaşdırma funksiyaları istənilən ardıcıllıqda qəbul oluna bilər.

$M=E_{KU}[D_{KR}[M]]$

Bu kifayətcə güclü olan tələblər, birtərəfli funksiya anlayışını daxil edir. Hər bir arqument üçün bir dənə əks qiyməti olan funksiya birtərəfli adlanır. Belə funksiyanın hesablanması sadə, amma əksinin hesablanması çətin olur.

$y=f(x)$ – asandır

$x=f^{-1}(y)$ – çətindir.

«Asan» o deməkdir ki, problemin həlli girişin uzunluğunun polinominal vaxtı ərzində həll oluna bilər.

Beləliklə, əgər girişin uzunluğu n-bitdirsə, funksiyanın hesablanma vaxtı n^a proporsionaldır, burada a – fiksə olunmuş sabitdir. Beləliklə, deyirlər ki, alqoritm P – poliminal alqoritmlər sinfinə daxildir.

«Çətin»dir termini isə daha mürəkkəblik anlayışı daşıyır, yəni problem həll olunmazdır, əgər onun həlli üçün girişin uzunluğundan asılı olan polinominal vaxtdan böyük güc tələb olunur. Məs., əgər giriş uzunluğu n bitdir və funksiyanın hesablama vaxtı 2^n proporsionaldır, onda məsələ hesablama nöqteyi nəzərdən mümkün olmaz məsələdir.

Birtərəfli funksiyanın bir istiqamətdə hesablanması asandır, əks istiqamətdə hesablanması isə, əlavə informasiya olmazsa, çətindir. Belə bir əlavə informasiya olduqda inversiyanın hesablanması polinominal vaxt ərzində mümkündür. Beləliklə, birtərəfli funksiya quyulu (s lyukom) adlanır elə birtərəfli funksiyalar ailəsinə f_k məxsusdur ki, onlar üçün

$y=f_k(x)$ – əgər k və x məlumdur, asan yerinə yetirilir,

$x= f_k^{-1}(y)$ – əgər k və y məlumdur, asan yerinə yetirilir,

$x= f_k^{-1}(y)$ – əgər y məlumdur, amma K-məlum deyil, çətindir.

Görünür ki, konkret açıq açarlı alqoritmin işlənilməsi uyğun quyulu birtərəfli funksiyanın açılışından asılıdır.

Açıq açarlı alqoritmlərin kriptoanalizi.

Simmetrik şifrləmə alqoritmlərində olduğu kimi açıq açarlı şifrləmədə üzbəüz hücumlar üçün təhlükəlidir. Buna, standart olaraq, əks tədbir – böyük açarlardan istifadədir.

Açıq açarlı kriptosistemdə inversiya olunmayan riyazi funksiyalardan istifadə edilir. Belə funksiyaların hesablanmasının mürəkkəbliyi, açarın uzunluğundan xətti asılılıq olmadığından, daha tez artır. Beləliklə, açarın uzunluğu lazımınca uzun olmalıdır ki, hücumu əlverişsiz etsin və kifayətcə kiçik olmalıdır ki, şifrləmə praktiki olaraq mümkün olsun. Təcrübədə açarın uzunluğu hücumu əlverişsiz edən götürülür, amma nəticədə alqoritmin ümumi məqsəd üçün istifadəsində, şifrləmə sürəti kifayətcə aşağı olur. Bu səbəbdən müasir dövrdə təcrübədə açıq açarlı şifrləmə açarlı idarə olunma əlavələri və imza ilə məhdudlaşdırılır. Bu əlavələrdə kiçik verilənlər blokunun şifrlənməsi tələb olunur.

Hücumun başqa bir forması – açıq açarı bilərək, bağlı açarın hesablama üsulunun tapılmasından ibarətdir. Riyazi sübut etmək mümkün deyil ki, belə bir hücum formasını konkret açıq açarlı alqoritm üçün istisna olunur. Beləliklə, bütün alqoritmlər RSA daxil olmaqla şübhəlidir.

Nəhayət, açıq açarlı sistemlər üçün spesifik olan hücum forması mövcuddur. Bu, ehtimal olunan məlumatın hücumudur. Məsələn, fərz edək ki, ötürülən məlumat 56-bitli açardan ibarətdir (simmntrik şifrələmə alqoritmi üçün). Rəqib (düşmən) açıq açardan istifadə edərək, bütün açarları şifrələyə bilər və ötürülən şifrələnmiş mətnə uyğün olan istənilən məlumatı aydınlaşdıra bilər. Beləliklə, açarın ölçüsündən asılı olmayaraq, açıq açarlı sxemin 56 bitli simmetrik açara olan hücumla məhdudlaşır. Belə hücumdan müdafiə, təsadüfi bitlərin əlavə olunmasından ibarət olur

Açıq açarlı alqoritmlərin əsas istifadə üsulları.

Açıq açarlı alqoritmlərin əsas istifadə olma üsulları bunlardır: şifrələmə/aydınlaşdırma, imzanın yaradılması və yoxlanılması, açarların dəyişdirilməsi.

Açıq açarlı şifrələmə açıq açarlı şifrələmə aşağıdakı addımlardan ibarətdir:

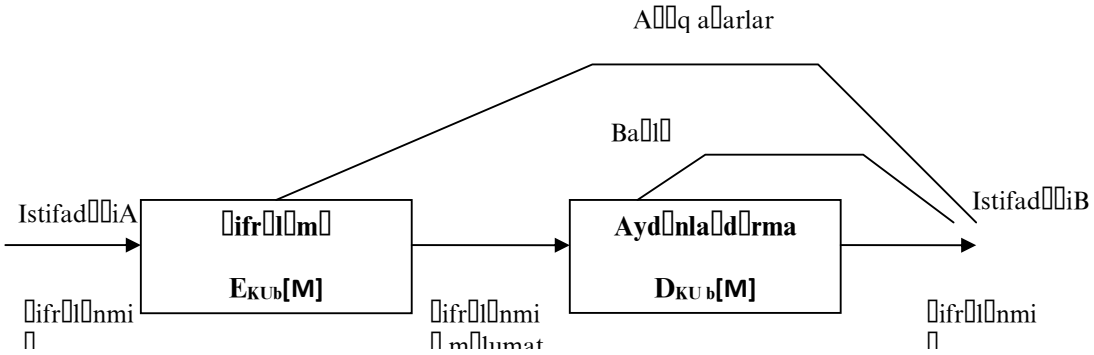
1. B istifadəçi, ötürülən məlumatın şifrələnməsi üçün və aydınlaşdırılması üçün istifadə olunan açarlar cütlüyü KU_b və yaradır.

2. B istifadəçi öz şifrələmə açarını, yəni KU_b açıq açarı, əlçatan edir. Onun cütlüyü olan bağlı açar KR_b bağlı saxlanır.

3. Əgər A, məlumatı B-yə göndərmək istəyirsə, o, B-nin açıq açarından KU_b istifadə edərək məlumatı şifrələyir.

4. B məlumatı aldıqda öz bağlı açarından istifadə edərək məlumatı aydınlaşdıra bilməz, çünki bağlı açarı ancaq B bilir.

Əgər istifadəçi öz bağlı açarını etibarlı saxlayırsa, heç kim ötürülə məlumatı əldə elə bilməz.



A□□q a□arlı □ifr□ı□m□

Imzanın yaradılması və yoxlanılması addımlarının ardıcıllığı aşağıdakı kimimdir:

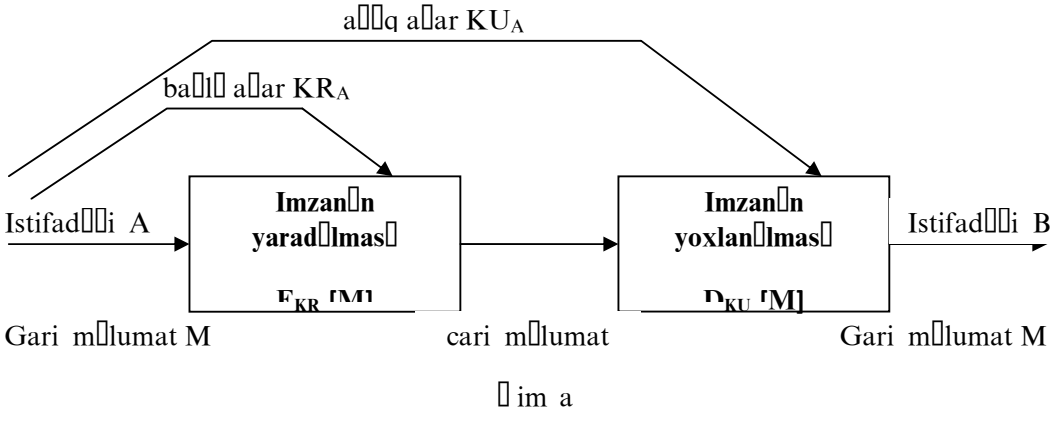
1. A istifadəçi tərəfəindən açarlar cütlüyü KR_A və KU_A yaradılır və ötürülən məlumatların imzalarının yaradılması və yoxlanılması üçün istifadə olunur.

2. A istifadəçi öz yoxlama (yəni açıq açarı KU_A) açarını hansı üsulla əlçatan edir. Onun cütlüyü olan açar KR_A məxfi saxlanır.

3. əgər A imzalanmış məlumatı B-yə göndərmək istəyərsə, onda o öz bağlı açarı ilə bu məlumat üçün imza $E_{KRa}[M]$ yaratmalıdır.

4. imzalanmış məlumatı B aldıqda, A-nın açıq açarının KU_A istifadəsi ilə imzanı $D_{KUa}[M]$ yoxlayır. Heç kəs məlumatı imzalaya bilməz, çünki bağlı açar ancaq A-ya məlumdur.

Istifadəçi və ya sistem bağlı açarı etibarlı saxladığı zamana kimi, imza etibarlı hesab olunur. Bundan başqa, A-nın bağlı açarının əldə edilməsi mümkün olmadığından, məlumatın dəyişdirilməsi mümkün deyil, yəni verilənlərin bütövlüyü və autentifikasiya əldə edilir.



Imzan□n yarad□lmas□ v□ yoxlanmas□

Bu sxemdə bütün məlumat imzalanır, məlumatın bütövlüyünün təsdiqlənməsi böyük yaddaş tələb edir. Hər məlumatı təcrübədə istifadə etmək üçün, o şifrələnməmiş saxlanılmalıdır. Bundan başqa, məlumatın surəti də (kopiyası) şifrələnmiş saxlanılmalıdır ki, lazım gəldikdə imza yoxlanılsın. Daha effektiv üsul – kiçik bloklarla şifrələmədir. Autentifikator adlanan belə blokun – autentifikator dəyişmədən məlumatın dəyişməsi mümkün olmamalıdır – xüsusiyyəti olmalıdır. Mətni göndərənin bağlı açarı ilə şifrələnmiş autentifikator – rəqəmli imzadır. Onun köməyi ilə cari mətni yoxlamaq olar. Sonradan bu texnologiya detalları ilə araşdırılacaq.

Rəqəm imzanın yaradılması prosesi məxfiliyi təmin etmir. Bu o deməkdir ki, məlumatın dəyişdirilməsi mümkün deyil, amma əldə edilməsi mümkündür.

Açarların dəyişdirilməsi: açarın dəyişdirilməsi üçün iki mübadilə tərəfi iştirak edir və sonradan açar simmetrik şifrələmə alqoritmində istifadə ediləcək. Bəzi alqoritmlər üç üsulla, bəziləri isə bir və iki üsulla işlənilə bilər. Ən geniş yayılan açıq açarın alqoritmlərə və onların istifadə üsullarına baxaq.

RSA alqoritmi

Deffi və Xelman şifrələməyə yeni bir yanaşma növü – açıq açarlı şifrələmə yaratdılar. Birinci nəticə olaraq 1977-ci ildə Ron Rivers, Adi Şamir və Len Adelman yeni alqoritmi yaradıldı və 1978-ci ildə nəticə nəşr olundu. O vaxtdan RSA alqoritmi təcrübədə geniş tətbiq olunub.

RSA alqoritmi – bloku şifrələmədən ibarətdir. Burada şəfrələnmiş və şifrələnməmiş verilənlər, verilmiş n üçün, 0 və n-1 arasında olani tam ədələr olur.

Alqoritmin təsviri

Alqoritmdə eksponentdən istifadə olunur. Verilənlər bloklarla şifrələnir. Hər bir bloka verilmiş n-dən kiçik olan ədəd kimi baxılır. Şifrələmə və aydınlaşdırma şifrələnməmiş M və şifrələnmiş C bloklar üçün belədir:

$C = M^e \pmod n$

$M = C^d \pmod n = (M^e)^d \pmod n = M^{ed} \pmod n$

Məlumatı ötürən, həm də alan N qiymətini bilməlidir. Məlumatı ötürən e qiymətini, alan d qiymətini bilir. Beləliklə,

açıq açar – KU = { e,n} və bağlı açar – KR = { d,n}

Bununla bərabər aşağıdakı şərtlər ödənilməlidir:

1. Elə c,d,və n tapmaq olar ki, $M^{ed} = M \bmod n$ bütün M>n.

2. Bütün M<n üçün M^e və C^d nisbi olaraq yüngül hesablanmaı

3. c və n məlum olduqda d-nin tapılmasının mümkün olmaması

Əvvəlcə birinci şərtə nəzər salaq. Bizə $M^{ed} = M \pmod n$ bərabərliyinin yerinə yetirilməsi vacibdir.

e, d və n tapılması üçün bəzi riyazi anlayışlara, xüsusiyyətlərə və teoremlərə baxaq.

1. Əgər $(a \bullet b) \equiv (a \bullet c) \bmod n$, onda $b \equiv c \bmod n$, əgər a və n qarşılıqlı sadə istifadəçilərdisə, yəni $g \in d\ (a,n)=1$.

2. P-dən kiçik olan və bütün qarşılıqlı sadə ədədləri Z_p – işarə edək. Əgər p-sadədirsə, onda Z_p – bütün qalıqlardır.

W^{-1} – elə bir ədədi işarə edək ki, $W \bullet W^{-1} \equiv 1 \bmod p$

Onda $\forall\ W \in Z_p \quad \exists z : w \bullet z \equiv 1 \bmod p$

Bunun sübutu ondan irəli gəlir ki, w və p – sadə olduqlarından, bütün Z_p elementləri W vurulduqda, qalıqlar bütün Z_p elementləri olacaq, ola bilsin ki, yerləri dəyişdirilmiş olsun. Beləliklə, qalıqlardan biri 1 bərabər olacaq.

3. Eyler funksiyasını belə müəyyən edək:

n-dən kiçik və n-lə qarşılıqlı sadə olan müsbət elementlərin sayını $\varphi(n)$ kimi işarə edək. Əgər p-sadədirsə, onda $\varphi(p)=p-1$

Əgər p və q-sadədirlər, onda $\varphi(p \bullet q)=(p-1) \bullet (q-1)$

O zaman : $Z_{p \bullet q}=(0,1,\ldots,(p \bullet q-1))$.

$(p \bullet q)$-ilə qarşılıqlı sadə olmayan qalıqları sadalayaraq:

$\{p, 2 \bullet p, \ldots, (q-1) \bullet p\}$

$\{q, 2 \bullet q, \ldots, (p-1) \bullet q\}$

Beləliklə $\varphi(p \bullet q)=p \bullet q-[(q-1)+(p-1)+1]$

$$=p \bullet q-(p+q)+1=(p-1) \bullet (q-1).$$

4. Forma teoremi

$a^{n-1} \equiv 1 \bmod n$, əgər n-sadədir.

Əgər bütün Z_n elementlərini n moduluna görə a-ya vursaq, nəticədə Z_n-bütün elementləri alınacaq, ola bilər ki ayrı ardıcıllıqda olsun. Növbəti ədədlərə baxaq.

Bu {a mod n, 2•a mod n,…,(n-1)•a mod n} ədədlər bu {1,2,…,(n-1)} ədədlərdir, ola bilər ki ayrı bir ardıcıllıqda olsun. Indi bu iki çoxluğun ədədlərini n moduluna görə bir birinə vuraq

$[(a \bmod n)\bullet(2a \bmod n)\bullet\ldots\bullet(n-1)a \bmod n] \bmod n \equiv (n-1)! \bmod n$

$(n-1)!\, a^{n-1} \equiv (n-1)! \bmod n$

n və (n-1) ! – qarşılıqlı sadədirlər, əgər n-sadədir, deməli $a^{n-1} \equiv 1 \bmod n$.

5. Eyler teoremi.

Bütün a və n qarşılıqlı sadə ədədlər üçün $a^{\varphi(n)} \equiv 1 \bmod n$ yerinə yetirilir. Bu düzdür, əgər n-sadədirsə, onda φ(n)=n-1.

$R=\{x_1,x_2,\ldots,x_{\varphi(n)}\}$ çoxluğuna baxaq. Indi isə bu çoxluğun elementlərinin hər birini n moduluna görə a-ya vuraq.

$S=\{a\bullet x_1 \bmod n,\ a\bullet x_2 \bmod n,\ldots,\ a\bullet x_{\varphi(n)} \bmod n\}$ çoxluğu alırıq.

Aşağıdakı səbəblərə görə bu çoxluq R çoxluğunun yerdəyişməsidir:

a və x_i – n ilə qarşılıqlı sadədirlər, onda $a\bullet x_i$ – həmçinin n ilə qarşılıqlı sadədirlər. Beləliklə S – tam, n-dən kiçik və n ilə qarşılıqlı sadə ədədlər çoxluğudur. S-də təkrarlanma yoxdur, çünki $a\bullet x_i \bullet \bmod n = a\bullet x_ж \bullet \bmod n \Rightarrow x_i = x_ж$

Deməli, R və S çoxluqlarının elementlərini vursaq, alırıq:

$(a^{\varphi(n)} \equiv 1) \bmod n$

Indi isə RSA alqoritminə baxaq. p və q-sadədirlər, n=p•q. Sübut etmək lazımdır ki, $\forall M< n : M^{\varphi(n)}=M^{(p-1)(q-1)} \equiv 1 \bmod n$

Əgər gcd (M,n)=1 yerinə yetirilir. Indi təsəvvür edək ki, gcd (M,n)≠1, yəni gcd (M, p•q) ≠1. Qoy gcd (M,p) ≠1, yəni M=c•p⇒gcd (M,q)=1, çünki əks halda M=c•p və M=1•q, amma şərtə görə M<p•q.

Deməli $M^{\varphi(q)} \equiv 1 \bmod q$.

$$(M^{\varphi(q)})^{\varphi(p)} \equiv 1 \bmod q$$

$$M^{\varphi(n)} \equiv 1 \bmod q$$

Modulun müəyyən olunmasına görə $M^{\varphi(n)} = 1+k•g$

Bərabərliyin iki tərəfini də M=c•p vurduqda, alırıq:

$M^{\varphi(n)+1}=c•p+k•q•c•p$

$M^{\varphi(n)} \equiv 1 \bmod n$ ya $M^{\varphi(n)+1} \equiv M \bmod n$

Beləliklə, belə e və d seçilməlidir ki, $e•d \equiv 1 \bmod \varphi(n)$

Ya $e \equiv d^{-1} \bmod \varphi(n)$. e və d - φ(n) moduluna görə vurmaya görə qarşılıqlı əks olunandılar.

Qeyd edək ki, modul riyaziyyatı qaydalarına uyğun olaraq, bu doğrudur ancaq onda, əgər d(və, deməli e) φ(n) ilə qarşılıqlı sadədirlər.

Beləliklə, gcd (φ(n), d)=1.

Indi isə RSA alqoritminin bütün elementlərinə baxaq.

P>q – iki tam sadə ədəddir – bağlıdırlar, seçilməlidirlər.

n=p•q - - açıqdır, hesablanmalıdır.

d, qcd (φ(n), d)=1; 1<d<φ(n) – bağlıdır, hesablanmalıdır.

$e \equiv d^{-1} \bmod \varphi(n)$ - açıqdır, seçilməlidir.

Bağlı açar (d, n) ibarətdir, açıq açar (e, n) ibarətdir.

Təsəvvür edək ki, A-istifadəçi öz açıq açarını nəşr edir və B-istifadəçi A-ya M məlumatını ötürmək istəyir. Onda B-C=M^e(mod n) hesablayır və ötürür. Bu şifrlənmiş mətni A istifadəçi aldıqda aydınlaşdırır M=C^d(mod n) bu hesablama ilə.

RSA alqoritminin cəmlənməsi:

<u>Açarların yaradılması</u>.

Sadə p və q seçilməsi

n=p•q – hesadlanması

d – seçilməsi gcd ($\varphi(n)$, d)=1; 1<d<$\varphi(n)$

e – hesablanması $e=d^{-1} \bmod \varphi(n)$

Açıq açar KU=(e, n)

Bağlı açar KR=(d, n)

<u>Şifrləmə</u>

Şifrlənməmiş mətn: M<n

Şifrlənmiş mətn: C=M^e(mod n)

<u>Aydınlaşdırma</u>.

Şifrlənmiş mətn: C

Şifrlənməmiş mətn: M=C^d(mod n)

Konkret misala baxaq:

Iki sadə ədəd seçək: p=7, q=17 hesablayaq n=p•q=7•17=119

Hesablayaq $\varphi(n)$=(p-1)•(q-1)=96

e elə seçək ki, e φ(n) ilə qarşılıqlı sadə olsun φ(n)=96 və φ(n)-dən kiçik olsun: e=5

d-ni elə müəyyən edək ki, d•e ≡ 1 mod 96 və d<96 d=7, çünki 77•5=385=4•96+1.

Nəticəvi açarlar:

-açıq KU=(5, 119)

-bağlı KR=(77, 119).

Məs., M=19 məlumatının şifrlənməsi tələb olunur.

19^5=66 (mod 119); C=66

Aydınlaşdırma üçün hesablayaq 66^{77}(mod 119)=19.

Hesablama aspektləri.

RSA alqoritmində, açarların yaradılması və şifrləmə/aydınlaşdırma zamanı, hesablamaların mürəkkəbliyinə baxaq.

Şifrləmə / aydınlaşdırma.

Şifrləmədə olduğu kimi, aydınlaşdırmada da tam ədədin n moduluna görə tam qüvvətə yüksəlməsi daxildir. Bu zaman aralıq qiymətlər böyük olur. Buna yol verməmək üçün modul riyaziyyatının xüsusiyyətindən istifadə edilir:

[(a mod n)•(b mod n)] mod n=(a•b) mod n

Başqa bir optimallaşdırma qüvvətə yüksəltmənin effektiv istifadəsindədir, çünki RSA-da qüvvətə yüksəltmənin qiymətlərinin çox böyük olmasındadır. Təsəvvür edək ki, x^{16} – hesablanmalıdır. Düz yanaşma 15 dəfə vurma tələb edir. Amma son nəticənin əldə edilməsi ancaq 4 vurma əməli ilə də mümkündür, əgər hər aralıq nəticənin kvadratından istifadə edilsə: x^2, x^4, x^8, x^{16}.

Açarların yaradılması.

Açarların yaradılmasına daxil olan məsələlər:

1. Iki sadə ədədlərin p və q tapılması
2. e seçilməsi və d hesablanması.

Əvvəldən baxaq p və q seçilməsi probleminə n=p•q hər bir potensial düşmən ola bilən adama məlum ola bilər. Buna imkan verməmək üçün sadə ədədlər p və q çox böyük olan bir çoxluqdan seçilməlidir, yəni p və q – böyük ədədlər olmalıdır. O biri tərəfdən bu seçilmə prosesi kifayətcə sadə olmalıdır.

Müasir dövrdə kifayətcə böyük olan sadə ədəd yaradan elə bir alqoritm yoxdur. Məlum olan və bunun üçün istifadə edilən prosedur, tələb olunan diapazondan təsadüfi tək ədəd seçir və onun sadə olmasını yoxlayır. Proses sadə ədəd tapılana kimi davam edir.

Ədədin sadə olmasını yoxlamaq üçün cürbəcür testlər işlənib. Bu testlər ehtimallıq testləridir, yəni test göstərir ki, verilən ədədin sadə olması ehtimal olunur. Ola bilər ki, bu testlər ehtimalı 1 yaxınlaşdırsın. Əgər n testdən «keçirsə» - o sadə ola da bilər olmaya da bilər. Əgər n testi ödəmir («keçmir») – n sadə deyil. Əgər n çox belə testləri ödəyirsə, onda yüksək dəqiqliklə demək olar ki, n sadə ədəddir. Belə proses çox uzun çəkəndir, amma belə proseduru çox nadir hallarda yerinə yetirmək lazım gəlir: ancaq yeni açarlar cütlüyü yaradılarsa (KU, KR).

Hesablamaların mürəkkəbliyinə qəbul edilməmiş (rədd edilmiş) ədədlərin sayı da (sadə ədəd tapılana kimi) təsir edir. Ədədlər nəzəriyyəsindən məlum olan nəticə kimi sadə ədədlər teoremində deyilir ki, hər bir ln(n) üçün orta hesabla bir sadə ədəd olmaqla, sadə ədədlər n yanında yerləşir. n yanında yerləşən sadə ədədlərin sayı orta hesabla hər bir ln(n) üçün birdənədir. Beləliklə, sadə ədədi tapmazdan əvvəl, ln(n) tam ədədlər ardıcıllığının yoxlanılması tələb olunur. Cüt ədədlərin yoxlanılmadan rədd edilməsi göstərir ki, ln(n) / 2 kimi yoxlama tələb olunur. Məs., əgər sadə ədəd 2^{200} diapazonunda axtarılırsa, onda ln(2^{200}) / 2=70 yaxın yoxlamanın yerinə yetirilməsi kifayətdir.

p və q – sadə ədədlərin seçilməsindən sonra e qiymətinin seçilməsi lazım gəlir. Bunun üçün gcd (φ(n),e)=1 və d qiyməti hesablanmalıdır, $d=e^{-1} \mod \varphi(n)$.

Genişləndirilmiş Evklid alqoritmi adlanan yeganə alqoritm var, hansı ki, fiksə olunmuş vaxt ərzində iki tam ədədin ən böyük ümumi bölənin hesablayır və əgər o birə bərabərdirsə, birinin invers qiyməti o birisinin moduluna görə hesablayır. Beləliklə, prosedur bir sıra təsadüfi ədədlərin henerasiyasından və onların hər birinin φ(n) görə o vaxta kimi yoxlanılmasından ibarətdir ki, ta φ(n) ilə qarşılıqlı sadə olan ədəd tapılsın.

<u>Kriptoanaliz haqqında</u>.

RSA alqoritminin kriptoanaliz dörd cür yanaşmaq olar:

1.Qarşıdan hücum: bütün bağlı açarların arasından seçib ayırmaqla.

2.n iki sadə vuruğa ayırmaqla. Bu φ(n) hesablamağa imkan yaradır.

$\varphi(n)=(p-1)(q-1)$ və $d=e^{-1} \pmod{\varphi(n)}$.

3.p və q-ni əvvəldən tapmadan, φ(n) tapılması.

Bu həmçinin d tapılmasına da imkan yaradır $d=e^{-1} \pmod{\varphi(n)}$.

4.φ(n) hesablamadan, d tapılması.

RSA və ona uyğun alqoritmlərin qarşıdan hücumdan müdafiəsi uzunluğu böyük olan açarlardan istifadədir. Beləliklə, e və d-də bitlərin sayı nə qədər çoxdursa bir o qədər yaxşıdır. Amma, hesablamaların həm açarları yaradarkən, həm də şifrləmə / aydınlaşdırmada lazım olduğundan, açarın ölçüsünün böyüklüyü, sistemin işini yavaşlayır.

Müasir dövrdə elə bir alqoritm məlum deyil ki, çox böyük olan (yəni bir neçə yüz onluq rəqəm) ədədi iki vurulanlarla əvəz etsin. Alqoritmlərdən ən yaxşısı proporsional nəticə verir:

$l(n) = e^{\mathrm{sqrt}(\ln \cdot n \cdot \ln(\ln n))}$

Hələ ki, ən yaxşı olan ədədi sadə ədədlərlə əvəz edən alqoritm işlənib hazırlanmayıb, hesab etmək olar ki, 100-dən 200 kimi rəqəmlərdən ibarət olan n kifayətcə təhlükəsizdir. Hesab edilir ki, 100 rəqəmdən ibarət olan ədədin iki vurulanla əvəzi iki həftə ərzində mümkündür. Bahalı konfiqurasiyalar (10 min) üçün 150 rəqəmli ədədi 1 ilə, 200rəqəmli mümkünsüzdür. Məs., əgər saniyədə 10^{12} əməliyyat əldə edilərsə (müasir texnologiyalar üçün bu mümkünsüzdür), 10 il tələb olunur ki, 200 rəqəmdən ibarət olan ədədi məlum olan alqoritmlərdən istifadə etməklə vurulanlar ilə əvəz etmək.

Məlum olan alqoritmlərdən verilən e və n əsasında φ(n) tapılması məsələsi ədədin verilənlərə paylamaq məsələsi ilə vaxtına görə müqayisə edilə bilər.

n-in qiymətini asan vurulanlarla əvəz edilməsindən xilas olmaq üçün, p və q qiymətləri məhdudlaşdırılmalıdır:

1. p və q bir-iki rəqəmlə fərqlənməlidirlər. Beləliklə, p və q qiymətləri 10^{75}-dən 10^{100} kimi olmalıdır.