AZƏRBAYCAN RESPUBLİKASI
ELM VƏ TƏHSİL NAZİRLİYİ

Azərbaycan Kibertəhlükəsizlik
Təşkilatları Assosiasiyası

KRİPTOQRAFİYANIN ƏSASLARI

fənni üzrə mühazirələr toplusu

Prof.Dr. Yadigar İmamverdiyev

Bu vəsait Azərbaycan Respublikasının Elm və Təhsil Nazirliyinin maliyyə dəstəyi ilə həyata keçirilən "İnformasiya təhlükəsizliyi ixtisası üzrə elmi-metodiki tədris (çap və onlayn) vəsaitlərinin hazırlanması" layihəsi çərçivəsində hazırlanmışdır.

Nəşrin məzmununa görə donor məsuliyyət daşımır.

Azərbaycan Respublikası Elm və Təhsil Nazirliyi
Azərbaycan Kibertəhlükəsizlik Təşkilatları Assosiasiyası
Azərbaycan Texniki Universiteti
Kibertəhlükəsizlik Kafedrası

Yadigar İmamverdiyev

KRİPTOQRAFİYANIN ƏSASLARI – MÜHAZİRƏLƏR

BAKI – 2025

İmamverdiyev Y.N. «Kriptoqrafiyanın əsasları - mühazirələr», Bakı: AzTU Press nəşriyyatı, 2025, 132 səh.

Kitabda “Kriptoqrafiyanın əsasları” fənni üzrə tədris materialları mühazirələr şəklində sistematik şərh olunur. Klassik şifrlər və onların kriptoanalizi, axın və blok şifrləri, asimmetrik kriptosistemlər, kriptoqrafik heş funksiyaları, kriptoqrafik protokollar əhatə olunur. Kriptovalyutalara və post-kvant kriptoqrafiyasına qısa giriş edilir. Hər mühazirədə mövzuya aid kriptoanaliz məsələlərinə də baxılır. Hər mühazirənin sonunda öyrənilən materialı möhkəmləndirmək üçün yoxlama sualları verilir. Əlavələrdə S-DES şifrinə aid sadə məsələlər, blokçeyn terminlərinin izahlı lüğəti, blokçeynlərə aid laboratoriya işləri və sərbəst işlərin mövzuları təqdim edilib.

Vəsait müasir kriptoqrafik texnologiyaların tətbiqi, tədqiqi və kriptosistemlərin işlənilməsi sahəsində ixtisaslaşan tələbələr, tədqiqatçılar və mühəndislər üçün nəzərdə tutulmuşdur.

Mündəricat

Giriş

Təqdim olunan vəsait dünyada hər zaman böyük maraq doğuran kriptoqrafiyanın əsaslarına həsr olunub.

Kriptoqrafiya informasiya təhlükəsizliyinin təmin edilməsinin baza texnologiyası kimi çıxış edir. Konfidensiallığın təmin edilməsi üçün şifrləmə, tamlığa nəzarət etmək üçn kriptoqrafik heş funksiyaları, müəlliflikdən imtinanın qeyri-mümkünlüyü üçün rəqəmsal imzalar, autentifikasiya üçün açarlı heş funksiyaları istifadə edilir. Kriptoqrafik protokolların əsasında və digər texnologiyaların dəstəyi ilə son onillikdə nəhəng kriptovalyutalar ekosistemi formalaşmışdır. Buna görə həm "İnformasiya təhlükəsizliyi" ixtisasında, həm də digər əlaqədar ixtisaslarda "Kriptoqrafiyanın əsasları" fənninin tədrisini zəruri edir. Adi istifadəçilərin də istifadə etdikləri təhlükəsizlik texnologiyalarının iş prinsiplərini başa düşərək informasiya təhlükəsizliyi sistemlərindən səmərəli istifadə etmələri üçün kriptoqrafiyadan zəruru minimal biliklərə yiyələnməsi zamanın tələbidir.

Qeyd edək ki, Azərbaycan Respublikası Prezidentinin 2023-cü il 28 avqust tarixli 4060 nömrəli Sərəncamı ilə təsdiq edilmiş "Azərbaycan Respublikasının informasiya təhlükəsizliyi və kibertəhlükəsizliyə dair 2023 – 2027-ci illər üçün Strategiyası"nın həyata keçirilməsi ilə bağlı Tədbirlər Planının prioritetləri arasında kriptoqrafiya ilə əlaqəli məsələlər də yer alır (8.2.2, 8.2.2.4, 8.6.1.4.). 8.6.1.4-də "Kriptoqrafiyanın əsaslarının dərindən öyrənilməsini, bu sahədə elmi tədqiqatların aparılmasını, milli kriptoqrafik alqoritmin tətbiq olunduğu informasiyanın kriptoqrafik mühafizə vasitələrinin layihələndirilməsini və təkmilləşdirilməsini təmin edəcək yüksəkixtisaslı kadrların və səriştəli mütəxəssislərin hazırlanmasını təşkil etmək" nəzərdə tutulur.

Bu vəsait, Azərbaycan Texniki Universitetində "İnformasiya təhlükəsizliyi" ixtisasında təhsil alan bakalavriat tələbələrinə oxunan mühazirələr əsasında hazırlanıb. Vəsaitin əsas məqsədi tələbələrə kriptoqrafiyaya dair sistemli biliklər vermək, kriptoqrafik sistemlərin işləmə, qurulma və analiz prinsiplərini, əsas primitivləri və gələcək inkişaf və tətbiq istiqamətlərini qısa və mümkün olduqca, aydın şəkildə izah etməkdir.

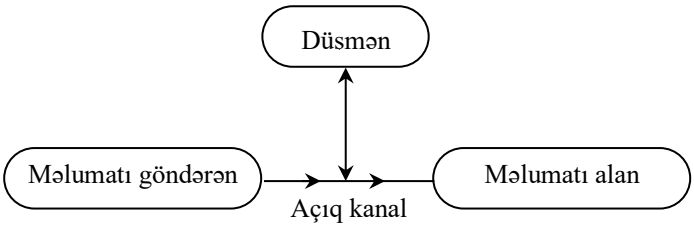
Rəqəmsal dünyada etibarlı kommunikasiya sistemlərinə tələbat artır və hər bir şəxs gündəlik həyatda kriptosistemlərdən geniş istifadə edir. Ümid edirik ki, bu vəsait tələbələrə təkcə kriptosistemlərin işləmə prinsiplərini öyrənməkdə deyil, həm də gələcəkdə bu sahədə ixtisaslaşmaları üçün bir təməl yarada biləcək.

Mühazirə 1. Kriptoqrafiyanın əsas anlayışları

1.1. Rabitənin baza modeli

Kriptoqrafiya informasiyanıi cazəsi olmayan istifadəçilərdən qorumaq məqsədi ilə informasiyanın çevrilməsi metodlarını öyrənir. «Kriptoqrafiya» sözü yunan dilində kriptos (kryptos) – «gizli» və γραφω (grapho) – «yazı» sözlərindən yaranıb.

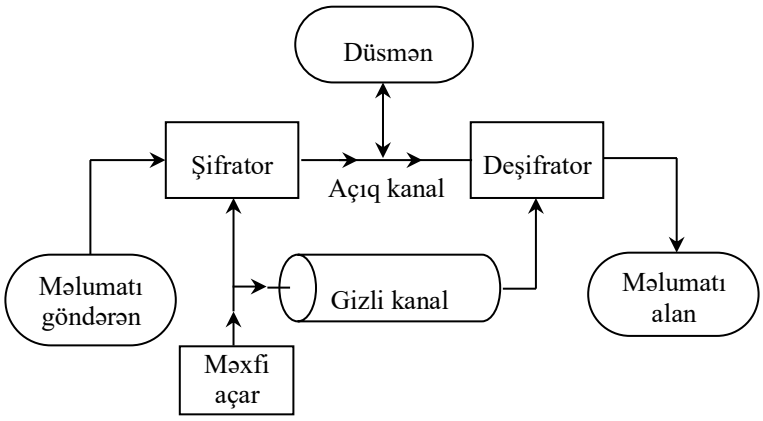
Kriptoqrafiyada, adətən, düşmənin rabitə kanalına tam nəzarət etməsi fərz olunur. Rabitənin baza modeli (şəkil 1) açıq rabitə kanalına girişi olan və qulaq asmaq yolu ilə göndərəndən alana ötürülən bütün məlumatları ələ keçirən düşmənin varlığını nəzərdə tutur.



Şəkil 1.1. Rabitənin baza modeli

Düşmən tərəfindən ötürülən məlumatlara qulaq asılması (sonradan analiz etmək üçün) *məlumatların passiv tutulması* adlanır. Düşmən informasiya ötürülməsi prosesinə aktiv müdaxilə də edə bilər – onları aktiv dəyişdirə bilər, abonentlərdən birinin adından saxta məlumat da göndərə bilər və s. Belə hərəkətlər *məlumatların aktiv tutulması* adlanır.

Düşmənə əks-təsir üsulları kriptoqrafik metodların tətbiqinə əsaslanır. Klassik



Şəkil 1.2. Biraçarlı kriptosistem

biraçarlı (simmetrik) *kriptoqrafik sistem* (kriptosistem) şəkil 1.2-də təsvir olunub.

Açıq kanala ötürülməzdən əvvəl məlumatın məzmununu gizlətmək üçün ilkin məlumat xüsusi çevirməyə məruz qalır. Qəbul edən tərəfdə ilkin məlumatı almaq

üçün tərs çevirməni yerinə yetirmək lazımdır. Ötürən tərəfdə düz çevirmə proseduru *şifrləmə*, qəbul edən tərəfdəki tərs çevirmə proseduru *deşifrləmə* adlanır. Biraçarlı kriptosistemlərdə şifrləmə və deşifrləmə prosedurlarını yerinə yetirmək üçün göndərən və alan ümumi gizli komponenti – *məxfi açarı* bilməlidir.

Məxfi açarın göndərəndən alana ötürülməsi üçün istifadə edilən gizli kanal informasiyanın sızması imkanını istisna etməlidir. Etibarlı gizli kanalın yaradılması çox vaxt xeyli maliyyə və təşkilati xərclər tələb edən mürəkkəb problemdir. Paylanmış şəbəkə arxitekturası şəraitində abonentlərin hər bir cütü üçün gizli kanalın təmin edilməsi praktk olaraq həlledilməz məsələyə çevrilir.

1.2. Kriptoqrafiyanın bəzi anlayışları

Əlifba – informasiyanı kodlamaq üçün istifadə edilən simvolların sonlu çoxluğudur. Müasir informasiya sistemlərində istifadə edilən əlifbalara misal: ingilis əlifbasının 26 hərfi; ASCII standart kodlarına daxil olan simvollar; ikilik əlifba – {0, 1} və s. göstərmək olar.

Mətn – əlifbanın elementlərinin nizamlı (sıralanmış) yığınıdır.

Bəzən *kriptoqram* anlayışından da istifadə olunur. Kriptoqram – daha geniş anlayışdır, şifrmətndən başqa əlavə verilənlər, məsələn, kriptoqrafik atributlar (kriptoalqoritmlərin, heş funksiyaların identifikatorları, sertifikatlar, müxtəlif nişanlar və s.) daxildir.

Şifrator – şifri realizə edən qurğu (və ya proqram).

Şifrləmə – şifrin köməyi ilə açıq mətnin şifrlənmiş mətnə çevrilməsi prosesidir;

Deşifrləmə – məlum açar əsasında şifrlənmiş mətni açıq mətnə çevirmə prosesidir.

Şifrin dözümü – şifrin açarı bilmədən açıq mətni almaq cəhdlərinə qarşı durmaq qabiliyyətidir.

Şifrin sındırılması (*açılması*) – açarı bilmədən açıq mətni tapmaq üsulunun tapılmasıdır (bu üsul açarın bütün variantlarının yoxlanmasından daha səmərəli olmalıdır).

Şifrə hücum – şifrin sındırılmasına cəhddir.

Son dövrlər «kriptoqrafiya» sözü ilə yanaşı «kriptologiya» sözü də tez-tez işlədilir, lakin onların arasındakı münasibət heç də həmişə düzgün başa düşülmür. Hazırda bu elmi fənlərin formalaşması baş verir, onların predmet və məsələləri dəqiqləşdirilir.

Kriptologiya iki hissədən – *kriptoqrafiya* və *kriptoanalizdən* ibarət elmdir.

- *Kriptoqrafiya* – bədniyyətlinin müəyyən hərəkətlərindən qorumaq məqsədi ilə informasiyanın çevrilməsi üsulları haqqında elmdir.

- *Kriptoanaliz* – qorunan informasiyanı əldə etmək məqsədi ilə kriptoqrafik çevirmələrin (şifrlərin) analizi metodları haqqında elmdir (və onların tətbiqi praktikasıdır).

Kriptologiya tətbiqi elmdir, o, fundamental elmlərin, ilk növbədə riyaziyyatın ən son yeniliklərindən istifadə edir. Digər tərəfdən kriptoqrafiyanın bütün konkret məsələləri texnikanın və texnologiyanın inkişaf səviyyəsindən, tətbiq edilən rabitə vasitələrindən və informasiyanın ötürülməsi üsullarından əhəmiyyətli dərəcədə asılıdır.

1.3. Kriptoanalizin məsələləri və prinsipləri

Məşhur kriptoqraf Uilyam Fridmanın tərifinə görə «kriptoanaliz istifadə edilən dilin, kriptosistemin növünün, açarın və ilkin mətnin müəyyən edilməsindən ibarətdir; adətən məhz bu ardıcıllıqda».

Kriptoanalizin əsas məqsədi istifadə edilən və yaradılan kriptosistemlərin dözümünü qiymətləndirməkdən ibarətdir. Kriptoanaliz məsələlərinin həlli zamanı istifadə edilən beş əsas prinsipi sadalayaq:

1. Düşməni lazımınca qiymətləndirməmək olmaz.
2. (*Kerkhoffs prinsipi*) Kriptosistemin təhlükəsizliyi haqqında yalnız kriptoanalitik hökm verə bilər.
3. (*Kerkhoffs-Şennon prinsipi*) Düşmən istifadə edilən kriptosistemi açar dəqiqliyi ilə bilir.
4. (*Jiverje prinsipi*) Kriptosistemin səthi mürəkkəbləşdirilməsi aldadıcı ola bilər, çünki bu dözümün yalançı qiymətləndirilməsini törədir.
5. Hər hansı kriptosistemin dözümünü qiymətləndirərkən kriptoqrafik səhvləri və təhlükəsizlik qaydalarının digər pozuntularını da nəzərə almaq gərəkdir.

Şennon-Kerkhoffs prinsipinə görə kriptoanalitikin əsas məsələsi istifadə edilən kriptosistemin özünün məlum olması şərtində açar informasiyasını qiymətləndirməkdir.

1.4. Kriptoanaliz hücumlarının növləri

Şifrləmə sisteminin kriptoanalizi cəhdini *hücum* aldandırırlar. Ümumi halda, kriptoanalitikin məsələsi şifrləmə zamanı istifadə olunmuş açıq mətnlərin və gizli açarların axtarışından ibarətdir. Qeyd etmək lazımdır ki, şifrləmə sisteminin *sındırılması* dedikdə, şifrin tam sınaq üsulu ilə müqayisədə az çətinliyə malik hücumu təşkil etməyə imkan verən "zəifliyinin" tapılması başa düşülür.

Kriptoanalizdə vacib fərziyyə Kerkhoffs qaydasıdır: Şifrin dözümü yalnız açarın gizliliyi ilə müəyyən olunmalıdır. Beləliklə, bütün hücumlar üçün ümumi fərziyyə kriptoanalitikə şifrləmə alqoritminin apriori məlum olması faktıdır. Digər

demək olar ki, ümumi qəbul edilmiş fərziyyə ondan ibarətdir ki, kriptoanalitikin ixtiyarında məlumatların şifrmətnləri var.

Yeni şifrlərin analizi zamanı və şifrləmə alqoritmlərinin kriptoqrafik dözümünün isbatı zamanı adətən hücumların aşağıdakı əsas növləri istifadə olunur:

Yalnız şifrmətni bilməklə hücum (known-ciphertext attack) – bu hücum növündə kriptoanalitikə eyni şifrləmə sistemindən istifadə etməklə eyni açarla şifrlənmiş bir və ya bir neçə şifrmətn məlumdur.

Bu hücum variantı rabitə xəttinə fiziki girişi olan, lakin şifrləmə və deşifrləmə aparatlarına girişi olmayan xarici kriptoanalitik modelinə uyğun gəlir.

Açıq mətni bilməklə hücum (known-plaintext attack) – bu hücum növündə kriptoanalitikə açıq mətn fraqmentləri və onlara şifrmətnin hansı fraqmentlərinin uyğun olması haqqında informasiya məlumdur.

Belə hücumun həyata keçirilməsi imkanı standart formalar üzrə hazırlanan standart sənədlərin şifrlənməsi zamanı yaranır, belə sənədlərdə müəyyən verilənlər bloku təkrarlanır və məlumdur. Bu hücum həmçinin qlobal şifrləmə rejimi istifadə edilən zaman tətbiq edilə bilər, bu halda maqnit daşıyıcıdakı bütün informasiya, yükləmə sektoru, sistem proqramlar və s. daxil olmaqla şifrlənmiş mətn şəklində yazılır.

Seçilmiş açıq mətnlərlə hücum (chosen-plaintext attack) – bu hücum növündə kriptoanalitikə nəinki açıq mətn və şifrmətn məlumdur, həm də istənilən açıq mətn üçün ona uyğun şifrmətn əldə edilə bilər.

Bu hücum variantı daxili pozucu modelinə uyğun gəlir. Praktikada belə vəziyyət gizli açarı bilməyən, lakin öz xidməti səlahiyyətlərinə görə öz məlumatlarının ötürülməsi üçün şifrləmədən istifadə etmək imkanı olan şəxslərin kriptoqrafik hücuma cəlb edilməsi zamanı baş verə bilər.

Xarici ədəbiyyatda çox vaxt bu hücumu «gecəyarısı hücum» (midnight attack) və ya *coffee-break attack* adlandırırlar, şəxsi heyətin şifrləmə qurğusunu işçi vəziyyətdə nəzarətsiz qoyduğu və düşmənin onu müvəqqəti ələ keçirdiyi real şəraitə uyğun gəlir. Gizli açara əli çatmasa da, düşmən özünün hazırladığı açıq mətnləri şifrləyə bilər ki, bu da kriptosistemə hücum üçün ona əlavə informasiya verir.

Seçilmiş şifrmətnlə hücum (chosen-ciphertext attack) – bu hücum zamanı kritoanalitik şifrmətni seçir və ona uyğun açıq mətni əldə edə bilir. Hücumun bu variantı məsələn, kriptoanalitik avtomatik deşifrləməni yerinə yetirən, icazəsiz açılmaqdan mühafizəli bloka giriş əldə etdikdə mümkündür. Kriptoanalitikin işi açarı tapmaqdan ibarətdir. Kriptoanalizin bu növü açıq açarlı alqoritmlərin açılması üçün xüsusi maraq kəsb edir.

Mətni seçməklə hücum (chosen-text attack). Düşmənin kriptosistemə «hər iki ucdan» hücum etmək imkanı var, yəni həm şifrmətnləri seçə (və onları deşifrləyə), həm də açıq mətnləri seçə (və onları şifrləyə) bilər.

Müasir kriptoqrafiyada bu hücumlardan savayı, yuxarıda sadalanan məlum verilənlərdən istifadə etməklə *mühəndis kriptoanalizinin* aşağıdakı metodlarına da baxılır:

- qəsdən törədilən aparat səhvləri əsasında hücum;
- sərf edilən elektrik enerjisinin ölçülməsi əsasında hücum;
- hesablamaların vaxtının ölçülməsi əsasında hücum.

Mühazirə üzrə yoxlama sualları

1. Kriptoqrafiya sözü hansı sözlərdən yaranıb?
2. Kerkhoffs prinsipi nədir?
3. Açıq mətn nədir?
4. Şifr nədir?
5. Şifrləmə açarı nədir?
6. Şifrləmə açarı nəyi müəyyən edir?
7. Şifrmətn nədir?
8. Şifrator nədir?
9. Şifrləmə nədir?
10. Deşifrləmə nədir?
11. Şifrin dözümü nədir?
12. Şifrin açılması nədir?
13. Şifrə hücum nədir?
14. Kriptologiya elmi hansı bölmələrdən ibarətdir?
15. Kriptoqrafiya nəyi öyrənir?
16. Kriptoanaliz nəyi öyrənir?
17. Yalnız şifrmətni bilməklə hücum (known-ciphertext attack) nədir?
18. Açıq mətni bilməklə hücum (known-plaintext attack) nədir?
19. Seçilmiş açıq mətnlə hücum (chosen-plaintext attack) nədir?
20. Seçilmiş şifrmətnlə hücum (chosen-ciphertext attack) nədir?
21. Mətni seçməklə hücum (chosen-text attack) nədir?
22. Mühəndis kriptoanalizinin hansı üsulları var?

Mühazirə 2. Klassik şifrlər

2.1. Yerdəyişmə və əvəzetmə şifrləri

Şifrlərin təsnifatını müxtəlif əlamətlərə görə aparmaq olar. Şifrlərin təsnifatını aparmaq üçün ilk əlamət kimi şifrləmə zamanı açıq mətn üzərində həyata keçirilən çevirmələrin növü götürülür. Bu əlamətə görə şifrlərin iki əsas növü var: əvəzetmə şifrləri və yerdəyişmə şifrləri.

Yerdəyişmə şifrləri açıq mətn elementlərinin (hərfləri, simvolları) müəyyən qaydada yerini dəyişdirir.

Məlumatı qruplara bölmək (məsələn, hərəsində üç hərf olmaqla) və hər bir qrupda eyni bir yerdəyişməni etmək olar. Məsələn:

LET USG OTO LON DON → ETL SGU TOO ONL OND.

Həmin məlumatı 3×5 ölçülü cədvələ yazmaq olar:

L	E	T	U	S
G	O	T	O	L
O	N	D	O	N

Sonra onu sütunlar üzrə oxumaq olar: LGO EON TTD UOO SLN. Bu şifrmətn (kriptoqram) olacaq.

Əvəzetmə şifrləri açıq mətn elementlərini müəyyən qayda üzrə başqa elementlərlə əvəz edir. Sadə, mürəkkəb, qoşa əvəzetmə, hərf-heca və sütunlu əvəzetmə şifrlərini fərqləndirirlər. Əvəzetmə şifrləri iki qrupa bölünür: birəlifbalı və çoxəlifbalı əvəzetmə şifrləri.

Birəlifbalı əvəzetmə şifrlərində ilkin mətnin hərfi öncədən müəyyən edilmiş bir hərflə əvəz edilir. Birəlifbalı əvəzetmə şifrlərini *sadə əvəzetmə şifrləri* də adlandırırlar.

Çoxəlifbalı əvəzetmə şifrlərində ilkin mətnin müəyyən simvolunun əvəz edilməsi üçün hər dəfə rast gəldikdə müəyyən çoxluqdan müxtəlif simvollar ardıcıl istifadə edilir.

Şifrləmənin etibarını artırmaq məqsədi ilə hər hansı şifrlə şifrlənmiş mətn yenidən digər şifrlə şifrlənə bilər. Şifrlərin bütün belə mümkün kompozisiyaları şifrlərin üçüncü sinfini əmələ gətirir, onları adətən *kompozisiya şifrləri* adlandırırlar. Qeyd edək ki, kompozisiya şifri nə əvəzetmə şifrləri sinfinə, nə də yerdəyişmə şifrləri sinfinə daxil olmaya bilər.

Əvəzetmə şifrinə tarixi misal Sezar şifridir (b. e. ə. I əsr). Qay Yuli Sezar (b. e. ə. 100-44-cü illər) yazışmalarında özünün ixtira etdiyi şifrdən istifadə edirdi. Müasir Azərbaycan dilinə tətbiq etdikdə bu şifr aşağıdakılardan ibarətdir. Əlifba yazılır: ABÇÇDE…; sonra onun altında həmin əlifba sola üç hərf dövri sürüşdürməklə yazılır:

A B C Ç D E Ə F G Ğ H X I İ J K Q L M N O Ö P R S Ş T U Ü V Y Z
Ç D E Ə F G Ğ H X I İ J K Q L M N O Ö P R S Ş T U Ü V Y Z A B C

Şifrləmə zamanı A hərfi Ç hərfi ilə əvəz edilir, B hərfi D ilə, C hərfi E ilə və s. əvəz edilir.

Sürüşmə şifrləri. Sezar şifrləri sadə əvəzetmə şifrlərinin – sürüşmə şifrlərinin xüsusi halıdır. Sürüşmə şifrlərində şifrləmə funksiyası:

$$y = E_k(x) = (x + k)\ mod\ m$$

burada: m – əlifbadakı hərflərin sayı; ingilis əlifbasında $m = 26$;

x – açıqmətnin simvolu; k – sürüşmənin miqdarıdır, Sezar şifrində $k = 3$;

y – şifrmətnin simvoludur.

Sürüşmə şifrlərində deşifrləmə funksiyası belədir: $x = D_k(y) = (y - k)\ mod\ m$

2.2. Affin şifrlər

Tutaq ki, $P=C= Z_{26}$. $K=\{(a, b)\in Z_{26}\times Z_{26} \mid \text{ƏBOB}(a, 26)=1\}$. $k=(a, b)\in K$ və x, $y\in Z_{26}$ üçün şifrləmə və deşifrləmə qaydası uyğun olaraq

$$E_k(x) = (ax + b) \bmod 26$$

və

$$D_k(y) = a^{-1}(y - b) \bmod 26$$

düsturları ilə təyin edilir. İstifadə edilən funksiyalar affin funksiyalardır, buna görə kriptosistem də *affin şifr* adlanır. Qeyd edək ki, affin şifrlər əvəzetmə şifrlərinin xüsusi halıdır. Mümkün 26! yerdəyişmədən yalnız 26·12=312-sı istifadə olunur (b – 26 qiymət, a – isə 12 qiymət ala bilər). a və 26-nin qarşılıqlı sadəliyi inikasın biyektivliyi üçün tələb olunur. Əks halda müxtəlif simvolların eyni bir simvola inikası və deşifrləmənin qeyri-müəyyənliyi baş verə bilər. a=1 olduqda *sürüşmə şifri* alınır.

Misal (*Affin şifr*). CRYPTOGRAPHY sözünü açarı $k = (7,\ 3)$ olan affin şifr ilə şifrləyək. Şfirləmə funksiyası

$$y = E_k(x) = 7x + 3$$

olacaq. CRYPTOGRAPHY sözünə x=(2, 17, 24, 15, 19, 14, 6, 17, 0, 15, 7, 24) ədədi ardıcıllığı uyğun gəlir. Şifrləmə funksiyasının qiymətini $y=E_k(x)=7x+3$ düsturuna əsasən hesablayaq:

$y = E_k(x)$ = (7·2+3, 7·17+3, 7·24+3, 7·15+3, 7·19+3, 7·14+3, 7·6+3, 7·17+3, 7·0+3, 7·15+3, 7·7+3, 7·24+3) = (17, 18, 15, 4, 6, 23, 19, 18, 3, 4, 0, 15).

Nömrələrdən hərflərə keçsək, RSPEGXTSDEAP şifrmətni alınır.

Deşifrləmə üçün 7^{-1}-i Z_{26}^* qrupunda hesablamaq lazımdır. Aydındır ki, $7^{-1} \bmod 26=15$. Deşifrləmə funksiyası

$$x = D_k(y) = 15y - 19$$

olacaq. *y*-i bu funksiyaya əsasən deşifrləyək:

$x=D_k(y)=(15\cdot17-19,\ 15\cdot18-19,\ 15\cdot15-19,\ 15\cdot4-19,\ 15\cdot6-19,\ 15\cdot23-19,\ 15\cdot19-19,\ 15\cdot18-19,\ 15\cdot3-19,\ 15\cdot4-19,\ 15\cdot0-19,\ 15\cdot15-19) = (2, 17, 24, 15, 19, 14, 6, 17, 0, 15, 7, 24)$.

2.3. Vijener şifri

Romada fransız səfiri olarkən Blaise de Vigenère orada kriptoqrafiya üzrə əsərlərlə tanış olmuş və 1585-ci ildə kriptoqrafiyanın əsaslarını şərh edən "Şifrlər haqqında traktat" (*Traicté de chiffres*) kitabını yazmışdı. Əslində, Vijener adını daşıyan şifri ilk dəfə 1553-cü ildə Giovan Battista Bellaso tərəfindən təsvir edilmişdi. Çoxəlifbalı sistemlərin çox vacib təkmilləşdirməsi olan Vijener şifri özündən əvvəlki şifrlərin bir çox ideyalarını birləşdirirdi.

Vijener şifrində şifrləmə/deşifrləmə üçün cədvəldən istifadə edilir (Cədvəl 2.1.). Cədvəlin hər bir sətri bir Sezar şifrini təyin edir. Buna görə də Vijener şifri çoxəlifbalı şifrdir. Qeyd edək ki, ümumi halda Vijener cədvəli dövri sürüşdürülən əlifbalardan ibarətdir, həm də birinci sətir istənilən qarışıq əlifba ola bilər.

Cədvəl 2.1. Vijener cədvəli

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

Vijener şifrinin bir neçə variantı var. Ən sadə variantda qısa açar sözdən istifadə edilir. Açar söz açıq mətnin uzunluğuna qədər təkrarlanır. Açıq mətnin hərflərinə əsasən sətir, açarın hərflərinə əsasən sütun seçilir. Kəsişmədəki hərf şifrmətn hərfini verəcək.

Misal. "The boy has the ball" açıq mətnini VIG açarı ilə şifrləyək.

Açar	V I G V I G V I G V I G V I G V
Açıqmətn	t h e b o y h a s t h e b a l l
Şifrmətn	O P K W W E C I Y O P K W I R G

Açıq mətnin birinci hərfi T-ni açar hərfi V ilə şifrləyirik: V sütunu ilə T sətrinin kəsişməsinə baxılır ("O" hərfi alınır).

Açıq mətnin ikinci hərfi H-ni açar hərfi I ilə şifrləyirik: I sütunu ilə H sətrinin kəsişməsinə gəlirik ("P" hərfi alınır) və s.

Vijener şifrinin cəbri yazılışını belə vermək olar. Tutaq ki, m qeyd olunmuş müəyyən ədəddir, $X = Y = K = (Z_{26})^m$. $k=(k_1, k_2, \ldots, k_m)$ açarı üçün şifrləmə və deşifrləmə qaydaları uyğun olaraq

$$E_k(x_1, x_2, \ldots, x_m)=(x_1+k_1, x_2+k_2, \ldots, x_m+k_m) \text{ və}$$
$$D_k(y_1, y_2, \ldots, y_m)=(y_1-k_1, y_2-k_2, \ldots, y_m-k_m)$$

düsturları ilə təyin edilir. Bütün əməliyyatlar Z_{26}-da yerinə yetirilir.

$A\leftrightarrow 0$, $B\leftrightarrow 1$, …, $Z\leftrightarrow 25$ uyğunluğundan istifadə edərək hər bir k açarını uzunluğu m olan, açar söz adlanan sətir ilə əlaqələndirə bilərik. Vijener şifri bir dəfəyə m hərfi şifrləyir, Fərz olunsa ki, açar söz m müxtəlif hərfdən ibarətdir, onda Vijener şifrində hər hərf m mümkün hərfdən birinə çevrilə bilər. Belə şifrlər çoxəlifbalı adlanır.

Misal (*Vijener şifri*). Fərz edək ki, m=6 və açar söz CIPHER-dir. Açar sözün ədədi ekvivalenti k=(2, 8, 15, 7, 4, 17)-dir. Tutaq ki,

thiscryptosystemisnotsecure

açıq mətndir. Şifrləmə üçün açıq mətnin elementlərini ədədlərə çeviririk, onları altı-altı qruplarla yazırıq və sonra açar sözün ədədi ekvivalenti ilə 26 moduluna görə toplayırıq:

19	7	8	18	2	17	24	15	19	14	18	24	18	19	4	12	8	18	
2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7	4	17	
21	15	23	25	6	8	0	23	8	21	22	15	20	1	19	19	12	9	

13	14	19	18	4	2	20	17	4
2	8	15	7	4	17	2	8	15

15 22 8 25 8 19 22 25 19

Şifrmətnin hərfi ekvivalenti: VPXZGIAXIVWPUBTTMJPWIZITWZT.

Deşifrləmə üçün həmin açar söz istifadə edilir, ancaq onu 26 modulu ilə şifrmətndən çıxmaq lazımdır.

Vijener şifrində uzunluğu m olan mümkün açar sözlərin sayı 26^m-dır, buna görə də m-in nisbətən kiçik qiymətlərində də tükədici açar axtarışı böyük vaxt tələb edəcək. Məsələn, m=5 üçün açar fəzasının ölçüsü $1{,}1\cdot10^7$-ni aşır. Bu isə əl ilə tükədici axtarış üçün artıq keçilməzdir. Ümumi halda çoxəlifbalı şifrlərin kriptoanalizi birəlifbalı şifrlərin analizindən çox çətindir. Vijener şifri hətta 1850-ci illərdə sındırılmasına baxmayaraq, 300 il müddətində deşifrlənməz şifr ("le chiffrage indéchiffrable") hesab edilirdi.

2.4. Hill şifri

1929-cu ildə amerikan riyaziyyatçısı və kriptoqrafı Lester Hill tərəfindən təklif olunmuş bu şifr çoxəlifbalı şifrlərə aiddir. Bir dəfədə hərflər qrupunu: diqraf, triqraf və poliqrafları şifrləyir.

Tutaq ki, m müsbət tam ədəddir, $X=Y=(Z_{26})^m$ qəbul edək. Şifrin ideyası bir açıq mətn elementində m hərfin xətti kombinasiyasını götürərək bir şifrmətn elementində m hərf hasil etməkdən ibarətdir. Misal üçün m=2 olarsa, biz açıq mətn elementini $x=(x_1, x_2)$ kimi, şifrmətn elementini isə $y=(y_1, y_2)$ kimi yazmalıyıq. Burada y_1 və y_2 x_1 və x_2-nin xətti kombinasiyası olmalıdır.

Məsələn, biz

$$y_1 = 11x_1+3x_2$$
$$y_2 = 8x_1+7x_2$$

götürə bilərik. Bunu matrisin köməyi ilə də yazmaq olar.

$$(y_1, y_2) = (x_1,\ x_2)\begin{pmatrix}11 & 8\\ 3 & 7\end{pmatrix}$$

Ümumi şəkildə biz $m\times m$ ölçülü K matrisini açar kimi götürə bilərik. $x=(x_1, \ldots, x_m)\in X$ və $K\in K$ üçün $y=E_K(x)=(y_1, y_2, \ldots, y_m)$ belə hesablanır.

$$(y_1 \ldots y_m)=(x_1 \ldots x_m)\left(\begin{bmatrix}k_{11} & \cdots & k_{1m}\\ \vdots & \ddots & \vdots\\ k_{m1} & \cdots & k_{mm}\end{bmatrix}\right)$$

Başqa sözlə, $y = xK$.

Hill şifrində şifrmətn açıq mətndən xətti çevirmənin köməyi ilə alınır. Deşifrləmə $x = yK^{-1}$ düsturu ilə hesablanır, burada K^{-1} tərs matrisdir.

Misal (*Hill şifri*). Fərz edək ki, açar $K = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$ matrisidir, *july* açıq mətnini şifrləyək. Bu mətn (9, 20, 11, 24) ədədlər ardıcıllığına ekvivalentdir. (9, 20) və (11, 24) açıq mətninin elementləri şifrlənməlidir:

$$(9,\ 20)\cdot\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = (99+60,\ 72+140) = (3,\ 4)$$

$$(11,24)\cdot\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = (121+72, 88+168) = (11,22).$$

Deməli, *july* sözünün şifrmətni *DELW*-dir. Deşifrləmə üçün əvvəlcə tərs matris hesablanır:

$$K^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}.$$

Deşifrləmə belə aparılır:

$$(3,4)\cdot\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = (9,20),$$

$$(11,22)\cdot\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = (11,24).$$

Düzgün açıq mətn alındı.

Göründüyü kimi deşifrləmənin mümkün olması üçün *K* matrisinin tərsinin olması zəruridir. Kvadrat matrisin tərsinin olması onun determinantının qiymətindən asılıdır. Elementləri həqiqi ədədlər olan matrisin tərsinin olması üçün onun determinantının sıfırdan fərqli olması zəruri və kafidir. Xatırladaq ki, biz Z_{26} üzərində işləyirik. Bu halda *K* matrisinin Z_{26} üzərində yalnız və yalnız ƏBOB(det *K*, 26)=1 olduqda tərsi var.

Qeyd. 2×2 ölçülü $A = (a_{ij})$ matrisinin determinantı $detA = a_{11}a_{22} - a_{12}a_{21}$ kimi hesablanır. det $A \neq 0$ olduqda tərs matris belə hesablanır:

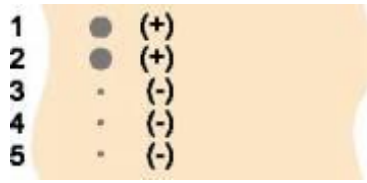
$$A^{-1} = (\det A)^{-1}\begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}.$$

Hill şifrinin yaxşı xüsusiyyətlərinə şifrmətnin ayrıca götürülmüş simvolları tezliyinin yaxşı bərabərləşdirilməsini aid etmək olar. Bununla yanaşı, baxılan sistem tənliklər sisteminin həllinə əsaslanan kriptoanalizə davam gətirmir. Əgər kriptoanalitikdə şifrmətndən əlavə açıq mətnin bir neçə *m*-qramı olarsa, açarın elementləri xətti tənliklər sisteminin həllindən tapmaq olar.

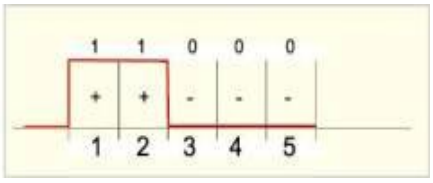
2.5. Vernam şifri

Amerikalı Cilbert Vernam (Gilbert Vernam) kriptoqrafiyanın inkişafına əhəmiyyətli töhfə vermişdir. Vernam 1917-ci ildə AT&T (American Telephone&Telegraph) teleqraf şirkətinin əməkdaşı işləyərkən, teleqraf məlumatlarının avtomatik şifrlənməsi ideyasını təklif etmişdir, onun mahiyyəti aşağıdakından ibarətdir. Açıq mətn Bodo kodunda (beşrəqəmli «impuls

kombinasiyaları» şəklində) təsvir olunur. Bu kodda, məsələn, «A» hərfi (+ + – – –) şəklində idi. Kağız lentdə bu hərf şəkil 2.1-də göstərildiyi kimi alınırdı:



Şəkil 2.1.



Şəkil 2.2.

«+» işarəsi deşiyi, «–» işarəsi isə onun yoxluğunu bildirirdi. Lentin oxunması zamanı beş metal çubuq deşiyi «tanıyırdı» (deşik olduqda çubuq elektrik dövrəsini qapayırdı). Rabitə xəttinə cərəyan impulsları göndərilirdi (şəkil 2.2).

Vernam məxfi məlumat işarələri impulslarını qammanın impulsları ilə koordinatlar üzrə elektromexaniki toplamağı təklif etmişdi. Qamma – elə həmin əlifba hərflərinin xaotik yığımından ibarət məxfi açardır. Toplama müasir terminologiya ilə «2 moduluna görə» həyata keçirilirdi: 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 0 (burada «0» «Bodo kodunun» «–» işarəsini, 1 isə «+» işarəsini bildirirdi). Məsələn, tutaq ki, qammanın işarəsi: + – + – – (10100) kimidir. Onda şifrləmə zamanı «A» hərfi: 01100 (– + + – –) ikilik kombinasiyasına keçir. Deşifrləmə zamanı həmin əməliyyatı təkrar etmək lazımdır: (01100)+(10100) = (11000) = (A)

Vernam göstərilən əməliyyatları avtomatik, insanın iştirakı olmadan yerinə yetirən qurğu yaratdı. Bununla da «xətti şifrləmənin» əsası qoyuldu, bu halda şifrləmə prosesi və məlumatın ötürülməsi prosesi eyni zamanda baş verir. O vaxtadək şifrləmə qabaqcadan yerinə yetirilməli idi, buna görə xətti şifrləmə rabitənin operativliyini əhəmiyyətli dərəcədə yüksəldirdi.

Vernam riyaziyyatçı-kriptoqraf deyildi, bunula belə o, təkid edirdi ki, şifrin açarı şifrləmə zamanı təkrarlanmamalıdır və bunda haqlı idi. Otuz ildən sonra K. Şennon aşağıdakı şərtlər ödəndikdə Vernam şifrinin mütləq dözümlü olmasını isbat etdi:

1. Açar tamamilə təsadüfi (bərabər ehtimallı) olmalıdır;
2. Açarın uzunluğu açıq mətnin uzunluğuna bərabər olmalıdır;
3. Açar yalnız bir dəfə istifadə edilməlidir.

Bu tələblərdən heç olmasa biri pozulduqda şifrin mütləq dözümlü olması xassəsi aradan qalxır.

2.6. Eniqma şifratoru

Dünya müharibələri arasında bütün qabaqcıl ölkələrdə elektromexaniki şifratorlar meydana çıxdı. Onların iki növü vardı – kommutasiya diskləri və ya rotorlar əsasında və dişli disklər əsasında. Rotorlu maşınlar yalnız XX əsrin

əvvəllərində praktik tətbiq edilməyə başlandı. Praktikada istifadə edilən ilk maşınlardan biri 1917-ci ildə Edvard Hebernin ixtira etdiyi alman "Enigma" şifratoru idi ("enigma" latın dilindən tərcümədə tapmaca deməkdir).

Rotorlu maşınlar ikinci dünya müharibəsində fəal istifadə edilirdi. "Enigma"dan başqa həmçinin Sigaba (ABŞ), Tupex (Böyük Britaniya), Red, Orange və Purple (Yaponiya) qurğuları da istifadə olunurdu. Rotorlu sistemlərə uğurlu kriptohücumlar yalnız 1940-cı illərin əvvəllərində elektron-hesablama maşınlarının meydana çıxması ilə mümkün oldu.

Şəkil 2.3. Enigma

"Enigma" əvvəlcə bir ox üzərində fırlanan 4 barabandan – diskdən ibarət idi. Diskin hər tərəfində çevrə üzrə 25 elektrik kontaktı yerləşirdi (əlifbada olan hərflərin sayı qədər). Hər iki tərəfdən kontaktları diskin daxilində kifayət qədər təsadüfi şəkildə 255 naqillə birləşdirirlər, onlar simvolların əvəz edilməsini formallaşdırırdı. Disklər birlikdə yığılırdı, onların kontaktları bir-birinə toxunurdu və cərəyanın bütün disk paketindən keçərək yazan qurğuya keçməsini təmin edirdi. Disklərin yan üzündə əlifba həkk olunmuşdu. İşə başlamazdan əvvəl disklər elə dönürdü ki, kod sözü qoşulsun. Klavişə basdıqda və növbəti simvol kodlaşdırıldıqda sol baraban bir addım dönürdü. Disk bir tam dövr etdikdən sonra, elektrik enerjisi sayğacında olduğu kimi növbəti baraban bir addım dönürdü. Şifrləmə prosesində sol diskin kontaktlarından birinə elektrik impulsu daxil olurdu. Barabanlar kontaktlar ilə toxunduqlarından elektrik impulsu dörd diskdən keçərək və dörd sadə əvəzetməyə məruz qalaraq çıxışa düşürdü. İcraçı qurğu (yazı makinası və ya perforator) çıxış diskinin hansı kontaktına elektrik impulsunun daxil olmasına uyğun olaraq şifrlənmiş mətn işarəsini qeyd edirdi. Hər bir şifrləmə taktında heç olmasa bir disk hərəkət etdiyindən, şifrləmənin həyata keçirildiyi əvəzləmə – sadə əvəzetməyə açıq mətnin hər bir simvolu üçün dəyişirdi. Uzunmüddətli açar disklərinin kommutasiyası disklərin daxilində naqillərin birləşməsi idi, o çox az-az dəyişdirilirdi. Şifratorun hər seans dəyişdirilən açarı disklərin başlanğıc vəziyyətinin məcmusu idi. Deşifrləməni

çətinləşdirmək üçün disklər günbəgün ya dəyişdirilirdi, ya da onların yerləri dəyişdirilirdi, yəni 10-20 diskdən ibarət komplektdən 4 disk seçilirdi.

"Enigma" portativ idi (yazı makinası ölçülərində), batareya ilə işləyirdi və ağac futlyarı vardı. Onun ciddi qüsuru ondan ibarət idi ki, o şifrmətni çap edə bilmirdi (yalnız hərfləri bildirən lampaları vardı) və sürətli iş üçün – məlumatın mətnini oxumaq və klaviaturada yığmaq, şifrmətnin közərən hərflərini diqtə etmək və yazmaq üçün üç, hətta dörd adam tələb olunurdu.

Mühazirə üzrə yoxlama sualları

1. Şifrləri hansı əlamətlərə görə təsnif etmək olar?
2. Yerdəyişmə şifri nədir? Yerdəyişmə şifrlərinə misallar söyləyin.
3. Əvəzetmə şifri nədir? Əvəzetmə şifrlərinə misallar söyləyin.
4. Birəlifbalı şifr nədir? Birəlifbalı şifrlərə misallar söyləyin.
5. Çoxəlifbalı şifr nədir? Çoxəlifbalı şifrlərə misallar söyləyin.
6. Birəlifbalı şifrlərin əsas zəifliyi nədir?
7. Sezar şifrində şifrləmə və deşifrləmə funksiyaları necə yazılır?
8. Sürüşmə şifrində şifrləmə və deşifrləmə funksiyaları necə yazılır?
9. Sezar şifrində mümkün açarların sayı nə qədərdir?
10. Affin şifrində şifrləmə və deşifrləmə funksiyaları necə yazılır?
11. Affin şifrində mümkün açarların sayı nəyə bərabərdir?
12. Vijener şifrində şifrləmə və deşifrləmə funksiyaları necə yazılır?
13. Vijener şifri niyə çoxəlifbalı şifr sayılır?
14. Vijener şifrinin hansı variantları var?
15. Vijener şifrində mümkün açar sözlərinin sayı nəyə bərabərdir?
16. Vijener şifrində açar açıq mətndən qısa olarsa, nə edirlər?
17. Hill şifrində şifrləmə və deşifrləmə funksiyaları necə yazılır?
18. Hill şifrinin deşifrlənməsi üçün hansı şərt vacibdir?
19. Vernam şifrinin mütləq dözümlü olması şərtləri hansılardır?
20. «Xətti şifrləmə» nədir?

Mühazirə 3. Klassik şifrlərin kriptoanalizi

3.1. Kobud güc hücumu

Kobud güc hücumu (ing. brute force attack) – açarın bütün mümkün variantlarını yoxlamaq üsulu ilə hücumdur.

Sürüşmə şifrinin **bütün açar variantlarını yoxlamaqla** sındırılmasına nümunə aşağıda göstərilir. Qeyd edək ki, ingilis dilli mətnlər üçün 25 açar (sürüşmə miqdarı) mümkündür.

Şifrmətn: PHHW PH DIWHU WKH WRJD SDUWB

Açar	**Deşifrlənmiş mətn**
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufc uif uphb qbsuz
3	meet me after the toga party
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxopv
7	iaap ia wbpan pda pksw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby niau julns
10	fxxm fx tymxk max mhzt itkmr
11	ewwl ew sxlwj lzw lgys hsjlq
12	dvvk dv rwkvi kyv kfxr grikp
13	cuuj cu qvjuh jxu jewq fqhjo
14	btti bt puitg iwt idvp epgin
15	assh as othsf hvs hcuo dofhm
16	zrrg zr nsgre gur gbtn cnegl
17	yqqf yq mrfqd ftq fasm bmdfk
18	xppe xp lqepc esp ezrl alcej
19	wood wo kpdob dro dyqk zkbdi
20	vnnc vn jocna cqn cxpj yjach
21	ummb um inbmz bpm bwoi xizbg
22	tlla tl hmaly aol avnh whyaf
23	skkz sk glzkx znk zumg vgxze
24	rjjy rj fkyjw ymj ytlf ufwyd
25	qiix qi ejxiv xli xske tevxc

Kobud güc hücumunda hesablamaların mürəkkəbliyi mümkün açarların sayından asılıdır. Cədvəldə bəzi klassik şifrlər üçün mümkün açarların sayı göstərilib.

Cədvəl . Açar fəzasının ölçüsü

Şifr	**Açarların sayı**
Sezar şifri	$\|K\| = 25$
Affin şifri	$\|K\| = 312$
Əvəzetmə şifri	$\|K\| = 26!$= 403291461126605635584000000 ≈ $\approx 4.032914611 \cdot 10^{26}$
Vijener şifri	$\|K\| = 26^m$, m – açarın uzunluğudur. Məsələn, $m = 5$ üçün $\|K\| = 26^5 \approx 1.1 \cdot 10^7$

Açar fəzasının ölçüsü həddindən artıq böyükdürsə, bütün variantların yoxlanması illər, hətta əsrlər ərzində nəticə verməyə bilər. Cədvəldə açar uzunluğundan asılı olaraq kobud güc hücumunun orta müddətləri göstərilib.

Cədvəl . Kobud güz hücumunun orta müddətləri

Açarın uzunluğu	Müxtəlif açarların sayı	1 *şifrləmə/msan* sürətində tələb olunan zaman	10^6 *şifrləmə/msan* sürətində tələb olunan zaman
32 bit	$2^{32} \approx 4.3 \cdot 10^9$	$2^{31} msan \approx 35.8$ dəq	$2.15\ msan$
56 bit (DES)	$2^{56} \approx 7.2 \cdot 10^{16}$	$2^{55} msan \approx 1142$ il	10.01 saat
128 bit (AES)	$2^{128} \approx 3.4 \cdot 10^{38}$	$2^{127} msan$ $\approx 5.4 \cdot 10^{24}$il	$5.4 \cdot 10^{18}$ il
26 simvol (yerdəyişmə)	$26! \approx 4 \cdot 10^{26}$	$2 \cdot 10^{26} msan$ $\approx 6.4 \cdot 10^{12}$il	$6.4 \cdot 10^6$ il

Kriptoanalizin əsas məqsədi – kobud güc hücumundan daha yaxşı üsulun tapılmasıdır.

Kobud güc hücumu kriptoqrafik alqoritmin təhlükəsizliyinin – kripto-dözümün göstəricisi kimi istifadə edilir.

3.2. Tezlik analizi

Birəlifbalı əvəzetmə şifrlərinin əsas kriptoanaliz metodu *tezlik analizi*dir. Tezlik analizini ilk dəfə ərəb filosofu, riyaziyyatçısı Əl-Kindi (801–873) təklif edib (tam adı Abu Yusuf Yaqub ibn Ishaq as-Sabbah al-Kindi).

Hərf tezliyi — müəyyən bir dilin əlifbasındakı hərflərin yazılı dildə orta hesabla hansı tezliklə işləndiyini göstərir. Cədvəldə ingilis dilində hərflərin rastgəlmə tezliyi göstərilib.

Cədvəl . İngilis dilində hərflərin rastgəlmə tezliyi
(Beker H, and Piper F., *Cipher Systems: The Protection of Communications*. Wiley, 1983)

Hərf	A	B	C	D	E	F	G	H	I	J	K	L	M
Tezlik	.082	.015	.028	.043	.127	.022	.020	.061	.070	.002	.008	.040	.024
Hərf	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Tezlik	.067	.075	.019	.001	.060	.063	.091	.028	.010	0.23	.001	.020	.001

Şifrmətndə hərflərin tezlik paylanması müəyyən edilir və ilkin mətn əlifbasında hərflərin tezliyi ilə müqayisə olunur. Şifrmətndə ən böyük tezliyə malik hərflər ilkin mətn əlifbasındakı ən böyük tezlikli hərflərlə əvəzlənirlər. Tezlik analizinin effektiv olması üçün kifayət qədər uzun şifrmətnin olması lazımdır. Şifrmətnin uzunluğu artdıqca, bu metodun uğur ehtimalı artır.

Misal. Verilmiş şifrmətni tezlik analizi ilə açın: IQ IFCC VQQR FB RDQ VFLLCQ NA RDQ CFJWHWZ HR BNNB HCC HWWHBSQVQBRE HWQ VHLQ.

Şifrmətndə hərflərin tezlik analizini aparaq (hərf rast gəldikcə qarşısında 1 yazırıq:

A 1	N 111
B 11111	O
C 111111	P
D 11	**Q 1111111111**
E 1	R 11111
F 1111	S 1
G	T
H 1111111	U
I 11	V 1111
J 1	W 11111
K	X
L 111	Y
M	Z 1

Göründüyü kimi, şifrmətndə ən çox rast gələn hərf Q-dür. Açıqmətnin ingilis dilində olduğunu fərz edək. İngilis dilində E hərfi digərlərindən çox rast gəlir. Digər çox rast gəlinən hərflər – T, A, O, N, R və S-dir. (Hərflərin bu qrupunu «SENORITA» sözü şəklində asan yadda saxlamaq olar.)

Q hərfini "e" ilə əvəz edək (açıqmətn hərflərini kiçik yazırıq).

Ie IFCC VeeR FB RDe VFLLCe NA RDe CFJWHWZ HR BNNB HCC HWWHBSeVeBRE HWe VHLe.

İngilis dili bilgilərimizdən istifadə edərək, "Ie"-nin "we" olduğunu tapa bilərik; "I"-ni "w" il' əvəzləyib "IFCC"-nin "will" olduğunu, sonra "VeeR"-in "meet" təxmin edə bilərik. Sözönlərinə fikir versək, iB→in, RDe→the, NA→on aydınlaşdırmaq olar. Uyğun əvəzləmələrdən sonra şifrmətn bu şəklə düşəcək.

we will meet in the miLLle on the liJWHWZ Ht noon Hll HWWHnSementE HWe mHLe.

Növbəti tapıntılar miLLle→middle, Hll→all, HWe→are ola bilər. Əvəzləmələri etsək:

we will meet in the middle on the liJrarZ at noon all arranSementE are made.

Qalan hərflərin tapılması asandır. "Amerikan kriptoqrafiyasının atası" hesab edilən Uilyam Fridmanın kriptoanalizə verdiyi tərifdə ilk sadalanan əməliyyatın "istifadə edilən dilin" müəyyən edilməsi təsadüfi deyil.

3.3. Kasiski testi

Kasiski testi ilə Vijener şifrinin sındırılması iki addımdan ibarətdir:

1. Açarın uzunluğunun tapılması
2. Tezlik analizi vasitəsilə sürüşmələrin (açarın) müəyyən edilməsi.

Kasiski testində əsas məqsəd Vijener şifrində açarın uzunluğunu tapmaqdır. Kasiski testində şifrmətndə *hərflərin təkrarlanan ardıcıllığı* axtarılır.

Təkrarlanan hərflər ardıcıllığının tapılması faktı iki şey deməkdir:

- Ya açıq mətnin eyni hərf ardıcıllığı açarın eyni hissəsi ilə şifrlənib.
- Ya da müxtəlif ardıcıllığın hərfləri açarın müxtəlif hissələri ilə şifrlənib, lakin onlar eyni şifrlənmiş hərflərlə nəticələnib. İkinci halın ehtimalı çox kiçikdir.

Kasiski testi aşağıdakı addımlarla yerinə yetirilir:

1. Şifrmətn triqramlara bölünür.
2. Hər bir triqram üçün mətndə onunla eyni olan triqramlar axtarılır və tapılmış triqramların mətndəki yerinin nömrəsi (ilk hərfinin) götürülür.
3. Triqram təkrarları arasındakı məsafələrin uzunluqlarının ortaq bölənləri tapılır.

4. Alınmış ortaq bölənlərdən ən kiçiyi, çox yəqin ki, şifrləmə açarının periodudur.
5. Hər bir blokun birinci simvoluna, sonra ikinci, üçüncü və s. periodun sonuna kimi, tezlik analizi tətbiq edilir. Beləliklə, blokun hər bir simvolunun əlifba üzrə sürüşməsi tapılır.

Kasiski testi açarın uzunluğu qısa və mətn kifayət qədər uzun olduqda effektiv olur.

Misal: ABCXYZABCKLMNOPQRSABC şifrmətnində üç dəfə ABC rast gəlir. ABC-nin şifrmətndə yerləri 0, 6 və 18-dir. Onların ortaq bölənləri 2, 3 və 6-dır, buna görə də açarın 2, 3 və ya 6 hərf uzunluğunda olması ehtimalı yüksəkdir.

3.4. Üst-üstə düşmə indeksi

Kasiski testi və üst-üstə düşmə indeksi hər ikisi Vijener şifrinin kriptoanalizi məsələsini həll edir. Kasiski testi ardıcıllıq təkrarlarına əsaslanır, üst-üstə düşmə indeksi isə simvol uyğunluğuna əsaslanır – mətnin özü ilə sürüşdürülmüş versiyası arasında simvol uyğunluğunu ölçür.

«Üst-üstə düşmə indeksi» anlayışını Uilyam Fridman 1920-ci ildə daxil edib. Şifrin birəlifbalı və ya çoxəlifbalı olduğunu müəyyən etmək üçün istifadə edilir. Üst-üstə düşmə indeksi $IC(x)$ ilə işarə edilir (index of coincidence, IC). Əgər $\underline{\boldsymbol{IC(x) = 0.065}}$ ətrafında olarsa, şifr **birəlifbalı**, $\underline{\boldsymbol{IC(x) = 0.038}}$ ətrafında olarsa, şifr **çoxəlifbalı**dır.

Tərif. Fərz edək ki, $x=x_1x_2\ldots x_n$ – n hərfdən ibarət sətirdir. x-in təsadüfi seçilmiş iki hərfinin eyni olması ehtimalına x-in üst-üstə düşmə indeksi deyilir.

Fərz edək ki, A, B, C, …, Z hərflərinin x-də rastgəlmə sayı, uyğun olaraq, n_i ilə işarə edilib.

x-in təsadüfi seçilmiş iki hərfinin AA olması ehtimalı:

$$\frac{n_A}{n} \cdot \frac{n_A - 1}{n - 1}.$$

Eyni qayda ilə, $\frac{n_B}{n} \cdot \frac{n_B-1}{n-1}$, $\frac{n_C}{n} \cdot \frac{n_C-1}{n-1}$, …, $\frac{n_Z}{n} \cdot \frac{n_Z-1}{n-1}$.

$$IC(x) = \frac{n_A}{n} \cdot \frac{n_A - 1}{n - 1} + \frac{n_B}{n} \cdot \frac{n_B - 1}{n - 1} + \frac{n_C}{n} \cdot \frac{n_C - 1}{n - 1} + \cdots + \frac{n_Z}{n} \cdot \frac{n_Z - 1}{n - 1}$$

Hərflərdən nömrələrinə keçsək:

$$IC(x) = \frac{\sum_{i=0}^{25} n_i(n_i - 1)}{n(n - 1)}$$

Üst-üstə düşmə indeksi – Birəlifbalı şifrlər. Birəlifbalı şifrlər hərflərin «təbii tezliyini» pozmur:

$$IC(x) = .082 \times .082 + .015 \times .015 + .028.\times 028 + \cdots + .001 \times .001$$

$$IC(x) \approx 0.0656010$$

Üst-üstə düşmə indeksi – Çoxəlifbalı şifrlər. Çoxəlifbalı şifrlər «təbii tezliyi» pozur – ehtimalları təxminən eynidir: $\frac{1}{26}$.

$$IC(x) = \underbrace{\left(\frac{1}{26}\times\frac{1}{26}\right)+\left(\frac{1}{26}\times\frac{1}{26}\right)\ldots+\left(\frac{1}{26}\times\frac{1}{26}\right)}_{26\ toplanan} = \frac{1}{26} \approx 0.038$$

3.5. Vijener şifrinin sındırılması

Üst-üstə düşmə indeksindən istifadə etməklə Vijener şifrinin sındırılması iki mərhələdən ibarətdir:

1. Açar sözün uzunluğu tapılır.
2. Açar sözün özü tapılır.

Açarın uzunluğunun tapılması alqoritmi aşağıdakı kimidir:

1. Açar sözün uzunluğu $n = 2$ fərz olunur.
2. Şifrmətni n qrupa elə bölürlər ki, eyni qrupdakı hərflər açarın eyni hərfi ilə şifrlənmiş olsun.
3. Hər qrupun IC indeksi hesablanır.
4. Bütün qrupların ortalama IC indeksi hesablanır.
5. Əgər ortalama IC indeksi baxılan dil üçün olan indeksə «yaxındırsa», onda n açarın uzunluğu qəbul olunur.
6. Əks halda, n-i bir vahid artırıb Addım 2-yə keçirlər.

Misal. Aşağıdakı şifrmətn üçün üst-üstə düşmə indekslərinin hesablanması
JBORP WXINF FTQMC MLWCX JBNEV ODLNA FXRQS EWOBI NOMLJ GUIMH YGXAZ YCSDL NJYPD AOSOO VMCHV IASNV DPLIA PFPUH RIVOD LNAFX RQSEW TSRMH RINEE ECWYR XTDLN GEVOO MIYSB JXFWW JHBMY ZOMWH OFAOV EWRMS VDEXN GDLNJ YPDAO XQSZL HGSGR GDHDB UWCVO FJZVM WOZSX ZYJFO DIAOX HVSFW DFOWC VOZXZ EQNCP XQSNM BDVEL SNAJH OVCVO IWUSR NSBPX CUWDD DLNJF PDSSR QWCFX CUSOZ SXCZO VNRBY FPOVK OVPEC VYVSC

Aşağıdakı cədvəl https://thezero.org/coincidence-index/ saytındakı onlayn kalkulyator istifadə edilməklə alınmışdır (sadəlik üçün vergüldən sonra 4 rəqəm saxlanıb).

Cədvəldən göründüyü kimi, açarın ən ehtimallı uzunluğu 4-dür.

Açarın tapılması

- Hər bir qrupa tezlik analizi tətbiq edilir.
- Hər bir qrupda sürüşmə tapılır, sürüşməyə uyğun hərf açar hərfidir (məsələn, sürüşmə 2-dirsə, açar hərfi C olacaq).

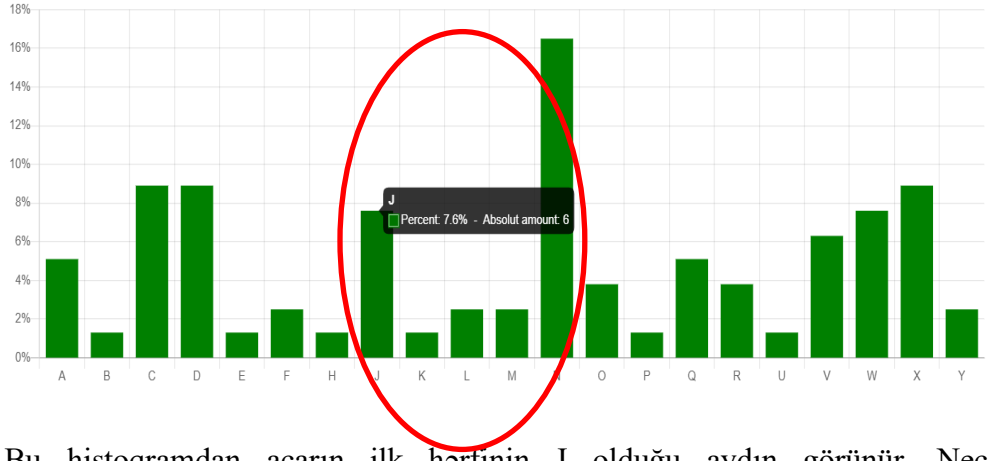
Açarın uzunluğu	Qruplar üçün IC indeksləri	Ortalama IC
1	0.0423	0.0423
2	0.0550; 0.0438	0.04945
3	0.0446; 0.0408; 0.0395	0.04169
4	**0.0658; 0.0613; 0.0658; 0.0606**	**0.06343**
5	0.0322; 0.0378; 0.0512; 0.0394; 0.0435	0.04086
6	0.0529; 0.0406; 0.0537; 0.0460; 0.0535; 0.03846	0.04646
7	0.0505; 0.0444; 0.0404; 0.0444; 0.0353; 0.0404; 0.0404	0.04227
8	0.0653; 0.0538; 0.0641; 0.0634; 0.0566; 0.0580; 0.0607; 0.0526	0.05935

Yuxarıdakı şifrmətnin kriptoanalizini davam etdirərək, açarı tapaq. Bunun üçün şifrmətni 4 qrupa bölüb tezlik analizini tətbiq etməliyik. Qrup 1 açarın birinci hərfi ilə, Qrup 2 açarın ikinci hərfi ilə və s. şifrlənmiş hərflərdən ibarətdir. Qrupları şifrmətnin elementlərini 4-4 olmaqla qruplara bölüb 4 sətri olan cədvələ yazmaqla əldə etmək olar:

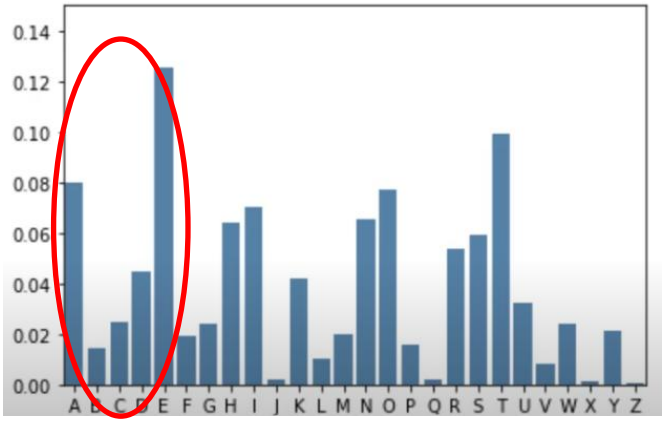
Qrup 1	J	P	N	Q	L	J	...
Qrup 2	B	W	F	M	W	B	...
Qrup 3	O	X	F	C	C	N	...
Qrup 4	R	I	T	M	X	E	...

Qrup 1: JPNQLJVNRWNJMXCNDOCADAUVNRWMNCXNOYXJYWA WVNNDQHRDCJWXFAVDCXNQBLJCWNXDNDQXOCNFKEV
Qrup 2. BWFMWBOAQOOGHASJAOHSPPHOAQTHEWTGOSFHZHO RDGJASGGBVZOZOOSFVZCSDSHVUSCDJSWCZZRPOCS
Qrup3. OXFCCNDFSBMUYZDYOVVNLFRDFSSREYDEMBWBOOVM EDYOZSDUOVZYDXFOOEPNVNOOSBUDFSCUSOBOVVC
Qrup 4. RITMXELXEILIGYLPSMIVIPILXERIERLVIJWMMFESXLPXL GHWFMSJIHWWZQXMEAVIRPWLPRFSXVYVPY

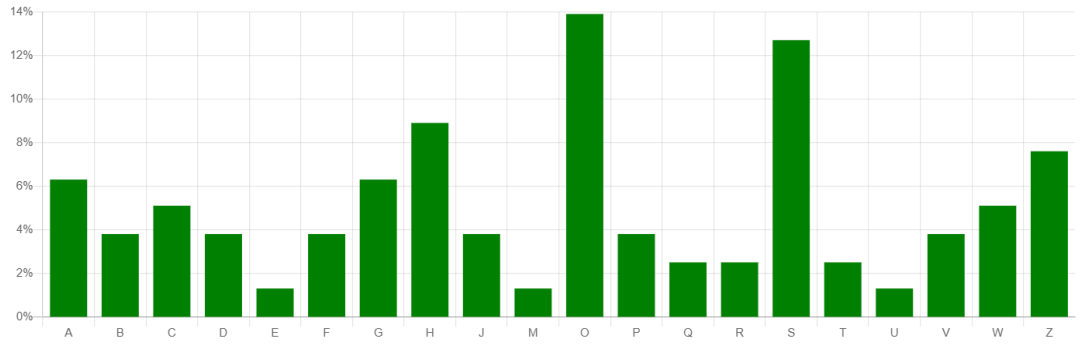
Qrup 1-ə tezlik analizini tətbiq edək:



Bu histoqramdan açarın ilk hərfinin J olduğu aydın görünür. Necə göründüyünü izah etmək üçün ingilis dilində hərflərin standart tezlik histoqramında "əlamətdar" sütun şəkillərinə oxşar şəkil elementini axtarmaq lazımdır. Ən asan gözədəyən hissə A hərfi ilə E hərfi arasındakı histoqram hissəsidir – solda və sağda iki hündür sütun, onların arasında üç qısa sütun.



Digər qruplara da tezlik analizini tətbiq edərək histoqramlardan açar hərflərinin uyğun olaraq O, K və E olduğunu müəyyən edə bilərik. Beləliklə, Açar söz JOKE olur.



Mühazirə üzrə yoxlama sualları

1. Kobud güc hücumu nədir?
2. Sezar şifrini sındırmaq üçün neçə açarı yoxlamaq lazımdır?
3. Affin şifrində mükün açarların sayı nəyə bərabərdir?
4. Əvəzetmə şifrlərində mümkün açarların sayı nəyə bərabərdir?
5. Vijener şifrinə kobud güc hücumunda neçə açarı yoxlamaq lazımdır?
6. DES-də mümkün açar variantlarının sayı nəyə bərabərdir?
7. AES-də açarın uzunluğu 128 bit olarsa, mümkün açarların sayı necə olar?
8. Kriptoanalizin əsas məqsədi nədir?
9. Tezlik analizi hücumu hansı şifrlərə tətbiq edilir?
10. Hansı şifr tezlik analizinə qarşı daha dayanıqlıdır?
11. Tezlik analizinin effektiv olması üçün hansı şərt lazımdır?
12. Kasiski testində əsas məqsəd nədir?
13. Kasiski testinin birinci addımında nəyi tapırlar?
14. Kasiski testinin ikinci addımında nəyi müəyyən edirlər?
15. Üst-üstə düşmə indeksi nəyə deyilir?
16. Birəlifbalı əvəzetmə şifrləri üçün üst-üstə düşmə indeksi nəyə bərabərdir?
17. Çoxəlifbalı əvəzetmə şifrləri üçün üst-üstə düşmə indeksi nəyə bərabərdir?
18. Üst-üstə düşmə indeksi əsasında Vijener şifrinin sındırılması hansı addımlardan ibarətdir?
19. Üst-üstə düşmə indeksi əsasında açar uzunluğunun tapılması alqoritminin addımları nədədən ibarətdir?
20. Üst-üstə düşmə indeksi çox yüksək olarsa, bu nəyi göstərə bilər?
21. Kasiski testi ilə üst-üstə düşmə indeksi yanaşmasının fərqi nədir?
22. Üst-üstə düşmə indeksi hesablanarkən şifrmətni qruplara necə bölürlər?

Mühazirə 4. Axın şifrləri

4.1. Axın şifrlərinin tərifi

Kriptosistemləri müxtəlif əlamətlərə görə təsnif etmək olar: qorunan informasiyanın növünə görə (mətn, nitq, video və s.) açarların istifadəsinə görə (simmetrik, asimmetrik, hibrid), konstruktiv prinsiplərə görə (blok və axın şifrləri), kriptoqrafik dözümə görə və s.

Axın şifrləri açıq mətni şifrmətnə simvol-simvol (bir bit və ya bayt), blok şifrləri isə sabit ölçülü bloklarla çevirir. Axın şifrləri xüsusi rejimlər tətbiq etməklə asanlıqla blok şifrlərindən almaq olar.

Axın şifrlərində şifrləmə belə aparılır. Açar axını generatoru bitlər axını verir: $k_1, k_2,\ldots,k_i$. Bu bitlər axını və açıq mətnin $p_1, p_2, \ldots, p_i$ bitlər axını 2 moduluna görə toplanır, nəticədə şifrmətnin bitlər axını alınır:

$$c_i = p_i \oplus k_i$$

Deşifrləmə zamanı 2 modulu üzrə toplama əməli şifrmətnin bitlər axını və elə həmin açar axını üzərində yerinə yetirilir:

$$p_i = c_i \oplus k_i$$

Misal. Tutaq ki, açıq mətn IF sözüdür, onun ASCII kodu 1001001 1000110 olacaq. Açar isə 1010110 0110001 olsun. Şifrmətn açıq mətnlə açarın bitlərini 2 moduluna görə toplamaqla tapılır.

Şifrləmə:

Açıq mətn	1001001 1000110
Açar	1010110 0110001
Şifrmətn	0011111 1110111

Deşifrləmə:

Şifrmətn	0011111 1110111
Açar	1010110 0110001
Açıq mətn	1001001 1000110

Vernam şifrini tarixən ilk axın şifri hesab etmək olar. Axın şifrlərinin blok şifrlərindən vacib üstünlüyü yüksək şifrləmə sürətidir, şifrləmə/deşifrləmə praktiki olaraq real zamanda təmin edilir. Bu cəhət də onların istifadə sahəsini müəyyənləşdirir – istehlakçıya operativ çatdırılmasını tələb edən verilənlərin – məsələn, audio və video informasiyanın şifrlənməsidir.

Axın şifrlərinə nümunələr: A5 (GSM-də şifrləmə üçün istifadə edilir), ZUC (LTE və 5G mobil şəbəkələri üçün hazırlanıb), RC4 (naqilsiz şəbəkələrdə WEP və WPA) protokollarında istifadə edilir), SEAL, ORYX, WAKE və s.

4.2. Sinxron və öz-özünə sinxronlaşan axın şifrləri

Axın şifrləri, adətən, *sinxron* və *öz-özünə sinxronlaşan* sistemlərə bölünür. Sinxron axın şifrlərində açar axını açıq mətn və şifrmətn axınlarından asılı olmadan

generasiya olunur, hər bir element (bit, simvol, bayt və s.) digər elementlərdən asılı olmadan şifrlənir.

Sinxron şifrlərin səhvi yaymamaq xassəsi var, şifrmətnin təhrif olunmuş simvolu yalnız açıq mətnin uyğun simvoluna təsir edir.

Sinxron axın şifrlərində şifrmətn ardıcıllığında elementlərin daxil edilməsi və ya silinməsi yolverilməzdir, çünki sinxronlaşma pozulur və sonrakı bütün elementlər səhv deşifrlənir.

Adətən, sinxronlaşdırma üçün ötürülən məlumata xüsusi markerlər daxil edilir. Başqa bir həll isə – əvvəlcədən razılaşdırılmış şərtlərə əsasən həm şifratoru, həm də deşifratoru yenidən başlanğıc vəziyyətə gətirməkdir.

Öz-özünə sinxronlaşan axın şifrləri sinxronlaşdırma itdikdən sonra informasiyanı bərpaetmə xassəsinə malikdir, buna görə diplomatik, hərbi və sənaye rabitə sistemlərində ən geniş yayılmış şifrləmə metodudur. Öz-özünə sinxronlaşan axın şifrlərinin ən geniş yayılmış iş rejimi – şifrmətnlə əks-əlaqə rejimidir.

Öz-özünə sinxronlaşan axın şifrlərində şifrlənmiş hər simvol (ardıcıl ötürmədə bit) *k* açarından və şifrmətnin yalnız axırıncı *n* simvolundan asılı olur. Axın şifrinin *n* mərtəbəlik yaddaşını müxtəlif üsullarla reallaşdırmaq olar, onlardan ən sadəsi *n*-mərtəbəli sürüşmə registrindən istifadə etməkdir.

4.3. Açar axınının generasiyası

Axın şifrinin dözümü tamamilə açar axını generatorunun xassələrindən asılıdır. Açar axını generatorunun çıxışı təsadüfiyə nə qədər yaxındırsa, kriptoanalitikin hücumu bir o qədər çətin (əməktutumlu) olacaq. Əgər açar axını generatoru sonsuz sıfırlar sətri verirsə, şifrlənmiş mətn açıq mətnlə üst-üstə düşəcək və şifrləmə mənasız olacaq.

İdealda gizli açar axını tamamilə təsadüfi olmalıdır. Bu tam *mükəmməl şifr* – birdəfəlik bloknot (ing. One time pad) şifrinin binar variantını verməlidir. Praktikada bu yanaşma birdəfəlik bloknotda olan iki problemdən əziyyət çəkir: tam təsadüfi verilənlər ardıcıllığının generasiyası və şifrləmə açarlarının paylanması.

Həqiqətdə, kompüter tam təsadüfi açar axını yarada bilməz, yalnız müxtəlif alqoritmlərdən istifadə etməklə "psevdotəsadüfi" açar axını yarada bilər.

Kompüterin istifadə etdiyi psevdotəsadüfi açar axını yaratma proseduruna və ya "alqoritminə" başlanğıc kimi bitlərin nisbətən qısa ardıcıllığını – "toxum" (ing. seed) vermək olar ki, müxtəlif toxumlar müxtəlif açar axını versin. Alqoritm “qeyri-xətti” olmalıdır ki, hətta bir biti dəyişdirilən toxum tamamilə fərqli açar axını törətsin. Toxumu şifrləmə açarı kimi də istifadə etmək olar.

4.4. Xətti konqruent generatorlar

$x_1, x_2, \ldots \in A$, $A=\{0, 1, \ldots, n-1\}$ psevdotəsadüfi ardıcıllığını

$$x_{t+1} = (ax_t + b) \bmod n,\ t=0, 1, \ldots$$

rekurrent münasibətlərin köməyi ilə törədən proqram generatoru parametrləri (x_0, a, c, n) olan *xətti konqruent generator* adlanır.

Bu generatorun parametrlərinin aşağıdakı mənaları var:

- $x_0 \in A$ – *ilkin* və ya *start qiyməti* (ing. seed);
- $a \in A \setminus \{0\}$ – sıfır olmayan *vuruq*;
- $b \in A$ – *artım*;
- $n \in N$ – *modul*, A əlifbasının gücünə bərabərdir.

$b=0$ olarsa, multiplikativ konqruent generator, $b \neq 0$ olarsa, qarışıq konqruent generator adlanır.

Alınan ardıcıllıq x_0 başlanğıc qiymətinin seçilməsindən asılıdır və onun müxtəlif qiymətlərində təsadüfi ədədlərin müxtəlif ardıcıllıqları alınır. Eyni zamanda, x_t ardıcıllığının bir çox xassələri düsturdakı əmsalların seçilməsi ilə müəyyən edilir və seçilmiş başlanğıc qiymətindən asılı deyil. Aydındır ki, xətti konqruent alqoritmlə generasiya olunan ədədlər ardıcıllığı periodikdir və onun periodu n–i aşmır.

Misal. $x_0= a=b=7$ və $m=10$ olduqda xətti konqruent generator periodu 4 olan aşağıdakı ardıcıllığı verir:

7, 6, 9, 0, 7, 6, 9, 0, …

4.5. Blum-Blum-Şub generatoru

BBS generatoru hal-hazırda bitlər ardıcıllığı generasiya edən ən sadə və səmərəli generatordur. İxtiraçıların familiyası ilə adlanır (Blum, Blum, Shub – BBS). Bəzən onu kvadratik qalıqlı generator da adlandırırlar. Bu generator n modulu üzrə *kvadratik çıxıq*lardan istifadə edir.

Əvvəlcə 4-ə bölündükdə qalıqda 3 verən iki böyük p və q sadə ədədləri seçilir: $p \equiv 3\ mod\ 4,\ q \equiv 3\ mod\ 4$. Sonra $n = pq$ hasili hesablanır. Belə n ədədi Blum ədədi adlanır.

n ilə qarşılıqlı sadə olan x_0 ədədi seçilir: ƏBOB(n, x_0) = 1. x_0 generatorun başlanğıc qiymətidir. BBS generatoru b_i bitlər ardıcıllığını aşağıdakı alqoritmin köməyi ilə alır:

for i=0 to ∞
  $x_{i+1} = (x_i)^2 \bmod n$
  $b_i = x_i \bmod 2$

Aşağıdakı cədvəldə BBS alqoritmi ilə alınan ardıcıllığın bir hissəsi göstərilib (Period 6-ya bərabərdir). Burada n=133 (p=7, q=19, n=pq), x_0 = 4 götürülüb. $x_1= 4^2 \bmod 133 = 16$.

i	0	1	2	3	4	5	6	...
x_i	4	16	123	100	25	93	4	...
b_i	0	0	1	0	1	1	0	...

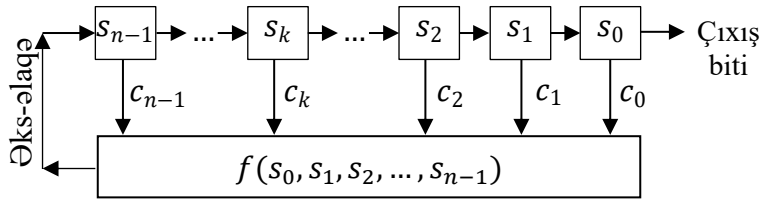
BBS generatorunun periodunu λ (Karmaykl) funksiyasının köməyi ilə qiymətləndirmək olar. Baxılan halda, $\lambda(n) = \lambda(pq) = ƏKOB(p-1, q-1)$ kimi hesablanır. Əgər x_0 başlanğıc qiyməti n moduluna görə kvadratik çıxıqdırsa, onda BBS ardıcıllığı b-nin periodu $per(b)$ aşağıdakı şərti ödəyir: $per(b)|\lambda(\lambda(n))$. Bu misalda $\lambda(133) = \lambda(7 \cdot 19) = ƏKOB(6,18) = 18$ və $\lambda(18) = 6$.

Maraqlıdır ki, i-ci biti almaq üçün (i–1) vəziyyətdən keçmək lazım deyil. Əgər p, q məlumdursa, onu dərhal hesablamaq olar: b_i – x_i-nin kiçik qiymətli bitidir, burada $x_i = x_0^k$, $k = 2^i \bmod((p-1)(q-1))$.

n ədədini açıq etmək olar, hər kəs bu generatordan istifadə edərək bitləri generasiya edə bilər. Əgər kriptoanalitik n-i sındırmasa, sonrakı biti qabaqcadan söyləyə bilməz. Daha dəqiq desək, BBS generatoru sola və sağa proqnoz edilməzdir (yəni ardıcıllığın müəyyən hissəsinə görə əvvəlki və ya sonrakı biti bilmək olmaz).

4.6. Xətti əks-əlaqəli sürüşmə registrləri

Praktikada istifadə edilən axın şifrlərinin bir çoxu əks-əlaqəli sürüşmə registrlərinə (Feedback Shift Register, FSR) əsaslanır. Sürüşmə registri açar ardıcıllığının generasiyası üçün tətbiq edilir. Əks-əlaqəli sürüşmə registri iki hissədən ibarətdir: sürüşmə registri və əks-əlaqə funksiyası (şəkil 4.1).



Şəkil 4.1. Əks-əlaqəli sürüşmə registri

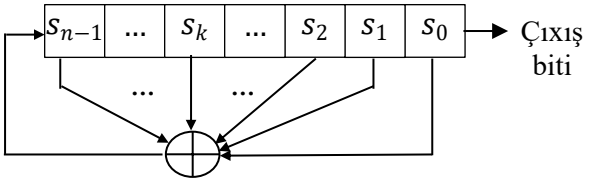
Sürüşmə registri bitlər ardıcıllığından ibarətdir. Bitlərin sayı sürüşmə registrinin uzunluğu ilə müəyyən edilir. Əgər uzunluq n bitə bərabərdirsə, registr *n-bitlik sürüşmə registri* adlanır.

Növbəti biti generasiya etmək üçün hər taktda aşağıdakı əməllər yerinə yetirilir:

1. Registrin bütün bitləri bir mövqe sağa sürüşdürülür, ən kiçik bit («kənara çıxan») generatorun çıxışı olur;
2. Ən böyük bit əks-əlaqə funksiyasının qiyməti ilə yüklənir, əks-əlaqə funksiyasının arqumentləri registrin ya bütün bitləri, ya da yalnız seçilmiş mövqelərdəki bitləridir.

Təkrarlanma baş verənə qədər alınan ardıcıllığın uzunluğu *registrin periodu* adlanır.

Əks-əlaqəli sürüşmə registrinin ən sadə növü xətti əks-əlaqəli sürüşmə registridir (Linear Feedback Shift Register, LFSR) (şəkil 4.2).



Şəkil 4.2. Xətti əks-əlaqəli sürüşmə registri

Tutaq ki, uzunluğu n-ə bərabər giriş ardıcıllığı (s_0, s_1, …, s_{n-1})-dir. LFSR üçün əks-əlaqə

$$f(s) = \sum_{i=0}^{n-1} c_i s_i$$

ilə təyin edilən $f(s_0, s_1, \ldots, s_{n-1})$ xətti funksiyasıdır, burada c_0, c_1, …, c_n sabit əmsallardır, "+" ilə 2 moduluna görə toplama işarə edilib. Registrin sıfırdan fərqli c_0, c_1, …, c_n əmsallarına uyğun olan bitləri *budaq ardıcıllığı* adlanır. LFSR-in çıxışı s_0, s_1, …, s_{n-1} ilkin qiymətləri və

$$s_{k+n} = \left(\sum_{i=0}^{n-1} c_i s_{i+k} \right), k \geq 0$$

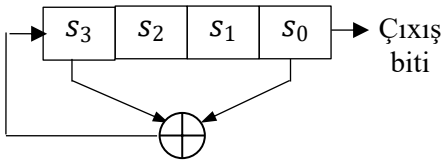
rekurrent münasibəti ilə təyin edilir. Bu münasibət

$$\sum_{i=0}^{n} c_i s_{i+k} = 0, k \geq 0$$

bərabərliyinə eynigüclüdür, burada tərifə görə $c_n = 1$.

LFSR-i istənilən bitlər ardıcıllığı ilə yükləmək olar. Bu ardıcıllıq *ilkin yükləmə* və ya açar adlanır.

Şəkil 4.3-də birinci və dördüncü bitlərindən budaq olan 4-bitlik LFSR göstərilir.



Şəkil 4.3. 4-bitlik LFSR

Əgər onu 1111 qiyməti ilə ilkin yükləsək, təkrarlanma alınana qədər registr aşağıdakı daxili vəziyyətləri qəbul edəcək:

1111 0111 1011 0101 1010 1101 0110 0011 1001 0100 0010 0001 1000 1100 1110

Kiçik mərtəbənin bitləri sətri çıxış ardıcıllığı olacaq:

111101011001000 . . .

n-bitlik LFSR 2^n–1 sayda daxili vəziyyətdən birində ola bilər. Bu o deməkdir ki, nəzəri olaraq belə registr periodu $2^n - 1$ bit olan psevdotəsadüfi ardıcıllıq generasiya edə bilər. (Daxili vəziyyətlərin sayı və period 2^n–1-ə bərabərdir, çünki LFSR-in sıfırlarla yüklənməsi ona gətirir ki, sürüşmə registri sıfırların sonsuz ardıcıllığını verir ki, bu da tamamilə faydasızdır). Yalnız bəzi çıxış ardıcıllıqlarında LFSR $2^n - 1$ daxili vəziyyətin hamısından dövri olaraq keçir. Belə LFSR-lər maksimal perioda malikdir. Alınan nəticə *M-ardıcıllıq* adlanır. Konkret LFSR-in maksimal perioda malik olması üçün budaq ardıcıllığı ilə əlaqələndirilən çoxhədli (xarakterik çoxhədli) 2 moduluna görə *primitiv* olmalıdır.

LFSR-in *xarakteristik çoxhədlisi*

$$f(x) = c_0 + c_1x + \ldots + c_{n-1}x^{n-1} + x^n = \sum_{i=0}^{n} c_i x^i$$

çoxhədlisi kimi təyin edilir, burada tərifə görə c_n=1. Xarakteristik çoxhədlidən istifadə etməklə ilkin ardıcıllığı bilmədən LFSR-in periodunu müəyyən etmək olar.

Məsələn, $x^{32} + x^7 + x^5 + x^3 + x^2 + x + 1$ çoxhədlisi 2 moduluna görə primitivdir. Bu çoxhədliyə maksimal periodlu LFSR terminlərində nəzər salaq. Çoxhədlinin dərəcəsi LFSR-in uzunluğunu verir. Çoxhədlinin sərbəst həddi həmişə 1-ə bərabərdir və onu buraxmaq olar. Çoxhədlinin formal dəyişəninin dərəcələri 0-dan (sıfırdan) savayı sürüşmə registrinin sol kənarından saymaqla budaq ardıcıllığını verir, yəni çoxhədlinin kiçik dərəcəli hədləri registrin sağ kənarına yaxın yerləşmiş mərtəbələrə uyğun gəlir.

Baxılan 32-bitlik sürüşmə registri üçün yeni bit 32-ci, 7-ci, 5-ci, 3-cü, 2-ci və 1-ci bitlərin XOR-u vasitəsi ilə generasiya edilir. Alınan LFSR maksimal uzunluğa malikdir, təkrarlanma alınana qədər $2^{32} - 1$ müxtəlif qiymətlərdən dövri keçəcək. Özlüyündə LFSR psevdotəsadüfi ardıcıllığın yaxşı generatoru olsa da, bir sıra arzuolunmaz qeyri-təsadüfi xassələrə malikdir. Uzunluğu *n* olan LFSR üçün daxili vəziyyət generatorun əvvəlki *n* çıxış bitlərindən ibarətdir. Hətta əks-əlaqə sxemi məxfi saxlansa belə, onu generatorun 2*n* çıxış bitinə görə Berlekemp-Messi alqoritminin köməyi ilə müəyyən etmək olar.

Bundan əlavə, bu ardıcıllığın ardıcıl gələn bitlərindən istifadə etməklə düzəldilən böyük təsadüfi ədədlər güclü korrelyasiya edirlər və bir sıra tətbiqi proqramlar üçün təsadüfiliyi təmin etmirlər.

Praktikada uzun və mümkün olduqca təsadüfi açar axınlarının törədilməsi üçün müvafiq LFSR sxemləri sahəsində intensiv tədqiqatlar aparılıb. Hazırda ən yaxşı LFSR də təhlükəsizliyi təmin etmək üçün artıq yaramır, buna görə praktikada açar

axını törətmək üçün bir neçə LFSR bir-biri ilə mürəkkəb, qeyri-xətti qaydada birləşdirilir. LFSR-lərin belə birləşmə sxemləri "sürüşmə registrləri kaskadı" adlanır.

Mühazirə üzrə yoxlama sualları

1. Axın şifrləri ilə blok şifrləri arasındakı əsas fərq nədir?
2. Axın şifrlərinin blok şifrlərindən vacib üstünlüyü nədir?
3. Axın şifrlərini hansı üsulla blok şifrlərindən almaq olar?
4. Hansı şifri tarixən ilk axın şifri hesab etmək olar?
5. Geniş yayılmış axın şifrlərinə nümunələr söyləyin.
6. Açar axını nədir?
7. Açar axını axın şifrlərində hansı funksiyanı yerinə yetirir?
8. Sinxron və asinxron axın şifrləri arasında hansı fərq var?
9. Sinxron axın şifrlərinin səhvi yaymamaq xassəsi nədir?
10. Sinxron axın şifrlərində sinxronlaşma pozulsa, nə baş verər?
11. Praktikada sinxronlaşdırma üçün hansı üsullar istifadə edilir?
12. Öz-özünə sinxronlaşan axın şifrlərinin bərpaetmə xassəsi nədir?
13. Öz-özünə sinxronlaşan axın şifrlərində şifrlənmiş hər simvol nədən asılıdır?
14. Axın şifrinin təhlükəsizliyi nədən asılıdır?
15. İdealda gizli açar axını necə olmalıdır?
16. Axın şifrlərində açarın təkrar istifadəsi hansı təhlükəyə səbəb olur?
17. Əks-əlaqəli sürüşmə registri hansı hissələrdən ibarətdir?
18. Əks-əlaqəli sürüşmə registri necə işləyir?
19. LFSR nədir və necə işləyir?
20. *n*-bitlik LFSR-in daxili vəziyyətlərinin sayı nəyə bərabərdir?
21. LFSR-in periodu nədir və hansı şərtlərdə maksimal olur?
22. LFSR-in xarakteristik çoxhədlisi nədir?
23. Primitiv çoxhədli nədir?
24. LFSR-in xarakteristik çoxhədlisi nə üçün primitiv olmalıdır?
25. BBS generatoru hansı çıxıqlardan istifadə edir?

Mühazirə 5. DES (Data Encryption Standard)

DES alqoritmi 1970-ci illərin əvvəlində IBM şirkəti tərəfindən yaradılmış "Lucifer" şifrinin əsasında işlənmişdi. ABŞ Milli Standartlar və Texnologiyalar İnstitutu (NIST) bu alqoritmi 1977-ci ildə müəyyən düzəlişlərlə DES standartı kimi qəbul etdi (FIPS Pub 46).

DES blok şifridir, şifrləmə blokunun uzunluğu 64 bitdir. DES 56 bit uzunluqda açardan istifadə edir (8-baytlıq). Hər baytın 7 biti istifadə edilir, baytın 8-ci biti cütlüyə nəzarət üçün istifadə edilir. Raundların sayı 16-dır.

DES şifrinin ABŞ standartı kimi qəbul edilməsini kriptoqrafiya tarixində "birinci inqilab" adlandırırlar. 50 il vaxt keçməsinə, şifrin köhnəlməsinə və sındırılmasına baxmayaraq, onun və bir çox digər şifrlərin əsasında dayanan kriptoqrafiya prinsiplərini öyrənmək üçün əvəzolunmaz bir obyektdir.

5.1. Blok şifrilərinin dizaynı prinsipləri

Blok şifrləri açıq mətni eyni uzunluqda (məsələn, *b* bit) bloklara bölür və eyni kriptoqrafik çevirməni hər bir bloka tətbiq edir. Adətən, uzunluğu 64 bit olan bloklardan istifadə edilir. Axırıncı bloku müəyyən üsulla tam bloka doldururlar. Adətən, axırıncı blok (100…0) ikilik vektoru ilə doldurulur.

Blok və axın şifrləri müxtəlif cür reallaşdırılır. Blok şifrlərini proqramla reallaşdırmaq asandır, çünki onlar kompüter üçün rahat verilənlər blokları ilə əməliyyat aparır. Axın şifrləri daha çox aparat realizəsi üçün yarayır. Bəzi rejimlərdə blok şifrləri axın şifrləri kimi effektiv olurlar.

K. Şennon dözümlü şifrlərin arzu olunan xarakteristikalarını təsvir etmək üçün *səpilmə* və *qarışdırma* anlayışlarını daxil etmişdi.

- ***Səpilmə*** (ing. Diffusion) – açıq mətnin və ya açarın bir işarəsinin təsirinin şifrmətnin çox sayda işarəsinə yayılmasıdır. Bu açıq mətnin statistik xassələrini gizlədir. Sadə diffuziya elementi **bit permutasiyası**dır, bu DES-də tez-tez istifadə edilir.
- ***Qarışdırma*** (ing. Confusion) − şifrmətnlə açar arasındakı statistik asılılığın maksimal dərəcədə çətinləşdirilməsidir. Qarışdırmanı əldə etmək üçün ümumi element **əvəzləmə**dir, onu DES və AES-də tapmaq olar.

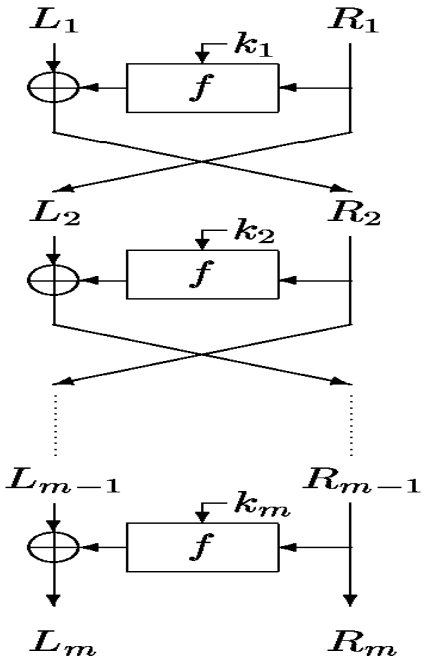
Kriptoqrafik cəhətdən dözümlü şifrlərin qurulması üçün K.Şennon əvəzləmə (Substitution) və yerdəyişmə (Permutation) çevirmələrini növbə ilə istifadə etməyi təklif edirdi. Hazırda praktik olaraq bütün müasir blok şifrləri bu ideyadan istifadə edir, yəni *kompozsiya şifrləridir* − sadə çevirmələrin kompozisiyasından ibarətdirlər: $\mathbf{F}=\mathbf{F}_1 \circ \mathbf{F}_2 \circ \mathbf{F}_3 \circ \mathbf{F}_4 \circ .. \circ \mathbf{F}_n$, burada **F** – şifrin çevirməsi, $\mathbf{F_i}$ – sadə çevirmələrdir, onları *i*-ci *şifrləmə raundu* da adlandırırlar. Çevirmə özlüyündə təkbaşına lazımi xassələri

təmin etməyə də bilər, lakin onların zənciri tələb olunan nəticəni almağa imkan verir. Əgər eyni bir çevirmə istifadə edilirsə, yəni i üçün $\boldsymbol{F_i}$ sabitdirsə, onda belə kompozisiya şifri *iterativ şifr* adlanır.

İterativlik prinsipi kriptoqrafik çevirmələrin yaradılmasında əsas yanaşmadır və açıq mətnin bir blokunun dəfələrlə, bir neçə dövrdən (raunddan) ibarət emalından ibarətdir. Verilənlər hər raundda verilmiş məxfi açardan alınmış raund açarının iştirakı ilə xüsusi çevirməyə məruz qalır. Raundların sayının seçilməsi blok şifrinin kriptoqrafik dözümü və realizəsinin effektivliyi tələbləri ilə müəyyən edilir. Bir qayda olaraq, raundların sayı nə qədər çoxdursa, kriptoqrafik dözüm bir o qədər yüksək və effektivlik aşağıdır və tərsinə.

5.2. Feystel şəbəkəsi

Feystel şəbəkəsi əsasında qurulan blok şifrləri daha geniş yayılıb. Şifrləmə zamanı açıq mətn bloku iki bərabər hissəyə – sağ və sol hissəyə bölünür. Aydındır ki, bu zaman blokun uzunluğu cüt olmalıdır. Hər raundda hissələrdən biri f funksiyasının və ilkin məxfi açardan alınmış k_i raund açarının köməyi ilə çevrilir. Əməliyyatın nəticəsi 2 modulu üzrə o biri hissə ilə toplanır. Sonra sol və sağ hissələrin yeri dəyişdirilir. Hər raundda çevirmələr eynidir, ancaq sonuncu raundda yerdəyişmə edilmir. Deşifrləmə proseduru şifrləmə proseduruna analojidir, lakin k_i raund açarları tərs ardıcıllıqda seçilir (Şifrləmə və deşifrləmənin yalnız açar cədvəli ilə fərqlənməsi Feystel şəbəkəsinin üstünlüklərindən biridir).



Feystel şəbəkəsi

DES-in kriptoqrafik çevirmə prosesi üç əsas mərhələdən ibarətdir (Cədvəllər Əlavə 1-də verilib).

1. Blokun bitlərinə IP başlanğıc permutasiyası tətbiq edilir. Cədvəl 5.1-ə uyğun olaraq 58-ci bit birinci, 50-ci bit ikinci və s. olur. Sonra alınmış nəticə iki hissəyə bölünür: L_0R_0, burada L_0 − 32 bitlik sol yarı, R_0 – 32 bitlik sağ yarıdır.
2. L_0R_0 Feystel sxemi üzrə 16 dəfə çevrilir:

$$L_i=R_{i-1},\ R_i=L_{i-1}\oplus f(R_{i-1},\ K_i),\ i=1,2,\ldots,16,$$

burada f funksiyası və K_1, K_2, …, K_{16} açar cədvəli ayrıca təsvir olunacaq.

1. $L_{16}R_{16}$ bloku IP^{-1} tərs permutasiyası ilə qarışdırılır.

Başlanğıc və son yerdəyişmələr bir-birinin tərsidirlər, alqoritmin kriptoqrafik dözümünə təsir etmirlər (onları istifadə etməsək, DES-in təhlükəsizliyi pisləşməz).

5.3. DES-in *f* funksiyası

f funksiyasının iki arqumenti var: R_{i-1} və K_i. Birinci arqument 32 bit, ikincisi 48 bitdir. Funksiyanın çıxışı 32 bitdir. f funksiyası dörd addımdan ibarətdir: 1) E genişlənməsi; 2) Raund açarı ilə XOR; 3) S-blok əvəzləməsi ; 4) Permutasiya.

1. 32 bitlik R_{i-1} arqumenti $E(R_{i-1})$ genişlənmə yerdəyişməsinin (Cədvəl 5.2) köməyi ilə 48-bitlik vektora çevrilir. Bu prosedur bütün raundlar üçün eynidir.
2. $E(R_{i-1}) \oplus K_i$ cəmi hesablanır və səkkiz 6-bitlik sözün konkatenasiyası şəklində yazılır: $E(R_{i-1}) \oplus K_i = B_1B_2B_3B_4B_5B_6B_7B_8$.
3. Hər bir B_i sözü uyğun S_i S-blokuna daxil olur. S_i bloku 6-bitlik B_i girişini 4-bitlik C_i çıxışına çevirir. S-blok elementləri 0-dan 15-ə qədər tam ədədlər olan 4×16 ölçülü matrisdir.

B_i sözünün ikilik yazılışına baxılsa, onun birinci və altıncı bitləri birlikdə S-blok matrisində sətrin nömrəsini müəyyən edir. Ortadakı dörd bit müəyyən sütunun nömrəsini verir. Bununla da matrisin müəyyən elementi tapılır. Onun ikilik yazılışı çıxış olur.

Əlavə 1-də cədvəl 5.3-də DES-in 8 S-blokunun hamısı göstərilib. S-blokların dizaynı DES üçün çox vacibdir, çünki DES-in bütün digər elementləri xəttidir, onların sındırılması çətin məsələ deyil və DES-in qeyri-xəttiliyini təmin edən əsas vasitə S-bloklardır.

4. $C = C_1C_2C_3C_4C_5C_6C_7C_8$ çıxışı P permutasiyası (Cədvəl 5.4) ilə qarışdırılır.

5.4. DES-in açar cədvəli

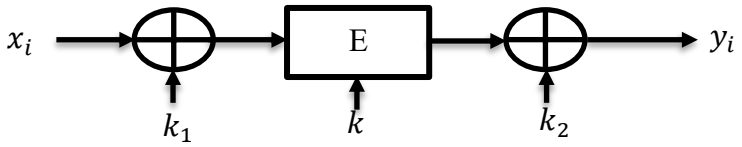
64-bitlik açarda 8, 16, …, 64 nömrəli bitlər silinir. Qalan bitlər PC-1 yerdəyişməsi ilə qarışdırılır (Cədvəl 5.5). 16 raundun hər biri üçün 56 bitlik açardan 48 bitlik altaçar generasiya olunur. Bunun üçün əvvəlcə 56 bitlik açar 28 bitlik iki yarıya bölünür. Sonra hər yarı raundun nömrəsindən asılı olaraq sola 1 və ya 2 mövqe dövri sürüşdürülür (raundun nömrəsi 1, 2, 9, 16 olduqda 1 bit sürüşdürülür). Bundan sonra 56 bitdən 48 bit verən PC-2 sıxma yerdəyişməsi (Cədvəl 5.6) yerinə yetirilir (raund nömrəsindən asılı deyil).

5.5. DES-in gücləndirilməsi

DES-in açarın uzunluğunun kiçik olması səbəbindən kobud güc hücumuna (bütün mümkün açarları yoxlamaq) qarşı zəifliyi bu şifr üçün alternativ axtarışlarına xeyli maraq doğurur. İki yanaşma mümkündür:

- Çoxqat (dəfələrlə) şifrləmə. Nəzəri olaraq daha təhlükəsizdir, lakin bəzən praktikada təhlükəsizliyi çox da artırmır.
- Açar ağardılması. DES kimi blok şifrlərini kobud güc hücumlarına daha dayanıqlı edir. Əlavə hesablama yükü olduqca kiçikdir.

Açar ağardılmasında açıqmətni və şifrmətni XOR-maskalamaq üçün k əsas şifr açarına əlavə olaraq, iki k_1 və k_2 ağartma açarları istifadə edilir.

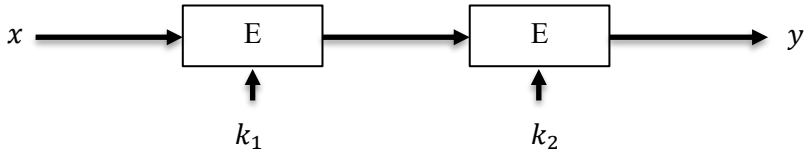


Açar ağardılması blok şifrlərini xətti və diferensial kriptoanaliz kimi bir çox analitik hücuma qarşı gücləndirmir. Açar ağardılması təbiətən zəif olan şifrlər üçün "əlac" deyil. Onun əsas tətbiqi analtik hücumlara qarşı nisbətən dayanıqlı olan, ancaq açar fəzası kiçik olan şifrlərdir (xüsusilə DES). DES-in açar ağardılması istifadə edən variantı DESX adlanır.

5.6. İkiqat şifrləmə (Double DES – 2DES)

Yanaşmalardan biri bir neçə açar tətbiq etməklə DES-in köməyi ilə dəfələrlə şifrləmədən ibarətdir. İkiqat DES iki açarla iki dəfə şifrləməni təklif edir. *x* açıqmətni əvvəlcə birinci k_1 açarı ilə şifrlənir, alınan şifrmətn yenidən ikinci k_2 açarı ilə şifrlənir:

$$c = E_{k_2}(E_{k_1}(x)).$$



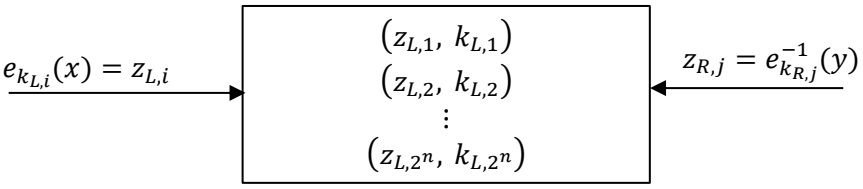
Açarın uzunluğunun *k* bit olduğunu fərz etsək, tükədici açar axtarışı $2^k \cdot 2^k = 2^{2k}$ şifrləmə və ya deşifrləmə tələb edəcək, bu da şifrin kriptoqrafik dözümünü əhəmiyyətli dərəcədə yüksəldir. Lakin ikiqat DES-in kriptoanalizi üçün effektiv «ortada görüş hücumu» (Meet-in-the-Middle) təklif olunmuşdur.

Meet-in-the-Middle hücumu $2^k + 2^k = 2^{k+1}$ əməliyyat tələb edir!

Mərhələ I: Verilmiş (x_1, y_1) üçün **sol** şifrləməyə bütün $k_{L,i}$, i=1,2, ..., 2^k üçün kobud güc hücumu edilir və 2^k girişi olan axtarış cədvəli hesablanır (hər birinin uzunluğu $n+k$ bitdir).

- Axtarış cədvəli şifrləmənin ($z_{L,i}$) nəticəsinə görə sıralanmalıdır

Məsələ II: **Sağ** şifrləməyə kobud güc hücumu edilir (Deşifrləmə istifadə edilməklə) və hər bir $z_{R,i}$ üçün yoxlanır ki, $z_{R,i}$ birinci mərhələdəki cədvəldə



hər hansı $z_{L,i}$ qiymətinə bərabərdirmi?

- Şifrləmələrin və deşifrləmələrin sayı = $2^k + 2^k = 2^{k+1}$.
- Saxlama yerlərinin sayı = 2^k.

Beləliklə, ikiqat şifrləmə birqat şifrləmədən çox da təhlükəsiz deyil!

5.7. Üçqat şifrləmə (Triple DES – 3DES)

Üçqat DES (TripleDES, 3-DES) praktikada DES-in effektiv açar uzunluğunu 112-yə çatdırmaq üçün istifadə edilir:

$$y = DES_{k_3}\left(DES_{k_2}\left(DES_{k_1}(x)\right)\right)$$

Alternativ 3DES versiyası EDE (Encrypt-Decrypt-Encrypt) adlanır:

$$y = DES_{k_3}\left(DES^{-1}_{k_2}\left(DES_{k_1}(x)\right)\right)$$

Bu versiyanın üstünlüyü: $k_1 = k_2 = k_3$ seçilsə, adi DES şifrləməsi alınır. Onu 3DES-ə hələlik praktiki hücumlar məlum deyil. TDES-in şifrləmə sürəti DES-inkindən üç dəfə yavaşdır, çünki üç DES şifrləməsindən istifadə edilir. Bir çox köhnə tətbiqlərdə, məsələn, bank sistemlərində istifadə edilir.

3DES-ə Meet-in-the middle hücumu etmək mümkündür, bu üçqat şifrləmənin *effektiv açar uzunluğunu* 3*K*-dan 2*K*-ya qısaldır! Hücum edən 3DES halında 2^{112} variant yoxlamalıdır.

Mühazirə üzrə yoxlama sualları

1. Blok şifri nədir?
2. Səpilmə nədir və səpilmə üçün hansı çevirmə istifadə oluna bilər?
3. Qarışdırma nədir və qarışdırma üçün hansı çevirmə istifadə oluna bilər?
4. Kompozisiya şifri nədir?
5. Hansı şərtdə kompozisiya şifri iterativ şifr olur?

6. Feystel şəbəkəsi necə işləyir?
7. Feystel şəbəkəsində şifrləmə və deşifrləmə nə ilə fərqlənir?
8. DES şifrinin əsas məqsədi nə idi?
9. DES şifrində blokun uzunluğu və açarın uzunluğu neçə bitdir?
10. DES şifrində neçə raund yerinə yetirilir?
11. DES-in kriptoqrafik çevirmə prosesi hansı əsas mərhələlərdən ibarətdir?
12. DES-in f funksiyasının neçə girişi var və neçə bitdirlər?
13. DES-in f funksiyası neçə addımdan ibarətdir?
14. f funksiyasında E genişlənməsi necə işləyir?
15. DES-də raund açarları necə generasiya olunur?
16. DES-də S-bloklar nədir və nə üçün istifadə olunur?
17. DES-də neçə S-blok istifadə olunur?
18. DES-də S-blokun hər biri neçə bitlə işləyir?
19. S-bloklarının funksiyası DES-in təhlükəsizliyinə necə təsir edir?
20. DES-də sola sürüşmə prosesi nə vaxt və necə tətbiq olunur?
21. Açar ağardılması necə işləyir?
22. İkiqat şifrləmədə Meet-in-the-Middle hücumu necə işləyir?
23. Triple DES (3DES) nədir?
24. 3DES-dən hansı şərtlər ödəndikdə adi DES alınır?
25. 3DES-in effektiv açar uzunluğu neçə bitdir?

Mühazirə 6. Advanced Encryption Standard (AES)

6.1. AES müsabiqəsi

25 il ərzində intensiv kriptoanaliz zamanı səmərəsinə görə DES-in açarlar fəzası üzrə tam axtarışdan əhəmiyyətli fərqlənən metod tapılmadı. Lakin bu illər ərzində hesablama texnikasının tərəqqisi o qədər böyük olmuşdu ki, açarların bütün mümkün 2^{56} (təxminən 10^{17}) variantının yoxlanması məsələsi o qədər də inanılmaz görünmürdü. 1997-ci ilin əvvəlində DES-i üç sutkada sındıran və qiyməti 210 min dollar olan EFF DES Cracker adlı aparat kompleksi qurulmuşdu.

1997-ci ildə ABŞ-ın Milli Standartlar və Texnologiyalar İnsititutu (NIST) 1974-cü ildən mövcud olan DES-in əvəzinə hökumət səviyyəsində vacib informasiyanın şifrlənməsi üçün yeni kriptoqrafik şifrləmə standartının qəbulu üzrə proqramın başlandığını elan etdi.

Müsabiqəyə 12 ölkə – Avstriya, Belçika, Böyük Britaniya, Almaniya, İsrail, Kanada, Kosta-Rika, Norveç, ABŞ, Fransa, Cənubi Koreya və Yaponiya kriptoqrafları tərəfindən yaradılmış 15 alqoritm qəbul olunmuşdu. Müsabiqənin finalına aşağıdakı alqoritmlər çıxmışdı: MARS, TWOFISH və RC6 (ABŞ), RIJNDEAL (Belçika), SERPENT (Böyük Britaniya, İsrail, Norveç).

2000-ci ilin oktyabrında müsabiqə başa çatdı. Dözümü, məhsuldarlığı, realizənin effektivliyi və çevikliyi ən yaxşı birləşdirən şifr kimi, Belçika şifri RIJNDAEL qalib elan edildi. Şifrin müəllifləri Joan Daemen və Vinsent Rijmendir, familiyalarının ilkin hərfləri alqoritmin adını – RIJNDAEL əmələ gətirir ("reyndal" kimi oxunur).

AES – 2001-ci ildən ABŞ-ın şifrləmə standartıdır. Blokun uzunluğu və açarın uzunluğu bir-birindən asılı olmadan 128, 192 və ya 256 bitə bərabər ola bilər. Raundların sayı açarın uzunluğundan və blokun uzunluğundan asılı olaraq 10, 12 və 14 olur. Raundların sayı aşağıdakı cədvəldə verilib:

	128	192	256
128	**10**	**12**	**14**
192	**12**	**12**	**14**
256	**14**	**14**	**14**

6.2. AES alqoritminin riyazi əsasları

Alqoritmin təsviri zamanı $GF(2^8)$ meydanından istifadə edilir. $GF(2^8)$ meydanın seçilməsi onunla əlaqədardır ki, onun elementləri baytlarla eyniləşdirilir. Bu isə onun realizəsini 8-bitlik prosessorlarda asanlaşdırır. Bu meydan gətirilə bilməyən $\varphi(x) = x^8 + x^4 + x^3 + x + 1$ çoxhədlisinin köklərinə görə $GF(2)$ meydanının genişlənməsi kimi qurulur. Bu çoxhədli meydan elementlərinin effektiv təsviri

mülahizələrindən seçilib. Alqoritmdə istifadə edilən elementar əməliyyatlar göstərilən meydanda yerinə yetirilir.

$GF(2^8)$ meydanının elementləri dərəcəsi 7-dən böyük olmayan çoxhədlilərdir, çoxhədlini əmsallarından ibarət sətir kimi göstərmək olar. Baytı bitlərin

$$\{a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0\},\ a_i \in \{0,1\}, i = \overline{0,7}$$

ardıcıllığı kimi göstərsək, meydanın elementi

$$a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

çoxhədlisi ilə təsvir ediləcək.

Məsələn, {01100011} baytına x^6+x^5+x+1 çoxhədlisi uyğundur.

Baytın qiymətlərini onaltılıq say sistemində yazmaq rahatdır. Bunun üçün hər 4 bitə şəkil 6.1-də göstərildiyi kimi onaltılıq simvol qarşı qoyulur. Məsələn, {01100011} elementi {63} kimi göstərilir, böyük bitlər qrupunu göstərən onaltılıq simvol yenə də solda gəlir.

Bitlər qrupu	16-lıq simvol	Bitlər qrupu	16-lıq simvol	Bitlər qrupu	16-lıq simvol	Bitlər qrupu	16-lıq simvol
0000	0	0100	4	1000	8	1100	c
0001	1	0101	5	1001	9	1101	d
0010	2	0110	6	1010	a	1110	e
0011	3	0111	7	1011	b	1111	f

Şəkil 6.1

Sonlu meydan elementləri üçün toplama və vurma əməlləri təyin olunub. $GF(2^8)$ meydanında toplama – çoxhədliləri adi toplama əməlidir, oxşar toplananlar islah edildikdə XOR əməli istifadə edilir; əgər meydanın elementləri ikilik yazılışda göstərilibsə, toplama bitlər üzrə XOR əməlinə uyğundur.

Məsələn, aşağıdakı ifadələr bir-birinə ekvivalentdir:

- $(x^6 + x^4 + x^2 + x+1) + (x^7 + x + 1) = (x^7 + x^6 + x^4 + x^2)$ (polinomial yazılış);
- {01010111} ⊕ {10000011} = {11010100} (ikilik yazılış);
- {57} ⊕ {83} = {d4} (onaltılıq yazılış).

Sonlu meydanda hər bir elementin additiv tərsi var. $GF(2^8)$ meydanında sıfırdan fərqli hər bir element özünün additiv tərsidir: $\alpha+\alpha=0$, burada sıfır elementi 00-dır.

Polinomial yazılışda $GF(2^8)$ meydanında vurma (· ilə işarə edilir) nəticəni dərəcəsi 8-ə bərabər olan gətirilə bilməyən çoxhədlinin moduluna gətirməklə çoxhədlilərin vurulmasına uyğun gəlir. AES alqoritmi üçün $\varphi(x) = x^8+x^4+x^3+x+1$ gətirilə bilməyən çoxhədlisi seçilib, onu onaltılıq işarələrlə {01}{1b} kimi yazırlar.

Misal a){a4}·{00} = {00}; b) {a4}·{01} = {a4}; c) {a4}·{02} = {13}.

Belə ki, {a4}=1010 0100 = $x^7 + x^5+x^2$ və {02}=0000 0010 = x.

Hasil $(x^7 + x^5+x^2)\cdot x = x^8 + x^6+x^3 = (x^8 + x^6+x^3) + (x^8 + x^4 + x^3 + x + 1) = x^6+x^4 + x + 1$ = {0101 0011} = {53} olacaq.

6.3. AES-in raund çevirməsi

Raund dörd müxtəlif çevirmədən ibarətdir:

SubBytes() – baytların əvəz edilməsi; 8×256 ölçülü sabit əvəzləmə cədvəlli S-bloklarda baytları əvəzetmə.

ShiftRows() – sətirlərin sürüşməsi; State massivi sətirlərinin müxtəlif sayda baytlar qədər baytların sürüşdürülməsi.

MixColumns() – sütunların qarışdırılması; $GF(2^8)$ üzərində çoxhədli kimi baxılan vəziyyət sütunlarının üçüncü dərəcəli $g(x)$ çoxhədlisinə x^4+1 modulu ilə vurulması.

AddRoundKey() – raund açarı ilə toplama; genişləndirilmiş açarın cari qiyməti ilə 2 modulu ilə toplama.

***SubBytes* çevirməsi.** *SubBytes* çevirməsi vəziyyətin hər bir baytı ilə asılı olmadan yerinə yetirilən, baytların qeyri-xətti əvəz olunmasından ibarətdir. S-blokun əvəzləmələr cədvəli inversiya olunandır və iki çevirmənin kompozisiyasından qurulmuşdur:

1. Hər bir bayt $GF(2^8)$ meydanında vurmaya nəzərən tərs elementi ilə əvəz edilir. '00' sıfır elementi öz-özünə keçir.
2. Hər bir bayt üzərində $GF(2)$ meydanında aşağıdakı kimi təyin olunan çevirmə tətbiq edilir:

$$\begin{vmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{vmatrix} \oplus \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}$$

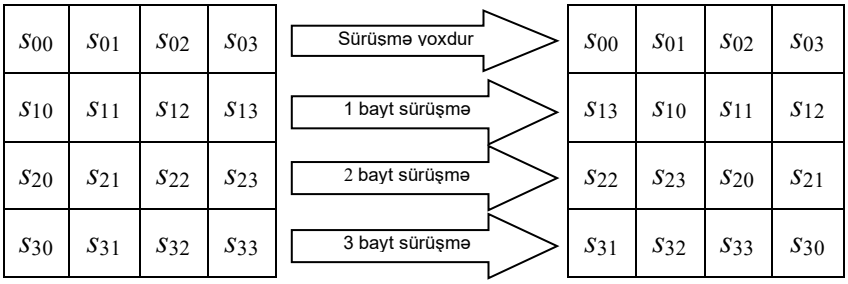
Bu affin çevirməni çoxhədlilərlə $Y = Cx + C_1 \pmod{x^8+1}$ şəklində yazmaq olar. Bu S-blokun vəziyyətin bütün baytlarına tətbiqi (State) kimi işarə edilir.

***ShiftRows* çevirməsi.** Vəziyyətin axırıncı üç sətri müxtəlif sayda baytlara dövri sürüşdürülür. Sətir 1 C_1 bayt, sətir 2 – C_2 bayt, sətir 3 – C_3 bayt sürüşdürülür. C_1, C_2 və C_3 sürüşmələrinin qiyməti blokun uzunluğu N_b-dən asılıdır. Onların qiyməti cədvəl 6.2-də göstərilir.

N_b	C_1	C_2	C_3
4	1	2	3
6	1	2	3
8	1	3	4

Cədvəl 6.2.

Axırıncı üç sətrin sürüşməsi əməliyyatı *ShiftRows*(State) kimi işarə edilir. Şəkil 6.1-də çevirmənin State-ə təsiri göstərilir.



Şəkil 6.1.

***MixColumns* çevirməsi.** Bu çevirmədə *State* massivinin sütunlarına $GF(2^8)$ meydanı üzərindəki çoxhədlilər kimi baxılır. Çevirmə sütunun x^4+1 moduluna görə sabit

$$c(x) = \{03\}x^3+\{01\}x^2+\{01\}x+\{02\}$$

çoxhədlisinə vurulmasından ibarətdir:

$$b(x) = c(x) \cdot a(x)(\text{mod } x^4+1).$$

$c(x)$ çoxhədlisi x^4+1 ilə qarşılıqlı sadədir və buna görə vurmanın tərsi var. Verilən çevirməni matris formada belə təsvir etmək olar:

$$\begin{vmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{vmatrix} = \begin{vmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{vmatrix} \cdot \begin{vmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{vmatrix}$$

Tərs çevirmə x^4+1 moduluna görə $c(x)$-in multiplikativ tərsi olan

$$d(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}$$

çoxhədlisinə vurmaqdan ibarətdir.

AddRoundKey çevirməsi. Raund açarının əlavə edilməsi *State* massivinin hər bir açar massivinin uyğun baytı ilə 2 moduluna görə bit-bit toplanması ilə həyata keçirilir. Bu çevirmə özü özünə tərsdir.

6.4. Açarın emalı alqoritmi

Raund açarları şifrləmə açarından açarın emalı alqoritminin köməyi ilə alınır. Bu alqoritm iki komponentdən ibarətdir: *açarın genişləndirilməsi* (Key Expansion)

və *raund açarının seçilməsi* (Round Key Selection). Alqoritmin əsas prinsipləri aşağıdakı kimidir:

- Raund açarlarının bitlərinin ümumi sayı blokun uzunluğunun bir vahid artırılmış raundlar sayına hasilinə bərabərdir (məsələn, uzunluğu 128 bit blok və 10 raund üçün raund açarının 1408 bit olar).
- Şifrləmə açarı geniş açara (Expanded Key) genişləndirilir.
- Raund açarları geniş açardan aşağıdakı kimi götürülür: birinci raund açarı ilk *Nb* sözdən, ikinci – sonrakı *Nb* sözdən və s. ibarətdir.

Açarın genişlənməsi (Key Expansion). Genişlənmiş açar 4-baytlıq sözlərin xətti massividir və *W*[*Nb* (*Nr*+1)] kimi işarə edilir. İlk *Nk* söz şifrlənmə açarını özündə saxlayır. Qalan digər sözlər kiçik indeksli sözlərdən rekursiv hesablanır. Açarın emalı alqoritmi *Nk* kəmiyyətindən asılıdır. *Nk*≤6 üçün aşağıdakı alqoritm işlədilir (psevdo-C dilində):

```
for (i = Nk; i<Nb·(Nr+1); i++)
{
  temp = W[i-1];
  if (i % Nk = = 0)
  temp = SubByte(RotByte(temp)⊕Rcon[i /Nk]);
W[i] = W[i–Nk]⊕temp; }
```

SubByte(W) funksiyası 4-baytlıq söz formalaşdırır, onun hər bir baytı öz növbəsində *RotByte* çevirməsinin tətbiq olunması ilə formalaşdırılır. *RotByte*(W) funksiyası giriş baytlarının sola dövri sürüşməsini həyata keçirir, məsələn, *RotByte*(*a*, *b*, *c*, *d*)=(*b*, *c*, *d*, *a*).

Görmək olar ki, ilk *Nk* söz şifrləmə açarı ilə doldurulur. Hər sonrakı *W*[*i*] sözü XOR əməli ilə əvvəlki *W*[*i*-1] sözü və *Nk* mövqe əvvəlki *W*[*i*-N_k] sözündən alınır. Mövqeləri *Nk*-ya bölünən sözlər üçün XOR-dan qabaq *W*[*i*–1]-ə çevirmə olunan sözdə baytların dövri sürüşməsi var, sonra baytların əvəzedilməsi gəlir.
N_k>6 (*Nk*=8) halında aşağıdakı alqoritm işlədilir.

```
for (i = Nk; i<Nb×(Nr+1); i++)
{temp = W[i-1];
  if (i%Nk == 0)
  temp = SubByte(RotByte(temp)⊕Rcon[i/Nk]);
  else if (i % Nk == 4)
  temp = SubByte(temp);
  W[i] = W[i–Nk]⊕temp;}
```

Əvvəlki baxılan sxemdən fərqi *Nk*-dan hər dördüncü bayt üçün Subbyte tətbiq olunmasıdır.

Rcon raund sabitləri *Nk*-dan asılı deyil və:

$$Rcon[i]=(RC[i], '00', '00', '00')$$

kimi təyin edilir, burada RC[*i*] $GF(2^8)$ meydanında x=’02’ doğuran elementinin tətbiq olunması ilə hesablanır, həm də $RC[1]='01'=1$, $RC[i] = x \cdot RC[i-1] = x^{i-1}$.

Raund açarının seçilməsi. *i*-ci raund açarı raund açarı massivinin $W[Nb \cdot i]$-dən $W[Nb(i+1)]$-ə qədər sözləri götürməklə alınır.

Şifrləmə açarının seçilməsinə heç bir məhdudiyyət yoxdur. Genişlənmiş açar həmişə şifrləmə açarından alınmalı və heç vaxt birbaşa göstərilməməlidir.

6.5. Deşifrləmə funksiyası

AES alqoritminin deşifrləmə prosedurlarının həyata keçirilməsi zamanı *State* massivi *InvShiftRows*(), *InvSubBytes*(), *InvMixColumns*(), və *AddRoundKey*() çevirmələri ilə emal edilir:

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word
w[Nb*(Nr+1)])
begin
   byte state[4,Nb]
   State = in
   AddRoundKey(State, w[Nr*Nb, (Nr+1)*Nb-1])
   for round = Nr-1 step -1 downto 1
      InvShiftRows(State)
      InvSubBytes(State)
      AddRoundKey(State, w[round*Nb, (round+1)*Nb-1])
      InvMixColumns(State)
   end for
   InvShiftRows(State)
   InvSubBytes(State)
   AddRoundKey(State, w[0, Nb-1])
   out = state
end
```

InvShiftRows() çevirməsi *ShiftRows*() çevirməsinin tərsidir. *State* massivinin axırıncı üç sətrindəki baytlar müxtəlif sayda baytlar qədər dövri sürüşdürülür. Birinci sətir (*Row* 0) sürüşdürülür. Axırdakı üç sətir $Nb - Shift(r, Nb)$ bayt qədər sürüşdürülür, burada sürüşmənin kəmiyyəti $shift(r, nB)$ sətrin nömrəsindən asılıdır.

InvSubBytes() çevirməsi baytların əvəzləmə çevirməsi üçün inversiyadır. Bu çevirmə zamanı *State* massivinin hər bir baytı tərs S-blokundan alınmış qiymətlə əvəz edilir. Bu tərs affin çevirməsinin tətbiqi və sonra $GF(2^8)$ meydanında multiplikativ tərsin hesablanması ilə əldə edilir**.**

***InvMixColumns*()** çevirməsi *MixColumns* çevirməsinin tərsidir. *InvMixColumns* çevirməsi növbə ilə *State* massivinin strukturlarına təsir edir. Sütunlara $GF(2^8)$ meydanı üzərində çoxhədli kimi baxılır və fiksə edilmiş $a^{-1}(x)$ çoxhədlisinə (x^4+1) moduluna gətirməklə vurulur.

$$a^{-1}(x)=\{0e\}x^3+\{0d\}x^2+\{09\}x+\{0e\}$$

AddRoundKey() çevirməsi özünün tərsidir, çünki yalnız XOR əməlindən istifadə edir.

Mühazirə üzrə yoxlama sualları

1. AES hansı məqsədlə hazırlanmışdı?
2. AES müsabiqə finalına hansı şifrlər çıxmışdı?
3. AES şifrində blok ölçüsü neçə bitdir (baytdır)?
4. AES şifrində açar uzunluqları hansılardır?
5. AES-də neçə raund icra olunur?
6. AES-də əməliyyatlar hansı meydan üzərində aparılır?
7. {01101011} baytına hansı çoxhədli uyğundur?
8. $x^6+x^5+x^4+x+1$ çoxhədlisinə hansı bayt yazılışı uyğundur?
9. AES-də başlanğıc vəziyyət matrisi necə formalaşır?
10. AES-də hər raunddda şifrləmə üçün hansı əməliyyatlar icra olunur?
11. SubBytes mərhələsində nə baş verir?
12. ShiftRows əməliyyatı AES-də hansı məqsədlə istifadə olunur?
13. MixColumns mərhələsi AES-də hansı funksiyanı yerinə yetirir?
14. AddRoundKey əməliyyatının funksiyası nədir?
15. AES-də sonuncu şifrləmə raundu əvvəlki raundlardan nə ilə fərqlənir?
16. AES-in S-blok komponenti nədir və necə işləyir?
17. S-blok AES-də hansı təhlükəsizlik prinsiplərinə cavab verir?
18. 128-bit AES açarı üçün açar cədvəlində neçə söz olacaq?
19. AES açar cədvəlində birinci altaçar nədən ibarət olur?
20. AES-də açar genişlənməsi (key expansion) necə həyata keçirilir?
21. AES-də altaçarların sayı necə müəyyən edilir?
22. Rcon raund sabitləri necə hesablanır?
23. AES-də raund açarları necə generasiya olunur?
24. AES-də şifrləmə və deşifrləmə əməliyyatları simmetrikdirmi?
25. AES-də hər raundda deşifrləmə üçün hansı əməliyyatlar icra olunur?

Mühazirə 7. Blok şifrlərinin iş rejimləri

Blok şifri uzunluğu əvvəlcədən müəyyən edilmiş bir bloku şifrləməyə imkan verir. Əgər şifrlənəcək məlumatın uzunluğu blokun uzunluğundan kiçikdirsə, onun sonuna əlavələr (məsələn, 0-lar) etməklə lazımi uzunluğa çatdırılır. Əgər məlumatın uzunluğu böyükdürsə, onu bloklara bölmək lazım gəlir. Belə bloklar ardıcıllığını şifrləmək üçün bir neçə üsul var, onları *blok şifrlərinin iş rejimləri* adlandırırlar. İş rejimi – blok şifrinin açıq mətnin ardıcıl bloklarını necə şifrləyəcəyini müəyyən edən metodologiyadır. Bu zaman bir bloku şifrləmək üçün digər blokun verilənləri istifadə edilə bilər. İş rejimi şifrin təhlükəsizliyini, məhsuldarlığını və səhv yayılmasını müəyyən edir.

1980-ci ildə qəbul edilmiş FIPS Pub 81 «DES Modes of Operation» standartı DES alqoritmi üçün 4 rejim müəyyən edirdi:

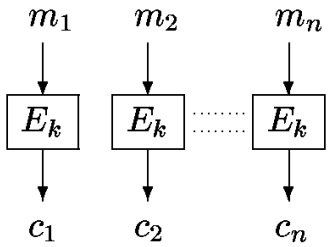
- Electronic Codebook (ECB) Mode – elektron kod kitabı rejimi;
- Cipher Block Chaining (CBC) Mode – şifr blokların ilişməsi rejimi;
- Cipher Feedback (CFB) Mode – şifr üzrə əks-əlaqə rejimi;
- Output Feedback (OFB) Mode – çıxış üzrə əks-əlaqə rejimi.

ECB rejimi. ECB rejimində (şəkil 7.1) açıq mətnin/şifrmətnin *i*-ci blokunun şifrlənməsi/deşifrlənməsi müstəqil yerinə yetirilir:

$$c_i = E_k(m_i),$$

$$m_i = D_k(c_i),$$

burada E_k və D_k ilə k məxfi açarında şifrləmə/deşifrləmə alqoritmləri işarə edilib.



Şəkil 7.1. ECB rejimində şifrləmə

ECB rejiminin əsas nöqsanı yüksək dərəcədə determinik şifrləmə olmasıdır: eyni açıqmətnlər eyni şifrmətnlərə inikas edir və açıq mətnin statistik xassələri şifrmətndə qorunub saxlanır (Linux-un loqosu ilə məşhur şəklə baxın).

Eyni açıqmətnlərin eyni şifrmətnlərə inikas etməsi şifrmətn bloklarını silməklə, təkrarlamaqla və yerlərini dəyişməklə açıq mətni asanlıqla manipulyasiya etməyə imkan verir. Əgər eyni mətn iki dəfə göndərilsə, hücum edən bunu aşkarlaya bilir. Hücum edən şifrmətn bloklarının yerini elə dəyişə bilər ki, düzgün açıqmətn alınsın.

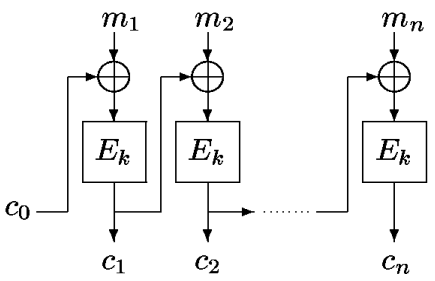
ECB-yə əvəzləmə hücumu. Əgər müəyyən bir açıqmətnin şifrmətnə blok inikası $m_i \to c_i$ məlum olarsa, şifrmətn bloklarının ardıcıllığını asanlıqla manipulyasiya etmək olar. *Elektron bank köçürməsi* nümunəsinə baxaq (şəkil 7.2):

İki bank arasında şifrləmə açarı, adətən, tez-tez dəyişmir. Hücum edən A bankındakı öz hesabından B bankındakı hesabına \$1.00 köçürmələrini bir neçə dəfə göndərir. O təkrarlanan şifrmətn bloklarını tapa bilər və belə köçürmələrin 1, 3 və 4-cü bloklarını yadda saxlayır. Bundan sonra digər transferlərin 4-cü blokunu yadda saxladığı 4-cü blokla əvəzləyə bilər. Bu zaman A bankındakı müəyyən hesabdan B bankındakı müəyyən hesaba olan *bütün köçürmələr* hücum edənin B bankındakı hesabına yönəldiləcək!

CBC rejimi. CBC rejimində (şəkil 7.3) açıq mətnin hər bir i-ci bloku şifrmətnin $(i-1)$-ci bloku ilə 2 modulu üzrə (XOR əməliyyatı) toplanır və sonra şifrlənir:

$$c_i = E_k(m_i \oplus c_{i-1}),$$
$$m_i = D_k(c_i) \oplus c_{i-1}.$$

Başlanğıc qiymət (bizim işarələrdə c_0) *ilkin yükləmə vektoru* ilə verilir.



Şəkil 7.3. CBC rejimində şifrləmə

CBC rejiminin arxasında iki əsas ideya var:

1. Bütün blokların şifrləmələri "birlikdə zəncirlənir".
   - c_i şifrmətni yalnız m_i blokundan deyil, əvvəlki açıqmətn bloklarından da asılı olur.
2. Şifrləmə **ilkin yükləmə vektoru**ndan (Initialization Vector, IV) istifadə etməklə təsadüfiləşdirilir.

CBC rejiminin tətbiqi ECB rejiminin nöqsanını aradan qaldırmağa imkan verir: açıq mətnin hər bir bloku əvvəlki mərhələdə alınmış şifrmətn bloku ilə «maskalanır». CBC rejimi istifadə edildikdə birinci və sonuncu blokun silinməsi istisna olmaqla şifrmətn blokları ilə istənilən manipulyasiyanı aşkarlamaq mümkündür.

CBC-ə əvəzləmə hücumu. Son misala yenidən baxaq (*Elektron bank köçürməsi*). Əgər hər bir köçürmə üçün IV düzgün seçilibsə, əvəzləmə hücumu, ümumiyyətlə, işləməyəcək! Əgər IV bir neçə köçürmə üçün eyni olsa, hücum edən özünün A bankındakı hesabından B-dəki hesabına köçürmələri tanıya bilər.

Blok #	1	2	3	4	5
	Göndərən Bank A	Göndərən Hesab #	Alan Bank B	Alan Hesab #	Miqdar \$

Şəkil 7.2.

Əgər hər dəfə şifrlədikdə yeni IV seçilərsə, CBC rejimi *ehtimallı şifrləmə* sxeminə çevrilir, yəni iki eyni açıqmətnin şifrlənməsi tamamilə müxtəlif olur.

IV-ni *məxfi* saxlamağa ehtiyac yoxdur! Adətən, IV məxfi olmayan təsadüfi sətir (yalnız bir dəfə istifadə edilən mənasız ardıcıllıq – *nonce*) olmalıdır.

"Bit flipping" hücumu. Şifrlənmiş məlumatı **bitlərini dəyişməklə** manipulyasiya edir. "Bitləri flip etmək" — 0-ı 1, 1-i 0 etməkdir, buna görə "bit flipping" deyilir.

Bu hücum növü *şifrləmə açarını bilmədən* məlumatı manipulyasiya etməyə imkan verir. CBC rejimində əvvəlki blokda biti dəyişdirsək, növbəti blokun deşifrləmə nəticəsini dəyişdirə bilərik.

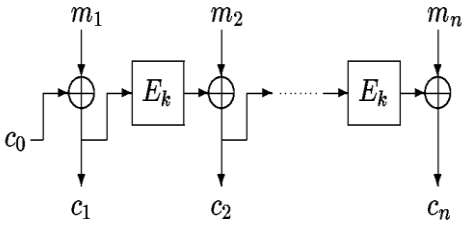
Misal. Tutaq ki, veb servis CBC ilə şifrlənmiş aşağıdakı cookie göndərir:

– username=guest; *admin*=0 (no)

Biz istəyirik ki, belə bir dəyişiklik edək:

– username=guest; *admin*=1 (yes)

CFB rejimi. CFB rejimində (şəkil 7.4) şifrmətnin hər bir *i*-ci bloku şifrmətnin (*i*–1)-ci blokunun şifrlənməsi və onun açıq mətnin *i*-ci bloku ilə 2 modulu üzrə (XOR əməliyyatı) toplanması yolu ilə formalaşır:



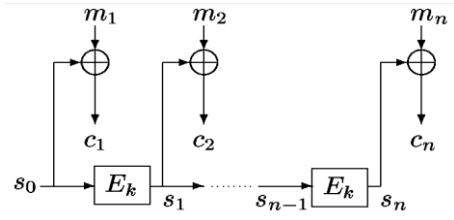
Şəkil 7.4. CFB rejimində şifrləmə

$$c_i = m_i \oplus E_k(c_{i-1}),$$

$$m_i = c_i \oplus D_k(c_{i-1}).$$

c_0 başlanğıc qiyməti CBC rejiminində olduğu kimi ilkin yükləmə vektoru ilə verilir. Eləcə də CBC rejimində olduğu kimi açıq mətnin blokları şifrmətn blokları ilə «maskalanır». Açıq mətnin dəyişdirilməsi imkanları da CBC rejimində olduğu kimidir. Əgər CFB rejimində iki identik şifrmətn bloku varsa, onda sonrakı şifrləmənin nəticəsi həminki kimi olacaq.

OFB rejimi. OFB rejimi (şəkil 7.5) CFB rejiminə analojidir, istisna ondadır ki, açıq mətn ilə toplanan bitlər açıq mətndən və şifrmətndən asılı olmadan generasiya olunmurlar. s_0 ilkin yükləmə vektoru s_i bloklar ardıcıllığının başlanğıc qiymətini verir, hər bir s_i bloku əvvəlki s_{i-1} blokunun şifrlənməsi ilə alınır. Açıq mətn açıq mətnin *i*-cı blokunun s_i ilə 2 modulu üzrə toplanması yolu ilə şifrlənir:



Şəkil 7.5. OFB rejimində şifrləmə

$$s_i = E_k(s_{i-1}),$$

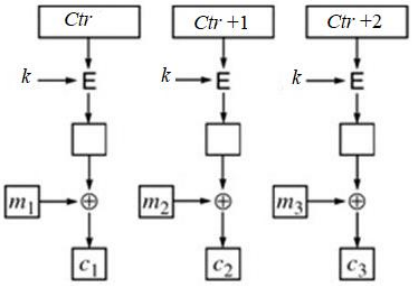
$$c_i = m_i \oplus s_i,$$

$$m_i = c_i \oplus s_i.$$

OFB rejimi CFB rejimi ilə müqayisədə aşağıdakı üstünlüyə malikdir: oküylü

kanal ilə ötürmə nəticəsində meydana çıxan səhvlərdeşifrləmə zamanı bütün şifrmətn üzrə «yayılmır», bir blok çərçivəsində lokallaşır. Lakin açıq mətn şifrmətn bloklarının müəyyən manipulyasiyaları ilə dəyişdirilə bilər.

Sayğac rejimi (Counter mode, CTR). CTR rejimi müvafiq blok şifri alqoritminin girişinə müəyyən sayğacın başlanğıc anından toplanan qiymətinin verilməsini nəzərdə tutur. Bu rejim blok şifrini axın şifrinə çevirir, yəni o, ardıcıllıq generasiya edir və ona açıq mətnlə XOR əməliyyatı tətbiq edilir. İlkin mətn və şifrlənmiş mətn bloku əsas şifr (məsələn, DES və ya AES) ilə eyni blok uzunluğuna malik olur.



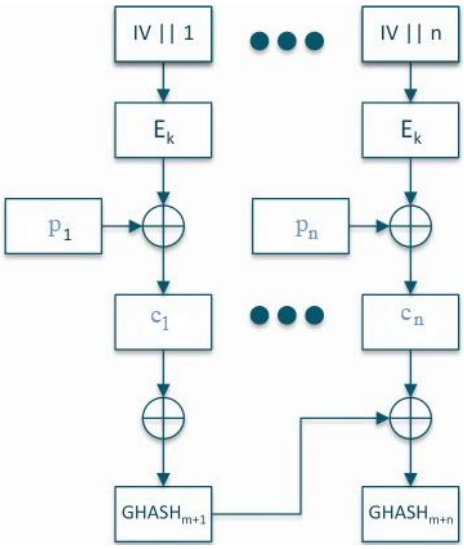
CTR rejimində şifrləmə: $c_i = m_i \oplus E_k(Ctr_i); i = 1,2,\dots,n$

CTR rejimində deşifrləmə: $m_i = c_i \oplus E_k(Ctr_i); i = 1,2,\dots,n$

Ctr_i – i-ci blok üçün sayğacın qiymətidir.

CFB və OFB rejimlərindən fərqli olaraq CTR rejimini paralelləşdirmək olar, çünki 2-ci şifrləməni 1-ci şifrləmə qurtarana kimi başlamaq olar.

Qalua/Sayğac rejimi (Galois/Counter Mode, GCM). 2007-ci ildən NIST standartıdır. GCM rejimi blokun uzunluğu 128 bit olan blok şifrləri üçün müəyyən edilib. Autentifikasiya ilə şifrləmə rejimidir (Authenticated Encryption with Associated Data, AEAD), verilənlərin həm konfidensiallığını, həm də autentifikasiyasını təmin edir. GMAC adlı variantında yalnız autentifikasiya təmin edilir.



CTR şifrləmə rejimində giriş blokları ardıcıl nömrələnir və blokun nömrəsi E blok şifri (adətən, AES) ilə şifrlənir. Şifrləmə funksiyasının çıxışı açıqmətn ilə XOR əməlində istifadə edilir və şifrmətn alınır. Sayğac əsasında digər rejimlərdə olduğu kimi, GCM də axın şifridir, buna görə də şifrlənən axının hər bir bloku üçün unikal ilkin yükləmə vektorunun istifadəsi tələb edilir.

GCM-də Qalua meydanında vurma əməli əsasında GHASH(H, A, C) funksiyası istifadə edilir. Bu funksiya autentifikasiya teqini almaq üçün şifrmətn bloklarını və autentifikasiya kodunu birləşdirir. Funksiyanın girişinə heşləmə açarı H verilir, o, K açarı ilə 128 sıfır bitin şifrlənməsinin nəticəsidir, yəni $H = E(K, 0^{128})$.

Autentifikasiya teqi məlumatın tamlığını yoxlamaq üçün istifadə edilir. Kanal ilə IV başlanğıc vektoru, şifrmətn blokları və autentifikasiya kodu (16 bayt) ötürülür. Öz xassələrinə görə GCM (GMAC) rejimi HMAC-a oxşardır.

Şifrləmə rejiminin seçilməsi. Şifrləmə rejiminin seçilməsi qarşıya qoyulmuş məqsəddən asılıdır. Adi açıq mətn üçün CBC, CFB və ya OFB istifadə etmək olar. Faylları şifrləmək üçün CBC daha yaxşıdır: təhlükəsizlik xeyli artır, saxlanılan verilənlərdə səhv meydana çıxdıqda sinxronlaşma, demək olar ki, pozulmur. Konkret rejimin seçilməsi istifadəçinin tələblərindən asılıdır. Bütövlükdə, şifrləmə metodunun seçilməsi təhlükəsizliklə məhsuldarlıq arasında tarazlığın axtarışıdır.

Mühazirə üzrə yoxlama sualları

1. Blok şifrlərinin iş rejimləri nədir?
2. «DES Modes of Operation» standartı hansı rejimləri müəyyən edir?
3. ECB rejimi necə işləyir?
4. ECB rejiminin əsas çatışmazlıqları nələrdir?
5. ECB rejiminə hansı hücum mümkündür?
6. CBC rejimi necə işləyir?
7. CBC rejimində ilk blok necə şifrlənir?
8. CBC rejiminin əsas ideyası nədir?
9. CBC rejimində ilk bloku şifrləmək üçün nə istifadə olunur?
10. CBC rejimində IV (initialization vector) hansı rolu oynayır?
11. CBC və ECB rejimləri arasında əsas fərq nədir?
12. CBC rejiminə əvəzləmə hücumu mümkündürmü?
13. CBC rejimində IV-nin təkrar istifadəsi hansı təhlükəyə səbəb ola bilər?
14. CFB rejimi necə işləyir?
15. OFB rejimi necə işləyir?
16. OFB rejimində səhv yayılması necə baş verir?
17. CTR rejimi necə işləyir?
18. GCM rejimi nədir?
19. GCM rejimi hansı əlavə funksionallığı təqdim edir?
20. GCM-in hansı variantında yalnız autentifikasiya təmin edilir?
21. Hansı rejimlərdə paralel şifrləmə mümkündür?
22. Hansı iş rejimləri axın şifri kimi işləyirlər?
23. CFB rejimi hansı hallarda istifadə olunur?
24. CFB rejiminin hansı üstünlükləri var?
25. OFB rejimi ilə CFB rejimi arasında əsas fərqlər nələrdir?

Mühazirə 8. Açıq açarlı kriptosistemlər

8.1. Açıq açarlı kriptoqrafiya anlayışı

Məxfi açarlı kriptosistemlər minilliklər boyu hazırki dövrə kimi istifadə edilirlər və informasiyanın konfidensiallığının təmin edilməsinin klassik prinsipinə: icazəsi olan şəxslərdən başqa bütün şəxslər üçün **istifadə edilən açarın məxfiliyi** prinsipinə əsaslanırlar. Belə kriptosistemləri **simmetrik** də adlandırırlar, çünki onlarda şifrləmə və deşifrləmə üçün istifadə olunan açarlar müəyyən simmetriyaya malikdirlər (çox vaxt onlar üst-üstə düşürlər).

Açıq açarlı kriptosistemlər Uitfild Diffi və Martin Hellman tərəfindən 1975-ci ildə təklif olunmuşdur. Onları *asimmetrik sistemlər* də adlandırırlar, çünki onlarda istifadə olunan şifrləmə və deşifrləmə açarları simmetriya və ya bərabərlik münasibətləri ilə bağlı deyillər.

Açıq açarlı kriptoqrafiyanın əsas ideyası açarın iki hissəyə ayrılmasıdır. Şifrləmə açarı açıq, hamıya məlum ola bilər, lakin məlumatı yalnız məxfi deşifrləmə açarına malik olan şəxs deşifrləyə bilər. Bu açarı simmetrik sistemin açarı ilə qarışdırmamaq üçün, adətən, **gizli açar** adlandırırlar.

Açıq açara görə gizli açarın hesablanması həlli yüksək çətinliklə xarakterizə olunan müəyyən riyazi məsələlər ilə bağlanır (əlaqələndirilir).

8.2. Biristiqamətli funksiyalar

Asimmetrik alqoritmlərin ideyası biristiqamətli funksiyalar nəzəriyyəsinin inkişafı ilə sıx bağlıdır.

Aşağıdakı iki xassəyə malik $F: X \to Y$ funksiyası ***biristiqamətli*** adlanır:

1) $F(x)$ qiymətlərinin hesablanması üçün polinomial alqoritm var.
2) F funksiyasının tərsini hesablamaq (yəni $F(x) = y$ tənliyinin x-ə nəzərən həll etmək) üçün polinomial alqoritm yoxdur.

Hazırda biristiqamətli funksiyaların varlığı nəzəri olaraq isbat olunmayıb. Praktikada biristiqamətli funksiya adına namizədlərdən istifadə edilir. Biristiqamətli funksiyaya namizəd kimi U. Diffi və M. Hellman diskret qüvvətə yüksəltmə funksiyasını təklif etmişdilər:

$$F(x) = a^x \bmod p$$

burada x – tam ədəd, $1 \le x \le p-1$, p sadə ədəddir, $a < p$ ədədi isə p moduluna görə ibtidai kökdür.

Hətta çox böyük p modulu üçün də x verildikdə $a^x \bmod p$ qiymətini hesablamaq çox asandır. Buna inanmaq üçün a^{25} qüvvətinin hesablanmasına baxaq: $a^{25} = (((a^2 \times a)^2)^2)^2 \times a$. Bu bərabərlik göstərir ki, a^{25} qüvvətini cəmisi dörd kvadrata

yüksəltmə və iki vurma əməlinin köməyi ilə hesablamaq mümkündür. Qüvvət üstü x-in uzunluğu L bit olarsa, L-dən $2L$-ə qədər sayda vurma və kvadrata yüksəltmə əməli tələb olunur.

Modula görə qüvvətə yüksəltməyə tərs məsələ diskret loqarifm məsələsi adlanır: $a^x = b \bmod n$ tənliyindən x-i tapmaq tələb olunur. Məsələn, $3^x = 15 \bmod 17$ tənliyindən $x = 6$ tapmaq olar. Diskret loqarifm məsələsinin həlli olmaya da bilər. Məsələn, $3^x = 7 \bmod 13$ tənliyinin həlli yoxdur. Diskret loqarifm məsələsini böyük ədədlər üçün həll etmək olduqca çətindir.

8.3. Gizli girişli biristiqamətli funksiyalar

Biristiqamətli funksiyanı şifrləmə funksiyası kimi istifadə etmək olmaz, çünki $F(x)$ məlumatına görə heç kim, hətta qanuni istifadəçi də x-i bərpa edə bilməz. Bu problemi həll etmək üçün U.Diffi və M.Hellman *gizli girişli biristiqamətli funksiyaları* (ing. one-way trapdoor function) istifadə etmək ideyasını irəli sürüblər.

K parametrindən asılı olan $F_K: X \to Y$ funksiyası aşağıdakı üç şərti ödədikdə *gizli girişli biristiqamətli funksiya* adlanır:

1) istənilən x və K üçün $F_K(x)$ qiymətini hesablamaq asandır;

2) K məlum olduqda F_K funksiyasının tərsini hesablamaq asandır;

3) K parametrini bilmədən F_K funksiyasının tərsini hesablamaq çətindir.

Gizli girişli biristiqamətli funksiyanın düz istiqamətdə hesablanması alqoritmini bilən şəxs gizli informasiyanı bilən şəxsə məlumat göndərmək üçün məlumatı biristiqamətli funksiya ilə çevirməlidir. Yalnız gizli informasiyanı bilən şəxs tərs çevirməni edə və məlumatı bərpa edə bilər. Funksiyanın düz istiqamətini bilmək tərsi hesablamağa imkan vermədiyindən bu funksiyanı açıq nəşr etmək olar. Bu imkan da sahəyə açıq açarlı kriptoqrafiya adını verdi. Təsvir olunan sistemi açıq açarlı kriptosistem adlandırırlar, çünki şifrləmə alqoritmi hamıya məlumdur, yəni açıqdır. Belə sistemləri asimmetrik sistemlər də adlandırırlar, çünki alqoritmlərdə asimmetriya var: şifrləmə və deşifrləmə alqoritmləri müxtəlifdir.

Müasir asimmetrik alqoritmlərin əksəriyyətinin təhlükəsizliyi həlli hazırda çətin hesab olunan üç riyazi məsələy əsaslanır:

- böyük ədədlərin vuruqlara ayrılması (faktorizasiyası) məsələsi;
- sonlu meydanlarda diskret loqarifm məsələsi;
- elliptik əyrilərdə diskret loqarifm məsələsi.

Simmetrik sistemlərin hamı tərəfindən qəbul edilmiş üstünlüyü şifrləmənin daha yüksək sürəti, istifadə olunan açarın uzunluğunun kiçik olması və kriptoqrafik dözümün təmin edilməsinə daha əsaslı zəmanətin olmasıdır. Asimmetrik kriptosistemlər isə informasiyanın autentifikasiyası, istifadəçilər arasında açarların

paylanması kimi vacib məsələlərin həlli üçün daha rahat protokollar təklif edirlər. Lakin açıq açarlı kriptosistemlər böyük ədədlərlə işlədiklərindən sürətləri olduqca yavaşdır. Buna görə informasiyanın təhlükəsizliyi üçün *hibrid kriptosistemlər* istifadə edilə bilər, bu sistemlərdə həm simmetrik, həm də asimmetrik sistemlərin prinsipləri istifadə edilir – asimmetrik sistemlər açarların mübadiləsi və rəqəmsal imza üçün, simmetrik sistemlər isə şifrləmə üçün istifadə edilir.

Açıq açarların təhlükəsiz paylanmasını təmin etmək üçün açıq açar sertifikatları və onların autentikliyinə zəmanət verən açıq açar infrastrukturu (ing. Public Key Infrastructure, PKI) istifadə olunur. Bu mövzu 11-ci mühazirədə müzakirə olunacaq.

8.4. Zəruri riyazi məlumatlar və alqoritmlər

Açıq açarlı kriptoqrafiyada istifadə edəcəyimiz aşağıdakı teoremlər və xassələr isbatsız verilir.

Fermanın kiçik teoremi. Əgər p – sadə ədəddirsə və a ədədi p-yə bölünmürsə, onda $a^{p-1} \equiv 1 \bmod p$.

Eyler funksiyası. n-dən kiçik və n ilə qarşılıqlı sadə olan ədədlərin sayını $\phi(n)$ ilə işarə edirlər və Eyler funksiyası adlandırırlar.

Qəbul edilir ki, $\phi(1)=1$.

Misal. Asanlıqla tapmaq olar ki, $\phi(2)=1$, $\phi(3)=2$, $\phi(4)=2$, $\phi(5)=4$, $\phi(6)=2$.

$\phi(n)$ Eyler funksiyasının aşağıdakı xassələri asanlıqla isbat edilir:

1. $\phi(p) = p - 1$.

2. $\phi(p^k) = p^k - p^{k-1}$, $k \in N$.

3. *Eyler funksuiyasının multiplikativliyi.* $\text{ƏBOB}(a, b) = 1 \Rightarrow \phi(ab) = \phi(a)\phi(b)$.

Buradan p və q sadə ədədləri üçün $\phi(pq) = (p-1)\cdot(q-1)$.

Eyler teoremi. Qarşılıqlı sadə a və $n > 1$ ədədləri üçün $a^{\phi(n)} \equiv 1 \pmod n$.

Ferma teoremi $n = p$ sadə ədəd olduqda Eyler teoreminin nəticəsi olur.

Qalıqlar haqqında Çin teoreminin nəticəsi. $x \equiv a \pmod{n_i}$, $i = 1, 2, \ldots, k$ müqayisələr sistemi $n_1, n_2, \ldots, n_k$ – ədədləri qarşılıqlı sadə olduqda $x \equiv a \pmod n$ müqayisəsinə ekvivalentdir, burada $n = n_1 n_2 \ldots n_k$ dir.

Evklid alqoritmi

İki tam ədədin ən boyük ortaq bölənini tapır.

Giriş: mənfi olmayan iki a və b tam ədədi, $a \geq b$.

Çıxış: a və b ədədlərinin ən böyük ortaq böləni.

- $b \neq 0$ olduqca aşağıdakılar yerinə yetirilir:
- $r \leftarrow a \bmod b$, $a \leftarrow b$, $b \leftarrow r$.
- **Nəticə**: a.

Misal. ƏBOB(1180, 482)-nin hesablanması addımları:

$1180 = 2 \cdot 482 + 216$

$482 = 2 \cdot 216 + 50$

$216 = 4 \cdot 50 + 16$

$50 = 3 \cdot 16 + 2$

$16 = 8 \cdot 2 + 0.$

ƏBOB 0-dan fərqli axırıncı qalıqdır – $\text{ƏBOB}(1180,\ 482) = 2$.

Genişlənmiş Evklid alqoritmi

Genişlənmiş Evklid alqoritmi ƏBOB-la yanaşı həm də elə x və y tam əmsallarını tapır ki:

$$ax + by = ƏBOB(a, b).$$

Misal. $a = 30$, $b = 20$.

$\text{ƏBOB} = 10$; $x = 1$, $y = -1$. (Fikir verin ki, $30 \cdot 1 + 20 \cdot (-1) = 10$)

Genişlənmiş Evklid alqoritmi.

Giriş: Mənfi olmayan a və b ədədləri, $a \geq b$.

Çıxış: d= ƏBOB(a, b) və $ax+by=d$ şərtini ödəyən x və y tam ədədləri.

1. **if** $b=0$ **then** $d \leftarrow a$, $x \leftarrow 1$, $y \leftarrow 0$ **return**: d, x, y.
2. $x_1 \leftarrow 0$, $x_2 \leftarrow 1$, $y_1 \leftarrow 1$, $y_2 \leftarrow 0$.
3. **while** $b>0$ **do**:
   3.1. $q \leftarrow \lfloor a/b \rfloor$, $r \leftarrow a - bq$, $x \leftarrow x_2 - qx_1$, $y \leftarrow y_2 - qy_1$.
   3.2. $a \leftarrow b$, $b \leftarrow r$, $x_1 \leftarrow x_2$, $x_2 \leftarrow x$, $y_1 \leftarrow y_2$, $y_2 \leftarrow y$.
4. $d \leftarrow a$, $x \leftarrow x_2$, $y \leftarrow y_2$. **return**: d, x, y.

Genişlənmiş Evklid alqoritminin hesablamalarını kağız-qələmlə aparan zaman aşağıda misalda göstərilən "sehirli cədvəli" belə doldurmaq tövsiyə edilir.

1) Cədvəlin ilk 3 sütunu Evklid alqoritmi ilə hesablamalardır, onu yuxarıdan aşağıya hesablayırıq. Son sətirdə a sütununda $d = \text{ƏBOB}(a, b)$ olacaq.
2) Son sətirdə x və y-in qiymətlərini uyğun olaraq 1 və 0 yazırıq.
3) x və y sütunlarını aşağıdan yuxarıya doğru $x = x' - \lfloor a/b \rfloor \cdot y'$ və $y = x'$ düsturları ilə hesablayırıq, burada x' və y' bir aşağıdakı sətirdən götürülən qiymətlərdir, $\lfloor a/b \rfloor$ isə hesablama aparılan sətirdən götürülür.

Modula görə tərsin tapılması. Genişlənmiş Evklid alqoritmi ədədin modula görə tərsini sürətli hesablamağa imkan verir, bu əksər açıq açarlı sxemlər üçün vacibdir.

Tutaq ki, $a^{-1} mod\ b$-ni tapmalıyıq. Bunun üçün $ƏBOB(a,b) = 1$ olmalıdır. Deməli, genişlənmiş Evklid alqoritmi elə x və y verəcək ki: $ax + by = 1$ olacaq. Buradan $a^{-1} mod\ b = x$.

($ax + by = 1$-in hər tərəfindən $mod\ b$ alsaq, $ax = 1 \bmod b$ alınar!)

Misal. $12^{-1}\ mod\ 67$ hesablayın

Həlli. «Sehrli cədvəl»i quraq:

a	b	$\lfloor a/b \rfloor$	x	y
67	12	5	**-5**	**28**
12	7	1	3	-5
7	5	1	-2	3
5	2	2	1	-2
2	1	2	0	1
1	0	→	1	0

Cədvəldən alınır ki: $-5 \cdot 67 + 28 \cdot 12 = 1$.

Deməli, $12^{-1}\ mod\ 67$ =28-dir.

Yoxlama: 28×12 mod 67= 336 mod 67=1 mod 67.

Mühazirə üzrə yoxlama sualları

1. Açıq açarlı kriptosistemin əsas ideyası nədir?
2. Niyə açıq açarlı alqoritmlər simmetrik alqoritmlərdən daha yavaşdır?
3. Biristiqamətli funksiya nədir?
4. Biristiqamətli funksiyalara misallar söyləyin.
5. Gizli girişli biristiqamətli funksiya nədir?
6. Gizli girişli funksiya üçün hansı üç şərt ödənməlidir?
7. Açıq açarın təhlükəsiz paylanması necə təmin olunur?
8. Açıq açarlı kriptosistemdə açıq mətn hansı açarla şifrlənir?
9. İmzanın yoxlanması zamanı hansı açar istifadə olunur?
10. Məxfi açarlı kriptosistemləri niyə simmetrik adlandırırlar?
11. Açıq açarlı kriptosistemləri niyə asimmetrik adlandırırlar?
12. Açıq açarlı kriptosistemdə autentifikasiya necə həyata keçirilə bilər?
13. Hibrid kriptosistem nədir və nə üçün istifadə olunur?
14. Asimmetrik sistemlərin təhlükəsizliyi hansı riyazi məsələlərə əsaslanır?
15. Açıq açarlı kriptosistemin əsas təhlükəsi nədir?
16. Açıq açarlı kriptosistemlərdə açar uzunluğu niyə vacibdir?
17. Açıq açara görə gizli açarı hesablamaq mümkündürmü?
18. Eyler funksiyanın multiplikativlik xassəsi nədən ibarətdir?
19. $\phi(25)$ nəyə bərabərdir?

20. $\phi(253)$-i hesablayın.
21. $\phi(400)$-ü hesablayın.
22. Evklid alqoritmi nəyi hesablayır?
23. Evklid alqoritmi ilə ƏBOB(220, 31)-i hesablayın.
24. Genişlənmiş Evklid alqoritmi nəyi hesablayır?
25. Genişlənmiş Evklid alqoritmi ilə $31^{-1} \bmod 220$-ni hesablayın.

Mühazirə 9. Diskret loqarifm əsasında kriptosistemlər

9.1. Sonlu dövri qruplar

Tutaq ki, G multiplikativ qrupunda elə a elementi var ki, hər bir $b \in G$ elementi a elementinin qüvvətidir, yəni elə k tam ədədi var ki, $b=a^k$. Belə G qrupuna *dövri qrup* deyilir.

a elementinə G qrupunun *primitiv elementi* (*generatoru*) deyilir.

Misal. $Z_{19}^*=\{1, 2,\ldots, 18\}$ multiplikativ qrupuna baxaq. Qrupun tərtibi 18-dir. Qrup dövridir və doğuranı, məsələn, a=2-dir. Bunu aşağıdakı cədvəldən görmək olar.

2^1 mod 19 =2	2^2 mod 19 =4	2^3 mod 19 = =8	2^4 mod 19 = =16	2^5 mod 19 = =13	2^6 mod 19 = 7
2^7 mod 19= 14	2^8 mod 19 =9	2^9 mod 19=1 8	2^{10} mod 19= 17	2^{11} mod 19=1 5	2^{12} mod 19= 11
2^{13} mod 19 =3	2^{14} mod 19 =6	2^{15} mod 19 = 12	2^{16} mod 19 =5	2^{17} mod 19 = 10	2^{18} mod 19 =1

İsbat etmədən bir neçə riyazi faktı diqqətə çatdıraq:

1) Z_n^* multiplikativ qrupu yalnız və yalnız $n = p^k$ və ya $n = p^{2k}$ olduqda, yaxud $n = 2$ və ya $n = 4$ olduqda dövri qrupdur.
2) p sadə ədəd olduqda Z_p^* multiplikativ qrupunun primitiv elementlərinin sayı $\phi(p-1)$-dir.

Məsələn, Z_{19}^* qrupunun primitiv elementlərinin sayı $\phi(18) = 6$-dır.

Primitiv elementləri tapmaq üçün $2, 3, \ldots, p-1$ elementlərini bir-bir qüvvətə yüksəltməklə onların primitivliyini yoxlamaq olar. Böyük p sadə ədədləri üçün primitiv elementləri tapmaq üçün ədədlər nəzəriyyəsinə əsaslanan daha mürəkkəb metodlar istifadə edilir.

9.2. Diskret loqarifm məsələsi

Z_p^*-də diskret loqarifm məsələsi (Discrete Logarithm Problem, DLP) belə ifadə olunur:

- Tərtibi $p-1$ olan sonlu dövri Z_p^* qrupu və $a \in Z_p^*$ primitiv elementi və digər $b \in Z_p^*$ elementi verilib.
- Diskret loqarifm məsələsi elə $1 \le x \le p-1$ tam ədədini tapmaqdır ki, $a^x = b \bmod p$ olsun.
- $x = \log_a b \bmod p$ hesablaması *diskret loqarifmləmə* adlanır.

Misal: $5^x = 41 \bmod 47$ -dən x-i tapın.

Ümumiləşdirilmiş diskret loqarifm məsələsi

∘ qrup əməli ilə tərtibi n olan sonlu dövri G qrupu verilib.

$a \in G$ primitiv elementinə və başqa bir $b \in G$ elementinə baxılır.

Ümumiləşdirilmiş diskret loqarifm məsələsi elə x, $1 \leq x \leq n$ tam ədədini tapmaqdır ki:

$$b = \underbrace{a \circ a \circ \ldots \circ a}_{x \text{ dəfə}} = a^x$$

Kriptoqrafiyada istifadə edilmək üçün aşağıdakı diskret loqarifm məsələləri təklif edilmişdir:

1. Z_p^* sadə meydanının multiplikativ qrupu və ya onun altqrupu. Məsələn, klassik DHKE (Diffie-Hellman Key Exchange) ilə yanaşı ElGamal şifrləmə və DSA (Digital Signature Algorithm) da bu qrupdan istifadə edir.
2. Elliptik əyri ilə formalaşan dövri qrup;
3. $GF(2^m)$ Galua meydanının multiplikativ qrupu və ya onun altqrupu. DHKE kimi sxemləri onlarla reallaşdırmaq olar.
4. Hiperelliptik əyrilər və ya cəbri çoxobrazlılar, onlara elliptik əyrilərin ümumiləşdirilməsi kimi baxmaq olar.

Qeyd: 1 və 2 qrupları praktikada çox geniş istifadə edilir.

9.3. Diskret loqarifm məsələsinin həlli alqoritmləri

Diskret loqarifm məsələsinin həlli üçün aşağıdakı alqoritmlər vardır

- Universal alqoritmlər: İstənilən dövri qrupda işləyirlər:
  - Bütün variantların axtarışı (kobud güc hücumu)
  - Şenksin "Kiçik addımlar-Böyük addımlar" metodu;
  - Pollardın ro metodu – DLP məsələsinin həlli üçün ən sürətli alqoritmdir;
  - Pohlig-Hellman metodu.
- Qeyri-universal alqoritmlər: Yalnız konkret qruplarda, xüsusi halda Z_p^*-də işləyirlər, məsələn, indeks hesabı metodu.

Qeyd: Elliptik əyrilərə yalnız universal alqoritmlərlə hücum etmək olar, onlar qeyri-universal alqoritmlərə nisbətən zəifdirlər. Deməli, daha qısa açarlarla elliptik əyrilər Z_p sadə meydanında DLP-yə nisbətən daha təhlükəszdirlər.

9.4. Kiçik addımlar – böyük addımlar alqoritmi

Giriş. Tərtibi n-ə bərabər G dövri qrupunun generatoru a və $b \in G$ elementi.
Çıxış. $x = \log_a b$ diskret loqarifmi ($a^x = b$)
Kiçik addımlar:

1. $m \sim \sqrt{n}$
2. $0 \leq i < m$ üçün elementləri $(i,\ a^i)$ olan cədvəl qurulur. Bu cədvəl ikinci komponentə görə nizamlanır.
3. **Böyük addımlar**

3. a^{-m} hesablanır və $\gamma = b$ qəbul edilir.

4. $j = 0$-dan m–1-ə qədər aşağıdakılar edilir:

4.1. γ-nın cədvəldə olması yoxlanılır.

4.2. Əgər cədvəldə $\gamma = a^i$ olarsa, onda Nəticə : $x = jm + i$.

4.3. $\gamma = \gamma \cdot a^{-m}$ qəbul edilir. Addım 4.1-ə keçid.

Alqoritmin zaman çətinliyi $O(\sqrt{n})$-dir, yaddaş çətinliyi də $O(\sqrt{n})$ tərtibindədir.

Misal. Tutaq ki, $p = 433$ və Z_p^*-da primitiv kök $a = 7$-dir. $\log_7 166 \ mod\ 443$-ü hesablamaq istəyirik.

$m = 21 \sim \sqrt{443}$.

Kiçik addımlar: $0 \leq i < m$ üçün $a^i mod\ p$ hesablayırıq.

i	a^i
0	$a^0 = 1$
1	$a^1 = 7$
2	$a^2 = 49$
3	$a^3 = 49$
4	$a^4 = 236$
5	$a^5 = 353$
6	$a^6 = 306$
7	$a^7 = 410$
8	$a^8 = 272$
9	$a^9 = 172$
10	$a^{10} = 338$

i	a^i
11	$a^{11} = 201$
12	$a^{12} = 108$
13	$a^{13} = 323$
14	$a^{14} = 96$
15	$a^{15} = 239$
16	$a^{16} = 374$
17	$a^{17} = 20$
18	$a^{18} = 140$
19	$a^{19} = 114$
20	$a^{20} = 365$

Böyük addımlar. $a^{-m} = a^{-21} = 292$.

$0 \leq j < m$ üçün $ba^{-mj} mod\ p$ hesablanır ($b = 166$) və kiçik addımlar cədvəlində axtarılır:

- $j = 0$: $b \cdot a^0 = 166$
- $j = 1$: $b \cdot a^{-21} = 409$

- $j = 2$: $b \cdot a^{-42} = 353\ (= a^5)$ – kiçik addımlar cədvəlində tapıldı!

Cavab: $x = 2 \cdot 42 + 5 = 47$.

Z_p* -da diskret loqarifmin hesablanması üzrə rekordlar

Onluq rəqəmlərin sayı	Bit uzunluğu	Tarix
58	193	1991
68	216	1996
85	282	1998
100	332	1999
120	399	2001
135	448	2006
160	532	2007

p ədədinin uzunluğu haqqında aşağıdakı tövsiyələri vermək olar.

DLP-nin həllinə yönəlmiş hücumların qarşısını almaq üçün Z_p^*-da Diffi-Hellman kimi sxemlərdə uzunluğu ən azı 1024 bit olan sadə ədədlər istifadə etmək tövsiyə olunur.

Verilənlərin 20 il ərzində qorunması üçün uzunluğu 2048 bit, 35 il – 3072 bit, 50 il – 4096 bit olan sadə ədəddən istifadə etmək lazımdır. Hazırda hesab olunur ki, 6800 bit uzunluq təhlükəsizliyə ən yüksək tələbləri ödəyir. Lakin belə uzunluq kriptosistemin məhsuldarlığını olduqca aşağı salır.

9.5. Diffi-Helman açar mübadiləsi sxemi (Diffie–Hellman Key Exchange, DHKE)

Açarların açıq rabitə kanalları üzərindən təhlükəsiz mübadiləsi üçün ilk alqoritm 1976-cı ildə U. Diffi və M. Hellman tərəfindən ”Kriptoqrafiyada yeni istiqamətlər” adlı məqalədə təklif olunmuşdu. Protokol iki istifadəçiyə açıq rabitə kanalından istifadə etməklə ümumi məxfi açar almağa imkan verir. DHKE açıq açarlı kriptoqrafiyanın fundamental metodudur və TLS, SSH və IPsec kimi müxtəlif təhlükəsizlik protokollarında geniş istifadə olunur.

Alqoritmin reallaşdırılması üçün iki parametr tələb olunur:

p −böyük sadə ədəd;

g − $Z_p^* = \{1,\ 2, \dots, p-1\}$ qrupunu doğuran primitiv element.

p və g ədədləri açıqdır və onlar bir neçə istifadəçi tərəfindən istifadə edilə bilər. Bu ədədlər əvvəlcədən əldə olunmalı və bütün abonentlərə göndərilməlidir (və ya əlyetər mənbələrdə nəşr edilməlidir).

Əgər A və B ümumi məxfi açar yaratmaq istəyirlərsə, onlar aşağıdakı addımları yerinə yetirməlidirlər:

1. A təsadüfi a, $1 \le a \le p-2$ tam ədədini seçir və B-yə $A = g^a \bmod p$ qiymətini göndərir.
2. B təsadüfi b, $1 \le b \le p-2$ tam ədədini seçir və A-ya $B = g^b \bmod p$ qiymətini göndərir.
3. A hesablayır: $k = B^a \bmod p$.
4. B hesablayır: $k' = A^b \bmod p$.

Asanlıqlq görmək olar ki, $k = k' = g^{ab} \bmod p$.

9.6. Oakley açar müəyyənetmə protokolu

Diffi-Hellman açar mübadiləsi protokolunun aşağıdakı məhdudiyyətləri var:

- Ortadakı adam hücumuna məruz qalır (autentifikasiya yoxdur);
- Tərəflərin kimliyi ilə bağlı heç bir məlumat vermir;
- Hesablamalar baxımından intensivdir. Nəticədə, rəqibin çox sayda açar tələb etdiyi *tıxanma hücumuna* qarşı həssasdır. Qurban real işdən daha çox faydasız modul üzrə qüvvətə yüksəltmə üçün xeyli hesablama resursları sərf edir.

IPSec-ə daxil olan Oakley protokolu bu məhdudiyyətləri aradan qaldırır:

- Tıxanma hücumlarının qarşısını almaq üçün kukilər kimi tanınan bir mexanizmdən istifadə edir. Kuki mübadiləsi tələb edir ki, hər bir tərəf ilkin mesajda psevdo-təsadüfi ədəd – kuki göndərsin, qarşı tərəf onu təsdiqləməlidir. Bu təsdiqləmə Diffi-Hellman açar mübadiləsinin ilk mesajında təkrarlanmalıdır. Mənbə ünvanı saxtadırsa, rəqib cavab almır. Beləliklə, rəqib istifadəçini Diffi-Hellman hesablamasını yerinə yetirməyə deyil, yalnız təsdiqləmələr yaratmağa məcbur edə bilər.
- Təkrarlama hücumlarına qarşı təminat vermək üçün *Nonce* istifadə edilir. Bu, Diffi-Hellman açıq açar qiymətlərinin mübadiləsinə imkan verir.
- Ortadakı adam hücumlarının qarşısını almaq üçün Diffi-Hellman mübadiləsi autentifikasiya edilir. Oakley ilə üç müxtəlif autentifikasiya metodu istifadə edilə bilər: 1) Rəqəmsal imzalar; 2) Açıq açarlı şifrləmə; 3) Simmetrik açarlı şifrləmə.

9.7. ElGamal şifrləmə sistemi

Taher ElGamal tərəfindən 1985-ci ildə təklif olunan kriptoqrafik sistem həm rəqəmsal imza, həm də şifrləmə üçün istifadə edilə bilər.

ElGamal kriptosistemində parametrlərin seçilməsi və açarların generasiyası. Tutaq ki, p – böyük sadə ədəd, g – GF(p) sonlu meydanının primitiv elementidir. Həm g-ni, həm də p-ni istifadəçilər qrupu üçün ortaq etmək olar.

Sistemin hər bir istifadəçisi açıq açar və uyğun gizli açar yaradır. Bunun üçün istifadəçi A aşağıdakı əməliyyatları yerinə yetirir.

1. $1 \leq x \leq p-1$ şərtini ödəyən təsadüfi x ədədi generasiya edir.
2. $y = g^x \bmod p$ hesablayır.
3. A-nın açıq açarı (p, g, y), gizli açarı x-dir.

Misal 1. ElGamal kriptosistemində parametrlərin seçilməsi və açarların yaradılması

1: Tutaq ki, $p = 23$

2: Primitiv element g = 11 seçilir.

3: $x = 6$ gizli açarı seçilir.

4: $y = 11^6 \pmod{23} = 9$ hesablanır.

Açıq açar 9; Gizli açar 6-dır.

ElGamal şifrləmə. Fərz edək ki, açıq mətn p moduluna görə ədədlər ardıcıllığı kimi göstərilib. M açıq mətnini şifrləmək üçün aşağıdakı addımlar edilir:

1. Təsadüfi k ədədi generasiya edilir. Onu *efemer açar* adlandırırlar.
2. İki C_1 və C_2 qiymətləri hesablanır, burada:
   $C_1 = g^k \bmod p$ və $C_2 = My^k \bmod p$.
3. C şifrmətni C_1 və C_2 qiymətlər cütündən ibarətdir.

1-ci addımda k ədədinin təsadüfi seçilməsi ElGamal şifrləmə sistemini deterministik yox, *ehtimallı* edir. k ədədi yalnız bir dəfə istifadə edilməli və sonra məhv edilməlidir.

ElGamal deşifrləmə. Şifrmətni alan x gizli açarı ilə C_1-i çevirir:

$$C_1{}^x = (g^k)^x \bmod p$$

QEYD: $C_1{}^x = (g^k)^x = (g^x)^k = (y)^k = y^k \bmod p$

C_2-ni bu ədədə bölməklə M açıq mətni tapılır. Başqa sözlə:

$$\frac{C_2}{y^k} = (My^k)/y^k = M \bmod p$$

Misal 2. ElGamal şifrləmə:

M = 10 açıq mətnini $y = 9$ açıq açarı ilə şifrləyək

1. Tutaq ki, $k = 3$ təsadüfi ədədi generasiya edilir.
2. $C_1 = 11^3 \bmod 23 = 20$; $C_2 = 10 \cdot 93 \bmod 23 = 10 \cdot 16 = 160 \bmod 23 = 22$ hesablanır.

3. Şifrmətn C = (20, 22) cütüdür.

Misal 3. ElGamal deşifrləmə:

C = (20 , 22) şifrmətnini deşifrləmək üçün**:**

1. $20^6 = 16 \bmod 23$ hesablanır.
2. $22/16 = 10 \bmod 23$ hesablanır.
3. Açıq mətn M = 10.

ElGamal kriptosisteminin təhlükəsizliyi sonlu meydanda diskret loqarifm məsələsinin hesablamaların həcmi cəhətdən çətinliyinə əsaslanır. $GF(p)$ sonlu meydanında **diskret loqarifm məsələsi** belə ifadə olunur: p, r sadə ədədləri və tərtibi r olan $g<p$ natural ədədi verilib, yəni $g^r \equiv 1 \pmod p$, $y = g^x \pmod p$ qiymətini bilərək $x \in Z$ qiymətini tapmaq tələb olunur.

ElGamal şifrləməsiinin homomorfizm xassəsi. Tutaq ki, m_1, m_2 açıq mətnlərdir. Fərz edək ki, $e_k(m)$ isə Z_p^*-də ElGamal şifrləmə funksiyasıdır. Onda

$$e_{k_1}(m_1) = (\mathrm{g}^{k_1}, m_1 \mathrm{A}^{k_1})$$
$$e_{k_2}(m_2) = (\mathrm{g}^{k_2}, m_2 \mathrm{A}^{k_2})$$

Şifrləmələrin hasili $Z_p^* \times Z_p^*$-də koordinatlar üzrə vurma kimi təyin edək:

$$e_{k_1}(m_1) \star e_{k_2}(m_2) = (\mathrm{g}^{k_1}, m_1 \mathrm{A}^{k_1}) \star \ (\mathrm{g}^{k_2}, m_2 \mathrm{A}^{k_2}) =$$
$$(\mathrm{g}^{k_1+k_2}, m_1 m_2 \mathrm{A}^{k_1+k_2}) = e_{k_1+k_2}(m_1 m_2).$$

Beləliklə, m_1, m_2 şifrmətnlərinin hasili $m_1 m_2$ hasilinin şifrləməsinə bərabərdir. Bu xassəyə ElGamal şifrləməsinin ***homomorfizm xassəsi*** deyilir. (Xatırladaq ki, RSA-nın da oxşar xassəsi var).

Mövzu üzrə yoxlama sualları

1. Multiplikativ qrup nədir?
2. Hansı qrupa dövri qrup deyilir?
3. Hansı elementə qrupun generatoru (primitiv elementi) deyilir?
4. Z_p^* multiplikativ qrupunun elementləri nələrdir?
5. Multiplikativ qruplarda diskret loqarifm məsələsi nədir?
6. Ümumiləşdirilmiş diskret loqarifm məsələsi necə ifadə olunur?
7. Hansı qruplarda diskret loqarifm məsələsi kriptoqrafiyada istifadə edilir?
8. Hansı qruplarda diskret loqarifm məsələsi praktikada geniş istifadə edilir?
9. Universal alqoritmlər hansı qruplarda işləyirlər?
10. Qeyri-universal alqoritmlər hansı qruplarda işləyirlər?
11. Hansı qruplarda diskret loqarifm məsələsi çətindir?
12. Diskret loqarifm məsələsinin həlli üçün hansı alqoritmlər var?
13. “Kiçik addımlar-Böyük addımlar” alqoritminin zaman mürəkkəbliyi necədir?

14. Diffi-Helman açar mübadiləsi sxemi necə işləyir?
15. Diffi-Hellman açar mübadiləsi hansı riyazi ideyaya əsaslanır?
16. Diffi-Hellman sxemi "ortada adam" hücumlarına qarşı dayanıqlıdırmı?
17. Diffi-Hellman açar mübadiləsi protokolunun hansı məhdudiyyətləri var?
18. Oakley tıxanma hücumlarına qarşı hansı mexanizmdən istifadə edir?
19. Oakley təkrarlama hücumlarına qarşı hansı mexanizmdən istifadə edir?
20. Oakley autentifikasiya üçün hansı üsullardan istifadə edir?
21. ElGamal kriptosistemində gizli açar və açıq açar necə yaradılır?
22. ElGamal şifrləmə sistemi determinikdir, yoxsa ehtimallı?
23. ElGamal şifrləməsində efemer açarın üzərinə hansı şərt qoyulur?
24. ElGamal şifrləmə sisteminin təhlükəsizliyi hansı məsələyə əsaslanır?
25. ElGamal şifrləməsində homomorfizm xassəsi nədir?

Mühazirə 10. RSA kriptosistemi

10.1. RSA-da açarların generasiyası və şifrləmə/deşifrləmə

Açıq açarlı şifrləmə alqoritmlərindən ən geniş yayılanı RSA alqoritmi müəlliflərinin familiyalarının ilk hərfi ilə adlandırılıb (**R**ivest R, **S**hamir A, **A**delman L.). Müəlliflər bu metodu 1977-ci ildə Massaçusets Texnologiya İnstitutunda birgə tədqiqatları zamanı təklif etmişlər. Alqoritmi həm şifrləmə, həm də rəqəmsal imza üçün istifadə etmək olar.

Açarların generasiyası. İnformasiya mübadiləsinin hər bir iştitakçısı aşağıdakı qaydalara uyğun olaraq açarlar cütü (açıq və gizli) generasiya edir:

1. Bir-birinə bərabər olmayan iki böyük p və q sadə ədədləri **seçilir**.
2. Sistemin modulu $n = p \cdot q$ və Eyler funksiyası $\phi(n) = (p-1)\cdot(q-1)$ **hesablanır**.
3. $1<e<\phi(n)$ şərtini ödəyən və $\phi(n)$ ilə qarşılıqlı sadə olan e ədədi **seçilir**.
4. $e \cdot d = 1 \pmod{\phi(n)}$, $1 < d < \phi(n)$ şərtlərini ödəyən d ədədi **hesablanır**.
5. (d, n) ədədlər cütü **gizli açar**, (e, n) ədədlər cütü **açıq açardır**.

Şifrləmə m məlumatı şifrlənir ($m < n$ tam ədədir): $c = E(m) = m^e \bmod n$.

Deşifrləmə: c şifrmətni deşifrlənir: $m = D(c) = c^d \bmod n$.

RSA-nın düzgün işlənməsinin əsaslandırılması. Istənilən $m \in Z_n$ üçün $D(E(m))=m$ olduğunu isbat etmək lazımdır.

$\phi(n)$-i p və q ilə ifadə edərək $ed = 1 + k\phi(n) = 1 + k(p-1)(q-1)$ alırıq. Əgər $(m, p)=1$ olarsa, onda Fermanın kiçik teoreminə görə

$$c^d \equiv (m^e)^d \equiv m^{1+k(p-1)(q-1)} \equiv m \cdot (m^{p-1})^{(q-1)} \equiv m \cdot 1^{q-1} (\text{mod } p) \equiv m \ (\text{mod } p).$$

Əgər $(m, p)=0$ olarsa, onda $m^{ed} \equiv m (\text{mod } p)$ olduğu aşkardır. Deməli, istənilən $m \in Z_n$ üçün

$$c^d \equiv m \ (\text{mod } p).$$

Analoji olaraq isbat edilir ki, istənilən $m \in Z_n$ üçün aşağıdakı müqayisə ödənir:

$$c^d \equiv m \ (\text{mod } q).$$

$n=pq$ olduğu üçün qalıqlar haqqında Çin teoreminin nəticəsinə görə alırıq:

$$c^d \equiv (m^e)^d \equiv m \ (\text{mod } n).$$

$m>n$ halında RSA özünü blok şifri kimi aparır və məlumatların şifrlənməsi aşağıdakı kimi aparılır.

1) məlumat m_i bloklara bölünür, blokun uzunluğu $10^{k-1} < n < 10^k$ bərabərsizliyinə uyğun gələn tam k ədədi ilə müəyyən olunur.

2) $c_i = m_i^e \ (\text{mod } n)$ qiymətləri hesablanır.

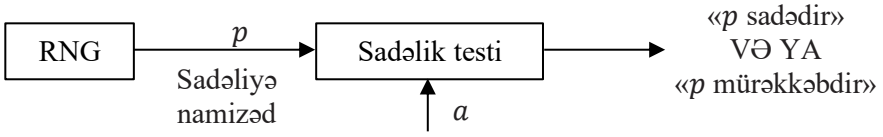
Deşifrləmə $m_i = c_i^d \ (\text{mod } n)$ qiymətlərinin hesablanmasından ibarətdir.

RSA alqoritminin əsas problemlərindən biri şifrləmə/deşifrləmə əməliyyatlarının sürətinin aşağı olmasıdır. RSA təxminən 100-1000 dəfə DES-dən yavaş işləyir. Uzun məlumatlar üçün onin istifadəsi səmərəli olmur. Buna görə asimmetrik alqoritmlər əsasən açıq rabitə kanalları ilə seans açarlarının paylanması üçün istifadə edilir. Simmetrik açar asimmetrik şifrləmə alqoritminin köməyi ilə şifrlənmiş şəkildə ötürülür, bundan sonra verilənlərin mübadiləsi simmetrik şifrləmə alqoritmlərdən istifadə etməklə aparılır. Açıq açarlı alqoritmlər həmçinin rəqəmsal imza üçün də istifadə edilir.

10.2. Sadə ədədlərin generasiyası

RSA-da açarların generasiyası iki böyük p və q sadə ədədlərini tələb edir.

Sadə ədədləri tapmaq üçün təsadüfi ədədlər generatoru ilə (Random number generator, RNG) ilə təsadüfi ədədlər generasiya edilir və onların sadəliyi test edilir (şəkil 10.1).



Şəkil 10.1.

RNG *proqnozlaşdırılmayan* olmalıdır, əks halda, hücum edən n-in vuruqlara ayrılmasını təxmin edə bilər.

Sadəlik testləri. Adətən, p və q-nü vuruqlara ayırmaqla sadəliyini test etmək hesablamaların həcmi baxımından mümkün olmur. Bizi vuruqlara ayırmaq maraqlandırmır, p və q-nün mürəkkəb olub-olmadığı maraqlandırır. Adətən, sadəlik testləri ehtimallı olur, yəni onlar 100% dəqiq deyillər, lakin çıxış yüksək ehtimalla doğrudur. Ehtimallı testin iki çıxışı var:

- "p mürəkkəbdir" – həmişə True olur;
- "p sadədir" – yalnız müəyyən ehtimalla True olur.

Məşhur sadəlik testləri: Ferma testi, Miller-Rabin testi, Solovey-Ştrassen testi, Lemann testi və s.

Miller-Rabin testi daha geniş istifadə edilir.

Tutaq ki, n − tək ədəddir və $n - 1 = 2^s t$, t − tək ədəddir. Əgər n ədədi sadədirsə, onda istənilən $a > 1$ üçün

$$a^{n-1} \equiv 1 \pmod{n}$$

müqayisəsi ödənir. Buna görə $\{a^t, a^{2t}, \ldots, a^{2^{s-1}t}\}$ elementlərinə baxaraq görmək olar ki, ya onların arasında $-1 \pmod{n}$-ə bərabər olanı tapılar, ya da $a^t \equiv 1 \pmod{n}$.

Aşağıdakı Miller-Rabin sadəlik testi bu qeydin əsasında qurulub:

1. {1, 2, …, $n-1$} intervalından a ədədini təsadüfi seçirik və Evklid alqoritminin köməyi ilə $(a, n) = 1$ şərtini yoxlayırıq;
2. Əgər şərt ödənmirsə, onda cavab «n − mürəkkəbdir»;
3. $a^t \pmod n$ hesablanır;
4. Əgər $a^t \equiv \pm 1 \pmod n$ olarsa, onda addım 1-ə keçirik;
5. -1 alınana qədər $a^{2t}, \ldots, a^{2^{s-1}t}$ hesablanır;
6. Əgər bu ədədlərdən heç biri -1-ə bərabər deyilsə, onda cavab «n − mürəkkəbdir»;
7. Əgər -1 alınsa, onda cavab məlum deyil (və testi bir daha təkrar etmək olar).

Bu test mürəkkəb ədədi $\frac{1}{4}$ ehtimalı ilə sadə ədəd kimi qəbul edir. Miller-Rabin testi s dəfə yerinə yetirilsə, mürəkkəb ədəd 4^{-s} ehtimalı ilə sadə ədəd kimi qəbul olunacaq. s=10 dəfə yerinə yetirilsə, bu ehtimal təxminən milyonda bir olur.

10.3. Qüvvətin hesablanması alqoritmi (ing. Square & multiply algorithm)

$a^b \bmod n$ hesablamaq tələb olunur. Tutaq ki, b ədədi ikilik şəkildə göstərilib: $b = (b_k b_{k-1} \ldots b_0)_2$.

$d = 1$
for $i \leftarrow k$ **downto** 0
  do
    $d \leftarrow (d \cdot d) \mod n$
    if $b_i = 1$
      then $d \leftarrow (d \cdot a) \bmod n$
return d

Qeyd edək ki, Square-and-multiply alqoritminin çətinliyi loqarifmikdir, yəni onun yerinə yetirilmə müddəti qüvvət üstünün bit uzunluğuna mütənasibdir (mütləq qiymətinə deyil): $b = (b_k b_{k-1} \ldots b_0)_2$ qüvvət üstündə $k + 1$ bit olarsa, onda kvadrata yüksəltmələrin sayı $\#SQ = k$, vurmaların orta sayı $\#MUL = 0.5k$ olacaq. Ümumi çətinlik: $\#SQ + \#MUL = 1.5k$ olur.

Modulyar qüvvətə yüksəltmə hesablama baxımından intensivdir. Hətta square-and-multiply alqoritmi ilə də RSA smart kartlar kimi məhdud resurslu qurğularda kifayət qədər yavaş ola bilər.

Sürətləndirmə üçün bəzi vacib üsullar aşağıdakılardır:

- Kiçik açıq qüvvət üstü e
- Çin qalıq teoremi (Chinese Remainder Theorem, CRT);
- Əvvəlcədən hesablanmaqla qüvvətə yüksəltmə.

10.4. RSA kriptosisteminin təhlükəsizliyi

RSA kriptosisteminin təhlükəsizliyi böyük tam ədədləri vuruqlara ayırma (faktorizasiya) məsələsinin çətinliyinə əsaslanır. Doğrudan da, deşifrləmə üçün biz d qüvvət üstünü bilməliyik. Məlum e-yə görə d-ni tapmaq üçün $\phi(n)$ Eyler funksiyasının qiymətini bilmək lazımdır. $\phi(n)$-ni hesablamaq üçün n ədədini p və q vuruqlarına ayırmaq lazım gələcək. Buna görə n-in vuruqlara ayrılması (faktorizasiyası) məlumdursa, onda şifrlənmiş mətnə görə məlumatı asanlıqla tapmaq olar.

Vuruqlara ayırmanın riyazi metodlarını inkişaf etdirmək məqsədi ilə RSA Data Security, Inc korporasiyası RSA-modulların faktorizasiyası üzrə açıq müsabiqə elan etmişdi. Bu müsabiqə barəsində İnternet-dən məlumat almaq olar.

RSA-nın təhlükəsizliyi təxminən 50 ilə yaxındır ki, araşdırılır və çoxsaylı hücumlar təklif edilib. Bu mühazirədə yalnız bir neçə hücum üsuluna baxılacaq.

Kiçik açıq açar. Şifrləmə sürətini artırmaq məqsədi ilə praktikada kiçik şifrləmə eksponentindən istifadə olunur. e eksponentinin kiçik seçilməsi neqativ nəticələrə gətirib çıxara bilər. Məsələ ondadır ki, bu halda bir neçə istifadəçidə eyni e açıq açarı ola bilər. Bu zaman həmin istifadəçilərə göndərilmiş şifrlənmiş eyni məlumatlara (məsələn, sirkulyar məktub) əsasən həmin məlumatı deşifrləmək mümkündür.

Kiçik gizli açar. Əgər $d < \sqrt[4]{n}$ olarsa, d-ni kəsilməz kəsrlərin köməyi ilə asanlıqla tapmaq olar. Bu faktı M. Viner isbat etmişdi. Belə fərziyyə var ki, $\sqrt{n}$-ə qədər bütün gizli açarlar təhlükəsiz deyil.

RSA-nın açarsız oxunması metodu. Düşmənə açıq açar (e, n) və c şifrmətni məlumdur. İlkin mətn m-i tapmaq lazımdır.

Düşmən $c^{ej}(mod\ n) = c$ bərabərliyini ödəyən j ədədini tapır, yəni düşmən ələ keçirilmiş şifrmətni bu bərabərlik alınana qədər açıq açar ilə ardıcıl şifrləyir:

$$y_1 = c\ mod\ n,\ y_2 = c^e mod\ n = y_1^e mod\ n,\ldots,\ y_j = c^{ej} = \left(y_{j-1}\right)^e mod\ n.$$

Belə j taparaq, düşmən $m = y_{j-1}$ qəbul edə bilər! Bu aşağıdakı bərabərlikdən alınır:

$$y_{j-1} = c^{e(j-1)} mod\ n = \left(c^{j-1}\right)^e (modn) = m^e mod\ n.$$

Homomorfizm xassəsi. RSA sistemi multiplikativlik xassəsinə malikdir. Tutaq ki, m_1 və m_2 – iki müxtəlif açıq mətn, c_1 və c_2 – onlara uyğun şifrmətnlərdir. Qeyd edək ki:

$$(m_1 m_2)^e mod\ n = (m_1)^e (m_2)^e mod\ n = c_1 c_2 mod\ n.$$

Başqa sözlə, $m=m_1m_2$ açıq mətninin şifrmətni $c=c_1c_2$(mod n)-dir. Bu xassə RSA-nın *homomorfizm xassəsi* adlandırılır.

Həmçinin qeyd edək ki, RSA şifrləməsi determinikdir, yəni eyni mətni eyni açarla şifrləsək, eyni şifrmətn alınar.

RSA-nın homomorfizm və determiniklik xassələrinə əsaslanan hücumların qarşısını almaq üçün RSA şifrləmədə *təsadüfi padding* sxemlərindən istifadə edilir. Misal olaraq, OAEP(Optimal Asymmetric Encryption Padding) göstərmək olar.

10.5. RSA-da parametrlərin seçilməsi

RSA-nın təhlükəsizliyini təmin etmək üçün kriptosistemin parametrlərinin seçilməsinə aşağıdakı tələblər irəli sürülür:

1) p və q sadə ədədləri böyük olmalıdırlar;
2) $|p - q|$ fərqi böyük olmalıdır;
3) $p \pm 1$, $q \pm 1$, $r-1$, $s-1$ ədədləri böyük sadə vuruqlara malik olmalıdırlar (r və s uyğun olaraq $p - 1$ və $q - 1$-in ən böyük bölənləridir);
4) ƏBOB($p-1$, $q-1$) kiçik olmalıdır.

Müasir qiymətləndirmələrə görə 20 il ərzində verilənlərin qorunmasını təmin etmək üçün n ədədinin uzunluğu təxminən 2048 bit olmalıdır. Mümkün olsa, uzunluğu 4096 bit ətrafında olan modullardan istifadə etmək tövsiyə olunur. Perspektivdə daha uzun modullar, məsələn, 8192 bit tələb edilə bilər.

Mühazirə üzrə yoxlama sualları

1. RSA-nın təhlükəsizliyi hansı riyazi məsələyə əsaslanır?
2. RSA-da açıq açar necə yaradılır?
3. RSA-da gizli açar necə hesablanır?
4. RSA-da şifrləmə əməliyyatı necə aparılır?
5. Nə zaman RSA blok şifr olur?
6. RSA-da deşifrləmə əməliyyatı necə aparılır?
7. RSA-da Eyler funksiyası $\varphi(n)$ necə hesablanır?
8. Niyə çox vaxt e kiçik seçilir?
9. Modulyar tərsin tapılması RSA üçün niyə vacibdir?
10. Modulyar tərsin tapılması üçün hansı alqoritm istifadə edilir?
11. RSA-nın işləməsi hansı teoremlə sübut olunur?
12. Sadə ədədlərin generasiyası üçün əsas yanaşma nədir?
13. Ədədin sadəliyi testlərinə aid nümunələr söyləyin.
14. Miller-Rabin alqoritmi nə üçün istifadə edilir?
15. RSA alqoritminin əsas problemlərindən biri nədir?
16. Məlumatın RSA ilə birbaşa imzalanması hansı təhlükəyə səbəb ola bilər?
17. Square-and-multiply alqoritminin çətinliyi necə qiymətləndirilir?
18. Square-and-multiply alqoritmində vurmaların sayı nəyə bərabərdir?
19. Square-and-multiply alqoritmində kvadrata yüksəltmələrin sayı nəyə bərabərdir?

20. CRT (Chinese Remainder Theorem) RSA-ya nə üstünlük verə bilər?
21. RSA-nın homomorfizm xassəsi nədir?
22. RSA-nın homomorfizm xassəsinə “qarşı” hansı üsul işlədilir?
23. RSA-nın açarsız oxunması metodu necə işləyir?

Mühazirə 11. Rəqəmsal imza sxemləri

11.1. Rəqəmsal imza sxemlərinin növləri

Rəqəmsal imzanın əsas məqsədləri məlumatın mənbəyinin autentikliyini, məlumatın tamlığını və müəllifliyin inkar edilə bilməməsini təmin etməkdir.

Rəqəmsal imza yalnız imzalayan şəxsə məlum olan müəyyən məxfi parametrdən (gizli açardan) və imzalanan məlumatın məzmunundan asılı olan sabit uzunluqlu verilənlər sətridir. Ümumilikdə, rəqəmsal imza belə işləyir: göndərən məlumatı gizli açarla imzalayır, qəbul edən tərəf isə imzanı açıq açarla yoxlayır.

Rəqəmsal imza sxemini qurmaq üçün üç alqoritmi müəyyən etmək zərurudir: 1) kriptoqrafik açarların generasiyası alqoritmi, 2) imzalama alqoritmi və 3) imzanı yoxlama alqoritmi.

Mövcud rəqəmsal imza sxemlərini iki sinfə bölürlər:

- məlumatı bərpa etməklə rəqəmsal imza sxemləri;
- məlumatı əlavə etməklə rəqəmsal imza sxemləri.

Məlumatı bərpa etməklə rəqəmsal imza sxemlərində imzalanmış məlumatın hamısı və ya bir hissəsi bilavasitə rəqəmsal imzadan bərpa edilir. Bu sxemlərdə imzaların uzunluğu çox böyük olur və praktikada, demək olar ki, istifadə edilmir.

Məlumatı əlavə etməklə rəqəmsal imza sxemlərində rəqəmsal imza məlumata qoşulur və bu şəkildə ünvana göndərilir. Belə rəqəmsal imzanı yoxlamaq üçün əldə imza və imzalanmış məlumat olmalıdır.

11.2. Məlumatı əlavə etməklə rəqəmsal imza

Məlumatı əlavə etməklə rəqəmsal imza sxemləri praktiki tətbiqlərdə ən geniş yayılmış sxemlərdir. Rəqəmsal imzaların bu sinfinə ElGamal, Schnorr, DSA, QOST 34.10 kimi imza sxemləri aiddir. Məlumatı əlavə etməklə rəqəmsal imzanın ümumi modelinə baxaq.

Tutaq ki, uzunluğu ixtiyari olan hər hansı m məlumatını imzalamaq lazımdır.

1. Kriptoqrafik heş funksiyadan istifadə etməklə məlumatın $h(m)$ heş qiyməti hesablanır.
2. İmzalayan özünün k_{pr} gizli açarından, SIG(•) imza yaratma alqoritmindən və məlumatın $h(m)$ heş qiymətindən istifadə etməklə $s = \mathrm{SIG}_{k_{pr}}(h(m))$ imzasını yaradır.
3. m məlumatı və ona qoşulmuş s imzası alana göndərilir.

Beləliklə, burada "heşlə-və-imzala" üsulu istifadə edilir – məlumat əvvəlcə heşlənir, sonra bu heş qiymət imzalanır. Heş funksiyanın buradakı rolu belədir – imzalanan məlumatın ölçüsünü azaldır və dəyişikliklərə qarşı dayanıqlığı təmin edir.

Alan tərəf imzanı aşağıdakı kimi yoxlayır.

1. İmzalayanın imza yoxlama açarının (açıq açarının) k_{pub} autentik surətini alır (açıq açarlar infrastrukturundan).
2. Alınmış m' məlumatına görə $h(m')$ heş-qiymətini hesablayır.
3. VER(•) imzanı yoxlama alqoritmindən istifadə edərək imzanın doğru və ya yalan olması haqqında qərar qəbul edir:

$$\text{VER}_{k_{pub}}(h(m'), \text{s}) \rightarrow \{\text{true, false}\}.$$

11.3. RSA imza sxemi

Gizli və açıq açarın yaradılması. Açarlar cütü RSA şifrləmədəki kimi generasiya edilir.

İmzanın yaradılması. x məlumat gizli açarla (d) şifrlənir:

- $s = sig_{K_{priv}}(x) = x^d mod\, n.$
- s məlumata əlavə olunur: (x, s)

İmzanın yoxlanması. İmza açıq açarla deşiifrlənir:

- $x' = ver_{K_{pub}}(s) = s^e mod\, n.$
- Əgər $x = x'$ olarsa**,** imza həqiqi hesab edilir.

Qeyd edək ki, RSA imza sxeminə "Existential forgery" hücumu etmək mümkündür. Bu zaman imzalanın heç vaxt imzalamadığı, amma düzgün görünən bir məlumat-imza cütü yaradılır. Bunun üçün hücum edən $s \in Z_n$ imzasını seçir və $x \equiv s^e\, mod\, n$ məlumatını hesablayır. $s^e = (x^d)^e \equiv x\, mod\, n$ olduğundan, imza həqiqidir. Lakin hücum edən yalnız s imzasını seçə bilir, x məlumatını seçə bilmir. Yəni, "Oskar-ın hesabına 1000 $ köçürün" kimi məlumatlar generasiya edə bilmir. Bu hücumun qarşısını almaq üçün x məlumatının *padding sxeminə* uyğun olaraq formata salınması istifadə edilir.

11.4. ElGamal rəqəmsal imza sxemi

ElGamal rəqəmsal imza sxemi dövri sonlu dövri qruplarda diskret loqarifm məsələsinin çətinliyinə əsaslanır.

Tutaq ki, p – böyük sadə ədəd, g – GF(p) sonlu meydanının primitiv elementidir. Həm g-ni, həm də p-ni istifadəçilər qrupu üçün ortaq etmək olar. Sistemin hər bir istifadəçisi açıq açar və uyğun gizli açar yaradır. Bunun üçün istifadəçi A aşağıdakı əməliyyatları yerinə yetirir.

1. $1 \le x \le p - 1$ şərtini ödəyən təsadüfi x ədədi generasiya edir.
2. $y = g^x \bmod p$ hesablayır.
3. A-nın açıq açarı (p, g, y), gizli açarı x-dir.

İmzalama alqoritmi. A m məlumatını imzalayır.

(*a*) $1 \le k \le p-2$ və $(k, p-1) = 1$ şərtlərini ödəyən k təsadüfi ədədini generasiya edir, onu gizli saxlayır və imza yaradılan kimi məhv edir.

(*b*) $r = g^k \bmod p$ hesablayır.

(*c*) $s = k^{-1}(m - xr) \bmod (p-1)$ hesablayır.

(*d*) m məlumatı üçün A-nın imzası (r, s) cütüdür.

İmzanı yoxlama alqoritmi. m məlumatı üçün A-nın (r, s) imzasını yoxlamaq üçün B aşağıdakı əməliyyatları yerinə yetirir.

(*a*) A-nın (p, g, y) autentik açıq açarını əldə edir.

(*b*) $1 \le r \le p-1$ şərtinin ödənməsini yoxlayır, əgər şərt ödənmirsə, onda imza rədd edilir.

(*c*) $v_1 = g^m \bmod p$ hesablayır.

(*d*) $v_2 = y^r r^s \bmod p$ hesablayır.

(*e*) Əgər $v_1 = v_2$, olarsa, onda imza doğru hesab edilir.

Misal. ElGamal imzasının yaradılması

Tutaq ki, $p = 19$, $g = 10$ verilib və A $x = 16$ gizli açarını seçib. Onun açıq açarı $y = 10^{16} \bmod 19 = 4$. Tutaq ki, A $m = 14$ məlumatını imzalayır:

1. Təsadüfi $k = 7$ ədədini seçilir, ƏBOB(18, 7) =1 şərti ödənir.
2. $r = 10^7 \bmod 19 = 15$ hesablanır.
3. $k^{-1} \bmod (p-1) = 7^{-1} \bmod 18 = 13$ hesablanır.
4. $s = 13 \cdot (14 - 16 \times 15) \bmod 18 = 14$ hesablanır.
5. İmza (15, 14) cütüdür.

ElGamal imzasının yoxlanması

İstənilən istifadəçi bu imzanı aşağıdakı kimi yoxlaya bilər:

1. A-nın autentik açıq açarını əldə edir, $y = 4$.
2. $1 \le 15 \le 18$ bərabərsizliyinin ödənməsini yoxlayır.
3. $v_1 = 10^{14} \bmod 19 = 16$ hesablayır.
4. $v_2 = 4^{15} \cdot 15^{14} \bmod 19 = 11 \cdot 17 \bmod 19 = 16$ hesablayır.
5. 16 = 16 olduğundan, imza həqiqidir.

İmza yoxlamasının düzgünlüyünün əsaslandırılması. Əgər imzanı A yaradıbsa, onda $s \equiv k^{-1}(m - xr) \pmod{p-1}$. Müqayisə tərəflərini k-ya vursaq, $ks \equiv m - xr \pmod{p-1}$ alırıq. Buradan $m \equiv xr + ks \pmod{p-1}$. Bu $g^m \equiv g^{xr+ks} \equiv (g^x)^r r^s \equiv y^r r^s \pmod{p}$, yəni tələb edildiyi kimi, $v_1 = v_2$ bərabərliyini verir.

ElGamal tipli imza sxemlərində imzaların hesablanması və yoxlanması zamanı yerinə yetirilən ən mürəkkəb əməliyyat $g^z \bmod n$ diskret qüvvətə yüksəltmə əməlidir.

1990-cı ildə alman kriptoqrafı C.P.Schnorr-un təklif etdiyi rəqəmsal imza sxemi ElGamal sxemi ilə müqayisədə bəzi üstünlüklərə malik idi – diskret hesablamalar daha səmərəli və açarın uzunluğu da qısa idi. Şnorr sxeminin təhlükəsizliyi də diskret loqarifmləmə məsələsinin çətinliyinə əsaslanırdı. Qeyd edək ki, bir sıra ölkələrin 1990-cı illərdə qəbul edilmiş rəqəm imza standartının əsasını ElGamal imza sxemi təşkil edirdi (Schnorr imzası patentləşdirilmişdi). Məsələn, ABŞ federal hökumət standartı DSA (Digital Signature Algorithm) və Rusiya Federasiyasının QOST 34.10 standartı.

Mühazirə üzrə yoxlama sualları

1. Rəqəmsal imzanın əsas məqsədi nədir?
2. Ümumilikdə, rəqəmsal imza necə işləyir?
3. Rəqəmsal imzanın hansı növləri var?
4. Rəqəmsal imza sxemi hansı alqoritmlərdən ibarətdir?
5. Məlumatı bərpa etməklə rəqəmsal imza sxemləri necə işləyir?
6. Məlumatı əlavə etməklə rəqəmsal imza sxemləri necə işləyir?
7. Məlumatı əlavə etməklə rəqəmsal imza sxemlərinə nümunələr söyləyin.
8. "Heşlə-və-imzala" üsulu nədən ibarətdir?
9. Rəqəmsal imzada heş funksiyanın rolu nədir?
10. Rəqəmsal imzanın yoxlanması hansı açarla aparılır?
11. Rəqəmsal imzanın yaradılması hansı açarla aparılır?
12. RSA imza sxemində imza hansı düsturla yaradılır?
13. RSA imza sxemində imza hansı düsturla yoxlanır?
14. RSA imza sxeminə "Existential forgery" hücumu nədən ibarətdir?
15. "Existential forgery" hücumuna qarşı hansı mexanizm istifadə edilir?
16. ElGamal imza sxemi hansı məsələnin çətinliyinə əsaslanır?
17. ElGamal imza sxemi açıq açar və uyğun gizli açar necə yaradılır?
18. ElGamal imza sxemində imzanın yaradılması alqoritmi necə yerinə yetirilir?
19. ElGamal imza sxemində eyni mətn eyni açarla imzalansa, imza eyni olarmı?
20. ElGamal imza sxemində imzanın yoxlanması alqoritmi necə yerinə yetirilir?
21. ElGamal tipli imza sxemlərində ən mürəkkəb əməl hansıdır?
22. Schnorr rəqəmsal imza sxemi hansı üstünlüklərə malik idi?
23. Hansı ölkələrin standartları ElGamal imza sxeminə əsaslanırdı?

Mühazirə 12. Kriptoqrafik heş funksiyaları

12.1. Kriptoqrafik heş funksiyaları

Heş funksiyaları kriptoqrafik protokolların vacib elementidir. Onları açıq açarların, sertifikatların, imzalanmış məlumatların, rabitə seansında verilənlərin həqiqiliyinin yoxlanması, əməliyyat sistemlərində parolların yoxlanması, təsadüfi ədədlərin və seans açarlarının generasiyası və s. üçün istifadə edirlər.

Kriptoqrafik heş funksiyaların iki mühüm növü – **açarsız heş funksiyaları** və **açarlı heş funksiyaları** fərqləndirilir. Açarsız heş funksiyaları səhvlərin aşkarlanması kodları (Modification Detection Code və ya Manipulation Detection Code, MDC) adlandırırlar. Onlar əlavə vasitələrlə (məsələn, şifrləmə, mühafizəli kanaldan istifadə və ya rəqəmsal imza) verilənlərin tamlığına zəmanət verməyə imkan verir. Bu heş funksiyalarından həm istifadəçiləri bir-birinə etibar edən sistemlərdə, həm də istifadəçiləri bir-birinə etibar etməyən sistemlərdə istifadə etmək olar.

Açarlı heş funksiyalar simmetrik açarlı sistemlərdə tətbiq olunurlar. Açarlı heş funksiyaları **məlumatın autentifikasiyası kodları** (Message Authentication Code, MAC) adlandırırlar. Məlumatın autentifikasiyası kodları əlavə vasitələr cəlb etmədən, istifadəçiləri bir-birinə etibar edən sistemlərdə həm verilənlərin mənbəyinin autentikliyinə, həm də verilənlərin tamlığına zəmanət verməyə imkan verir.

12.2. Açarsız heş funksiyalarının tərifi və xassələri

Tərif. A əlifbasında uzunluğu sonlu olan sözlərin uzunluğu n-ə bərabər olan sözlərə h: $A^* \rightarrow A^n$ inikasına *heş funksiyası* deyilir.

Sadə sözlərlə desək, heş funksiyası sonlu uzunluqlu və dəyişən girişə görə sabit ölçülü çıxış verir. Heş funksiyasının girişinə "pro-obraz" çıxışına heş qiyməti, yaxud "digest" deyilir.

$Y = h(X)$ heş qiyməti X-in tamlığına nəzarət üçün, məsələn, rəqəmsal imza sistemlərində istifadə edilir. Fərz olunur ki, heş qiymətin hesablanması alqoritmi effektivdir və istifadəçilərə açıqdır.

Heş-funksiya elə qurulur ki, onun köməyi ilə alınmış heş qiyməti məlumatla "möhkəm" bağlı olur. Məlumatın hətta bir biti dəyişdikdə belə, heş qiymətinin bitlərinin ən azı yarısı dəyişir. Bu xassə "çığ effekti" (ing. avalanche) adlanır. MD5 ilə hesablanmış iki heş qiyməti bu xassəni izah edir.

MD5('1234567890')= `E807 F1FC F82D 132F 9BB0 18CA 6738 A19F`
MD5('1234567891')= `0F7E 44A9 22DF 352C 05C5 F73C B40B A115`

Kriptoqrafik heş funksiyalar üçün aşağıdakı kriptoanalitik məsələlərin həllinin hesablama cəhətdən çətin olması tələb olunur:

H1) Verilmiş Y çıxış qiymətinə görə elə X tapmaq tələb edilir ki, $Y = h(X)$ olsun. Bu xassə "pro-obraza müqavimət" (ing. preimage resistance) adlanır.

H2) Verilmiş X_1 qiyməti üçün elə $X_2 \neq X_1$ tapmaq lazımdır ki, $h(X_1)=h(X_2)$ olsun. Bu xassə "ikinci pro-obraza müqavimət" (ing. second preimage resistance) adlanır.

H3) Elə iki müxtəlif X_1 və X_2 məlumatlarını tapmaq lazımdır ki, $h(X_1)=h(X_2)$ olsun. Bu xassə "toqquşmaya müqavimət" (ing. collision resistance) adlanır.

Əgər H1, H2, H3-ün həlli üçün n-ə nəzərən polinomial çətinlikli alqoritm yoxdursa, onda h uyğun olaraq *biristiqamətli* (ing. one-way), *kolliziyadan azad* (ing. collision free) və *kolliziyadan ciddi azad* (ing. strict collision free) adlanır.

Uzunluğu n-ə bərabər heş qiymət üçün tələb olunan əməliyyatların tərtibi aşağıdakı kəmiyyətlərə mütənasibdir.

Biristiqamətli	2^n
Kolliziyadan azad	2^n
Kolliziyadan ciddi azad	$2^{n/2}$

12.3. Heş funksiyaların qurulmasının iterativ modeli

Heş funksiyaların bir çoxu iterativ modelin əsasında qurulub. Bu model çox zaman Merkle–Damgård konstruksiyası adlanır. 1979-cu ildə Ralph Merkle-in dissertasiya işində irəli sürülüb. Ralph Merkle və İvan Damgård bir-birindən asılı olmadan konstruksiyanın "sağlam" olduğunu isbat ediblər: uyğun padding sxemindən istifadə edilərsə və sıxılma funksiyası toqquşmaya dayanıqlı olarsa, heş funksiyası da toqquşmaya dayanıqlı olacaq.

İterativ modelə görə heş qiymətin hesablanması aşağıdakı qaydada aparılır. h heş funksiyasının girişinə M ilkin məlumatı – ixtiyari uzunluqlu verilənlər sətri daxil olur. Bu sətir uzunluğu r bit olan bloklara bölünür: $M = M_1M_2\ldots M_t$. Emaldan öncə zəruri olduqda axırıncı blok r bitə qədər tamamlanır. Bundan başqa, dözümün artırılması mülahizələrinə görə ilkin sətrə yeni r-mərtəbəli blok əlavə oluna bilər, bu blokda, məsələn, ilkin məlumatın uzunluğunun ikilik göstərişi verilə bilər.

Heş funksiyanın özəyi raund funksiyası adlanan $f(x, y)$ sıxma funksiyasıdır, ümumi halda bu funksiya girişdə uzunluqları uyğun olaraq r və s bit olan iki x və y sətirlərini alır və çıxışda s-mərtəbəli sətir formalaşdırır. Hər bir M_i bloku raund funksiyası üçün giriş arqumentidir. Digər arqument heşləmənin əvvəlki addımında alınmış s-bitlik aralıq qiymətdir (ilişmə dəyişəni). Tutaq ki, H_i ilə i-ci addımda (iterasiyada) heşləmənin qiyməti işarə edilir, onda $M = M_1M_2\ldots M_t$ girişi ilə iterativ heş funksiyanın ümumi modeli aşağıdakı münasibətlərlə təsvir oluna bilər:

$H_0 = IV$;

$H_i = f(M_i, H_{i-1})$, $1 \le$ və $\le t$;

$H(M) = g(H_t)$,

burada H_{i-1} − emalın $i-1$-ci və i-ci addımları arasında s-mərtəbəli ilişmə dəyişənləri; H_0 − ilkin yükləmə vektorudur.

f raund funksiyasının çıxışında s-mərtəbəli heş qiymət formalaşır. Zəruru olduqda onu $g(H_t)$ əlavə emal vasitəsi ilə lazım olan uzunluğa kimi kəsmək olar. Çox vaxt $g(H_t) = H_t$ götürülür.

MD4-ailəsi iterativ model əsasında qurulub, bu ailəyənə MD5, RIPEMD-128, RIPEMD-160, HAVAL, SHA-1, SHA-2 heş funksiyaları daxildir.

12.4. SHA-1 heş funksiyası

Təhlükəsiz heş funksiyası SHA-1 (Secure Hash Alqorithm) 1992-ci ildə ABŞ-da DSA (Digital Signature Algorithm) rəqəmsal imza alqoritmində istifadə etmək üçün standart kimi qəbul olunmuşdu. 2030-cu ildən sonra istifadəsi tövsiyə edilmir.

SHA-1 alqoritmiminin necə işləməsinə baxaq. Əvvəlcə ilkin məlumata elə əlavələr edilir ki, onun uzunluğu 512-yə bölünsün. Məlumatın uzunluğu 512-yə bölünsə belə, yenə də əlavələr edilir. Əlavəetmə belə aparılır: vahid əlavə edilir, sonra axıra 64 mövqe qalana qədər sıfır əlavə edilir, daha sonra ilkin məlumatın uzunluğunun 64-bitlik təsviri əlavə edilir.

32-bitlik beş dəyişən aşağıdakı 16-lıq sabitlərlə ilkin qiymətlər alır:

A = 67452301, B = EFCDAB89, C = 98BADCFE, D = 10325476,

E = C3D2E1F0.

Sonra alqoritmin **əsas addımı** başlayır. Bu addımda məlumatın bütün 512 bitlik blokları ardıcıl emal edilir. A, B, C, D, E dəyişənləri digər beş a, b, c, d, e dəyişəninə mənimsədilir:

$a=A$, $b=B$, $c=C$, $d=D$, $e=E$.

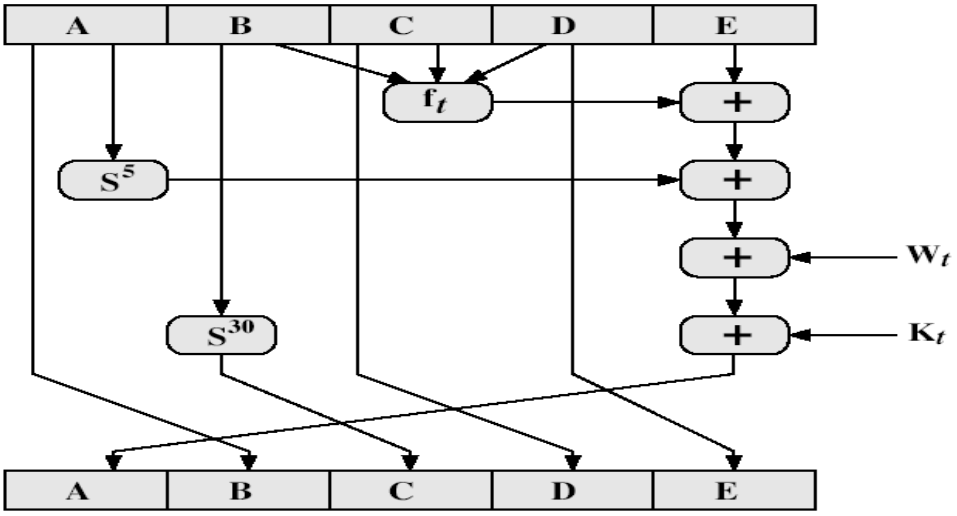
Aşağıdakı alqoritmin köməyi ilə 512-bitlik məlumat bloku on altı 32-bitlik (M_0...M_{15}) sözündən səksən 32-bitlik (W_0...W_{79}) sözünə genişləndirilir:

$W_t = M_t$, $t = 0...15$ olduqda,

$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1$, $t = 16...79$ olduqda,

burada t – əməliyyatın nömrəsidir ($t = 0...79$); W_t – genişlənmiş məlumatın t-ci altblokudur; $<<< S$ – sola S bit dövri sürüşmədir.

Əsas addımda hər birində 20 əməliyyat olmaqla 4 raund var. Hər əməliyyat beş a, b, c, d, e dəyişənlərinin üçündən asılı olan qeyri-xətti funksiyanı realizə edir, sonra sürüşmə və toplamanı yerinə yetirir. Bir əməliyyatın yerinə yetirilməsi sxemi şəkil 12.1.-də göstərilib.



Şəkil 12.1. SHA-1-in bir əməliyyatı

SHA-1 alqoritmində aşağıdakı qeyri-xətti funksiyalardan istifadə edilir:

$f_t(b, c, d) = (b \wedge c) \vee ((\neg b) \wedge d)$, $t = 0...19$,

$f_t(b, c, d) = b \oplus c \oplus d$, $t = 20...39$,

$f_t(b, c, d) = (b \wedge c) \vee (b \wedge d) \vee (c \wedge d)$, $t = 40...59$,

$f_t(b, c, d) = b \oplus c \oplus d$, $t = 60...79$,

burada t - əməliyyatın nömrəsidir.

Alqoritmdə dörd sabit də istifadə edilir:

K_t = 0x5A827999, $t = 0...19$ olduqda,

K_t = 0x6ED9EBA1, $t = 20...39$ olduqda,

K_t = 0x8F1BBCDC, $t = 40...59$ olduqda,

K_t = 0xCA62C1D6, $t = 60...79$ olduqda.

Daxil edilən işarələri nəzərə almaqla səksən əməliyyatdan ibarət əsas addımı belə təsvir etmək olar:

for t = 0 **do** 79

$temp = (a <<< 5) + f_t(b, c, d) + e + W_t + K_t$

$e = d$

$d = c$

$c = (b <<< 30)$

$b = a$

$a = temp$

Əsas addım bitdikdən sonra a, b, c, d və e qiymətləri uyğun olaraq A, B, C, D və E ilə 2^{32} modulu üzrə toplanır və alqoritm növbəti 512-bitlik blokun emalına keçir.

Heş funksiyanın yekun çıxışı A, B, C, D və E qiymətlərinin konkatenasiyası ilə alınır.

SHA-1 alqoritmi 160-bitlik heş qiymət verdiyindən hesab olunur ki, o tam seçim hücumlarına və "ad günü" hücumlarına qarşı 128-bitlik heş qiymət verən digər heş funksiya alqoritmlərindən daha dözümlüdür.

12.5. SHA-2 heş funksiyaları ailəsi – ümumi məlumat

AES standartında üç əsas təhlükəsizlik səviyyəsi nəzərdə tutulur: 2^{128}, 2^{192} və 2^{256} (128, 192, və 256 bit uzunluğunda açarlar). Məlumdur ki, heş funksiyaların kolliziyalar axtarışına dözümü təqribən $2^{n/2}$-yə bərabərdir, burada n – funksiyanın çıxış qiymətinin uzunluğudur. Deməli, göstərilən təhlükəsizlik səviyyələrini təmin etmək üçün heş-qiymətin minimal uzunluğu 256, 384 və 512 bit olmalıdır. SHA-1 alqoritminə həmin dövrə qədər heç bir analitik hücum tapılmadığına görə, onu 256, 384 və 512 bit uzunluğunda heş-qiymətin alınması imkanına qədər inkişaf etdirmək qərara alındı. SHA-2 alqoritmi uyğun olaraq üç alqoritmə: SHA-256, SHA-384 və SHA-512 alqoritmlərinə bölünür.

SHA-256 alqoritminin qısa təsvirinə baxaq. Onu iki hissəyə bölmək olar: sıxma funksiyasının təsviri və məlumatın emalı alqoritminin təsviri. Sıxma funksiyası mahiyyətcə 256 bitlik bloklarında heş funksiyanın aralıq nəticələri olan və növbəti mətn bloku açar kimi işlədilən blok şifri alqoritmidir.

12.6. SHA-3 standartı – ümumi məlumat

2004-2006-cı illərdə bir sıra heş-funksiyaların sındırılmasından sonra ABŞ Milli Standartlar və Texnologiyalar İnstitutu (National Institute of Standards and Technology, NIST) SHA-1 и SHA-2 alqoritmlərini əvəzləyəcək yeni kriptoqrafik heş-funksiya standartının işlənilməsini qərara aldı.

Kriptoqrafik heş-funksiya standartının seçilməsi üzrə müsabiqə 2007-ci ildə başlamışdı, müsabiqəyə 64 alqoritm təqdim olunmuşdu. Müsabiqə üç turdan ibarət idi və 2011-ci ildə üçüncü tura (finala) 5 alqoritm çıxmışdı: BLAKE; Grøstl; JH; Keccak; Skein.

2 oktyabr 2012-ci ildə müsabiqənin qalibi Keccak alqoritmi ("catch-ack" kimi tələffüz edilir) elan olundu, onun müəllifləri Guido Bertoni, Joan Daemen və Gilles Van Assche (hər üçü STMicroelectronics-dən) və Michaël Peeters (NXP Semiconductors) idi. Qeyd edək ki, Joan Daemen AES (Advanced Encryption Standard) standartında istifadə edilən Rijndael alqoritminin müəlliflərindən biridir.

Keccak-ın aparat realizələrində sürəti SHA-2 və digər finalçılardan xeyli yüksəkdir, lakin proqram realizəsi sürətinə görə Skein və BLAKE alqoritmlərindən iki dəfə geri qalır. Bununla yanaşı, onun əsas üstünlüyü "zərif dizaynı və müxtəlif qurğularda işləyə bilmək imkanıdır". Keccak-ın əlavə üstünlüyü onda SHA-2-dəki boşluqların olmamasıdır. Keccak alqoritmi Merkl-Damqard strukturuna əsaslanan

SHA-1, SHA-2 və MD5-dən əhəmiyyətli dərəcədə fərqlənir. Alqoritmlərin fərqli prinsiplər əsasında yaradılmasına görə SHA-2-yə qarşı olan hücumların Keccak üçün işləməyəcəyi söylənilir.

Keccak alqoritminin əsasında *Sponge* (süngər) adlanan konstruksiya durur. Konstruksiya iki mərhələdən ibarətdir:

- **Hopdurma** (ing. Absorbing). Heş-kodu hesablanan məlumat çoxraundlu yerdəyişmələrə məruz qalır.
- **Sıxma** (ing. Squeezing). Yerdəyişmələrdən alınan nəticə heş-kodun tələb edilən uzunluğu alınana qədər sıxılır.

Sponge konstruksiyasının unikal prinsipləri Keccak alqoritmini çıxışının uzunluğu ixtiyari olan heş-funksiya, axın şifri, paroldan kriptoqrafik açar generasiya edən funksiya, məlumatın autentifikasiyası kodu, psevdo-təsadüfi ədədlər generatoru kimi istifadə etməyə imkan verir.

12.7. HMAC (Hash-based Message Authentication Code)

HMAC sxemini 1996-da Mihir Bellare, Ran Canetti və Hugo Krawczyk təklif edib. HMAC *isbatlanan təhlükəsiz*dir (ing. provable secure): MAC-ı yalnız heş funksiyada kolliziya tapıldıqda sındırmaq olar.

HMAC hesablanması üçün heş funksiya istifadə edilir, heş funksiyanın girişinə verilən məlumata müəyyən qaydada məxfi açar əlavə edilir. Aşağıdakı işarələri qəbul edək:

H – istifadə edilən heş funksiya;

b – istifadə edilən heş funksiyada blokun uzunluğu;

n – heş qiymətin uzunluğu;

k – məxfi açardır. b-bitlik k^+ açarını almaq üçün bu açara soldan sıfırlar əlavə edilir;

İki köməkçi qiymət daxil edilir:

- *ipad* – '0011 0110' qiyməti $b/8$ dəfə təkrarlanır (onaltılıq '36' qiyməti).
- *opad* – '0101 1100' qiyməti $b/8$ dəfə təkrarlanır (onaltılıq '5C' qiyməti).

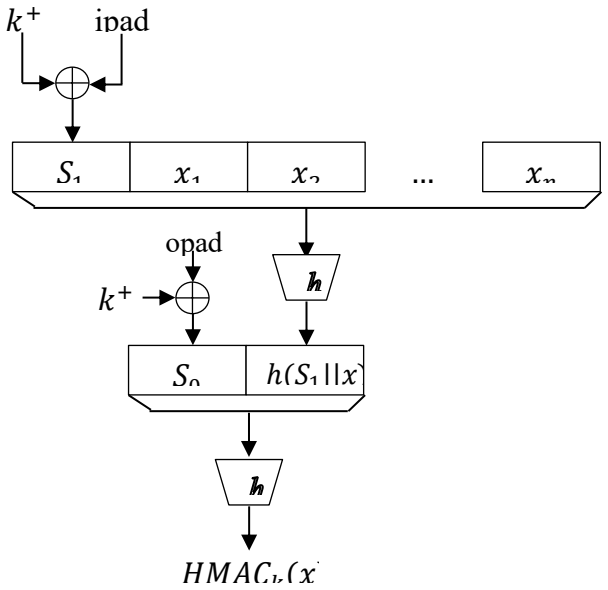
HMAC sxemi daxili və xarici heşdən ibarətdir:

$$HMAC_k(x) = \underbrace{h[(k^+ \oplus opad)||}_{xarici} \underbrace{h[(k^+ \oplus ipad) \,||\, x]]}_{daxili}$$

HMAC sxemi şəkil 12.2-də göstərilib və hesablamalar aşağıdakı addımlardan ibarətdir.

1. k-nın qiymətinə soldan sıfırlar əlavə olunur ki, b-bitlik k^+ sətri alınsın (məsələn, əgər k-nın uzunluğu 160 bit və b = 512-dirsə, onda k-ya 44 dəfə 0x00 baytı əlavə olunur).

2. k^+ qiyməti XOR əməli ilə $ipad$ ilə toplanır, nəticədə b-bitlik S_1 bloku alınır.
3. S_1-ə x qoşulur.
4. 3-cü addımda alınmış sətrə H funksiyası tətbiq olunur.
5. k^+ qiyməti XOR əməli ilə $opad$ ilə toplanır, nəticədə b-bitlik S_0 bloku alınır.
6. 4-cü addımda alınmış heş-qiymət S_0-a qoşulur.
7. 6-cı addımda alınmış sətrə H funksiyası tətbiq olunur, nəticə çıxışa verilir.



Şəkil 12.2. HMAC sxemi

MDx-MAC sxemləri. MDx-MAC konstruksiyaları HMAC-a alternativdir, onlar MD5, SHA, RIPEMD və ya oxşar heş funksiyaya əsaslanır. Burada özülə qoyulan heş funksiya kiçik dəyişikliklərlə MAC-a çevrilir, məxfi açar başlanğıcda, sonda və heş funksiyanın hər bir iterasiyasında cəlb edilir. Bu heş funksiyanın istifadə etdiyi başlanğıc qiymətlərin və additiv sabitlərin, çıxış çevirmələrinin açardan asılı dəyişiklikləri ilə əldə edilir. Bu daha ümumi yanaşmadır, çünki bu MAC-lar adətən blok şifrlərinə əsaslanan MAC-lardan sürətlidir. Lakin açar materialının heş funksiyanın girişinə verilən məlumata sadəcə əlavə edilməsi təhlükəsiz MAC-a gətirib çıxara bilməz. MDx-MAC-ın təhlükəsizliyi özülə qoyulan sıxma funksiyasının psevdotəsadüfi olması əsasında isbat oluna bilər.

12.8. Heş-funksiyalara mümkün hücumlar

Heş-funksiyalara yönəlmiş bütün hücumları iki qrupa bölmək olar:

– alqoritmdən asılı olmayan hücumlar;
– çevirmələr alqoritminin zəifliklərinə əsaslanan (analitik) hücumlar;

Alqoritmdən asılı olmayan hücumlara "kobud güc" hücümu, "ad günü" metodu ilə hücum, açarların tam saf-çürük edilməsi aid edilir. Belə hücumlara qarşı bütün alqoritmlər zəifliyə malikdir, onlardan yeganə xilas yolu – biristiqamətli və ya kolliziyasız heş-funksiyalarda heş-qiymətin uzunluğunu, MAC-funksiyalarında isə məxfi açarın uzunluğunu artırmaqdır.

Analitik hücumlara "ortada görüş", blokun korreksiyası ilə hücum, qeyd olunmuş nöqtə ilə hücum, baza şifrləməsi alqoritminə hücum, differensial analiz hücumu aid edilir.

Saxta məlumat yaratmaq məqsədi ilə istənilən heş-funksiyaya tətbiq oluna bilən ən sadə hücum aşağıdakılardan ibarətdir. Bədniyyətli müəyyən (r_1) sayda məlumatlar generasiya edə bilər, onların heş-qiymətlərini hesablayıb alınan qiymətləri əvvəllər ötürülmüş (r_2) sayda məlumatların müəyyən çoxluğundan məlum heş-qiymətləri ilə tutuşdura bilər. Heç olmasa bir üst-üstə düşmə alınsa, hücum uğurlu olacaq. Uğur ehtimalı R-i "ad günü" paradoksunun əsasında qiymətləndirmək olar. Məlumdur ki, bu ehtimal

$$R = 1 - e^{\frac{r_1 r_2}{2}}$$

düsturu ilə hesablanır, burada n – heş-qiymətin uzunluğu, e natural loqarifmlərin əsasıdır. Ehtimalın qiyməti $r_1=r_2=2^{n/2}$ olduqda ən böyükdür. Bu halda ehtimalın qiyməti təqribən 0,63-ə bərabərdir.

"Ad günü" paradoksu aşağıdakı sualın cavabından ibarətdir. Təsadüfi seçilmiş qrupda neçə nəfər olmalıdır ki, bu qrupda ad günü eyni olan iki nəfərin olması ehtimalı 0,5-ə bərabər olsun. Sualın cavabına görə qrupda cəmisi 23 nəfər olmalıdır. "Ad günü" paradoksuna görə, N elementi olan çoxluqdan həcmi $\sqrt{N}$ ilə müqayisə olunan seçmə varsa, seçmədə iki eyni elementin olması ehtimalı $^1/_2$ ilə müqayisə olunandır. Bu paradoks göstərir ki, açıq mətnin təsadüfi seçilməsi halında heş-qiymətlərin təkrarlanmasını almaq üçün orta hesabla $\sqrt{N}$ sayda açıq mətn götürmək kifayətdir, burada N–nəzəri cəhətdən rast gəlinə bilən heş-qiymətlərin ümumi sayıdır. Beləliklə, uzunluğu 64-bit olan heş-funksiyanın ($N=2^{64}$) kolliziyasını tapmaq üçün $2^{32}\approx4*10^9$ sayda heş-qiyməti hesablamaq kifayətdir.

Heş-funksiyaların qurulmasının iterativ üsulu onun tərsinin tapılması və ya kolliziyanın qurulması zamanı "ortada görüş" metodundan istifadə etməyə imkan verir. "Ortada görüş" hücumu ad günü metodu ilə hücumun modifikasiyasıdır, əgər dövr (tsikl) funksiyası aralıq X qiyməti və ya məlumat bloku M_i üçün inversiya olunandırsa, dövri strukturlu heş-funksiya üçün istifadə olunur. Bu hücum çətinliyinə görə ad günü hücumu ilə yanaşı qoyula bilər. Bu təhlükədən müdafiə

üçün adətən məlumatın sonuna nəzarət cəmi və məlumatın uzunluğu bloku əlavə olunur.

Mühazirə üzrə yoxlama sualları

1. Heş funksiyaları niyə biristiqamətli adlanır?
2. Heş funksiyası ilə şifrləmə arasındakı əsas fərq nədir?
3. Heş funksiyalarının əsas tətbiq sahələri hansılardır?
4. Heş funksiyalarında “pro-obraza müqavimət” nə deməkdir?
5. Heş funksiyalarında “ikinci pro-obraza müqavimət” nəyi ifadə edir?
6. üçün “toqquşmaya müqavimət” nə deməkdir?
7. Niyə heş funksiyası biristiqamətli adlanır?
8. Çıxışı 160-bit olan heş funksiyası üçün toqquşma ehtimalı nəyə bərabərdir?
9. Merkle–Damgård konstruksiyasının mahiyyəti nədir?
10. MD4 ailəsinə hansı heş funksiyalar daxildir?
11. SHA-1-də padding necə edilir?
12. SHA-1-in sıxma funksiyası nədən ibarətdir?
13. SHA-1-də blokun uzunluğu neçə bitdir?
14. SHA-2 ailəsinə hansı heş funksiyaları daxildir?
15. SHA-2 ailəsi hansı əsas dəyişənlərə görə fərqlənirlər?
16. SHA-256-da toqquşma (“kolliziya”) tapmaq üçün təxminən neçə cəhd lazımdır?
17. SHA-256 ilə SHA-3 arasında əsas fərq nədir?
18. HMAC hansı məqsədlə istifadə olunur?
19. ipad sabiti nəyə bərabərdir?
20. ipad sabiti neçə dəfə təkrarlanır?
21. opad sabiti nəyə bərabərdir?
22. opad sabiti neçə dəfə təkrarlanır?
23. k-nın uzunluğu 256 bit, blokun uzunluğu 512 bitdir. HMAC-da k^+ almaq üçün 0x00 baytı neçə dəfə əlavə olunacaq?
24. HMAC-ın isbatlanan təhlükəsizliyi nə deməkdir?
25. Heş funksiyaları üçün "ad günü hücumu" necə işləyir?

Mühazirə 13. Kriptoqrafik protokollar

13.1. Kriptoqrafik protokol anlayışı

Kriptoqrafik protokol – müəyyən kriptoqrafik məsələnin həlli üçün iki və ya daha çox tərəfin yerinə yetirdiyi addımlar ardıcıllığıdır.

Protokolun alqoritmdən əsas fərqi ondan ibarətdir ki, alqoritmin icrası bir subyektin aktiv əməllərini nəzərdə tutur, protokol isə bir neçə subyektin (*protokolun tərəfləri*) qarşılıqlı təsiri gedişində həyata keçirilir. Kriptoqrafik protokolun hər bir əməli məzmunca ya protokolun fəaliyyət göstərən subyektlərinin yerinə yetirdiyi əməlləri, ya da onlar arasında məlumatların göndərişini təsvir edir.

Kriptoqrafik protokolların aşağıdakı xassələri var:

1. Hər bir iştirakçı protokolu və bütün zəruri addımları qabaqcadan bilməlidir.
2. Hər bir iştirakçı ona əməl etməyə razı olmalıdır.
3. Protokol dəqiq və birmənalı təyin olunmalıdır.
4. Protokol tam olmalıdır (istənilən situasiya üçün əməllər müəyyən olunmalıdır).

Kriptoqrafik protokollar nəzəriyyəsinin tədqiqat obyekti, bir qayda olaraq açıq rabitə kanalları ilə qarşılıqlı əlaqədə olan məsafədəki abonentlərdir. Məsafədəki abonentlər tərəfindən həll edilən bir neçə məsələ nümunəsinə baxaq.

1. Bir-birinə inanmayan iki abonent qarşılıqlı əlaqədədir. Onlar saziş imzalamaq istəyirlər. Bunu elə etmək lazımdır ki, aşağıdakı vəziyyətə yol verilməsin: abonentlərdən biri digərinin imzasını alıb, özü isə imzalamayıb.
   Bu məsələnin həlli protokolu *sazişin imzalanması protokolu* adlandırılır.
2. Bir-birinə inanmayan iki abonent qarşılıqlı əlaqədədir. Onlar qəpiyin köməyi ilə püşk atmaq istəyirlər. Bunu elə etmək lazımdır ki, qəpiyi atan abonent, püşkatmanın nəticəsini tapmağa çalışan abonentdən cavab aldıqdan sonra nəticəni dəyişdirə bilməsin.
   Bu məsələnin həlli protokolu *püşkatma protokolu* adlanır.
3. İki *A* və *B* abonenti qarşılıqlı əlaqədədir (tipik nümunə *A* bankın müştərisi, *B* bank). *A* abonenti *B* abonentinə isbat etmək istəyir ki, məhz *A*-dır, düşmən deyil.
   Bu məsələnin həlli protokolu *abonentin identifikasiyası* protokolu adlanır.
4. Bir mərkəzdən əmr almış uzaqdakı bir neçə abonent qarşılıqlı əlaqədədir. Mərkəz də daxil olmaqla abonentlərin bir hissəsi düşmən ola bilərlər. Abonent üçün uduşlu vahid hərəkət strategiyasını yaratmaq tələb olunur.
   Bu məsələ Bizans generalları haqqında məsələ, onun həlli protokolu isə *Bizans sazişi protokolu* adlanır.

Beləliklə, kriptoqrafik məsələnin şərtləri uyğun protokolun xüsusiyyətlərini müəyyən edir.

13.2. Kriptoqrafik protokolların növləri

Əgər qarşılıqlı əlaqədə olan tərəflər bir-birinə etibar edirlərsə və kriptoqrafik məsələni birgə həll etməyə hazırdırlarsa, bu halda *ikitərəfli protokollar* adlanan *vasitəçisiz protokollar* istifadə olunur.

Əgər tərəflər arasında münaqişə baş verə bilərsə və ya onlara üçüncü tərəfin dəstəyi lazımdırsa, *vasitəçili protokollardan* (ing. adcudicated) istifadə olunur və üçtərəfli protokollar adlandırılır. Vasitəçinin (marağı olmayan etibarlı tərəf) vəzifəsi – qurtarma da daxil olmaqla protokolun bütün mərhələlərinin ifa olunmasını təmin etməkdir.

- Bir-birinə şübhə ilə yanaşan iki tərəf həmin şübhə ilə də vasitəçiyə yanaşacaqlar.
- Kompüter şəbəkəsi vasitəçiyə dəstəyi təmin etməlidir.
- Bütün vasitəçili protokollara *gecikmə* xasdır.
- Vasitəçi bütün tranzaksiyalarda iştirak etməlidir, bu istənilən protokolun geniş miqyaslı reallaşdırılmasında zəif yerdir.
- Şəbəkədə hər kəs vasitəçiyə inanmalı olduğu üçün vasitəçi şəbəkənin sındırılmasına cəhd zamanı zəif yerdir.

Hakimli protokollar (ing. arbitrated) da var, burada hakim dedikdə xüsusi tip vasitəçi nəzərdə tutulur: o, məcburi olaraq hər protokolun ifasında iştirak etmir, yalnız protokolun yerinə yetirilməsinin düzgünlüyünü yoxlamaq üçün dəvət olunur.

Protokolların ən cəlbedici növü – *özükafi protokollardır* (ing. self-enforcing), onların konstruksiyası protokolun düzgün yerinə yetirilməsinə nəzarəti təmin edir. Belə protokollar heç də hər məsələ üçün mövcud deyil.

13.3. Autentifikasiya protokolları

Autentifikasiyanın əsas məqsədlərindən biri müəyyən resurslara girişin idarə edilməsini təmin etməkdir. Autentifikasiya adətən açarların paylanması protokollarının da ayrılmaz hissəsidir. Bütün autentifikasiya protokollarında iki iştirakçı olur:

1) P – *isbat edən* – autentifikasiyadan keçən iştirakçı;
2) V – *yoxlayan* – isbat edənin autentikliyini yoxlayan iştirakçı.

Protokolun məqsədi yoxlanılanın həqiqətən də P olduğunu müəyyən etməkdir. Yoxlayan baxımından protokolun mümkün nəticəsi ya isbat edən P-nın identikliyi barədə qərarın qəbulu, ya da belə qərar qəbul etmədən protokolun bitməsidir.

Parolla autentifikasiya. Parolların köməyi ilə autentifikasiya ən geniş yayılmış autentifikasiya protokoludur. Belə bir parolla autentifikasiya protokoluna baxaq.

Hər bir U istifadəçisi qeydiyyat zamanı x_U parolunu seçir və onu serverə göndərir. Server h heş funksiyasını tətbiq edir və $y_U = f(x_U)$ hesablayır, (U, y_U) cütünü yadda saxlayır. Sonra bu istifadəçi hər dəfə öz hesabına giriş etmək istədikdə, özünün x_U parolunu serverə göndərir, server $y_U = f(x_U)$ bərabərliyinin ödənməsini yoxlayır.

Təəssüf ki, bu autentifikasiya protokolu müasir təhlükəsizlik tələblərinə cavab vermir. Doğrudan da, istənilən gizli qulaq asan məxfi parolu (şifrlənməmiş ötürülən) bərpa edə bilər, deməli, bu sxem passiv hücumlara qarşı tamamilə zəifdir.

Birdəfəlik parollar. Birdəfəlik protokollar qorunmasız kanaldan parolları tutan düşməndən qorunmanı təmin edir. Birdəfəlik parollar autentifikasiya üçün yalnız bir dəfə istifadə oluna bilərlər. Birdəfəlik parollarla autentifikasiyanı reallaşdıran Lamport protokolu aşağıdakı kimi işləyir.

İlkin yükləmə (hazırlıq) mərhələsi:

1. Hər bir istifadəçi U məxfi $x_U = x_0$ qiymətini seçir və yadda saxlayır.

2. $x_1 = f(x_0)$, $x_2 = f(x_1)$, …, $x_{1000} = f(x_{999})$ qiymətlərini hesablayır, burada f – biristiqamətli funksiyadır.

3. Bundan sonra $y_U = x_{1000}$ qiymətini serverə göndərir, server (U, y_U) cütünü saxlayır.

Autentifikasiya mərhələsi:

24. Birinci autentifikasiya zamanı isbat edən öz autentifikasiya verilənləri ilə birlikdə $x = x_{999}$ qiymətini göndərir. Yoxlayan $f(x) = y_U$ olmasını yoxlayır və $y_U = x$ saxlayaraq y_U qiymətini təzələyir.
25. Sonrakı autentifikasiyalar əvvəlki addıma analoji olaraq yerinə yetirilir, isbat edən $x_{998}, x_{997}, \ldots$ və s. göndərir. f-in biristiqamətli olması fərz olunduğundan, yalnız isbat edən bu qiymətlər ardıcıllığını hesablaya bilər.

Bu autentifikasiya sxemində iki problem var. Protokolun bir neçə dəfə yerinə yetirilməsindən sonra $x_{1000}, x_{999}, \ldots$ ardıcıl qiymətləri açılmış olurlar, onda bu protokolu təhlükəsiz təkrar etmək mümkün deyil. Xüsusi halda, diskin imtinası zamanı əgər axırıncı göndərilən qiymət (deyək ki, x_{765}) yaddaşda saxlanmayıbsa, lakin x_{900}-a qədər qiymətlər saxlanıbsa, onda x_{899} və x_{765} arasındakı qiymətləri yazan istənilən şəxs server əvvəlki autentifikasiya raundlarını təkrar edərsə, isbat edənin əvəzinə özünü autentifikasiyadan keçirə bilər.

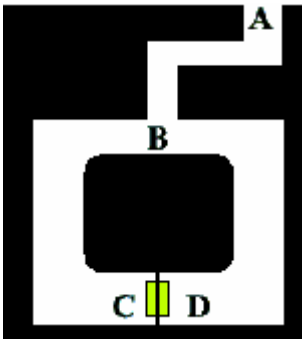
Digər problem onunla bağlıdır ki, bu autentifikasiya protokolu yalnız bir isbat edənlə istifadə oluna bilər (məsələn, bu elektron tranzaksiyalar üçün problem olur).

13.4. Sıfır biliklə isbat protokolu

Sabit parol əsasında protokolların nöqsanı ondan ibarətdir ki, isbat edən P öz parolunu yoxlayan V-yə verir. "Sorğu-cavab" tipli protokollar bu nöqsanı aradan

qaldırır. Onların icrası zamanı P V-nin zamana görə dəyişən sorğularına cavab verir, V-yə onun adından çıxış etmək üçün istifadə edə biləcəyi informasiya vermir. Bununla belə, P öz sirri barəsində qismən müəyyən informasiya verə bilər. Bu problemi sıfır biliklə isbat (ing. *Zero-knowledge proof*) protokolları həll edirlər. Bu protokollarda P müəyyən gizli biliklərə (məsələn, parola) malik olduğunu biliyi açıqlamadan V-yə isbat edir. Sıfır biliklə isbat protokolu interaktiv ola bilər, yəni isbat prosesi tərəflər arasında qarşılıqlı sorğu-cavab formatında baş verir.

Sıfır biliklə isbat protokollarının izahı üçün artıq klassik hesab edilən misala baxaq. 1989-cu ildə Crypto'89 konfransında fransız kriptoqrafları Jan-Jak Kuiskuoter (Jean-Jacques Quisquater) və Lui Qillou (Louis Guillou) "Sıfır biliklə isbatını uşaqlara necə izah etməli" adlı məruzə ilə çıxış etmişdilər. Məruzədə onlar "Ələddinin mağarası" kimi məşhur model təklif edirlər.



Fərz edək ki, P deyir ki, o, Ələddinin mağarasında **C-D** qapısını açan sehirli sözü bilir və bunu ona inanmayan B-yə isbat etmək istəyir. Bu zaman P istəmir ki, V bu sözü öyrənərək mağaranın sirrini ələ keçirsin.

P sirri açmadan V-ni necə inandıra bilər ki, o sirri bilir? Bunun üçün P və V aşağıdakı protokola uyğun olaraq addımları yerinə yetirməlidirlər:

1. V **A** nöqtəsində qalır.
2. P mağaraya girir və **B** nöqtəsində sola və ya sağa dönür (V onu görmədiyindən, onun məhz hansı tərəfə döndüyünü bilmir.
3. V **B** nöqtəsinə keçir.
4. V xahiş variantını təsadüfi seçərək mağaraya qışqırıb P-dən **C** tərəfdən və ya **D** tərəfdən qayıtmağı xahiş edir.
5. P onun xahişini yerinə yetirir (əgər qapıdan digər tərəfə keçibsə, sehirli sözdən istifadə edərək).

Sıfır biliklə isbat texnologiyası real həyatda anonim ödəniş sistemləri (Zcash kriptovalyutası), elektron səsvermə, identifikasiya sistemlərində istifadə olunur.

13.5. Fiat-Şamir protokolu

Sıfır biliklə ən sadə autentifikasiya protokolu Fiat-Şamir sxemidir. Belə protokollar intellektual kartlarda istifadə edilir. Əgər iddiaçı məxfi açarı bilməsini isbat edirsə, kart sahibinin şəxsiyyəti həqiqi qəbul edilir.

Fiat-Şamir protokolunun təhlükəsizliyi vuruqlara ayrılışı məlum olmayan kifayət qədər böyük mürəkkəb n ədədinin moduluna görə kvadrat kök alınmasının çətinliyinə əsaslanır.

Fiat-Şamir protokolunun reallaşdırılması üçün etibarlı Mərkəz tələb olunur, Mərkəz $n = pq$ şəklində tam ədədi seçərək nəşr edir, burada p və q – sadə ədədlərdir, onlar gizli saxlanırlar. Fərz edək ki, P isbat edən, V – yoxlayandır. Fiat-Şamir protokolu iki mərhələdən ibarətdir.

Hazırlıq mərhələsi. P gizli və açıq açar formalaşdırır:

1. P $(1, n-1)$ intervalından n ilə qarşılıqlı sadə olan gizli s ədədini seçir və $v = s^2 (mod\ n)$ açıq açarını hesablayır.
2. Mərkəz v-ni P-nin açıq açarı kimi qeydiyyata alır, s P-nin gizli açarı olur.

İsbat mərhələsi.

Fiat-Şamir autentifikasiya protokolunda aşağıdakı addımlar t dəfə təkrar edilir.

1. P $(1, n-1)$ intervalından təsadüfi r ədədini seçir və V-yə $x = r^2 (mod\ n)$ qiymətini göndərir.
2. V təsadüfi $e \in \{0,1\}$ bitini seçir və P-yə göndərir.
3. P $y = r \cdot s^e (mod\ n)$ hesablayır və onu V-yə göndərir.
4. V $y^2 = x \cdot v^e (mod\ n)$ bərabərliyini yoxlayır. Əgər bərabərlik doğrudursa, protokolun növbəti raunduna keçir, əks halda isbat qəbul edilmir.

Əgər yoxlama bütün t raundda uğurla keçirsə, onda V isbatı qəbul edir. Aldatma ehtimalı $\left(\frac{1}{2}\right)^t$-yə bərabərdir.

e-nin çoxluqdan seçilməsi nəzərdə tutur ki, əgər P doğrudan da gizli açarı bilirsə, seçilmiş e-dən asılı olmayaraq, düzgün cavab verə bilər. Tutaq ki, P V-ni aldatmaq istəyir, təsadüfi r seçir və $x = r^2/v$-ni göndərir. Əgər $e = 0$ olsa, onda P V-yə $y = r$ qaytaracaq və bu doğru olacaq. $e = 1$ halında isə P düzgün cavab verə bilməyəcək, çünki s-i bilmir və n moduluna görə kvadrat kök almaq kifayət qədər çətindir.

P-nin s sirrini bilmədən V-yə bunun əksini isbat etməsi ehtimalı $p = 2^{-t}$ kimi qiymətləndirmək olar, burada t – raundların sayıdır. Yüksək əminlik əldə etmək üçün t-ni böyük seçmək lazımdır $(t = 20 - 40)$. Beləliklə, V P-nin bildiyini yalnız və yalnız bütün t raund uğurla keçdikdən sonra təsdiqləyir.

Bu protokolun düzgün yerinə yetirilməsi üçün P heç zaman x-in qiymətini təkrar istifadə etməməlidir. Əgər P belə edərsə və V digər raund zamanı 2-ci addımda fərqli e təsadüfi bitini göndərərsə, V-də P-nin hər iki cavabı olacaq. Bundan sonra V s-in qiymətini hesablaya və P-nin gizli açarını öyrənə bilər.

Mühazirə üzrə yoxlama sualları

1. Kriptoqrafik protokol nədir?
2. Protokolun alqoritmdən əsas fərqi nədir?
3. Kriptoqrafik protokolun hansı xassələri var?
4. Sazişin imzalanması protokolu hansı məsələni həll edir?
5. Abonentin identifikasiyası protokolu hansı məsələni həll edir?
6. Püşkatma protokolu hansı məsələnin həll üçündür?
7. Bizans sazişi protokolu hansı hansı məsələnin həlli üçün nəzərdə tutulub?
8. Vasitəçisiz protokollarda neçə tərəf iştirak edir?
9. Vasitəçili protokollarda neçə tərəf iştirak edir?
10. Vasitəçili protokollarda vasitəçinin vəzifəsi nədir?
11. Hakimli protokollarda hakimin vəzifəsi nədir?
12. Özükafi protokollar nəyi təmin edirlər?
13. Autentifikasiya protokollarında iştirakçılar necə adlanırlar?
14. Parolla autentifikasiya protokolu necə işləyir?
15. Parolla autentifikasiya protokolu passiv hücumlara qarşı dayanıqlıdırmı?
16. Lamport protokolu hansı parollarla autentifikasiyanı reallaşdırır?
17. Lamport protokolu ilkin yükləmə mərhələsində necə işləyir?
18. Lamport protokolunda autentifikasiya necə işləyir?
19. Lamport protokolunun hansı problemləri var?
20. "Sorğu-cavab" tipli protokollar hansı nöqsanı aradan qaldırır?
21. Sifır biliklə isbat protokolu necə işləyir?
22. Fiat-Şamir protokolu hansı məsələnin çətinliyinə əsaslanır?
23. Fiat-Şamir protokolunda P-nin gizli və açıq açar cütü necə yaradılır?
24. Fiat-Şamir protokolunda V hansı bərabərliyi yoxlayır?
25. Fiat-Şamir protokolunda sirri bilmədən isbat ehtimalı nəyə bərabərdir?

Mühazirə 14. Kriptovalyutalar və blokçeynlər

14.1. Kriptovalyutalar sahəsində bəzi anlayışlar

Kriptovalyuta – rəqəmsal pulların bir növüdür, emissiyası və uçotu, adətən, mərkəziyətsiz olur.

Konfidensial ödənişlər üçün rəqəmsal pullar 1990-cı ildən DigiCash sistemində istifadə edilməyə başlanmışdı. Ödəniş sistemi mərkəzləşmiş idi, bu şirkət 1998-ci ildə müflis olmuşdu.

«Kriptovalyuta» termini 2011-ci ildə Forbes jurnalında Bitkoin haqqında «Crypto currency» məqaləsi nəşr olunduqdan sonra yayılmağa başladı. Bitkoin-in yaradıcısının özü «Electronic cash» terminini işlətmişdi. Bitcon-inn əsas məqsədi mərkəziyətsiz maliyyə sisteminin yaradılması idi.

Bitcoi-in əsasında duran texnologiyanı izah edən texniki sənəd – White paper ("Bitcoin: A peer-to-peer electronic cash system" adlı) – 31 oktyabr 2008-ci ildə Satoshi Nakomoto tərəfindən paylaşılmışdı. Onun şəxs və ya şəxslər qrupu olması hələ də məlum deyil. "Satoshi" adı yapon dilindən «müdrik», Nakamoto soyadı – «mürəkkəb (qapalı) sistemin daxilində yerləşən» kimi tərcümə edilir.

Sonrakı bir neçə ildə Bitcoin sürətlə məşhurlaşdı. Salvador 2021-ci ilin iyununda Bitcoin-i rəsmi ödəniş vasitəsi kimi tanıyan ilk ölkə olmuşdu.

Kritovalyutalarda təhlükəsizliyin təmin olunmasının əsasında blokçeyn (reyestr) – tranzaksiyaların zəncirlənmiş blokları dayanır. Blokçeyn – tranzaksiyaların paylanmış, açıq girişli reyestridir, şəbəkənin bütün qovşaqları tərəfindən birgə istifadə edilir.

Kriptovalyuta sistemi paylanmış kompüter şəbəkəsində fəaliyyət göstərir, tranzaksiyalar birranqlı (peer-to-peer, P2P) şəbəkəsində həyata keçirilir. Şəbəkənin hər bir tam qoşağında blokçeynin eyni kopyası olur.

Bu zaman tranzaksiyalar barədə məlumatlar şifrlənmir və açıq şəkildə əlyetər olur. Kriptoqrafiya tranzaksiyalar barədə məlumatlara girişi məhdudlaşdırmaq üçün deyil, tranzaksiya bloklarının dəyişməzliyinə zəmanət vermək üçün istifadə edilir.

Altkoinlər. Bitcoin xaricindəki bütün digər kriptovalyutalar *altkoin* adlanır. Hazırda altkoinlərin sayı minlərlədir.

Kriptovalyutaların volabilliyi çox yüksəkdir, bu onları spekylyantlar üçün cəlbedici və geniş istifadə üçün yararsız edir. İdeal kriptovalyuta öz alıcılıq qabiliyyəti baxımından stabilliyini saxlamalı, yaxud kiçik inflyasiyaya məruz qalmalıdır. Belə ideal kriptovalyuta *stabilkoin* (ing. stablecoin) adlanır.

Stabilkoinlərin stabilliyinə təminat ilə nail olurlar. Onların əksəriyyətinə valyutalar və qiymətli metallar, kriptovalyutalar, neft, brilyant kimi aktivlərlə

təminat verilir. Real valyutalara bağlı stabil-konlərə misallar: Tether, TrueUSD, Basecoin və s.

Etherium. Bazar kapitallaşmasına görə Bitcoin-dən sonra ikinci yeri Etherium tutur. Etherium yalnız kriptovalyuta deyil, ağıllı müqavilələr üçün bir platformadır.

Ethereum-un əsas ideyası blokçeynin funksional imkanlarının genişləndirilməsi idi. Bu ideya "ağıllı müqavilə"lər vasitəsilə həyata keçirilib. Ağıllı müqavilə blokçeyndə saxlanan (istənilən) proqram kodudur. O, istənilən rəqəmsal aktivin transferini təmsil edə bilər. Onlar müəyyən şərtlər ödəndikdə icra olunur, buna görə "ağıllı" adlanırlar.

Ethereum-da yerinə yetirilən ağıllı müqavilələri tranzaksiyalar işə salır. İstifadəçi müqaviləyə tranzaksiya göndərdikdə, şəbəkədəki hər bir qovşaq müqavilə kodunu işə salır və çıxışı yazır. Bunun üçün ağıllı müqavilələri kompüterin başa düşə biləcəyi əmrlərə çevirən Ethereum Virtual Machine (EVM) istifadə olunur.

Konsensus protokolu. Tam qovşaqların tranzaksiyaların sırası barədə razılığa nail olmalarını təmin edən mexanizm və ya qaydalar çoxluğudur. Konsensus protokollarının müxtəlif blokçeyn tətbiqlərində istifadə edilən bir çox növü var. Bitcoin şəbəkəsində **PoW** (*Proof-of-Work*) konsensus alqoritmi işləyir. PoW alqoritmi formalaşmış bloku blokçeynə hansı qovşağın yazacağını müəyyən edir və bununla mayninq şəbəkəsində konsensusu təmin edir. Etherium şəbəkəsi də PoW ilə işləyirdi. Etherium 2.0-da **PoS** (*Proof of Stake*) konsensus mexanizminə keçid həyata keçirilir. PoS-da konsensus token sahiblərinin paylarına əsasən təmin edilir.

Bitkoin mayninqinin mahiyyəti. Bitkoin hasilatı (istehsalı) mayninq vasitəsilə həyata keçirilir. Mayninq – Bitkoin şəbəkəsində zəruri və vacib prosesdir, onun nəticəsində aşağıdakı iki əsas məsələ həll edilir:

1. Yeni tranzaksiyalar blokunun blokçeynə yazılması.
2. Yeni bitkoinlərin emissiyası.

Beləliklə, mayninq zamanı yeni bitkoinlər mədəndən "çıxarılır", blokçeynə yazılacaq yeni bloku tapan mayner mükafat olaraq nəzərdə tutulmuş sayda bitkoin alır.

Bir blokun mayninqinə təxminən 10 dəqiqə gedir. Hazırda hər 10 dəqiqədən bir 3,125 yeni bitkoin buraxılır; onu bu bloku tapan, yəni son 10 dəqiqədə olan tranzaksiyaların blokunu yaradan alır. Bu emissiya sürəti Bitkoin protokolunda nəzərdə tutulub və kriptovalyutanın mədənçilik mükafatının yarıya endirilməsi prosesidir – mükafat hər 4 ildən sonra 2 dəfə azalır, bu "**halving**" adlanır.

2009-cu ilin yanvarında Bitcoinin ilk bloku *Genesis* yaradılan zaman mədənçilik mükafatı 50 bitkoin idi. Birinci "halving" 28 noyabr 2012-ci ildə baş verdi və mükafat 25 bitkoin oldu. 2016-cı ilin iyununa kimi hər 10 dəqiqədən sonra 25 bitkoin buraxılırdı, sonra bu sürət yarıya azaldıldı və hər 10 dəqiqədə 12,5 bitkoin

buraxıldı. Bu 2020-ci ilə kimi belə davam etdi. Sonra sürət yenidən yarıya bölündü və 2024-cü ilə kimi 6,25 bitkoin buraxılırdı. İlkin nəzərdə tutulduğuna görə, 2140-cı ilə kimi 21 milyon bitkoin buraxılacaq.

Kriptovalyutalarda hardfork və softfork. Adətən, kriptovalyutaların ilkin kodu açıqdır və istənilən şəxs onun zəif yerlərini tapa və təkmilləşdirə, yəni "fork" yarada bilər.

Blokçeynin qaydaları müxtəlif iki ayrı zəncirə bölünməsi prosesi forkinq adlanır. Forkinqin – soft və hard olmaqla iki növü var.

Softfork – kriptovalyutanın proqram kodunun «yumşaq» dəyişdirilməsi prosesidir. Softfork zamanı edilən dəyişikliklər ona qədər yaradılmış verilənlərlə uyarlığı saxlayır, lakin yeni verilənlərin yaranması prosesi dəyişir. Zəruriyyət yarandıqda softforku asanlıqla ləğv etmək olar.

Hardfork proqram koduna «sərt» dəyişiklik edilməsidir, bundan sonra köhnə proqram təminatı ilə uyarlıq itir. Faktiki olaraq, hardfork gedişində əvvəlkindən fərqli, tamalilə yeni prinsip ilə işləyən yeni kriptovalyuta yaradılır.

Kriptovalyuta pulqabılarının növləri. Kriptovalyutaları saxlamaq üçün kripto pulqabılardan istifadə edirlər. Kripto-pulqabı gizli açarları saxlayan proqramdır.

Kriptovalyuta pulqabıların üç əsas növü var: proqram, aparat və kağız pulqabıları. Onları həmçinin işləmə üsuluna görə *soyuq pulqabı* və *isti pulqabı* kimi də təsnif edirlər. İsti pulqabılar İnternetə qoşula və beləliklə, hücumlara daha çox məruz qala bilərlər. Soyuq pulqabılarda açarlar hər hansı İnternet bağlantısı olmadan generasiya edilir, buna görə kiberhücumlara daha dayanıqlıdırlar. *Proqram pulqabıları*nın veb, desktop və mobil pulqabılar kimi növləri var.

Tokenomika. Token – kriptovalyuta layihəsinə investisiyalar cəlb etmək məqsədilə buraxılan rəqəmsal aktivdir. Kriptovalyutanın bir növüdür, lakin öz xassələrinə görə səhmlərə və ya istiqrazlara yaxındır. Tokenlər ICO (Initial Coin Offering) gedişində buraxılır və sonra kriptovalyuta birjalarında satılır.

Tokenomika – tokenlərin yaradılması, paylanması, alqısı və satqısı ilə bağlı məsələləri əhatə edən iqtisadi modeldir.

DeFi anlayışı. DeFi (Decentralized Finance) – blokçeyn üzərində yaradılmış xidmətlər və tətbiqlər şəklində mərkəziyətsiz maliyyə alətləridir. DeFi – bank sektoruna müasir alternativdir, maliyyə sisteminin ənənəvi texnologiyalarını açıq kodlu protokollarla əvəzləyir. DeFi çox sayda insana mərkəziyətsiz kredit və yeni investisiya platformalarına giriş verir, həmçinin kriptovalyuta aktivlərindən passiv gəlir əldə etməyə imkan yaradır.

Bəzi populyar DeFi-tokenləri: Uniswap (UNI), SushiSwap (SUSHI), PancakeSwap (CAKE), Wrapped Bitcoin (WBTC), Dai (DAI), Compound (COMP), Avalanche (AVAX), Chainlink (LINK).

NFT (Non-Fungible Token – qarşılıqlı dəyişdirilə bilməyən token) son illər kriptovalyuta sənayesinin əsas trendlərindən biridir. Tənqidlərə baxmayaraq, NFT-lər populyarlıq qazanmağa və GameFi kimi əlaqəli sahələrdə tətbiqlər tapmağa davam edir.

Qeyd edək ki, FT (Fungible Token) – ERC-20 kimi standartlar əsasında yaradılmış dəyişdirilə bilən tokenlərdir (məsələn, kriptovalyutalar).

NFT tokenləri hər hansı unikal elementin rəqəmsal obrazını yaradan sertifikatdır (uçot vahididir). Onlar: rəsmlər, fotolar, videolar, musiqilər, giflər – bir sözlə, ən azı hər hansı unikallığa iddia edən istənilən kontent ola bilər. Onlar kolleksiyaçılar, oyunçular və sənət həvəskarları arasında yüksək qiymətləndirilir və hərraclar vasitəsilə alınıb satılır.

NFT standart kriptovalyuta tokenindən onunla fərqlənir ki, o, heç bir şəkildə bölünə, dəyişdirilə və ya saxtalaşdırıla bilməz. Beləliklə, hər bir NFT virtual obyekt üçün sertifikatdır, ona eksklüziv mülkiyyət hüquqlar verir və orijinallığını təmin edir.

14.2. Blokçeyn arxitekturasının səviyyələri

Blokçeynin müxtəlif arxitektura modelləri var. Ən sadə yanaşmada arxitektura adətən üç qatdan ibarətdir: 1) Konsensus qatı; 2) Verilənlər qatı; 3) Tətbiq qatı. Bu qatların funksiyaları belədir:

1) **Konsensus qatı:** konsensus mexanizminin işlədiyi və tranzaksiyaların yoxlanıldığı qatdır. Bu qat şəbəkənin təhlükəsizliyini təmin etmək üçün çox vacibdir.
2) **Verilənlər qatı:** tranzaksiya, blok və ağıllı müqavilə məlumatlarının blokçeyn şəbəkəsi üzərindən əlçatan olmasını və yoxlanılmasını təmin edən qatdır.
3) **Tətbiq qatı:** Ağıllı müqavilələrin və blokçeyn tətbiqlərinin icrasını təmin edən qatdır; İstifadəçilərə ağıllı müqavilələr və mərkəziyətsiz tətbiqlərlə qarşılıqlı əlaqə qura biləcəkləri interfeys təqdim edir.

Blokçeyn şəbəkəsinin infrastruktur qatlarını təsvir edən L0, L1, L2 və L3 modeli daha geniş yayılıb. Bu qatlar fərqli funksiyaları yerinə yetirir və şəbəkənin **miqyaslanma, təhlükəsizlik, emal sürəti** və **istifadəçi təcrübəsi** kimi əsas məsələlərini həll etmək üçün nəzərdə tutulub.

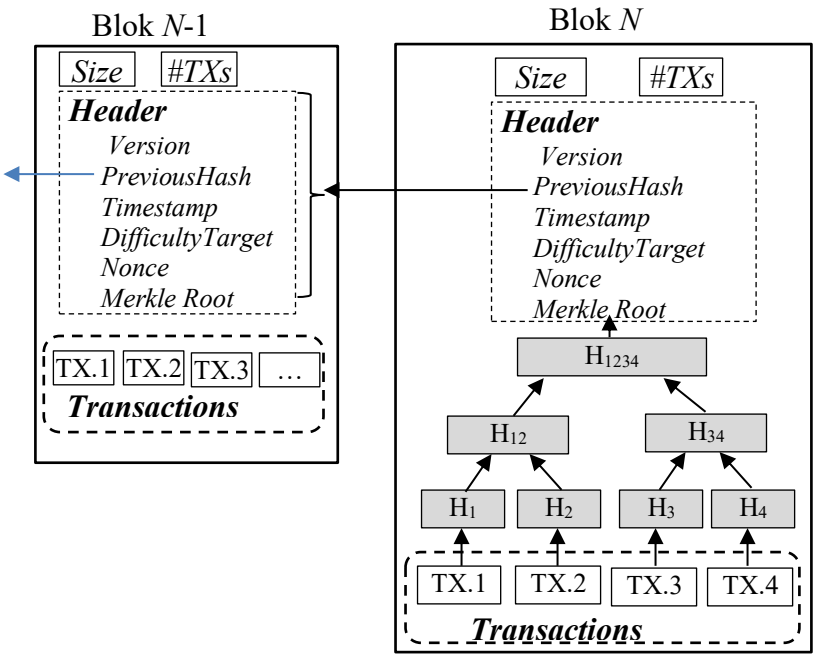
Layer 0 (L0) – əsas infrastruktur qatıdır. Müxtəlif blokçeyn şəbəkələrini və ya L1-ləri birləşdirən və əlaqələndirən şəbəkə infrastrukturu və protokol qatıdır. Vəzifəsi qovşaqlar arasında rabitəni və məlumat ötürülməsini təşkil etmək, fərqli L1-lərin əlaqəsini təmin etmək, multi-zəncir sistemlər yaratmaqdır.

Layer 1 (L1) – əsas blokçeyn qatıdır. Əsas blokçeyn infrastrukturudur – Konsensus, ağıllı müqavilələrin icrası, tranzaksiyaların emalı və təhlükəsizlik bu qatda təmin olunur. Bitcoin, Ethereum, Solana, BNB Chain bu qatda işləyirlər.
Layer 2 (L2) – miqyaslama həlləri qatıdır. Layer 1 şəbəkəsinin yüklənməsini azaltmaq üçün onun üzərində qurulmuş texnologiyalardır. Əsas vəzifəsi daha sürətli və ucuz tranzaksiyalar aparmaq, əsas şəbəkəyə əlavə yük yükləmədən funksionallığı artırmaqdır. Bu qatda olan texnologiyalara nümunə kimi Rollup-lar (Optimistik və ZK-Rollup-lar), Plasma, Validium göstərmək olar.
Layer 3 (L3) – tətbiq və inteqrasiya qatıdır. Layer 2 üzərində çalışan, istifadəçi yönümlü tətbiqləri və protokolları əhatə edən qatdır. DeFi, NFT, oyunlar, sosial tətbiqlər kimi spesifik xidmətləri təklif edə bilər. Bəzən interoperabilik və xidmət yönümlü middleware-lər də bura daxil edilir.

14.3. Bitcoin blokçeyninin strukturu

Texniki baxımdan, adından da göründüyü kimi, blokçeyn ("block" – blok + "chain" – zəncir) informasiya bloklarının zənciridir. Şəkil 14.1-də göstərildiyi kimi, hər bir Bitcoin bloku dörd komponentdən ibarətdir: blokun ölçüsü (*Size*), tranzaksiya sayğaçı (*#TXs*), blokun başlığı (*Header*) və tranzaksiyalar (*Tranzactions*). Blokun ölçüsü və tranzaksiya sayğacı, uyğun olaraq, blokdakı baytların və tranzaksiyaların saylarını göstərir. Tranzaksiyalar – bloka daxil olan tranzaksiyaların siyahısıdır. Blok başlığına altı sahə daxildir:

- *Version* – versiya nömrəsi (konsensus protokolunun yenilənməsini izləmək üçün istifadə edilir);
- *PreviousHash* – zəncirdə bilavasitə əvvəlki blokun heşi;
- *Timestamp* – blokun yaradılması tarixi;
- *Difficulty Target* – hədəf qiyməti, mayninq prosesində bu qiymətdən kiçik olan heş axtarılır;
- *Nonce* – mayninq alqoritmində istifadə edilən sayğac;
- *Merkle Root* – bloka daxil olan bütün tranzaksiyaların heşləri əsasında yaradılan binar ağacdır.



Şəkil 14.1. Bitcoin blokçeyninin strukturu

14.4. Bitkoin tranzaksiyaları

Öz mahiyyətinə görə, blokçeyn əsaslı kriptovalyutalar rəqəmsal imzaların zəncirindən ibarətdir və pul vəsaitlərinin sistemdə yolunu əks etdirir. Bu kontekstdə əsas anlayış tranzaksiyadır. Faktiki olaraq, tranzaksiya – blokun ünvanı olan minimal elementidir, blok isə öz növbəsində blokçeynin ünvanlı elementidir (ing. transaction, lat. transactio – müqavilə, razılaşma).

Tranzaksiyaların iki növü var:

- "Standart" tranzaksiyalar – sistemin istifadəçilərindən vəsaitlərin ötürülməsini kodlaşdırır,
- "Generasiya edən" tranzaksiyalar blokun generasiyası üçün mükafat olaraq mədənçiyə yeni hasil edilmiş bitkoinlər verilir.

Tranzaksiyaların strukturu. Şərti olaraq, tranzaksiyada iki sahə var – Input və Output. Əyanilik üçün şəkil 3-də göstərilmiş misala baxaq.

```
Input:
Previous tx: f5d8ee39a430901c91a5917b9f2dc19d6d1a0e9cea205b009ca73dd04470b9a6
Index: 0
scriptSig: 304502206e21798a42fae0e854281abd38bacd1aeed3ee3738d9e1446618c4571d10
90db022100e2ac980643b0b82c0e88ffdfec6b64e3e6ba35e7ba5fdd7d5d6cc8d25c6b241501

Output:
Value: 5000000000
scriptPubKey: OP_DUP OP_HASH160 404371705fa9bd789a2fcd52d2c580b65d35549d
OP_EQUALVERIFY OP_CHECKSIG
```

Şəkil 14.2. Bitkoin şəbəkəsində tranzaksiyaya misal

Baxılan tranzaksiya f5d8ee3… tranzaksiyasından bitkoinləri istifadə edir və onları 404371705fa9bd7… ünvanına göndərir.

Alan tərəf bu tranzaksiyadan olan bitkoinləri istifadə etmək istədikdə, sadəcə, öz tranzaksiyasının Input sahəsində bu tranzaksiyanı göstərir.

Input sahəsi – digər tranzaksiyanın konkret Output sahəsinə istinad edir (tranzaksiyada bir neçə müstəqil Output sahəsi ola bilər). Input sahəsində göstərilmiş tranzaksiyadan olan bütün bitkoinlər toplanır və bu cəm baxılan tranzaksiyada tam istifadə edilməlidir. **Previous tx** əvvəlki tranzaksiyaya istinaddır, **Index** sahəsi isə bu tranzaksiyanın məhz hansı Output-unun istifadə edildiyini göstərir.

scriptSig sahəsi iki hissədən – açıq açar və imzadan ibarətdir. Göstərilmiş açarın heşi əvvəlki tranzaksiyada göstərilmiş alan tərəfin heşinə uyğun olmalıdır. Açıq açar tranzaksiyada göstərilən imzanın yoxlanması üçündür. İmza verilmiş tranzaksiyanın bəzi sahələrinin əsasında generasiya edilir. Beləliklə, açıq açar imza ilə birlikdə təsdiq edir ki, tranzaksiya əvvəlki tranzaksiyada göstərilən həqiqi alan tərəfindən yaradılmışdır. Qeyd etmək lazımdır ki, imza heç zaman tranzaksiyaya bütövlükdə tətbiq edilmir və istifadəçilər tranzaksiyanın hansı hissələrinin imzalandığını göstərə bilərlər (əlbəttə, scriptSig sahəsindən başqa).

Output sahəsi – Output sahəsində bitkoinlərin göndərilməsi üçün təlimatlar olur. **Value** sahəsi bu outputa istinad edildikdə onda olacaq Satoshi miqdarını göstərir (1 BTC=100 000 000 Satoshi).

scriptPubKey bitkoinləri bu tranzaksiyadan digərinə keçirmək üçün "Doğru" qiyməti verməli olan skriptdir. Skripti yoxlamaq üçün scriptSig və scriptPubKey sahələri birləşdirilir (konkatenasiya) edilir, sonra skript yerinə yetirilir. Qeyd etmək lazımdır ki, skript komandalar stekidir, burada bir komandanın çıxış verilənləri digər komandanın girişinə daxil olur. Belə skript sistemi sayəsində göndərən bitkoinlərin verilməsi üçün müxtəlif tipli şərtlər yarada bilər. Məsələn, elə Output yaratmaq olar

ki, bu tranzaksiyadan bitkoinləri istənilən istifadəçi ala bilsin. Həmçinin elə skript yaratmaq olar ki, onun yerinə yetirilməsi üçün açar əvəzinə parol istifadə edilsin.

Tranzaksiyada heç də həmişə yalnız bir Input və Output olmur. Tranzaksiyada bir neçə Input (bir neçə tranzaksiyaya istinad edilir) və bir neçə Output (bitkoinlər bir neçə şəxsə göndərilir) ola bilər, həm də hər bir Output üçün özünün unikal skripti ola və digər çıxışlardan asılı olmadan yerinə yetirilə bilər. Həmçinin tranzaksiyada onun nə zaman icra oluna biləcəyini göstərən zaman məhdudiyyətləri də ola bilər.

Tranzaksiyaların emalı. Tranzaksiya şəbəkəyə daxil olduqda əvvəlcə bütün əlyetər Bitkoin qovşaqları tərəfindən yoxlanır və təsdiqlənir. Yoxlamadan uğurla keçdikdən sonra o, *mempul* adlanan yaddaş sahəsində növbəyə durur (ing. memory pool – yaddaş hovuzu) və mədənçinin onu növbəti bloka daxil etməsinə qədər burada gözləməyə məcburdur. Beləliklə, mempul – bütün gözləyən tranzaksiyaların saxlandığı yaddaş sahəsidir.

Mempul BIP 35-in hissəsidir və *yüngül pulqabılara* tranzaksiyalar etməyə imkan verir. Eyni zamanda , mempul Bitkoin şəbəkəsinin aktual problemlərindən biridir. Tranzaksiyalar ondan nə qədər sürətlə keçirsə, istifadəçilər şəbəkənin işindən bir o qədər razı qalırlar. Əgər yeni blokların yaradılması sürəti tranzaksiyaların mempula əlavə edilməsi sürətindən aşağıdırsa, onda böyük tıxac yarana bilər, onun nəticəsində tranzaksiyalar ölçüsündən və tranzaksiya haqqından asılı olaraq saatlarla (hətta sutkalarla) öz təsdiqlənmələrini gözləyə bilərlər.

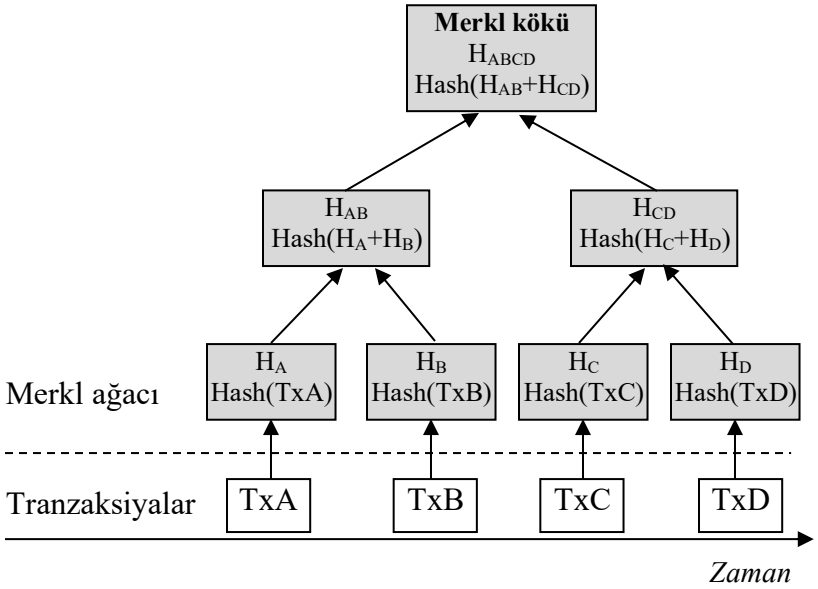
Məsələn, 2017-ci ilin mayında Bitkoin şəbəkəsi çox ciddi problemlə üzləşmişdi. Mempulda təsdiqlənməmiş tranzaksiyaların sayı 200 minə çatmışdı, emal edilməmiş verilənlərin həcmi isə 120 Mb-dan çox idi. Nəzərə alsaq ki, Bitkoin şəbəkəsində 1 blokun həcmi 1 Mb-dır və blokun yaradılması 10 dəqiqə çəkir, onda 120 blokdan ibarət növbə bir neçə sutkaya uzanmışdı, çünki təsdiqlənməmiş yeni tranzaksiyalar da növbəyə daim qoşulurlar.

14.5. Merkl ağacı

Bitkoin şəbəkəsində qovşaqlar bərabərhüquqlu və anonimdirlər, onlar bir-birinə etibar etməyə bilərlər. İnamın olmadığı vəziyyətdə verilənlərin verifikasiyası problemi meydana çıxır: blokda düzgün tranzaksiyaların yazılmasını necə yoxlamaq olar? Hər blokun yoxlanmasına böyük zaman və hesablama resursları sərf etmək lazım gələrdi. Merkl ağacları bu problemi həll etməyə və prosesi asanlaşdırmağa imkan verir.

Merkl ağaclarının konsepsiyası 1979-cu ildə Ralf Merkl (Ralph Charles Merkle) tərəfindən patentləşdirilmişdi.

Merkl ağacı böyük ölçüdə verilənləri toplamaq və təsdiq etmək üçün istifadə olunan verilənlər strukturudur. Merkl ağacına "Binar heş ağacı" da deyirlər. Merkl ağacı kriptoqrafik heş funksiyasının nəticələrindən ibarətdir. Bitkoin sistemində Merkl ağacı tranzaksiyaları blokda toplamaq üçün istifadə olunur. Tranzaksiyaların heş cütləri birləşərək ağacın düyünlərini əmələ gətirirlər. Bu proses "Merkl kökü"-nə qədər davam edir (Şəkil 14.3).



Şəkil 14.3. Merkl ağacında düyünlərin hesablanması

Fərz edək ki, TxA, TxB, TxC, TxD tranzaksiyaları var. Merkl ağacında kriptoqrafik heş funksiya kimi ikiqat SHA-256 (ing. double-SHA-256) tətbiq edilir. Bu tranzaksiyaların ikiqat heş qiymətlərini H_A, H_B, H_C, H_D kimi işarə edək. H_A, H_B, H_C və H_D Merkl ağacının *yarpaqlarında* yerləşir.

Merkl ağacının qurulması aşağıdakı qaydada aparılır:

1. Bloka yerləşdirilmiş tranzaksiyaların heşləri hesablanır: H_A, H_B, H_C, ...
$$H_A = SHA256(SHA256(TxA)),$$
$$H_B = SHA256(SHA256(TxB)), ...$$
2. Tranzaksiya heşlərinin cəmlərindən ("konkatenasiyasından") heşlər hesablanır.
$$H_{AB} = SHA256(SHA256(H_A + H_B)),$$
$$H_{CD} = SHA256(SHA256(H_C + H_D)), ...$$
3. Sonra heşlərin cəmlərindən heşlər yenidən hesablanır. Proses bir heş – *ağacın kökü* (ing. Merkle root) alınana kimi təkrarlanır. O, blokun tamlığının (yəni bütün tranzaksiyaların göstərilən ardıcıllıqda

yerləşdiyinin) kriptoqrafik isbatıdır. Kökün qiyməti blokun başlığında yerləşdirilir.

Merkl ağacı binar ağacdır, hər bir iterasiyada elementlərin sayı cüt olmalıdır. Əgər elementlərin sayı tək olarsa, sonuncu element təkrar istifadə olunur, yəni özü ilə birləşdirilir və heşlənir.

Göstərilən üsulu istənilən sayda tranzaksiyaya tətbiq etmək olar. Bir Bitkoin blokunda yüzlərlə, bəzi hallarda isə mindən artıq tranzaksiya ola bilər.

Merkl kökü ağacda ən böyük qiymət kimi görünsə də, sadəcə, 256 bitdir (32 baytdır).

Merkl ağacı verilmiş tranzaksiyaların MerkleRoot-a daxil olmasını digər hissələr olmadan yoxlamağa imkan verir. Məsələn, SPV (*Simplified Payment Verification*) Bitkoin protokolu Merkl ağaclarına əsaslanır və istifadəçilərə bütün blokçeyni yükləmədən tranzaksiyaları yoxlamağa imkan verir.

14.6. Kriptovalyutalarda multi-imza sxemləri

Məlumdur ki, Bitkoin şəbəkəsində tranzaksiyalar geriyə dönməzdir, yəni onları ləğv etmək və ya geri qaytarmaq mümkün deyil. Multi-imza əlavə imzanın hesabına pulqabıların təhlükəsizliyini gücləndirməyə xidmət edir. Əlavə imzanı üçüncü tərəf qoymalıdır, əks halda pulqabına daxil olmaq mümkün olmur.

Multi-imza (ing. *multisignature*) – elektron imzanın reallaşdırılması sxemlərindən biridir, onun həqiqiliyi üçün n üzvdən ibarət qrupdan t üzvün imzası tələb edilir ($t < n$, "n-dən t").

Multi-imza sxemi ilk dəfə 2012-ci ildə təqdim olunub, lakin texniki xüsusiyyətləri səbəbindən yalnız 2014-cü ildə geniş tətbiq olunmağa başlamışdı.

3-dən 2 variantı çox geniş yayılıb. Bu halda tranzaksiyanın yerinə yetirilməsi üçün hesab sahiblərindən ən azı ikisinin imzası tələb edilir. (*Çox zaman "imza" əvəzinə "açar" sözü işlədilir.*)

Multi-imzanın iş prinsipi alqı-satqı əməliyyatına nəzarətedici üçüncü şəxsin – hakimin daxil edilməsindən ibarətdir. Əməliyyatın birbaşa tərəfləri alqı-satqı haqqında razılaşdıqdan sonra onların və hakimin açıq açarları əsasında yeni ünvan generasiya edilir və onları heç kim dəyişdirə bilməz.

Qeyd etmək lazımdır ki, bir qayda olaraq, hakim üçün mükafat nəzərdə tutulur. Onu müqavilədə nəzərdə tutulmuş tərəflərdən istənilən biri ödəyə bilər.

Bitkoin multi-imzalarının əsas üstünlüklərinə aşağıdakıları aid etmək olar:

- Alqı-satqının təhlükəsizliyi yüksəlir.
- Müqavilənin şərtləri pozulduqda vəsaiti qaytarmaq imkanı var.
- Müxtəlif qruplarda (şirkətlər, maliyyə təşkilatları və s.) istifadə edilə bilər.

Multi-imzaların əsas nöqsanı ondan ibarətdir ki, tranzaksiya və əməliyyatlara üçüncü şəxs qoşulur. Onun yalnız nəzarətçi olmasına baxmayaraq, bir çoxları bu ideyanı bəyənmir, çünki bu, kriptovalyutaların təbliğ etdikləri demərkəzləşməni qismən pozur.

Bitkoində P2SH multi-imza mexanizmi var. Ethereum bu məqsədlə bir imza ilə əlavə ünvanlardan istifadə edir, onlara ağıllı müqavilələr nəzarət edir. Bu prosesi mürəkkəbləşdirsə də, proqramçılar üçün böyük çeviklik təmin edir.

Kriptovalyuta pulqabılarının bir çoxunda multi-imza funksiyası mövcuddur. Müxtəlif istehsalçılar onu fərqli şəkildə reallaşdırırlar. Bəziləri gizli açarları bilavasitə istifadəçidə saxlamağı təklif edir. Bəzi həllər gizli açarın surətini özündə saxlayır, onun pul göndərmək hüququ yoxdur, məqsəd açar itirildikdə onu bərpa edə bilməkdir.

14.7. Bitcoin şəbəkəsinə qarşı hücumlar

Kriptovalyuta heç bir fiziki mövcudluğu olmayan, tamamilə rəqəmsal valyutadır. Valyuta təhlükəsizliyi ilə bağlı məsələlər həmişə müzakirə mərkəzindədir. Virtual valyutada tranzaksiya və mayninq proseslərində təhlükəsizliyi artırmaq üçün səylər göstərilsə də, virtual valyutanın qarşısında hələ də bəzi təhlükələr mövcuddur. Eyni zamanda müştərilər üçün təklif olunan onlayn pulqabı xidmətlərinə qarşı da bir sıra təhlükələr var. Hətta mübadilə xidmətləri də hücumların hədəfi ola bilər.

Bitkoin dizayn edildikdə nəzərə alınmış əsas hücum "Double spending" idi (qeyd edək ki, ənənəvi pullarda bu hücum mümkün deyil).

"Double-spending" (İki dəfə xərcləmə) hücumu: Bitkoin tranzaksiyası şəbəkədə saniyələr içərisində yayılır, lakin təsdiqlənməsi zaman tələb edir. Sahib olduğu bitkoinləri rəqəmsal şəkildə imzalayaraq xərcləyən istifadəçinin başlatdığı tranzaksiya etibarlı tranzaksiya sayılır. Bitkoin ünvanına etibarlı bir ödəniş bir neçə saniyə ərzində şəbəkədəki bütün istifadəçilərə görünə bilər, lakin bunun təsdiq olunacağına zəmanət yoxdur. Tranzaksiyanın təsdiq olunmasının gecikməsi səbəbindən ikidəfə xərcləmə imkanı mövcuddur. Eyni bitkoini bir dəfədən çox xərcləmə prosesi *iki dəfə xərcləmə* adlanır. Tranzaksiya başlayıb və hələ təsdiq edilməyibsə, eyni bitkoinlər alıcı və ya satıcı tərəfindən yenidən istifadə edilə bilər. Mümkün iki dəfə xərcləmə hücumlarına *"Race"* hücumu, *Finney* hücumu, *kobud güc* hücumu, *51%* hücumu daxildir.

"Race" (*yarış*) *hücumu* – Yeni başlamış tranzaksiyadakı bitkoinin təsdiqlənməyi bir müddət vaxt tələb etdiyinə görə hücumçu bu bitkoini təkrar istifadə edə bilər. Eyni bitkoin bir neçə tranzaksiyada istifadə olunubsa hansı tranzaksiyanın təsdiqlənəcəyini tapmaq çətindir. Heç bir zərər görmək istəməyən satıcı

təsdiqlənməmiş ödənişi qəbul etməməli və ən azı 6 təsdiqlənməni gözləməlidir. Daha sadə dillə desək, müştərinin göndəriyi bitkoinləri təsdiqlənməmiş halda pulqabısına qəbul edən satıcı 6 təsdiqlənməni almadan müştəriyə xidmət edərsə, bu zaman zərərə uğrama ehtimalı var. Çünki bu proses ikidəfə xərcləmə ola bilər, sistem sonra bu tranzaksiyanı rədd edə bilər.

Kobud güc (*Brute-force*) *hücumu* – Bir qrup saxta mədənçi ikidəfə xərcləmə tranzaksiyalarının daxil olduğu blokları təsdiqləyir və təsdiq gözləyən alıcıya göndərir. Hücumun uğurlu olması ehtimalı alıcının neçə təsdiq gözləməsindən və saxta mədənçi qrupunun şəbəkənin neçə faizini təşkil etməsindən asılıdır. Məsələn, saxta mədənçi qrupu şəbəkənin 10 %-ni təşkil edirsə və alıcı 6 təsdiq gözləyirsə, o zaman hücumun uğurlu olma ehtimal 0.1 %-dir.

51% hücumu – Əgər bir qrup mədənçi şəbəkənin ümumi heş gücünün 51%-dən çoxunu ələ keçirsə, o zaman onlar öz bloklarını istədikləri kimi təsdiqləyə, digər tranzaksiyaları bloklaya və hətta ***iki dəfə xərcləmə*** həyata keçirə bilərlər**.** Bitcoin şəbəkəsinin geniş paylanmış olması və böyük mədənçilik hovuzlarının tarazlığı qoruması bu hücumun qarşısını qismən alır.

Timejacking hücumu – Bitkoin qovşaqları blokların zaman möhürü (ing. **timestamp)** qiymələrini əsas götürərək sinxronlaşdırırlar. Hücumçu saxta vaxt qiymətləri göndərərək qovşaqları **köhnə blok zəncirlərinə qoşulmağa** məcbur edə bilər. Çözüm olaraq, yeni Bitcoin versiyalarında timestamp yoxlamaları gücləndirilib.

"Timejacking" hücumu barədə əlavə qeyd. Yeni blokları təsdiqləmək üçün şəbəkə saatından istifadə olunur. Blokun yaradılma zamanı blok başlığında göstərilir. Hər bir Bitkoin qovşağında şəbəkə zaman sayğacı olur. Bu qovşaq cütlüklərinin median zamanına əsaslanır. Əgər median zaman sistem zamanından 70 dəqiqədən çox fərqlənirsə, şəbəkə zaman sayğacı sistem zamanına qayıdır. Hücumçu bir neçə müxtəlif qovşaqdan şəbəkəyə qoşularaq şəbəkə zaman sayğacında dəyişiklik edə bilər. Sayğac ya yavaşladılır, ya da sürətləndirilir.

Əgər bir blokun yaranma zamanı şəbəkə zamanından 2 saat öndə olarsa, qovşaqlar onu rədd edir. Hücumçu zamanı həqiqi zamandan 190 dəqiqə fərqlənən yeni blok yaradır. Mədənçi məntiqinə görə isə, blokun yaranma zamanı şəbəkə sayğacından 120 dəqiqədən çox keçməyibsə, mədənçi bu bloku qəbul edə bilər. Nəticə olaraq timejacking hücumda hücumçu şəbəkə zaman sayğacını dəyişdirir və aldadılmış qovşaq bloku alternativ bir blok zəncirinə qəbul edə bilər. Bu isə iki dəfə xərcləmə hücumuna və hesablama resurslarının israfına səbəb ola bilər.

Finney hücumu – Hücumçu əvvəlcədən bir tranzaksiyanı mədənçilik etdiyi bloka yerləşdirir**,** amma həmin tranzaksiyanı hələ şəbəkədə paylaşmır. Sonra mağazadan

Bitcoin ilə məhsul alıb, tranzaksiya yayılandan sonra "köhnə" blok ilə ödəməni ləğv edir. Çözüm olaraq, Bitcoin ödənişlərinin bir neçə blok təsdiqi alması tövsiyə edilir.
Vector76 hücumu – Finney və yarış hücumlarının kombinasiyasıdır, 2011-ci ildə təsvir olunub, tranzaksiyaların təsdiqi mexanizmlərindəki zəifliklərdən və blokların Bitcoin şəbəkəsi üzrə yayılmasında vaxt gecikmələrindən istifadə edir. Vector76 hücumunda bədniyyətli əvvəlcə iki tranzaksiya yaradır: biri öz Bitcoin ünvanına vəsait göndərmək, digəri isə eyni vəsaiti satıcının Bitcoin ünvanına göndərmək üçün. Sonra o, satıcını təsdiqlənməmiş tranzaksiyanı qəbul etməyə inandırmağa çalışır, eyni zamanda şəbəkədə o biri tranzaksiyanı yayır. Bədniyyətli öz tranzaksiyasını satıcının təsdiqləməsindən daha tez başa çatdıra bilsə, vəsait satıcının deyil, bədniyyətlinin ünvanına göndəriləcək.

Mühazirə üzrə yoxlama sualları

1. Kriptovalyuta nədir?
2. Blokçeyn nədir? Blokçeynin 3 əsas xüsusiyyətini sadalayın.
3. Blokçeyndə “blok” nədir?
4. Blokçeyndə blokları nə ilə “zəncirləyirlər”?
5. Bitcoin-də hansı heş funksiyası istifadə edilir?
6. İlk kriptovalyuta hansıdır və nə vaxt yaradılıb?
7. Bitcoin-də “halving” nədir?
8. Kriptovalyuta kontekstində “mayninq” nədir?
9. Kriptovalyutalarda “fork” nədir və hansı növləri var?
10. Altcoin nədir?
11. Stablecoin nədir?
12. Blokçeynlərdə “konsensus” nədir və niyə vacibdir?
13. Proof-of-Work (PoW) konsensus mexanizmi necə işləyir?
14. Proof-of-Stake (PoS) konsensus mexanizmi necə işləyir?
15. PoW və PoS konsensus modelləri arasında əsas texniki fərq nədir?
16. Layer 0 (L0) qatı nəyə xidmət edir?
17. Layer 1 (L1) qatı nəyə xidmət edir?
18. Layer 2 (L2) qatı nəyə xidmət edir?
19. Layer 3 (L3) qatı nəyə xidmət edir?
20. Ağıllı müqavilə nədir?
21. Ağıllı müqavilələr hansı maşında yerinə yetirilir?
22. Bitcoin və Ethereum-un əsas fərqi nədir?
23. Merkl ağacı blokçeyndə hansı məqsədlə istifadə olunur?
24. Merkl ağacı necə qurulur?
25. Merkl ağacının uzunluğu neçə baytdır?

15. Post-kvant kriptoqrafiyası

15.1. Kvant kriptoqarfiyası – ümumi baxış

Kvant kriptoqrafiyası ilə post-kvant kriptoqrafiyasını fərqləndirmək lazımdır. Kvant kriptoqrafiyası informasiyanın kvant vəziyyətlərində kodlaşdırılması metodudur, məlumatı göndərmək üçün kvant mexanikası qanunlarından istifadə edir. Kvant kriptoqrafiyasının informasiyanın digər mühafizəsi metodları ilə müqayisədə əsas üstünlüyü kanala gizli qulaq asmanın mümkünsüz olmasıdır.

Kvant rabitəsini təkcə optik lifli kabellə deyil, lazer şüası ilə "hava ilə" də etmək mümkündür, bu rabitə variantının əsas üstünlüyü – maneə və küyün az olmasıdır, verilənləri ötürmə sürəti də daha yüksək ola bilər.

Kvant kriptoqrafiyasında ilk addım 1984-cü ildə C. Bennet və G. Brassard tərəfindən *açarların kvant paylanması* (Quantum Key Distribution, QKD) sxemi ilə atılıb. BB84 adlanan bu protokolda iki tərəf arasında məxfi açarı yaymaq üçün autentifikasiya edilməli olan kvant kanalı və etibarlı olmayan klassik kanalın kombinasiyası istifadə edilir.

Hazırda açarların kvant paylanması ilə yanaşı aşağıdakı mövzularda da tədqiqatlar aparılır:

- Təhlükəsiz birbaşa kvant rabitəsi (Quantum Secure Direct Communication, QSDC);
- Kvant axın şifrləməsi (Quantum Stream Cipher, QSC);
- Kvant steqanoqrafiyası (Quantum Steganography, QS);
- Kvant açarlar infrastrukturu (Quantum Key Infrastructure, QKI);
- Kvant rəqəmsal imzası (Quantum Digital Signature, QDS).

Açarların kvant paylanması sahəsində aparat kompleksləri təklif edən bir neçə şirkət var.

Kvant kompüterləri və onlarda icra olunacaq kvant alqoritmləri müasir kriptoqrafiyanı "ölümlə hədələyən" problemlərdən biridir. Hazırda kvant kompüterləri sahəsində xeyli irəliləyişlər var. Hətta adi istifadəçilər də bir neçə kubitlə də olsa, real kvant kompüterlərində işləyə bilərlər (IBM, Google kvant kompüterlərində). 50-100 kubitlik kvant kompüterləri bugünkü klassik rəqəmsal kompüterlərin imkanlarını üstələyən tapşırıqları yerinə yetirə bilər, lakin kvant ventillərindəki küy etibarlı yerinə yetirilə bilən kvant dövrələrinin ölçüsünü məhdudlaşdırır. Mütəxəssislər NISQ (Noisy Intermediate-Scale Quantum) texnologiyasının yaxın gələcəkdə əlyetər olacağını vəd edirlər.

Geniş istifadə edilən müasir açıq açarlı kriptosistemlər tam ədədlərin vuruqlara ayrılması və ya diskret loqarifm məsələlərinin hesablama çətinliyinə əsaslanırlar. Lakin bu məsələləri Şor alqoritmindən istifadə etməklə kvant kompüterlərində

asanlıqla həll etmək olar. Yetərincə məhsuldar kvant kompüterlərinin gəlişi ilə bu alqoritmlərin istifadəsi mənasız olacaq. Buna görə son onilliklərdə post-kvant kriptoqrafiyası sahəsində xeyli tədqiqatlar aparılıb, standartlaşdırma işləri yekunlaşıb. Bu mühazirənin məqsədi post-kvant kriptoqrafiyasının müasir vəziyyəti barədə məlumat verməkdir.

15.2. Kvant alqoritmləri

1990-cı illərdə klassik analoqları olmayan ilk kvant alqoritmləri təklif edilmiş və onların istənilən klassik alqoritmdən effektiv olmaları isbat edilmişdi. Kriptoqrafiya üçün xüsusi əhəmiyyət kəsb edən kvant alqoritmlərindən axtarış üçün Qrover alqoritmini və ədədin vuruqlara ayrılması üçün Şor alqoritmini göstərmək olar.

Simmetrik kriptosistemlərdə bütün variantları yoxlamaqla məxfi açarın tapılmasının zaman mürəkkəbliyi $O(2^n)$-dir, Qrover alqoritmi isə bu məsələni $O(2^{n/2})$ zaman müddətində həll etməyə imkan verir.

Şor alqoritmi, bir çox digər kvant alqoritmi kimi, kvant hissəsindən və adi kompüterdə işləyən klassik hissədən ibarətdir. Şor alqoritminin kvant hissəsi müəyyən funksiyanın periodunu tapmaqdan ibarətdir və kvant Furye çevirməsi (Quantum Fourier Transform, QFT) ilə yerinə yetirilir.

Fərz edək ki, Şor alqoritminin klassik hissəsində N ədədini vuruqlara ayırmaq istəyirik. Bunun üçün təsadüfi a ədədi seçilir və $f(x) = a^x mod\ N$ funksiyasının ω periodu axtarılır. N ədədinin ən böyük böləni $a^{\frac{\omega}{2}} - 1$ ədədi olacaq.

Qeyd etmək lazımdır ki, a-nın təsadüfi seçimi həmişə N-in bölənlərini tapmağa imkan verməyə bilər. Bu halda a yenidən seçilir və a-nın az sayda qiyməti yoxlandıqdan sonra N-in böləni yüksək ehtimalla tapılacaqdır.

Şor alqoritmi N ədədini $O(log\ N)$ kubitdən istifadə edərək $O(log^3 N)$ müddətində vuruqlara ayırmağa mkan verir. Bir neçə min kubitlik kvant kompüterindən istifadə edərək açıq açarlı kriptoqrafik sistemləri sındırmaq mümkündür (məsələn, RSA-nı). Bu cür sistemləri sındırmaq üsullarından biri açıq açarın sadə vuruqlara ayrılmasıdır. Adi kompüterlər üçün klassik alqoritmlər belə sistemi $O\left(N^{\frac{1}{4}}\right)$ polinomial müddətində sındırmağa imkan verir, həm də bu müddət sadə ədədlərin vurulması müddəti ilə müqayisə oluna biləndir.

n bitlik DLP məsələsinin Şor alqoritmi ilə kvant kompüterində həllinin çətinliyi $O(n^2 l \log_2 n\,(\log_2 \log_2 n))$-dir, yəni Şor alqoritmi polinomial çətinlikdədir.

n bitlik əyridə ECDLP (Elliptic Curve Discrete Logarithm Problem) məsələsinin həlli üçün mövcud registr bölgülü alqoritm $f(n) = 5n + 8\sqrt{n} +$

$4\log_2(n) + \varepsilon$ kubit tələb edir, burada ε – təxminən 10-a bərabər olan sabitdir. Buradan n-in hazırda aktual qiymətləri üçün $f(256) = 1500$ kubit, $f(512) = 2800$ kubit alırıq.

15.3. Post-kvant kriptoqrafiyası – qısa icmal

Bəzi açıq açarlı kriptoqrafik sistemlər kvant hesablamalarına dayanıqlı hesab olunur:

- Heş funksiyalara əsaslanan kriptoqrafiya
- Kodlaşdırma nəzəriyyəsinə əsaslanan kriptoqrafiya
- Qəfəslərə əsaslanan kriptoqrafiya
- Çoxölçülü kvadratik sistemlərə əsaslanan kriptoqrafiya
- Supersinqulyar izogeniyalar əsasında kriptoqrafiya

Heş funksiyalara əsaslanan kriptoqrafiya. Heş əsaslı kriptoqrafiya digər post-kvant kriptoqrafiya sxemlərindən fərqli olaraq, vuruqlara ayırma və diskret loqarifm kimi çətin riyazi məsələlərin həllinə deyil, kriptoqrafik heş funksiyalarının təhlükəsizliyinə (proobraz axtarışı məsələsinin çətinliyinə) əsaslanır. Kvant kompüteri heş funksiya üçün proobrazın tapılması məsələsinin həllində yalnız kvadratik sürətlənmə təmin edir və buna görə də heş funksiyalarına əsaslanan kriptoqrafik alqoritmlər post-kvant kriptoqrafiyasında uğurla istifadə oluna bilər.

Özlüyündə heş funksiyalarına əsaslanan kriptoqrafik alqoritmlər olduqca mürəkkəbdir. Belə yanaşmaya klassik misal Merkl imzalarıdır. Onun iş prinsipi 3 addımdan ibarətdir:

1. Gizli açarların X və açıq açarların Y massivləri yaradılır. (X_i, Y_i) cütləri birdəfəlik imza üçün açıq-gizli açar cütləri kimi istifadə olunur. Y-i yaratmaq üçün Merkl ağacından istifadə olunur: hər bir Y_i üçün H kriptoqrafik heş funksiyasından istifadə etməklə $H(Y_i)$ – a_0 ağacının sıfırıncı təbəqəsi hesablanır. Hər bir növbəti təbəqə $H(a_{i,2n} \parallel a_{i,2n+1})$ kimi hesablanır (burada $\parallel$ konkatenasiyadır), hesablamalar təbəqədə yalnız bir açar qalana kimi davam etdirilir, həmin açar pub_key ilə işarə edilir və açıq açar adlanır.
2. İmza yaratmaq üçün (X_i, Y_i) açarlar cütü seçilir. d məlumatı üçün birdəfəlik imza b və Y_i-dən ağacın kökünə gedən yol hesablanır. İmzaya Y_i yoxlama açarı, b imzası və ağacın kökünə gedən yol daxildir.
3. İmzanın yoxlanılması 2 mərhələdə həyata keçirilir: əvvəlcə, alıcı b imzasının d məlumatına uyğun olduğunu yoxlayır. Əgər yoxlamadan keçirsə, ağacın kökünə kimi $H(Y_i)$ yolu qurulur. Ağacın alınmış kökü pub_key ilə üst-üstə düşürsə, imzanın yoxlanılması uğurlu sayılır.

RSA kimi alqoritmlərdən fərqli olaraq, Merkl imzası post-kvantdır, çünki o, H kriptoqrafik heş funksiyasının və birdəfəlik imzaların təhlükəsizliyinə əsaslanır. Heş funksiyalarına əsaslanan belə alqoritmlərin əhəmiyyətli çatışmazlığı hər məlumat üçün bir dəfə istifadə edilən imzaların məhdud sayda olmasıdır ki, bu da belə yanaşmanın kütləvi istifadəsinə mane olur. Bununla belə, çoxdəfəlik imzaların oxşar şəkildə yaradılması üsulları artıq mövcuddur.

Kodlara əsaslanan kriptoqrafiya. Belə sistemlərə bir nümunə tam xətti kodların dekodlaşdırılmasının çətinliyinə və şifrləmə prosesində təsadüfilik elementindən istifadə edilməsinə əsaslanan Mak-Elis (McEliece) açıq açarlı kriptosistemidir. Bu sistemdə n, k, t təhlükəsizlik parametrləri bütün istifadəçilər tərəfindən birgə istifadə edilir. Kvant hücumlarına qarşı sdayanıqlıq üçün $n = 6960,\ k = 5431, t = 119$ istifadə etmək təklif olunur, bu halda açıq açarın ölçüsü 8 373 911 bitə çatır. Mak-Elis kimi sistemlərin əsas nöqsanı açıq açarın böyük ölçüsü və şifrmətnin uzunluğunun əhəmiyyətli dərəcədə artmasıdır. Bununla belə, post-kvant kriptoqrafiyasında bu yanaşmanın ciddi nöqsanı yoxdur (məhdud sayda imza kimi), verilənlərin ötürülməsi sürətinin artması fonunda ötürülən məlumatların ölçüsü problem deyil, bu səbəbdən Mak-Elis perspektivli hesab olunur.

Mak-Elis kriptosisteminin iş prinsipi aşağıdakı kimidir:

Tutaq ki, V_n – sonlu F_m meydanı üzərində xətti fəzadır. İxtiyari $C \in V_n(F_m)$ altfəzası xətti blok kodu adlanır.

Tutaq ki, $C \in V_n(F_m)$ – ölçüsü k olan xətti koddur. C altfəzasının bazis vektorlarından təşkil edilmiş $k \times n$ ölçülü G matrisinə C kodunun törədici matrisi deyilir.

Mak-Elis kriptosistemində açıq və gizli açarlar aşağıdakı kimi yaradılır:

1. Tutaq ki, G – t səhvi düzəldən (n, k) xətti kodunun $k \times n$ ölçülü törədici matrisidir.
2. $k \times k$ ölçülü cırlaşmayan, təsadüfi S matrisi götürülür.
3. $n \times n$ ölçülü təsadüfi P terdəyişmə matrisi götürülür.
4. Açıq açar: (G', t) cütüdür, burada $G' = SGP$.
5. Gizli açar: (S, G, P) üçlüyüdür.

Mak-Elis kriptosistemində şifrləmə və deşifrləmə aşağıdakı kimi yerinə yetirilir:

Şifrləmə:

1. Açıq mətn m binar simvolların k uzunluqda blokları şəklində göstərilir.
2. $c' = mG'$ vektoru hesablanır.

Deşifrləmə:

1. P^{-1} və S^{-1} tərs matrisləri
2. $c' = cP^{-1}$ hesablanır.
3. C kodunun dekodlaşdırma alqoritmi ilə c'-dən m' alınır.

3. Uzunluğu n və t ədəd vahid olan təsadüfi z vektoru generasiya edilir.
4. Şifrmətn $c = c' + z$ kimi hesablanır.

4. Açıq mətn $m = m'S^{-1}$ kimi hesablanır.

Adətən, Mak-Elis kriptosistemində binar Qoppa kodları istifadə edilir (Valeri Denisoviç Qoppa, 1970-ci ildə təklif edilib). Binar Qoppa kodları Peterson alqoritminin köməyi ilə asanlıqla dekodlaşdırılır. Mak-Elis kriptosisteminin orijinal variantında uzunluğu $n = 1024$, ölçüsü $k = 524$ və minimal kod məsafəsi $d \geq 101$ olan Qoppa kodu istifadə edilirdi, bu kod $t = 50$ səhvi düzəldə bilərdi.

Qəfəslərə əsaslanan kriptoqrafiya. Qəfəslər (ing. lattice) xüsusi riyazi strukturlardır. Qəfəs – $\mathbb{R}^n$ Evklid fəzasının toplama üzrə diskret qrup təşkil edən vektorları çoxluğudur. Tamqiymətli qəfəs istənilən iki vektorunun skalyar hasili tam ədəd olan qəfəsdir. Qəfəslər üzərində kriptoqrafik alqoritmlər aşağıdakı iki NP-çətin məsələnin həllinin çətinliyinə əsaslanır: 1) Uzun bazis üzrə qısa bazisin müəyyən edilməsi (Short Basis Problem) və 2) Uzun bazislə verilmiş qəfəsin n-ölçülü fəzanın ixtiyari nöqtəsinə ən yaxın nöqtəsinin axtarılması (Closest Vector Problem, CVP). Qəfəslərə əsaslanan kriptoqrafiyada CVP məsələsinin variasiyaları daha populyardır.

Çoxölçülü kvadratik sistemlərə əsaslanan kriptoqrafiya. Çoxdəyişənli xətti tənliklərin kvadratik sisteminin (multivariate quadratics, MQ) həllinin tapılması məsələsi NP-çətindir və bunun əsasında kriptoqrafik alqoritmlər qurmaq olar. Bu bölmənin sistemləri yaxşı sürətə malikdir və hesablama resurslarına tələbləri yüksək deyil, lakin açıq açarların uzunluğu kifayət qədər böyükdür. Ən məşhur nümunə 1996-cı ildə J.Patarin tərəfindən təklif edilən HFE (Hidden Fields Equations) kriptosistemidir, gizli meydan tənliklərinə əsaslanır.

Supersinqulyar izogeniyalara əsaslanan kriptoqrafiya. Aşağıdakı məsələnin çətinliyinə əsaslanır: j–invariantları müxtəlif olan iki izogen E_1 və E_2 **elliptik** əyriləri verilib. E_1 və E_2 **əyriləri arasında izogeniyanı tapmaq tələb olunur** (j-invariant elliptik əyrinin məlum xarakteristikasıdır)**.** Belə sistemlərə nümunə supersinqulyar izogeniya əsasında Diffi-Hellman açar mübadiləsi protokoludur (Supersingular Isogeny Diffie-Hellman key exchange, SIDH).

15.4. Post-kvant kriptoqrafiyası üzrə standartlaşdırma

2016-cı ildə ABŞ Milli Standartlar və Texnologiya İnstitutu (NIST) post-kvant kriptoqrafiyası sxemlərini standartlaşdırmaq üçün bir neçə mərhələdən ibarət müsabiqə elan etmişdi və hazırda dördüncü mərhələdə namizəd alqoritmlər müzakirə edilir. Post-kvant kriptoqrafiyasında yanaşmalar köklü surətdə fərqlənirlər, bu

səbəbdən alqoritmlərin birbaşa müqayisəsi çox vaxt mümkün olmur, buna görə də NIST-in fəaliyyəti təqdim olunan alqoritmlərin məhsuldarlığına və təhlükəsizliyinə yönəlib. 69 namizəd alqoritmdən üçüncü mərhələyə 7 əsas və 8 alternativ alqoritm keçmişdi və onlardan 4-ü standart kimi seçilib (2022, iyul).

Rəqəmsal imza standartları CRYSTALS-Dilithium, FALCON və SPHINCS+-dır. CRYSTALS-Dilithium və Falcon tamqiymətli qəfəslərin, SPHINCS+ isə heş funksiyaların əsasında qurulub. CRYSTALS qısaltması "Cryptographic Suite for Algebraic Lattices" kimi açılır. CRYSTALS-Kyber açar inkapsulyasiyası mexanizmidir.

NIST bir neçə imza alqoritminin seçilməsinə baxmayaraq, CRYSTALS-Dilithium-u əsas alqoritm kimi tövsiyə edir. Bundan əlavə, açar yaratma alqoritmlərinə dörd alternativ namizəd dördüncü qiymətləndirmə mərhələsinə keçmişdi: BIKE, Classic McEliece, HQC (səhvin korreksiyası kodlarına əsaslanır) və SIKE (izogeniyalara əsaslanır). Bu namizədlər gələcək standartlar kimi nəzərdən keçirilirdi. NIST, həmçinin imza alqoritmləri portfelini artırmaq və şaxələndirmək üçün açıq açarlı rəqəmsal imza alqoritmləri üçün yeni təkliflər çağırışı da elan etmişdi.

Mühazirə üzrə yoxlama sualları

1. Kvant kompüterləri fizikanın hansı sahəsinə əsaslanır?
2. Kvant dövrələrinin ölçüsünü nə məhdudlaşdırır?
3. Kvant kriptoqrafiyası nədir?
4. Kvant kriptoqrafiyasının əsas üstünlüyü nədir?
5. Post-kvant kriptoqrafiyası nədir?
6. İlk kvant kriptoqrafiyası protokolu hansıdır?
7. İlk kvant kriptoqrafiyası protokolu neçənci ildə təklif edilib?
8. İlk kvant kriptoqrafiyası protokolunun müəllifləri kimlərdir?
9. Qrover alqoritmi hansı məsələnin həlli üçün təklif edilib?
10. Grover alqoritmi hansı klassik sistemlərə necə təsir edir?
11. Şor alqoritmi hansı kriptosistemləri təhdid edir?
12. Şor alqoritmi ilə N-in vuruqlara ayrılmasının çətinliyi necə qiymətləndirilir?
13. Şor alqoritmi ilə DLP məsələsinin həllinin çətinliyi necə qiymətləndirilir?
14. Post-kvant kriptoqrafiyasının əsas yanaşmaları hansılardır?
15. Heş funksiyalara əsaslanan kriptoqrafiyanın mahiyyəti nədir?
16. Merkl imzası nə üçün post-kvant hesab edilir?
17. Kodlara əsaslanan kriptoqrafiyanın mahiyyəti nədir?
18. Mak-Elis kriptosisteminin iş prinsipi nədən ibarətdir?
19. Mak-Elis kriptosistemində hansı binar kodlar istifadə edilir?

20. Qəfəslərə əsaslanan kriptoqrafiyanın mahiyyəti nədir?
21. Çoxölçülü kvadratik sistemlərə əsaslanan kriptoqrafiyanın mahiyyəti nədir?
22. Supersinqulyar izogeniyalara əsaslanan kriptoqrafiyanın mahiyyəti nədir?
23. Post-kvant kriptoqrafiyası üzrə standartlaşdırma müsabiqəsi nə vaxt elan olunmuşdu?
24. Post-kvant kriptoqrafiyası üzrə standartlardan hansını göstərə bilərsiniz?
25. CRYSTALS qısaltması necə açıqlanır?

İxtisarlar

	A
AES	Advanced Encryption Standard *Qabaqcıl şifrləmə standartı*
AH	Authentication Header *Autentifikasiya başlığı*
ANSI	American National Standards Institute *Amerika Milli Standartları İnsititutu*
API	Application Program Interface *Tətbiqi proqramlaşdırma interfeysi*
ASCII	American Standard Code for Information Interchange *İnformasiya mübadiləsi üçün Amerika standart kodu*
ASN.1	Abstract Syntax Notation One *Mücərrəd sintaksis işarələnməsi 1*
	B
BC	Block Chaining *Blokların qoşulması*
BER	Basic Encoding Rules *Əsas kodlaşdırma qaydaları*
	C
CA	Certification Authority *Sertifikat mərkəzi*
CBC	Cipher Block Chaining *Şifr blokların qoşulması*
CFB	Cipher Feedback *Şifr əks əlaqəsi*
CHAP	Challenge Handshake Authentication Protocol *Çağırış-əlsıxma autentifikasiya protokolu*
CPS	Certificate Practices Statement *Sertifikatın verilməsi qaydası*
CRL	Certificate Revocation List *Ləğv edilmiş sertifikatlar siyahısı*
CryptoAPI	Cryptographic Application Programming Interface *Kriptoqrafik tətbiqi proqramlaşdırma interfeysi*
CTR	Counter Mode *Sayğac rejimi*
	D
DER	Distinguished Encoding Rules *Fərqləndirici kodlaşdırma qaydaları*
DES	Data Encryption Standard *Verilənləri şifrləmə standartı*
DH	Diffie-Hellman *Diffi-Helman*
DSA	Digital Signature Algorithm

	Rəqəmsal imza alqoritmi
DSS	Digital Signature Standard *Rəqəmsal imza standartı*
	E
EAP	Extensible Authentication Protocol *Genişlənən autentifikasiya protokolu*
ECB	Electronic codebook *Elektron kod kitabı*
ECC	Elliptic Curve Cryptosystem/Cryptography *Elliptik əyrilər üzərində kriptosistem/kriptoqrafiya*
ECDL	Elliptic Curve Discrete Logarithm *Elliptik əyrilər üzərində diskret loqarifm*
ECDSA	Elliptic Curve Digital Signature Algorithm *Elliptik əyrilər üzərində rəqəmsal imza alqoritmi*
ESP	Encapsulating Security Payload *Şifrlənmiş verilənlərin inkapsulyasiyası protokolu*
ETSI	European Telecommunications Standardization Institute *Avropa Telekommunikasiya Standartlaşdırma İnstitutu*
	F
FIPS	Federal Information Processing Standard *Federal informasiya emalı standartı*
	G
GPS	Global Positioning System *Qlobal Mövqetəyinetmə Sistemi*
GSM	Global System for Mobile *Mobil rabitə üçün qlobal sistem*
	H
HMAC	Hash- Message Authentication Code *Məlumatın heş əsasında autentifikasiya kodu*
HTTPS	HyperText Transfer Protocol Secure *Hipermətni təhlükəsiz ötürmə protokolu*
	I
IBE	Identity Based Encryption *Identifikator əsasında şifrləmə*
ID	Identifier (identification) *İdentifikator (eyniləşdirmə)*
IDEA	International Data Encryption Algorithm *Verilənlərin şifrlənməsi üzrə beynəlxalq alqoritm*
IKE	Internet Key Exchange *İnternet açar mübadiləsi*
IKP	Internet Keyed Payments Protocol *Internet açarlı ödənişlər protokolu*

IPSec	Internet Protocol Security *İnternet protokolun təhlükəsizliyi*
ISAKMP	Internet Security Association and Key Management Protocol *Internet təhlükəsizlik assosiasiyası və açar idarəetmə protokolu*
IV	Initialization Vector *İlkin yükləmə vektoru*
	J
JCE	Java Cryptographic Extension *Java şifrləmə əlavəsi*
JSON	JavaScript Object Notation *Java Skript Obyekt İşarələri*
	K
KDC	KeyDistribution Center *Açar paylaşdırma mərkəzi*
KMC	KeyManagement Center *Açar idarəetmə mərkəzi*
KMI	Key Management Infrastructure *Açar idarəetmə infrastrukturu*
KMP	Key Management Protocol *Açar idarəetmə protokolu*
KMS	Key Management System *Açar idarəetmə sistemi*
	L
LDAP	Lightweight Directory Access Protocol *Kataloqa sadə giriş protokolu*
LFSR	Linear Feedback Shift Register *Xətti əks-əlaqə sürüşmə registri*
	M
MAC	1. Message Authentication Code *Məlumatın autentifikasiya kodu*
MD	Message Digest *Məlumat heş-kodu*
MDC	Manipulation Detection Code *Manipulyasiyanı aşkarlama kodu*
MIC	Message Integrity Check *Məlumatın tamlığını yoxlama*
MITM	Man-in-the-middle *Ortada adam*
	N
NIST	National Institute of Standards and Technology *Milli standartlar və texnologiyalar institutu*
NSA	National Security Agency *Milli Təhlükəsizlik Agentliyi*
	O

OAEP	Optimal Asymmetric Encryption Padding *Optimal asimmetrik şifrləmə əlavəsi*
OCSP	On-line Certificate Status Protocol *Onlayn sertifikat statusu protokolu*
OFB	Output Feedback *Çıxışla əks-əlaqə rejimi*
OFBNLF	Output Feedback With A Nonlinear Function *Qeyri-xətti funksiya ilə çıxışla əks-əlaqə rejimi*
OTP	1. One-Time Password *Birdəfəlik parol* 2. One-Time Pad *Birdəfəlik bloknot*
	P
PAP	1. Password Authentication Protocol *Parol autentifikasiya protokolu*
PBC	Plaintext Block Chaining *Açıq mətn bloklarının qoşulması*
P-BEST	Production-Based Expert System Toolset *İstehsalat-əsaslı ekspert sistemləri alətləri*
PCBC	Propagating Cipher Block Chaining *Şifrlənmiş blok zəncirinin yayılması rejimi*
PEAP	Protected EAP *Qorunan EAP*
PEM	Privacy Enhanced Mail *Yüksək gizlilik poçtu*
PFB	Plaintext Feedback *Açıqmətnlə əks əlaqə rejimi*
PFS	Perfect Forward Secrecy *Birbaşa mükəmməl məxfilik*
PGP	Pretty Good Privacy *Yüksək gizlilik proqramı*
PKC	Public Key Cryptography *Açıq açarlı kriptoqrafiya*
PKCS	Public-Key Cryptography Standards *Açıq açarlı kriptoqrafiya standartı*
PKI	Public Key Infrastructure *Açıq açar infrastrukturu*
PKIX	Public Key Infrastructure X.509 *Açıq açar infrastrukturu X.509*
PRNG	Pseudo Random Number Generator *Psevdotəsadüfi ədədlər generator*
	Q

QKD	Quantum Key Distribution *Açarların kvant paylanması*
	R
RA	Registration Authority *Qeydiyyat mərkəzi*
RC4/RC5/RC6	Rivest Chiper
RIPEMD	RIPE Message Digest
RSA	Rivest, Shamir, and Adleman
	S
SAFER	Secure And Fast Encryption Routine *Təhlükəsiz və sürətli şifrləmə alqoritmi*
SEAL	Software-optimized Encryption Algorithm *Proqram təminatına optimallaşdırılmış şifrləmə alqoritmi*
SHA	Secure Hash Algorithm *Təhlükəsiz heş alqoritmi*
SHS	Secure Hash Standard *Təhlükəsiz heş standartı*
SKIP	Simple Key-management for Internet Protocols *Internet protokolu üçün sadə açar idarəetməsi*
SPKI	Simple Public Key Infrastructure *Sadə açıq açar infrastrukturu*
SSL	Secure Socket Layer *Təhlükəsiz soket səviyyəsi*
	T
TLS	Transport Layer Security *Nəqliyyat səviyyəsinin təhlükəsizliyi*
Triple-DES	Triple Data Encryption Standard *Üç qat məlumat şifrləməsi standartı*
TSA	Time-Stamping Authority *Vaxt möhürü mərkəzi (xidməti)*
TTP	Trusted third party *Etibar edilən üçüncü tərəf*
	V
VPN	Virtual Private Network *Virtual xüsusi şəbəkə*
	W
WAP	Wireless Application Protocol *Simsiz tətbiqi proqram protokolu*
WEP	Wired Equivalent Privacy *Naqilli şəbəkəyə ekvivalent gizlilik*
	0-9
3DES	Triple DES *Üçqat DES*

Əlavə 1. DES-də istifadə edilən cədvəllər

Cədvəl 5.1. İlkin və son permutasiya cədvəlləri

IP							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

IP^{-1}							
40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

Cədvəl 5.2. *E* genişlənmə funksiyası

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

Cədvəl 5.4. Yerdəyişmə funksiyası P							
16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Cədvəl 5.5. Açarın ilkin emalı matrisi *PC*-1

57 49 41 33 25 17 09 01 58 50 42 34 26 18
10 02 59 51 43 35 27 19 11 03 60 52 44 36
63 55 47 39 31 23 15 07 62 54 46 38 30 22
14 06 61 53 45 37 29 21 13 05 28 20 12 04

Cədvəl 5.6. Açarın son emalı matrisi *PC*-2

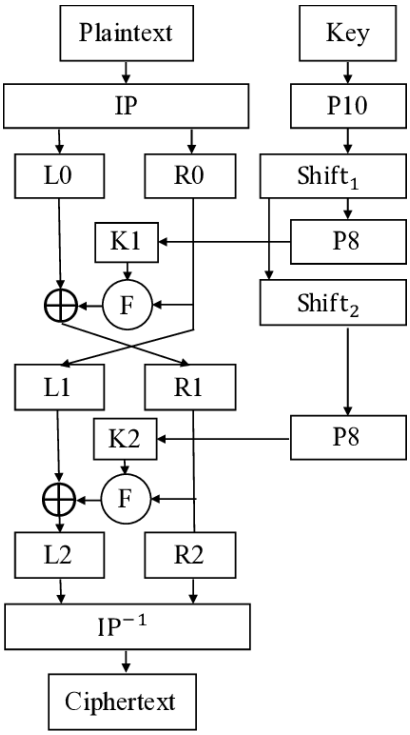
14 17 11 24 01 05 03 28 15 06 21 10
23 19 12 04 26 08 16 07 27 20 13 02
41 52 31 37 47 55 30 40 51 45 33 48
44 49 39 56 34 53 46 42 50 36 29 32

Cədvəl 5.3. *S*1, *S*2, ..., *S*8 çevirmə funksiyaları

		Sütunun nömrəsi																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Sətrin nömrəsi	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S1
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S2
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	S3
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	S4
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	S5
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	S6
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	S7
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	5	
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	S8
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Əlavə 2. S-DES şifrinə aid sadə məsələlər

Aşağıda S-DES (Simplified DES) şifri ilə bağlı müxtəlif növ məsələlər təqdim edilir. S-DES, DES-in sadələşdirilmiş versiyasıdır və şifrləmə üçün deyil, yalnız tədris məqsədləri üçün istifadə olunur. S-DES-in şəkildə göstərilən sxemini başa düşmək çətin deyil. Aşağıdakı məsələlərin "kağız-qələmlə" həlli DES-in işləmə prinsiplərini bir daha "beyin süzgəcindən" keçirməyə, dərindən başa düşməyə kömək edəcək və alternativlər yaratmaq üzərində düşünməyə təkan verəcək.



M1. Verilmiş 11010111 açıq mətnini IP=(2, 6, 3, 1, 4, 8, 5, 7) ilkin permutasiyası ilə qarışdırın. **Cavab:** 1 1 0 1 1 1 0 1

M2. IP nəticəsini sol və sağ hissələrə ayırın. **Cavab:** L=1101 və R=1101.

M3. Əsas açar 1010000010-dır. Ona P10: (3, 5, 2, 7, 4, 10, 1, 9, 8, 6) permutasiyasını tətbiq edin.

M4. Əsas açara P10 tətbiqindən sonra alınmış 1000001100 nəticəsinin 5 bitlik sol və sağ hissələrə bölün və LS-1 (sol sürüşdürmə) tətbiq edin.
Cavab: Sol hissə üçün LS-1 → 00001; Sağ hissə üçün: LS-1 → 11000.

M5. Əvvəlki məsələdə LS-1 nəticələrini birləşdirin, onu P8: (6, 3, 7, 4, 8, 5, 10, 9) permutasiyası ilə qarışdıraraq birinci raundun açarını alın (K1).
Cavab: K1=10000100

M6. LS-1 nəticələrinə LS-2 (iki sol sürüşdürmə) tətbiq edin.
Cavab: Sol hissə üçün LS-2(00001) → 00100; Sağ hissə üçün LS-2(11000) → 00011.

M7. LS-2 nəticələrini birləşdirin, onu P8 permutasiyası ilə qarışdıraraq ikinci raundun açarını alın (K2).
Cavab: K2=01000011.

M8. Açıq mətnin R = 1101 sağ hissəsini EP=(4, 1, 2, 3, 2, 3, 4, 1) permutasiyası ilə genişləndirin/qarışdırın.
Cavab: 1110 1011.

M9. EP nəticəsini K1 raund açarı ilə XOR edin.
Cavab: 01101111.

M10. XOR nəticəsini sol və sağ hissəyə ayırıb uyğun olaraq S0 və S1 S-bloklarını tətbiq edin.

$$S0 = \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \qquad S1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix}$$

Cavab:
`0110: sıra=0, sütun=3 → S0[0,3]= 2 → 10`
`1111: sıra=1, sütun=3 → S1[1,3]= 3 → 11`
S-blok nəticəsi 1011 olacaq.

M11. S-blok nəticəsinə P4: (2, 4, 3, 1) permutasiyasını tətbiq edin.
Cavab: P4(1011) = 0111

M12. P4 nəticəsini açıq mətnin sol hissəsi (1101) ilə XOR edin.
Cavab: 1010

M13. Birinci raundun sonunda sol hissə (1010) və sağ hissəni (1101) dəyişdirin.
Cavab: L=1101, R=1010

M14. Eyni qayda ilə ikinci raund çevirmələrini edin.

M15. Son raunddan sonra sol və sağ hissələrin birləşməsindən sonra alınmış 01101010 nəticəsinə IP^{-1}=(4, 1, 3, 5, 7, 2, 8, 6) tərs permutasiyasını tətbiq edərək şifrmətni alın. **Cavab:** 10111100.

Əlavə 3. Blokçeyn terminlərinin izahlı lüğəti

1. **Airdrop** – yeni tokenlərin istifadəçilərə pulsuz paylanması prosesi
2. **Altcoin** – Bitcoin xaricindəki bütün digər kriptovalyutalar.
3. **Atomic swap** – iki fərqli kriptovalyutanın birbaşa və etibarlı mübadiləsini təmin edən texnologiya
4. **BFT** (*Byzantine Fault Tolerance*) – şəbəkədəki bəzi qovşaqların səhv işləməsinə baxmayaraq, şəbəkənin düzgün işləməsini təmin edən konsensus mexanizmi
5. **BIP** (*Bitcoin Improvement Proposal*) – Bitkoinin təkmilləşdirilməsi layihəsi
6. **Block** – blokçeyndə verilənləri saxlayan və birləşdirən struktur vahididir.
7. **Block depth** – blokun blokçeyndə ən son əlavə edilmiş bloka nəzərən sıra nömrəsi
8. **Block height** – blokun blokçeyndə genesis (0-cı) blokuna nəzərən sıra nömrəsi
9. **Block reward** – blok yaradıcılarına verilən mükafatdır
10. **Blockchain** – verilənlərin bloklar şəklində şəbəkə üzərində paylanaraq saxlanıldığı mərkəzsiz verilənlər bazası
11. **Burning** – tokenlərin istifadə edilə bilməyəcək şəkildə "yandırılması" prosesidir.
12. **Coin** – öz blokçeyn şəbəkəsinə malik olan rəqəmsal valyuta.
13. **Cold wallet** – İnternetə bağlı olmayan, fiziki cihazlarda saxlanan kriptovalyuta pulqabı
14. **Consensus mechanism** – paylanmış şəbəkədə verilənlərin doğruluğunu təsdiqləmə mexanizmidir.
15. **Contract address** – ağıllı müqavilənin yerləşdiyi unikal blokçeyn ünvanı
16. **DAO** (*Decentralized Autonomous Organization*) – blokçeyn üzərində mərkəzsiz idarə olunan avtonom təşkilat
17. **Dapp** (*Decentralized Application*) – mərkəziyətsiz tətbiq – avtonom işləyən və öz verilənlərini blokçeyndə saxlayan açıq kodlu proqram
18. **DeFi** (*Decentralized Finance*) – mərkəziyətsiz maliyyə xidmətləri sistemi
19. **Double spending** – eyni rəqəmsal valyutanın bir neçə dəfə xərclənməsidir.
20. **ERC-20** – Ethereum şəbəkəsində tokenlərin yaradılması üçün standart
21. **ERC-721** – unikal tokenlərin yaradılması üçün standart
22. **Ethereum** – ağıllı müqavilələr və DApp-lər üçün mərkəziyətsiz platforma
23. **EVM** (*Ethereum Virtual Machine*) – Ethereium virtual maşını
24. **Faucet** – yeni istifadəçilərə kiçik miqdarda kriptovalyuta paylayan platforma
25. **Fiat currency** – dövlət tərəfindən qəbul edilən rəsmi valyuta
26. **Fork** – Blokçeyn şəbəkəsinin iki fərqli yola ayrılması
27. **Fungible token** – Eyni dəyərə malik, dəyişdirilə bilən tokenlər

28. **Gas** – Ethereum şəbəkəsində tranzaksiyaların yerinə yetirilməsi üçün tələb olunan hesablama resursu
29. **Genesis block** – blokçeyn şəbəkəsinin ilk bloku
30. **Governance token** – şəbəkə qərarlarında səsvermə hüququ verən tokenlər
31. **Halving** – kriptovalyutanın mədənçilik mükafatının yarıya endirilməsi prosesidir.
32. **Hard fork** – şəbəkə qaydalarının köklü şəkildə dəyişdirilməsi və əvvəlki versiyalarla uyğunluğun itirilməsi vəziyyəti
33. **Hashrate** – mədənçinin saniyədə hesablaya bildiyi heş sayı
34. **ICO** (*Initial Coin Offering*) – yeni tokenlərin ilkin satışı; layihə üçün maliyyə toplanması məqsədi daşıyır.
35. **Immutable** – dəyişdirilə və ya silinə bilməyən
36. **Query layer** – məlumatların indekslənməsi və blokçeyndən axtarış üçün istifadə edilən orta qat texnologiyası (məsələn, The Graph).
37. **Quickswap** – Polygon şəbəkəsində fəaliyyət göstərən sürətli və ucuz DEX platforması
38. **Quorum** – icazəli blokçeyn şəbəkəsi üçün hazırlanmış enterprise səviyyəli Ethereum forku
39. **Launchpad** – yeni kriptovalyuta layihələrinin başladılmasını və vəsait toplama proseslərini dəstəkləmək üçün istifadə edilən platforma və ya xidmətdir.
40. **Launchpool** – **y**eni kriptovalyuta layihələrində erkən mərhələdə iştirak etmək üçün yaradılan hovuzlardır.
41. **Layer 1** – əsas blokçeyn infrastrukturudur (məsələn, Ethereum, Bitcoin).
42. **Layer 2** – Layer 1 şəbəkəsinin üzərində qurulmuş və onu miqyaslandıran əlavə protokollardır (məsələn, Optimism, Arbitrum)
43. **Ledger** – tranzaksiyaların dəyişməz şəkildə qeydə alındığı rəqəmsal reyestrdir.
44. **Lightning Network** – Bitcoin üçün hazırlanmış, sürətli və ucuz ödənişlərə imkan verən Layer 2 həllidir.
45. **Liquidity** – bir aktivin sürətli və dəyər itirmədən pula çevrilə bilmə qabiliyyətidir.
46. **Liquidity pool** – istifadəçilərin tokenlərini depozit qoyduğu və avtomatlaşdırılmış ticarəti təmin edən hovuzdur.
47. **Lock-up period** – tokenlərin müəyyən müddət ərzində satıla və ya transfer edilə bilmədiyi dövr
48. **Mainnet** – tranzaksiyaların real zamanda baş verdiyi əsas blokçeyn şəbəkəsi
49. **Merkle root** – Merkle ağacının əsas heş qiyməti; verilənlərin dəyişdirilmədiyini sübut edir.

50. **Merkle tree** – böyük verilənlərin effektiv və təhlükəsiz yoxlanması üçün istifadə edilən kriptoqrafik verilənlər strukturu
51. **Metamask** – Ethereum və digər EVM əsaslı şəbəkələr üçün məşhur veb brauzer pulqabı
52. **Mining** – yeni blokların yaradılması və şəbəkənin təhlükəsizliyini təmin edən hesablama prosesi
53. **Minting** – yeni token və ya NFT-nin yaradılması əməliyyatı
54. **Multisig (Multi-signature)** – əməliyyatın icrası üçün birdən çox şəxsin təsdiqini tələb edən təhlükəsizlik metodu
55. **Nakamoto consensus** – Bitcoin və digər PoW şəbəkələrində istifadə edilən konsensus mexanizmi
56. **Native token** – bir blokçeyn şəbəkəsinin əsas kriptovalyutası (məsələn, Ethereum-da ETH).
57. **Network fee** – tranzaksiyaların blokçeyn üzərində yerinə yetirilməsi üçün ödənilən xidmət haqqı
58. **Node** – blokçeyn şəbəkəsində məlumat paylaşan və ya doğrulayan kompüter və ya cihaz
59. **Nonce** – blokçeyn tranzaksiyalarında və mədənçilikdə yalnız bir istifadə edilən təsadüfi ədəd
60. **Off-chain** – blokçeyn xaricində baş verən, lakin blokçeynlə əlaqəli məlumat və tranzaksiyalar
61. **On-chain** – bütün tranzaksiya və məlumatların blokçeyn şəbəkəsində qeydə alınması
62. **Oracle** – blokçeynə real dünya məlumatlarını ötürən üçüncü tərəf xidməti
63. **P2P** (*Peer to Peer*) – birranqlı kompüter şəbəkəsi
64. **P2SH** (*Pay to Script Hash*) – multi-imza texnologiyası
65. **PoS** (*Proof of Stake*) – token sahiblərinin paylarına əsasən şəbəkəni doğrulamasını təmin edən konsensus üsulu
66. **PoW** (*Proof-of-Work*) – yerinə yetirilmiş işin isbatı – Bitkoin mayninqi şəbəkəsində konsensus alqoritmi
67. **Private key** – pulqabılara və aktivlərə giriş imkanı verən gizli açardır.
68. **Public key** – açıq şəkildə paylaşılaraq tranzaksiyalar üçün istifadə olunan açardır.
69. **Pump and dump** – token qiymətinin süni şəkildə artırılıb sonradan sürətlə satılması strategiyası
70. **Reentrancy attack** – ağıllı müqavilələrdə icra zamanı funksiyanı təkrar çağırmaqla vəsait oğurluğuna səbəb olan hücum

71. **Relay chain** – Polkadot kimi multizəncir platformalarda fərqli zəncirləri birləşdirən əsas şəbəkə
72. **Replay attack** – əvvəlki bir tranzaksiyanının icazəsiz şəkildə başqa bir şəbəkədə təkrar icrası
73. **Rollup** – tranzaksiyaların blokçeyn xaricində toplanıb blokçeyndə yığcam formada qeydə alınması üsulu
74. **Satoshi** – 1 BTC-nin yüz milyonuncu (0.00000001) hissəsi; ən kiçik Bitcoin vahidi
75. **Security token** – ənənəvi maliyyə aktivlərini təmsil edən və tənzimlənən kriptovalyuta
76. **Sharding** – blokçeyn məlumat bazasının hissələrə bölünərək emal edilməsi üsulu
77. **Sidechain** – əsas blokçeynə paralel işləyən, onunla etibarlı şəkildə bağlı olan şəbəkə
78. **Slashing** – staking zamanı şəbəkə qaydalarını pozan validatorların cəzalandırılması prosesi
79. **SPV** (*Simplified Payment Verification*) – ödənişlərin sadələşdirilmiş verifikasiyası – Bitkoin protokolunun xüsusiyyəti, bütün blokçeyni yükləmədən tranzaksiyanı təsdiqləməyə imkan verir.
80. **Stablecoin** – qiyməti sabit qalmaq üçün fiat və ya digər aktivlərə bağlanmış kriptovalyuta
81. **State channel** – Blokçeyn xaricində tranzaksiyaların aparıldığı və yalnız son nəticənin zəncirdə qeyd edildiyi Layer 2 həlli
82. **Tangle** – DAG (Directed Acyclic Graph) strukturuna əsaslanan, IOTA tərəfindən istifadə olunan blokçeynsiz sistem
83. **Tendermint** – PBFT əsaslı konsensus mexanizmi və Cosmos SDK-nin əsas hissəsi
84. **Testnet** – əsas şəbəkəyə çıxmazdan əvvəl yeni tətbiqlərin sınaqdan keçirildiyi blokçeyn şəbəkəsi
85. **Token** – əsas blokçeyn üzərində yaradılan, müxtəlif istifadə məqsədləri olan rəqəmsal aktiv
86. **Tokenomics** – tokenlərin yaradılması, paylanması, alqısı və satqısı ilə bağlı məsələləri əhatə edən iqtisadi modeldir.
87. **Transaction fee** – blokçeyn şəbəkəsində tranzaksiya həyata keçirmək üçün ödənilən haqq
88. **Trustless** – tranzaksiyaların üçüncü tərəfə etibar etmədən icra edilə bilmə xüsusiyyəti

89. **Turing complete** – ağıllı müqavilələrin istənilən məntiqi əməliyyatı icra edə biləcək funksionallığa sahib olması
90. **Uniswap** – Ethereum üzərində qurulmuş mərkəziyətsiz mübadilə platforması
91. **Universal wallet** – çoxsaylı blokçeyn şəbəkələrini dəstəkləyən pulqabı növü
92. **UTXO** (*Unspent transaction output*) – xərclənməmiş tranzaksiya çıxışı – Bitkoində hesab və balans anlayışları yoxdur; yalnız UTXO anlayışı istifadə edilir.
93. **Utility token** – müəyyən bir ekosistem daxilində xidmət və ya funksionallıq təmin edən token növü
94. **Validator** – PoS şəbəkələrində tranzaksiyaları yoxlayan və bloklar əlavə edən iştirakçı
95. **Wallet** – kriptovalyutaları saxlamaq, göndərmək və qəbul etmək üçün istifadə edilən proqram və ya cihaz
96. **Web3** – İnternetin mərkəziyətsiz və istifadəçi yönümlü növbəti mərhələsini ifadə edən anlayış
97. **Whitepaper** – kriptovalyuta və ya blokçeyn layihəsinin texniki və iqtisadi əsaslarını izah edən sənəd
98. **Wrapped token** – bir blokçeyndə başqa bir blokçeynə aid aktivin tokenləşdirilmiş versiyası (məsələn, WBTC)
99. **Yield farming** – DeFi platformalarında aktivlərin istifadəyə verilməsi müqabilində gəlir əldə etmə üsulu
100. **ZKP** (*Zero Knowledge Proof*) – sirri açıqlamadan onu bildiyini isbatlamağa imkan verən kriptoqrafiya üsulu
101. **Zk-Rollup** – ZKP-əsaslı və Ethereum üzərində miqyaslanmanı təmin edən Layer 2 texnologiyası

Əlavə 4. Blokçeynlər – laboratoriya işləri

Aşağıdakı laboratoriya işlərini yerinə yetirərkən onlayn resurslardan istifadə etmək tövsiyə edilir. Qeyd etmək zəruridir ki, bu laboratoriya işlərinin hazırlanması zamanı ChatGPT istifadə edilmişdir.

Lab 1. ERC-20 standartına uyğun token yaradılması

Tapşırıqlar: Ethereum üzərində rəqəmsal token yaradın. Tokenləri Remix-də deploy edin və Metamask-la test edin.

Lab 2: Remix-də sadə ağıllı müqavilə yazmaq və yayımlamaq

Tapşırıqlar:

- Solidity-də `uint` tipli dəyişən saxlayan müqavilə (`Storage.sol`) yazın.
- `set()` və `get()` funksiyalarını əlavə edin.
- Web3 Injected Provider ilə testnet (Goerli) üzərində yayımlayın

Lab 3. İki pulqabı arasında ETH transferinin həyata keçirilməsi

Tapşırıqlar:

- MetaMask pulqabıları yaradın.
- Faucet-dən test ETH alın (etherscan.io).
- Digər pulqabına ETH göndərin və əməliyyatı testnet-də izləyin.

Lab 4. NFT kolleksiyası yaratmaq və mint etmək (OpenZeppelin ilə)

Tapşırıqlar:

- ERC-721 tokeni yaradın (`MyNFT`).
- `mintNFT(address recipient, string memory tokenURI)` funksiyasını əlavə edin.
- Bir və ya daha çox NFT mint edin və `tokenURI`-yə şəkil əlavə edin (IPFS və ya Pinata.io).

Lab 5. NFT-lərin satış və alışı üçün sadə market yerinin simulyasiyası

Tapşırıqlar:

- `NFTMarketplace.sol` yazın.
- NFT-ləri siyahıya salmaq, almaq və silmək funksiyaları əlavə edin.
- Ən azı bir NFT ilə əməliyyatı sınaqdan keçirin.

Lab 6. NFT metadata-sının JSON formatında saxlanması

Tapşırıqlar:

- Aşağıdakı formatda JSON faylı yaradın:

```
json
```

```
{
  "name": "Lab NFT #6",
  "description": "NFT laboratoriyası üçün test",
  "image": "ipfs://<CID>"
}
```

- Faylı IPFS-də yerləşdirin və `tokenURI` kimi istifadə edin.
- OpenSea testnet platformasında yoxlayın.

Lab 7. Universiteti bitirənlərə NFT şəklində diplom verilməsi
Tapşırıqlar:

- ERC-721 müqaviləsi yazın (`EduCertNFT.sol`).
- Hər NFT-də tələbə adı, ixtisası, bitirmə tarixi metadata şəklində göstərilsin.
- Müəllim yalnız özü NFT mint edə bilsin (girişə nəzarət).
- QR kod və ya “block explorer” linki ilə yoxlama təmin edin.

Əlavə 5. Sərbəst işlərin mövzuları

1. Əvəzetmə şifrinin kriptoanalizi
2. Yerdəyişmə şifrinin kriptoanalizi
3. Affin şifrinin kriptoanalizi
4. Vigener şifrinin kriptoanalizi
5. ADFGVX şifrinin kriptoanalizi
6. Hill şifrinin kriptoanalizi
7. Playfair şifrinin kriptoanalizi
8. Enigma-nın kriptoanalizi
9. Homofon şifrlər və onların kriptoanalizi
10. Feystel şəbəkəsi və onun növləri
11. SP-şəbəkələr və onların növləri
12. AES-in riyazi əsasları
13. SQUARE arxitekturası
14. Sadə və genişlənmiş sonlu meydanlar
15. Mak-Elis şifrləmə sistemi
16. Sadə ədədlərin generasiyası – Miller-Rabin testi
17. Sadə ədədlərin generasiyası – Solovey-Ştrassen testi
18. Sadə ədədlərin generasiyası – Ferma testi
19. Ədədin vuruqlara ayrılması – Pollardın (p-1) metodu
20. Ədədin vuruqlara ayrılması – Ədədi meydan xəlbiri metodu
21. Ədədin vuruqlara ayrılması – Ferma metodu
22. Diskret loqarifm məsələsi – Pollardın ro alqoritmi
23. Diskret loqarifm məsələsi – Poliq-Hellman alqoritmi
24. Diskret loqarifm məsələsi – İndeks hesabı alqoritmi
25. Elliptik əyri üzərində diskret loqarifm məsələsi
26. MD5 heş funksiyası
27. SHA-1 heş funksiyası
28. SHA-2 heş funksiyası
29. SHA-3 heş funksiyası
30. Whirlpool heş funksiyası
31. HMAC mexanizmi ilə autentifikasiya
32. Göyqurşağı (Rainbow) cədvəlləri (heş funksiyalar)
33. RC4 şifri
34. SEAL şifri
35. CHAMELEON şifri
36. SNOW şifri
37. WAKE şifri

38. SOBER şifri
39. LILI-128 şifri
40. ChaCha20 şifri
41. Bitcoin-də blokçeynin strukturu
42. DES-in variantları (Double DES, Triple DES, S-DES)
43. DES-ə əsaslanan şifrlər (G-DES, n-DES, DES-X, Ladder-DES, NewDES və s.)
44. AES finalçısı – Rijndael
45. AES finalçısı – Twofish:
46. AES finalçısı – RC6
47. AES finalçısı – MARS
48. AES finalçısı – Serpent
49. RC2 şifri
50. RC5 şifri
51. IDEA şifri
52. SAFER şifri
53. DEAL şifri
54. Skipjack şifri
55. Blowfish şifri
56. Bitcoin-də Şnorr imza sxemi
57. Bitcoin-də Taproot softforku
58. Blokçeynlərdə konsensus protokolları
59. Blokçeynlərdə ağıllı müqavilələr
60. Ağıllı müqavilələrin kriptoqrafik aspektləri
61. Kriptovalyutalarda ZK-SNARK protokolu
62. Kriptovalyutalarda ZK-STARK protokolu
63. Kriptovalyutalarda Bulletproofs protokolu
64. Xüsusi imza sxemləri – Qrup imzası
65. Xüsusi imza sxemləri – Kölgəli imza sxemi
66. CrypTool alətinin analizi
67. Java-da kriptoqrafiya kitabxanaları
68. Python-da kriptoqrafiya kitabxanaları
69. OpenSSL kitabxanası
70. Open Quantum Safe (liboqs) kitabxanası
71. OpenPGP ilə e-poçt təhlükəsizliyi
72. Kriptoqrafik kitabxanaların müqayisəli analizi
73. PKCS (Public Key Cryptography Standards) standartları
74. WhatsApp-da ucdan-uca şifrləmə

75. TLS protokolunun analizi
76. HTTPS protokolunun analizi
77. Kerberos protokolunun analizi
78. PKI (Public Key Infrastructure*)* strukturu və protokolları
79. PBKDF2 standartı
80. Yan kanal hücumları və müdafiə mexanizmləri
81. Homomorfik şifrləmə və onun növləri
82. Kriptoqrafik protokolların formal verifikasiyası
83. Ehtimallı kriptoqrafiya
84. Kvant kriptoqrafiyası: Mövcud alqoritmlər
85. Post-kvant kriptoqrafiyası: Mövcud yanaşmalar
86. Sezar şifri ilə şifrləmə/deşifrləmə üçün veb səhifə yaradılması
87. Affin şifri ilə şifrləmə/deşifrləmə üçün veb səhifə yaradılması
88. Hill şifri ilə şifrləmə/deşifrləmə üçün veb səhifə yaradılması
89. Vijener şifri ilə şifrləmə/deşifrləmə üçün veb səhifə yaradılması
90. Playfair şifri ilə şifrləmə/deşifrləmə üçün veb səhifə yaradılması
91. Üst-üstə düşmə indeksini hesablamaq üçün veb səhifə yaradılması
92. Şifrmətnin tezlik analizi üçün veb səhifə yaradılması
93. Kasiski testi ilə Vijener şifrinin açar uzunluğunun tapılması – veb-səhifə yaradılması
94. DES şifrləmə/deşifrləmə üçün veb səhifə yaradılması
95. AES şifrləmə/deşifrləmə üçün veb səhifə yaradılması
96. Heş funksiyaları reallaşdıran veb səhifə yaradılması
97. RSA açarların generasiyası və şifrləmə üçün veb səhifə yaradılması
98. Diskret loqarifm məsələsinin həlli üçün veb səhifə yaradılması
99. Böyük sadə ədədlərin generasiyası üçün veb səhifə
100. Qrafik interfeysli (GUI) klassik şifrlərə aid alətin yaradılması
101. Klassik şifrlərə aid CTF (Capture The Flag) tapşırıqların yaradılması

İstifadə olunmuş ədəbiyyat

1. Əliquliyev R.M., İmamverdiyev Y.N. Kriptoqrafiyanın əsasları. Bakı: İnformasiya texnologiyaları, 2006, 698 s.
2. Əliquliyev R.M., İmamverdiyev Y.N. Kriptoqrafiya tarixi. Bakı: İnformasiya texnologiyaları, 2006, 192 s.
3. Akif Orucəliyev. Kriptoqrafiyanın əsasları. Dərslik. Bakı: DTX-nin Heydər Əliyev adına Akademiyasının nəşriyyatı, 2019, 454 s.
4. Paar Ch., Pelzl J. Understanding Cryptography, A Textbook for Students and Practitioners. Springer, 2010, 372 p.
5. Stallings W. Cryptography and Network Security: Principles and Practice (6th Edition). Pearson, 2013, 752 p.
6. Narayanan A., Bonneau J., Felten E., Miller A., Goldfeder S., Bitcoin and cryptocurrency technologies: A comprehensive introduction. Princeton University Press, 2016, 336 p.
7. Bashir I., Mastering Blockchain: Inner workings of blockchain, from cryptography and decentralized identities, to DeFi, NFTs and Web3. 4th ed., Packt Publishing, 2023, 818 p.