Latvijas Universitāte
Fizikas un matemātikas fakultāte
Matemātiskās analīzes katedra

Jānis Buls

KLASISKĀ KRIPTOGRĀFIJA

Lekciju konspekts — 2007

Ievads

Kursa mērķis — sniegt matemātiskos pamatus, kas nepieciešami klasiskās kriptogrāfijas apgūšanai, kā arī izklāstīt klasiskās kriptogrāfijas pamatus. Kriptoloģijas matemātiskie aspekti būtiski balstās gan uz varbūtību teorijas, gan skaitļu teorijas un algebras, gan algoritmu teorijas atziņām. Taču saprotams, ka šai zinātnei ir savi specifiskie uzdevumi un mērķi. Un pirmkārt, tai ir sava specifiska terminoloģija un jēdzienu sistēma, kuras apgūšanai arī veltīts šis kurss.

Kriptoloģijas nosaukums cēlies no grieķu valodas. Tas ir saliktenis, kura pamatā ir divi grieķu vārdi: "cryptos" — slepens, un "logos" — vārds.

Mūsdienu izpratnē kriptoloģijas priekšmets ir informācijas pārveidojumi (attēlojumi), kas lietojami tās aizsardzībai no nesankcionētām ļaundara darbībām. Kriptoloģija no pašiem pirmsākumiem ir veidojusies kā duāla zinātne, kuras sastāvdaļas ir bijušas kriptogrāfija un kriptogrāfiskā analīze (īsāk — kriptoanalīze). Tā ir nepārtraukta cīņa (sacensība) starp kriptogrāfiem un kriptoanalītiķiem.

Kriptogrāfija — zinātne par informācijas attēlojumu veidošanu, kurus iespējams lietot tās aizsardzībai (tādus informācijas pārveidojumus sauksim par *kriptogrāfiskiem*), bet *kriptoanalīze* — zinātne par kriptogrāfisku attēlojumu analīzes metodēm ar mērķi atklāt aizsargājamo informāciju.

Līdz pat XX gadsimta 80. gadiem pat matemātiķi izvairījās nodarboties ar kriptoloģiju, jo tas nozīmēja, ka jūs ielaužaties aizliegtajā zonā, kurā saimniekoja galvenokārt militāristi. Sakarā ar e-pārvaldes un e-komercijas ieviešanu aktualizējas datu aizsardzības problēmas. Tā rezultātā kriptoloģija mūsdienās vairs nav tikai diplomātu un militāristu interešu lokā, kā tas tradicionāli ir bijis gadu simtiem.

Apzīmējumi

$\neg$ — negācija,
$\vee$ — disjunkcija, $\wedge$ — konjunkcija,
$\Rightarrow$ — implikācija, $\Leftrightarrow$ — ekvivalence,
$\mathfrak{A} \sim a$ — izteikums $\mathfrak{A}$ ir aplams,
$\mathfrak{A} \sim p$ — izteikums $\mathfrak{A}$ ir patiess,
$\exists$ — eksistences kvantors, $\forall$ — universālkvantors,
$\exists! x\ P(x)$ — eksistē viens vienīgs tāds x, kam izpildās nosacījums $P(x)$,

$x \in X$ — elements x pieder kopai X jeb x ir kopas X elements,
$A \subseteq B$ — kopa A ir kopas B apakškopa,
$A \cup B$, $A \cap B$, $A \setminus B$ — kopu A un B apvienojums, šķēlums, starpība,
$\min K$ — kopas K minimālais elements,
$\max K$ — kopas K maksimālais elements,

$\rightleftharpoons$, $\rightrightharpoons$ — vienādības saskaņā ar definīciju,
$\overline{1,n} \rightleftharpoons \{1, 2, \ldots, n\}$; $\overline{k,n} \rightleftharpoons \{k, k+1, \ldots, n\}$, te $k \leq n$,
$\mathbb{Z}$ — veselo skaitļu kopa, $\mathbb{Z}_+ \rightleftharpoons \{x \,|\, x \in \mathbb{Z} \wedge x > 0\}$,
$\mathbb{N} \rightleftharpoons \mathbb{Z}_+ \cup \{0\}$, $\mathbb{N}_- \rightleftharpoons \mathbb{Z} \setminus \mathbb{Z}_+$,
$\mathbb{P}$ — visu pirmskaitļu kopa,
$\mathbb{Q}$ — racionālo skaitļu kopa,
$\mathbb{R}$ — reālo skaitļu kopa, $\mathbb{C}$ — komplekso skaitļu kopa,
$|K|$ — kopas K apjoms,
$\aleph_0$ — kopas $\mathbb{N}$ apjoms, $\mathfrak{c}$ — reālo skaitļu kopas $\mathbb{R}$ apjoms,

$\langle x, y\rangle \rightleftharpoons (x, y) \rightleftharpoons \{\{x\}, \{x, y\}\}$,
$x_1 x_2 \ldots x_n \rightleftharpoons \langle x_1, x_2, \ldots, x_n\rangle \rightleftharpoons (x_1, x_2, \ldots, x_n) \rightleftharpoons ((x_1, x_2, \ldots, x_{n-1}), x_n)$,
$A_1 \times A_2 \times \ldots \times A_n \rightleftharpoons \{(x_1, x_2, \ldots, x_n) \,|\, \forall i \in \overline{1,n}\, (x_i \in A_i)\}$, A^n, $|u|$,
$f : x \mapsto y$, $f : X \multimap Y$, $X \overset{f}{\multimap} Y$,
$\mathrm{Dom}(f) \rightleftharpoons \{x \,|\, \exists y \in Y\, (f : x \mapsto y)\}$, $\mathrm{Ran}(f) \rightleftharpoons \{y \,|\, \exists x \in X\, (f : x \mapsto y)\}$,
$f : X \to Y$, $X \overset{f}{\to} Y$, $f : X \twoheadrightarrow Y$, $f : X \hookrightarrow Y$,
$\mathrm{pr}_i \varrho$, $\mathrm{pr}_i g$,
A^+, λ, A^*, u^n,

$$\sum_{i=k}^{m} a_i \rightleftharpoons a_k + a_{k+1} + \ldots + a_m,$$

$\prod_{i=k}^{m} a_i \rightleftharpoons a_k a_{k+1} \dots a_m,$

$a \diagdown b$ — skaitlis b ir skaitļa a daudzkārtnis,
$D(a_1, a_2, \dots, a_n) \rightleftharpoons \{q \,|\, \forall i \in \overline{1,n}\ q \diagdown a_i\},$
$\mathrm{ld}(a_1, a_2, \dots, a_n) \rightleftharpoons \max D(a_1, a_2, \dots, a_n),$
$a \equiv b (\mathrm{mod}\ m)$ — skaitļi a un b ir kongruenti pēc moduļa m,
$\mathbb{Z}_m \rightleftharpoons \{0, 1, \dots, m-1\},$
$\mathbb{Z}_m^* \rightleftharpoons \{\ a \mid \mathrm{ld}(a, m) = 1\ \wedge\ a \in \mathbb{Z}_m\},$

$\square$ — pierādījuma sākums,
$\blacksquare$ — pierādījuma beigas;
$\Rightarrow$ — implikācijas zīmi pierādījuma sākumā mēs izmantojam, lai norādītu, ka tagad sākas teorēmas nepieciešamā nosacījuma pierādījums,
$\Leftarrow$ — šo zīmi pierādījumos mēs izmantojam, lai norādītu, ka tagad sākas teorēmas pietiekamā nosacījuma pierādījums.

1. Pamatjēdzieni un kriptogrāfijas uzdevumi

Pamatjēdzieni un kriptogrāfijas uzdevumi. Kriptosistēma, šifrēšana, dešifrēšana, kriptoanalītisks uzbrukums. Simetriskas un asimetriskas kriptosistēmas. Kriptogrāfisks protokols. Drošu sakaru organizēšana. Ziņojuma autentiskuma nodrošināšana. Paraksts.

1.1. Kriptogrāfijas uzdevumi

Kriptogrāfiskos pārveidojumus izmanto sekojošu informācijas aizsardzības pamatuzdevumu risināšanai:

1. Informācijas *slepenības* nodrošināšanai, proti, tās aizsardzībai no nesankcionētas tās satura aplūkošanas.

2. Informācijas *integritātes* nodrošināšanai, proti, tās satura aizsardzībai no nesankcionētu izmaiņu veikšanas, pie kurām pieder tās sākotnējā satura papildināšana un fragmentu izņemšana vai aizvietošana.

3. Informācijas *autentifikācijai*, proti, pušu identitātes un pašas informācijas autentiskuma apliecināšanai, informācijas izveidošanas laika apliecināšanai, un tamlīdzīgi.

4. Lai nodrošinātu informācijas *izcelsmes neviltojamību*, proti, lai būtu iespējams pierādīt, kas ir informācijas autors.

Ja pirmais uzdevums ir bijis kriptogrāfijai tradicionāls vairāku tūkstošu gadu garumā, tad pārējie uzdevumi ir nonākuši kriptogrāfijas interešu lokā tikai 20. gs. saistībā ar elektronisko informācijas tehnoloģiju attīstību. Kriptogrāfiskos attēlojumus sāka izmantot elektronisko dokumentu apmaiņā, lai aizsargātu elektroniskos maksājumus, komercdarījumus un citām vajadzībām.

Aplūkosim kriptoloģijas jēdzienu sistēmu saistībā ar informācijas aizsardzības uzdevumiem.

1.2. Kriptoloģijas pamatjēdzieni

Par *šifru* sauc invertējamu kriptogrāfisku informācijas attēlojumu kopu E. Ar katru šifra attēlojumu saista kaut kāda parametra k vērtību, ko sauc par *atslēgu*, proti,

$$E = \{E_k \,|\, k \in K\},$$

kur K — galīga atslēgu pieļaujamo vērtību kopa, ko sauc par *atslēgu kopu*. Izvēlēta atslēga k viennozīmīgi nosaka šifra E attēlojumu E_k. Praktiska atslēgas izmantošana paredz tā sauktā atslēgas *dzīves cikla* realizāciju, proti, tādu darbību veikšanu kā atslēgas ģenerēšanu, tās izplatīšanu, uzglabāšanu, lietošanu, veidojot šifra attēlojumus, nomaiņu un iznīcināšanu. Informāciju, kas tiek pārveidota izmantojot šifra attēlojumu, sauc par *pamattekstu* (plaintext). Šifra attēlojuma pielietošanu pamattekstam sauc par *šifrēšanu* (encryption). Pamatteksta šifrēšanas rezultātu sauc par *šifrētu tekstu* (cyphertext) jeb *kriptotekstu*, *kriptogrammu*. Ja ar $\mathcal{P}$ un $\mathcal{C}$ apzīmējam attiecīgi pamattekstu un šifrēto tekstu kopas, tad šifru var aplūkot kā attēlojumu $\mathcal{E}$ no kopas $\mathcal{P} \times K$ kopā $\mathcal{C}$. Šifra attēlojumu apgriežamība nodrošina iespēju atjaunot pamattekstu no šifrētā teksta. Inversā attēlojuma pielietošanu kriptogrammai sauc par *atšifrēšanu* jeb *dešifrēšanu*. Atšifrēšanas rezultātā notiek kopas $\mathcal{C} \times K$ attēlošana kopā $\mathcal{P}$. Formāli to visu definē šādi.

Definīcija 1.1. *Kortežu*

$$\langle \mathcal{P}, \mathcal{C}, K, \mathcal{E}, \mathcal{D} \rangle$$

sauc par kriptosistēmu, ja

- $\mathcal{P}$ — *pamattekstu kopa;*
- $\mathcal{C}$ — *kriptotekstu kopa;*
- K — *atslēgu kopa;*
- $\mathcal{E} : \mathcal{P} \times K \to \mathcal{C}$ — *šifrs;*
- $\mathcal{D} : \mathcal{C} \times K \to \mathcal{P}$ — *dešifrējošais attēlojums*

un katram pamattekstam $x \in \mathcal{P}$, katrai atslēgai $k \in K$ ir spēkā vienādība

$$\mathcal{D}(\mathcal{E}(x, k), k) = x\,. \tag{1}$$

Tā rezultātā katram $k \in K$ atbilst attēlojums $E_k : \mathcal{P} \to \mathcal{C}$, kas definēts ar vienādību $E_k(x) \rightleftharpoons \mathcal{E}(x, k)$. Savukārt katram $k \in K$ atbilst inversais attēlojums $E_k^{-1} : \mathcal{C} \to \mathcal{P}$, kas definēts ar vienādību $E_k^{-1}(x) \rightleftharpoons \mathcal{D}(x, k)$.

No abstraktās algebras redzes viedokļa $\mathcal{P}$, $\mathcal{C}$ un K ir patvaļīgas kopas, parasti gan pieņem, ka tās ir galīgas kopas. Savukārt

$$\mathcal{E} : \mathcal{P} \times K \to \mathcal{C} \qquad \text{un} \qquad \mathcal{D} : \mathcal{C} \times K \to \mathcal{P}$$

ir patvaļīgi attēlojumi, kuriem ir spēkā vienādība (1). Līdz ar to no abstraktās algebras redzes viedokļa kriptosistēma ir trīs sugu algebra.

Šifra lietošana tā vai cita kriptogrāfijas uzdevuma atrisināšanai paredz attiecīgu darbojošos personu iesaisti (piemēram, abonentus, kas lieto slepenus sakarus) un noteiktu to mijiedarbības kārtību, ko sauc par *kriptogrāfisku protokolu*. Plašākā nozīmē kortežu $\langle \mathcal{P}, \mathcal{C}, K, \mathcal{E}, \mathcal{D} \rangle$ kopā ar lietotajiem protokoliem sauc par *kriptosistēmu* jeb *šifru sistēmu*. Parasti jau no konteksta ir skaidrs kādā nozīmē lietots termins kriptosistēma.

Atslēgu kopa un *atslēgu protokoli*, (t.i., protokoli, kas vada atslēgu dzīves ciklu) veido *šifra atslēgu sistēmu*. Šifrētas informācijas atklāšanu ar kriptoanalīzes palīdzību sauc par *dešifrēšanu* (pie tam kriptoanalītiķim nav pieejama atslēga atšifrēšanai, proti, nav zināms kādu attēlojumu tieši jāizmanto atšifrēšanai). Kriptoanalītiķa izstrādātā metode šifra vai šifrētās informācijas atklāšanai tiek saukta par *kriptoanalītisku uzbrukumu*. Kriptosistēmas spēju turēties pretī kirptoanalītiķa uzbrukumiem sauc par *kriptosistēmas kriptogrāfisko drošību*. Droša kriptosistēma nodrošina informācijas aizsardzību ilgā laika posmā, neskatoties uz pretinieka pūlēm, kuram ir pieejami nozīmīgi materiāli, intelektuāli un skaitļošanas resursi. Tas nozīmē, ka drošai kriptosistēmai ir jābūt balstītai uz šifru, kas sastāv no liela daudzuma dažādu attēlojumu, pretējā gadījumā slepeno informāciju iespējams atklāt ar iespējamo attēlojumu pilno pārlasi. Bez tam, drošai kriptosistēmai jābūt veidotai tā, lai visu kriptogrammu dešifrēšana (vai gandrīz visu) būtu darbietilpīgs uzdevums, kuru nebūtu iespējams atrisināt pat izmantojot pašas modernākās tehnoloģijas praktiski pieņemamā laika periodā.

1.3. Simetriskās un asimetriskās kriptosistēmas

Kriptosistēmas tiek iedalītas pēc to darbības principa sistēmās ar *slepenu atslēgu* vai *publisku atslēgu*.

Sistēmas ar slepenu atslēgu tiek lietotas jau vairākus gadu tūkstošus un ir balstītas uz klasisko informācijas slepenības nodrošināšanas principu: *lietotās atslēgas slepenību* no visiem izņemot personas, kas tiek pielaistas informācijai. Šādas sistēmas mēdz saukt arī par *simetriskām*, jo atslēgas kas tiek izmantotas gan informācijas šifrēšanai gan atšifrēšanai ir zināmā nozīmē simetriskas (tās bieži sakrīt).

Informācijas aizsardzība simetriskās sistēmās tiek nodrošināta ar atslēgas slepenību. Sistēmas ar publisko atslēgu 1975. gadā piedāvāja Diffijs un

Hellmans, un tās jau tiek aktīvi lietotas. Šīm sistēmām tiek lietots arī cits apzīmējums - *asimetriskas sistēmas*, jo tajās šifrēšanas un atšifrēšanas atslēgas nav saistītas ar simetrijas vai vienādības attiecību. Tāpēc šifrēšanas atslēga var būt *publiska* un zināma visiem, taču atšifrēt ziņojumu var tikai slepenās atšifrēšanas atslēgas turētājs. Lai izvairītos no neskaidrībām atšifrēšanas atslēgu simetrisko sistēmu gadījumā parasti sauc par *privāto atslēgu*.

Publiskās kriptosistēmas atšifrēšanas atslēgas noskaidrošana pēc šifrēšanas atslēgas, proti, šifra atklāšana, ir saistīta ar ļoti sarežģītu matemātisku uzdevumu atrisināšanu. Tādu uzdevumu skaitā figurē, piemēram, lielu naturālu skaitļu dalītāju atrašana un lielas kārtas galīga lauka elementu logaritmu noteikšana.

Vispāratzīta simetrisko sistēmu priekšrocība ir ātrāka šifrēšana, mazāki izmantojamās atslēgas izmēri un pamatotākas kriptogrāfijas drošības garantijas. No otras puses, asimetriskās kriptosistēmas lieto ērtākus protokolus, konkrēti tādu svarīgu uzdevumu risināšanai kā informācijas autentifikācijai un atslēgu izplatīšanai starp lietotājiem. Tādēļ informācijas aizsardzībai var izmantot *hibrīdas kriptosistēmas*, kurās izmanto gan simetrisko, gan asimetrisko sistēmu principus. Konkrētas rekomendācijas hibrīdu kriptosistēmu veidošanai aplūkosim, kad apskatīsim kriptogrāfiskos protokolus.

1.4. Kriptogrāfisku protokolu jēdziens

Fiksētu noteikto darbību kārtību, kas jāveic lai atrisinātu kādu kriptogrāfijas uzdevumu, sauc par *kriptogrāfisku protokolu*. Kriptogrāfiskie protokoli ir svarīga kriptogrāfiskās sistēmas sastāvdaļa. Ir iespējamas situācijas, kurās informācijas drošības nodrošināšanas uzdevumi netiek atrisināti nepilnību dēļ protokolā, neskatoties uz to, ka tiek izmantoti piemēroti kriptogrāfiskie attēlojumi.

Protokola pamatatšķirība no algoritma ir tā, ka algoritma realizācija paredz viena subjekta aktīvas darbības, kamēr protokols tiek realizēts caur vairāku subjektu (protokola iesaistītās personas) mijiedarbību. Katra kriptogrāfiskā protokola paredzētā darbība pēc satura ir vai nu skaitļošanas operācijas, kuras veic protokola subjekti, vai ziņojumu nosūtīšana starp tiem. Ja iesaistītās puses uzticas viena otrai un ir gatavas kopīgi risināt kriptogrāfisko uzdevumu, tad šādā gadījumā tiek izmantoti protokoli *bez starpnieka*, kurus sauc par *divpusējiem protokoliem*. Ja starp pusēm var rasties domstarpības vai tām nepieciešama trešās puses palīdzība, tad tiek izmantoti protokoli

ar starpnieku (neieinteresētu trešo pusi), kurus sauc par *trīspusējiem protokoliem*. Starpnieka uzdevums ir nodrošināt visu protokola soļu izpildi līdz pat transakcijas beigām.

Izgudroti arī *protokoli ar arbitru*, kur ar arbitru tiek saprasts īpaša tipa starpnieks: viņš ne obligāti piedalās katra protokola realizācijā, taču tiek pieaicināts tikai, lai pārbaudītu protokola izpildes korektumu. Vispievilcīgākais protokolu veids ir *pašpietiekamie* protokoli, kuru konstrukcija nodrošina pareizu protokola izpildi. Diemžēl šādi protokoli daudziem uzdevumiem nav pieejami.

Uzbrukumi protokoliem no pretinieka puses var būt virzīti gan kā pret kriptogrāfiskajiem algoritmiem, kas tiek izmantoti protokolos, gan pret pašiem protokoliem. Šādus uzbrukumus nosacīti iedala *pasīvos* un *aktīvos*. Pasīvā uzbrukuma gadījumā pretinieks tikai novēro protokola pušu darbības un cenšas no novērojumiem iegūt noderīgu informāciju, neiejaucoties protokola realizācijā. Aktīvā uzbrukuma gadījumā pretinieks ievieš protokola izmaiņas savās interesēs, kas var izpausties kā jaunu ziņojumu iefiltrēšana protokolā, "likumīgu" ziņojumu izņemšana no protokola, vienu ziņojumu aizvietošana ar citiem, saziņas kanāla vai atmiņas, kurā glabājas informācija, izmaiņas. Uzbrucējs var būt ne tikai trešā puse, bet arī protokola dalībnieks, pie tam pretinieks var būt arī persona grupā, kas var savā starpā komunicēt.

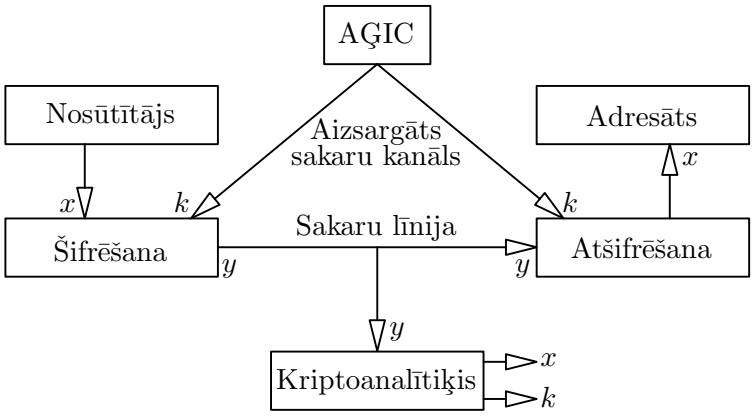
1.5. Slepenu sakaru organizēšana, kriptoanalītiķa uzdevumi

Sākumā aplūkosim slepenu sakaru organizēšanu izmantojot simetrisku kriptosistēmu. Protokolā iesaistītās personas ir ziņojuma sūtītājs (Alise), adresāts (Bucis), pasīvs uzbrucējs (Oskars), aktīvs uzbrucējs (Uģis) un citi. Protokola uzdevums ir nodot Alises ziņojumu x adresātam Bucim. Darbību secība ir sekojoša:

1. Alise ar Buci vienojas, kādu simetrisku kriptosistēmu $\langle \mathcal{P}, \mathcal{C}, K, \mathcal{E}, \mathcal{D} \rangle$ izmantot. Praksē tas parasti reducējas uz vienošanos par šifru $E = \{E_k \mid k \in K\}$.

2. Alise ar Buci vienojas par sakaru slepeno atslēgu k, proti, kuru attēlojumu $E_k \in E$ viņi izmantos.

3. Alise šifrē pamattekstu x ar attēlojuma E_k palīdzību, tādā veidā izveidojot kriptogrammu $y = E_k(x)$.



1. zīm.: Slepenu sakaru shēma starp diviem abonentiem izmantojot simetrisku kriptosistēmu.

4. Kriptogramma y tiek nosūtīta pa sakaru līniju adresātam Bucim.

5. Bucis atšifrē kriptogrammu y, izmantojot to pašu atslēgu k un attēlojuma E_k inverso attēlojumu E_k^{-1}, un nolasa ziņojumu x:

$$x = E_k^{-1}(y).$$

Precizēsim dažas svarīgas aplūkotā protokola īpašības ar shēmas palīdzību (1. zīm.). Protokola otrais solis tiek realizēts vai nu ar starpnieka (trešās puses) palīdzību, kuru nosacīti var nosaukt par *atslēgu ģenerēšanas un izplatīšanas centru* (AĢIC), vai arī abonenti paši pilda AĢIC funkcijas. Pirmajā gadījumā šifrēto sakaru protokolu sauc par *trīspusēju*, bet otrajā gadījumā — par *divpusēju*.

Nozīmīga protokola īpašība ir atslēgas k *slepenība*, kurš nosūtītājam (Alisei) un adresātam (Bucim) tiek nodots vai nu atklātā veidā pa sakaru kanālu, kas ir aizsargāts no uzbrucēja darbībām, vai arī šifrētā veidā pa sakaru līniju. Aizsargātais sakaru kanāls var būt salīdzinoši lēns, taču tam ir jāspēj nodrošināt uzticamu atslēgas aizsardzību no nesankcionētas piekļuves.

Atslēgai ir jāpaliek slepenai gan protokola realizācijas laikā, gan arī pirms un pēc tam, jo pretējā gadījumā uzbrucējs, ieguvis atslēgu, var atšifrēt kriptogrammu un izlasīt ziņojumu. Alise un Bucis pirmo protokola soli var veikt publiski (kriptosistēmas slepenība nav obligāta), bet otro soli viņiem jāizpilda slepeni (atslēgas slepenība ir obligāta). Šāda nepieciešamība skaidrojama ar

to, ka sakaru līnijas, it īpaši garas sakaru līnijas, ir nedrošas gan attiecībā pret pasīvo, gan aktīvo uzbrucēju.

Pasīvs uzbrucējs (kriptoanalītiķis), vēloties iegūt ziņojumu x, kontrolē sakaru līniju protokola 4. solī. Neiejaucoties protokola realizācijā, viņš pārtver kriptogrammu y, lai varētu atklāt šifru.

Izstrādājot kriptosistēmu, kriptogrāfs parasti pieņem, ka kriptoanalītiķim piemīt sekojošas īpašības:

1. kriptoanalītiķis kontrolē sakaru līniju;
2. kriptoanalītiķim ir zināmas šifra E īpašības;
3. kriptoanalītiķis nezin atslēgu k, proti nezin attēlojumu E_k, kas ir ticis izmantots, lai iegūtu kriptogrammu y.

Šādos apstākļos kriptoanalītiķis (Oskars) cenšas atrisināt sekojošus uzdevumus, kurus sauc par *dešifrēšanas uzdevumiem*:

1. Noteikt pamattekstu x un izmantoto atslēgu k pēc pārtvertās kriptogrammas y, proti, izveidot tādu dešifrēšanas algoritmu ψ, ka
$$\psi(y) = (x, k).$$
Šāda uzdevuma nostādne paredz, ka kriptoanalītiķis izmantos statistiskas pamatteksta īpašības. Zinātniskajā literatūrā šādu uzdevumu sauc par *uzbrukumu, izmantojot tikai kriptotekstu* (ciphertext–only attack).

2. Noteikt izmantoto atslēgu k pēc zināmiem pamattekstiem un attiecīgajiem šifrētajiem tekstiem, proti, izveidot tādu dešifrēšanas algoritmu φ, ka
$$\varphi(x, y) = k.$$
Šādai uzdevuma nostādnei ir jēga, ja kriptoanalītiķim ir izdevies pārtvert vairākas kriptogrammas, kas ir izveidotas ar atslēgu k, taču pats kriptoanalītiķis zin ne visu kriptogrammu pamattekstus. Šādā gadījumā, ja kriptoanalītiķim izdodas atrisināt otrā tipa dešifrēšanas uzdevumu, viņš varēs izlasīt visus pamattekstus, kas šifrēti izmantojot atslēgu k. Zinātniskajā literatūrā šādu uzdevumu sauc par *uzbrukumu, izmantojot zināmu pamattekstu* (known–plaintext attack).

3. Noteikt izmantoto atslēgu k pēc speciāli izvēlēta pamatteksta x un attiecīgā šifrētā teksta y, proti, izveidot tādu algoritmu φ_x, ka

$$\varphi_x(y) = k.$$

Šādai uzdevuma nostādnei ir nozīme, ja kriptoanalītiķim ir iespēja testēt kriptosistēmu, proti, iespēja ģenerēt kriptogrammas speciāli izvēlētam pamattekstam. Šāda iespēja biežāk rodas analizējot asimetriskas sistēmas. Zinātniskajā literatūrā šādu uzdevumu sauc par *uzbrukumu, izmantojot izvēlētu pamattekstu* (chosen–plaintext attack).

Eksistē šī uzdevuma paveids, kad tiek izmantots speciāli izvēlēts šifrētais teksts. Zinātniskajā literatūrā šādu uzdevumu sauc par *uzbrukumu, izmantojot izvēlētu kriptotekstu* (chose–ciphertext attack).

Pirmā tipa dešifrēšanas uzdevumi atšķiras no otrā un trešā tipa uzdevumiem ar augstāku skaitļošanas sarežģītību. Visvienkāršākie ir testēšanas tipa uzdevumi.

Aktīvs uzbrucējs (Uģis) iejaucas protokola realizācijā. Viņš var pārtraukt sakarus 4. solī, pieņemot, ka nosūtītājs vairs nevarēs neko paziņot adresātam. Tāpat viņš var pārķert ziņojumu un aizstāt to ar savu. Ja aktīvais uzbrucējs uzzinātu atslēgu (kontrolējot otro soli, vai iekļūstot kriptosistēmā), viņš varētu aizšifrēt savu ziņojumu un aizsūtīt to adresātam pārtvertā ziņojuma vietā, kas adresātam neizraisītu ne mazākās aizdomas. Ja aktīvais uzbrucējs nezin atslēgu, tad viņš var izveidot tikai tādu šifrētu ziņojumu, kas atšifrējot pārvēršas par nejaušu simbolu virkni.

Aplūkotais protokols pieņem, ka nosūtītājs, adresāts un trešā puse (AĢIC) savstarpēji uzticas. Tā ir dotā protokola vājība, jo ne vienmēr iespējams izslēgt protokolā iesaistītu personu krāpniecību. Vispār, absolūtas garantijas par tā vai cita protokola neievainojamību neeksistē, jo jebkura protokola izpilde ir saistīta ar cilvēku piedalīšanos un ir atkarīga no iesaistīto personu uzticamības. Līdz ar to, pēc sakaru organizācijas ar simetrisku kriptosistēmu, var izdarīt sekojošus secinājumus:

1. Protokolam jāaizsargā pamatteksts un atslēga no nesankcionētas piekļūšanas visos informācijas nodošanas posmos sākot no avota un beidzot ar ziņojuma saņēmēju. Atslēgas slepenība ir svarīgāka nekā konkrēto ziņojumu slepenība, kas tiek šifrēti ar šo atslēgu. Ja atslēga ir kompromitēta (nozagta, uzminēta, atklāta vai nopirkta), tad uzbrucējs, kam ir pieejama atslēga, var atšifrēt visus ziņojumus, kas ir šifrēti ar

šo atslēgu. Bez tam, uzbrucējs varēs imitēt kādu no pusēm un ģenerēt maldinošus ziņojumus ar mērķi apmānīt otru saziņas pusi. Biežas atslēgu maiņas gadījumā šī problēma tiek samazināta līdz minimumam.

2. Protokols nedrīkst pieļaut "liekas" informācijas noplūdi sakaru kanālā, kas atļautu kriptoanalītiķim vieglāk dešifrēt kriptogrammas. Protokolam jāaizsargā informāciju ne tikai no trešajām personām, bet arī no savstarpēji iesaistīto personu maldināšanas.

3. Ja pieļaujam, ka katrs sakaru tīkla lietotāju pāris lieto katrs savu atslēgu, tad n lietotāju gadījumā nepieciešamo atslēgu skaits vienāds ar $n(n-1)/2$. Tas nozīmē, ka pie lieliem n atslēgu ģenerēšana, glabāšana un izplatīšana kļūst par darbietilpīgu problēmu.

Tagad aplūkosim slepenu sakaru organizāciju izmantojot asimetrisku kriptosistēmu. Darbību secība izskatās šādi:

1. Nosūtītājs un adresāts vienojas par izmantojamo asimetrisko kriptosistēmu.

2. Adresāts nosūta nosūtītājam publisko atslēgu k.

3. Nosūtītājs šifrē pamattekstu x, izmantojot publisko atslēgu k, proti, izveido kriptogrammu $y = E_k(x)$.

4. Kriptogramma y tiek nosūtīta adresātam pa sakaru līniju.

5. Adresāts atšifrē kriptogrammu y, izmantojot privāto atslēgu z, un izlasa ziņojumu x:
$$x = E_z(y).$$

Skaidrs, ka dotais protokols aizsargā ziņojumu x: pasīvs uzbrucējs (Oskars) var pārtvert publisko atslēgu k un kriptogrammu y, bet, tā kā viņam nav privātās atslēgas k, nevar atšifrēt ziņojumu.

Protokola priekšrocība ir tā, ka tam nav nepieciešama slepenu atslēgu izplatīšana. Bieži vien visu lietotāju publiskās atslēgas tiek novietotas visiem pieejamā sakaru tīkla datu bāzē, bet privātās atslēgas glabājas pie lietotājiem. Tas vienkāršo protokolu, jo nosūtītājs pats izpilda protokola otro soli, negaidot adresāta rīcības. Adresāts neiesaistās protokolā līdz brīdim, kamēr nevēlas izlasīt saņemto ziņojumu.

Praksē publiskās atslēgas algoritmi neaizstāj simetriskos algoritmus. Kā likums, tie tiek izmantoti nevis pašu ziņojumu šifrēšanai, bet atslēgu vai citas ne pārāk garas "palīginformācijas" šifrēšanai. Tas ir saistīts ar sekojošo:

1. Publiskās atslēgas algoritmu ātrdarbība ir vismaz 1000 reižu lēnāka, nekā simetriskajiem algoritmiem. Tas nozīmē, ka tie ir slikti piemēroti lielu informācijas apjomu šifrēšanai.

2. Kriptosistēmas ar publisko atslēgu ir pakļautas uzbrukumiem ar izvēlētu pamattekstu, it īpaši, kad pamatteksta bloku skaits ir ierobežots un ir iespējama šo variantu pilna pārlase.

Šo iemeslu dēļ visracionālākais variants ir slepeno sakaru protokols, kas izmanto hibrīdu kriptosistēmu, kurā asimetriskais algoritms tiek izmantots atslēgu slepenības nodrošināšanai un to izplatīšanai, bet algoritms ar slepeno atslēgu tiek izmantots pašu ziņojumu aizsardzībai. Bez tam, šāds protokols pieļauj seansa slepeno atslēgu iznīcināšanu tūlīt pēc seansa beigšanas, kas stipri samazina to kompromitācijas risku.

1.6. Ziņojuma integritātes nodrošināšana

Viens no centrālajiem jēdzieniem publiskās atslēgas kriptogrāfijā ir *vienvirziena funkcija* (one-way function), proti, funkcija f, kuras vērtību $f(x)$ ir viegli izskaitļot jebkuram x no tās definīcijas apgabala, bet pirmtēla x atrašana pēc vērtības $f(x)$ ir darbietilpīga. Piemēram, viegli aprēķināt funkcijas a^x vērtību galīgā laukā L, ja ir dots $a \in L$, turpretim logaritmu aprēķināšana laukā L saistīta ar daudz nozīmīgākām grūtībām.

Vienvirziena funkcija ar sētas durvīm (one-way function with a trapdoor) ir atkarīga ne tikai no x, bet arī no kaut kāda parametra k. Pirmtēla aprēķināšana šādai funkcijai ir viegla tikai tad, ja zināms parametrs k.

Jaucējfunkcija (hash function) ir kopas $\bigcup_{l>m} X^l$ attēlojums kopā X^m (kur X - galīga kopa, m un l - naturāls skaitlis, l parasti daudz lielāks par m), proti, virknes ar garumu ne mazāku par m attēlojums par virkni ar garumu m. Jaucējfunkcijas vērtību sauc par *kontrolsummu* (hash).

Jaucējfunkcijas kriptogrāfijā tiek izmantotas, lai pārbaudītu vai divas virknes ir vienādas, salīdzinot to kontrolsummas. Šāda pārbaude nav pilnīgi droša, bet to iespējams veikt ar pieņemamu precizitāti.

Īpaši interesantas ir vienvirziena jaucējfunkcijas, kurām praktiski nav iespējams atjaunot sākotnējo virkni pēc kontrolsummas. Bez tam, īpaši labas

ir jaucējfunkcijas, kas ir *bez kolīzijām*, proti nepieļauj vienkāršu divu dažādu virkņu konstruēšanu ar vienādām kontrolsummām.

Kriptogrāfijā lietotās jaucējfunkcijas realizē pilnīgu attēlojumu (tiek veikta pilnīga informācijas sajaukšana). Tas nozīmē, ka pat viena bita izmainīšana ziņojumā (sākotnējā virknē) vidēji izmaina apmēram pusi bitu kontrolsummā.

Ja nepieciešams pārbaudīt kāda faila atbilstību oriģinālam, var aprobežoties ar šī faila kontrolsummu. Ja nepieciešams, lai pārbaude būtu pieejama tikai noteiktai personai, tad tiek izmantota jaucējfunkcija, kurai pievienota slepenā atslēga, vai liels šifrs ar slepenu atslēgu. Kontrolsummas aprēķināšanai, ko šinī gadījumā sauc par *ziņojuma autentiskuma kodu* (message authentication code - MAC), ir jāzin izmantotā atslēga.

1.7. Elektroniskais paraksts

Tāpat kā tradicionālais paraksts ar roku, arī elektroniskais paraksts paredzēts informācijas autentifikācijai, kā arī lai nodrošinātu izcelsmes nevilotjamību. Paraksts uz dokumenta kopš seniem laikiem tika uzskatīts kā piekrišanas apliecinājums dokumenta saturam vai arī tā autorības pierādījums. Tiek uzskatīts, ka parakstam piemīt sekojošas īpašības:

1. Paraksts ir *neatkārtojams*, jo tas liecina par to, kura persona ir parakstījusi dokumentu.

2. Paraksts ir *īsts*, jo tas liecina, ka tieši šī persona ir parakstījusi dokumentu.

3. Paraksts *netiek izmantots atkārtoti*, jo tas ir dokumenta daļa, to nav iespējams pārvietot uz citu dokumentu.

4. Parakstītais dokuments *nav izmainām*.

5. No paraksta *nav iespējams atteikties*.

Realitātē šīs īpašības var tikt pārkāptas, kaut gan "pārkāpējs" riskē ar atmaskošanu.

Aplūkosim elektroniskā paraksta protokolu ar starpnieku, kuram abas puses pilnībā uzticas. Mēs aprakstīsim kā izmantojama simetriska kriptosistēma. Pieņemsim, ka sūtītāja A (Alise) grib parakstīt elektronisku ziņojumu un nosūtīt to adresātam B (Bucim). Izmantojot simetrisku šifru

$E = \{E_k \mid k \in K\}$, Alise to var izdarīt ar starpnieka palīdzību, kuram ir atslēga k_1 sakariem ar Alisi un atslēga k_2 sakariem ar Buci. Par šīm atslēgām iesaistītajām personām jāvienojas pirms protokola izpildes sākuma, un tās var izmantot atkārtoti vairākiem parakstiem.

Aplūkosim protokolu pa soļiem.

1. Nosūtītājs A šifrē ziņojumu a adresātam B ar atslēgu k_1 un nosūta kriptogrammu b starpniekam, kur
$$E_{k_1}(a) = b.$$

2. Starpnieks atšifrē kriptogrammu b, izmantojot atslēgu k_1:
$$E_{k_1}^{-1}(b) = a.$$

3. Starpnieks izveido informācijas bloku $I_A = [a, b, p_A]$, kas sastāv no atšifrētā ziņojuma a, kriptogrammas kopijas b un informācijas bloka p_A, kas apliecina, ka ziņojums nāk no A (Alises identifikators). Pēc tam starpnieks ar atslēgu k_2 šifrē bloku I_A un nosūta kriptogrammu c adresātam B; te
$$E_{k_2}(I_A) = c.$$

4. Bucis atšifrē kriptogrammu c, izmantojot atslēgu k_2:
$$E_{k_2}^{-1}(c) = I_A = [a, b, p_A].$$
Tagad viņš var izlasīt ziņojumu a un starpnieka sertifikātu p_A, kas apliecina, ka ziņojumu nosūtījusi Alise.

Uzskaitīsim šī protokola priekšrocības:

1. Tikai starpniekam un Alisei ir pieejama atslēga k_1, tāpēc tikai Alise varēja nosūtīt starpniekam ziņojumu, kas ir šifrēts ar atslēgu k_1 (*paraksts nav viltojams*), un tāpēc starpnieka apliecinājums ir *uzticams*. Ja aktīvs uzbrucējs (Uģis) mēģinās izlikties par Alisi, starpnieks to sapratīs otrajā solī, un nenosūtīs ziņojumu Bucim.

2. *Paraksts nav pavairojams* un *parakstītais dokuments nav izmainām*. Ja Bucis mēģinātu savienot starpnieka sertifikātu p_A ar izmainītu ziņojumu

a', tad viņam nāktos paziņot, ka ir saņēmis bloku ar saturu a' un sertifikātu p_A. Taču starpnieks šo neatbilstību var konstatēt. Starpniekam atliek tikai pieprasīt no Buča ziņojumu a' un kriptogrammu b'.

Saskaņā ar protokolu Buča rīcībā jābūt trijniekam $I'_A = [a', b', p_A]$. Šifrējot nepareizo ziņojumu a' ar atslēgu k_1, starpnieks ievēros, ka $E_{k_1}(a') \neq b'$, proti, iegūtā kriptogramma nesakrīt ar kriptogrammu b', kuru atsūtījis Bucis. Skaidrs, ka Bucis nevar uzrādīt kriptogrammu b', kas būtu saderīga ar a', jo nezin atslēgu k_1.

3. No šī paraksta *nav iespējams atteikties*. Ja Alise vēlāk paziņotu, ka nav parakstījusi ziņojumu a, kas glabājas pie Buča, tad ar starpnieka palīdzību Bucis varēs parādīt pretējo.

Šādā veidā dotais protokols novērš nepilnības, kas piemīt tradicionālajam parakstam uz papīra dokumenta.

Aplūkosim vēl vienu protokolu ar starpnieku, kura mērķis ir nodrošināt iespēju saņēmējam B demonstrēt trešajai pusei C, ka viņš ir saņēmis parakstītu dokumentu a no nosūtītāja A.

1. B izveido bloku $I = [a, p_A]$ no saņemtā ziņojuma a un starpnieka sertifikāta p_A, un šifrē to ar atslēgu k_2,

$$E_{k_2}(I) = w,$$

un nosūta kriptogrammu w starpniekam (lai tas varētu to pārsūtīt C).

2. Starpnieks šifrē bloku I ar slepeno atslēgu k_3, kas tiek izmantota sakariem ar C, un nosūta kriptogrammu v adresātam C:

$$E_{k_3}(I) = v.$$

3. C atšifrē kriptogrammu v, izmantojot k_3:

$$E^{-1}_{k_3}(v) = I = [a, p_A].$$

Tagad C var izlasīt gan ziņojumu a, gan starpnieka sertifikātu p_A, kas apliecina, ka to ir sūtījis A.

Kaut gan šie protokoli principā ir realizējami, tiem nepieciešama darbietilpīga un nekļūdīga starpnieka iesaiste. Starpniekam nākas nepārtraukti atšifrēt un

šifrēt ziņojumus starp katru abonentu pāri, kas vēlas viens otram nosūtīt parakstītu dokumentu. Starpnieks šādā sakaru sistēmā ir *šaurā vieta*, pat ja tā ir vienkārši datorprogramma.

Starpniekam ir jābūt *nevainojamam*, jo pat viena vienīga kļūda kādā no miljoniem ziņojumu sagrauj viņa uzticamību. Bez tam, starpniekam jāstrādā *pilnīgā drošībā*. Ja viņa slepeno atslēgu datu bāze kļūtu pieejama uzbrucējam vai kāds varētu izmainīt viņa atslēgas, sistēmas darbība būs sagrauta. Šis protokols ir labs drīzāk teorētiski, nekā praksē.

Aplūkosim praktiskākus elektroniskā paraksta protokolus, kas izmanto asimetriskās kriptosistēmas. Vairākās publiskās atslēgas kriptosistēmās, piemēram, El Gamal un RSA, elektroniskā paraksta izveide ir līdzvērtīga dokumenta šifrēšanai ar privāto atslēgu. Šādā gadījumā katrs var izmantot publisko atslēgu, lai atšifrētu un līdz ar to pārbaudītu paraksta autentiskumu. Šāds protokols atbilst visām paraksta prasībām un tam nav nepieciešams starpnieks. Tomēr nepieciešamas dažas protokola izmaiņas, lai novērstu nepilnības.

Lielu dokumentu parakstīšana šifrējot tos ar asimetrisku algoritmu nav efektīva sakarā ar nelielo šifrēšanas ātrumu. Tāpēc izdevīgāk ir parakstīt nevis pašu dokumentu, bet gad tā kontrolsummu, kas iegūta izmantojot vienvirziena jaucējfunkciju. Saņēmējam tiek nosūtīts dokuments un tā parakstītā kontrolsumma. Svarīgi izmantot tieši vienvirziena jaucējfunkciju, lai minimizētu īstā dokumenta aizvietošanas iespēju ar citu, kuram ir tāda pati kontrolsumma. Dotais variants ir ērts, jo paraksta glabāšanai ir nepieciešams mazāk atmiņas un to iespējams glabāt atsevišķi no dokumenta.

Lai uzbrucējs nevarētu daudzkārt izmantot parakstītu dokumentu, kas paredzēts vienreizējai izmantošanai (piemēram, naudas pārskaitījumam), dokumentam jāpievieno informācijas bloks, kurā fiksēts dokumenta parakstīšanas datums un laiks, tā sauktais *laika spiedols* (timestamp). Bez tam, dokumentam pievienots laika spiedolas samazina risku, ka autors varētu atteikties no paraksta (ja parakstītājs paziņos, ka viņam nozagta privātā atslēga un viņš nav parakstījis dokumentu). Atbilstošs protokols ar starpnieku paredz sekojošus soļus:

1. *A* paraksta ziņojumu.

2. *A* ģenerē galveni (header), kas satur identifikācijas informāciju, pievieno galveni parakstītajam dokumentam, vēlreiz visu paraksta, un visu kopā nosūta starpniekam.

3. Starpnieks pārbauda ārējā paraksta pareizību un apstiprina identifikācijas informāciju. Pēc tam starpnieks pievieno ziņojumam un identifikācijas informācijai laika spiedolu. Pēc tam viņš paraksta visu paketi un nosūta to gan autoram A, gan adresātam B.

4. B pārbauda starpnieka paraksta pareizību, identifikācijas informāciju un A parakstu.

5. A pārbauda ziņojuma patiesumu, kuru starpnieks nosūtījis B. Ja A neatzīst savu autorību, viņš nekavējoties par to paziņo.

Atzīmēsim, ka svarīgs jautājums ir visiem pieejamās publisko atslēgu datu bāzes aizsardzība. Viens no aizsardzības variantiem paredz izmantot starpnieku kā atslēgu izplatīšanas centru (AIC), kurš paraksta katru publisko atslēgu ar savu privāto atslēgu. Šāds protokols darbojas labi, ja ir nodrošināta AIC publiskās atslēgas aizsardzība.

2. Šifra atslēgu sistēma

Šifra atslēgu sistēma. Atslēgu kopa. Atslēgu kopas varbūtiskais modelis. Atslēgu ģenerācija. Atslēgu slepenības nodrošināšana. Atslēgu apmaiņas protokols.

Atslēgu sistēmas uzbūve ir viens no kriptosistēmas kriptogrāfiskās noturības noteicošajiem faktoriem. Pie tam ir svarīgas abas atslēgu sistēmas komponentes: šifra atslēgu kopas uzbūve, kā arī procedūras un protokoli, kas vada visus atslēgas dzīves cikla posmus.

2.1. Atslēgu kopas uzbūve un kārta

Par atslēgu var tikt izmantota vai nu šifra attēlojuma ieejas datu daļa vai funkcionālie elementi, kas tiek lietoti šifra attēlojumu veidošanā, vai kaut kāds šo abu variantu apvienojums. Visas iespējamās atslēgas vērtības veido atslēgu kopu. Dažas atslēgu kopas ir neviendabīgas, tām piemīt noteiktas strukturālas īpašības. Piemēram atslēgu kopa K var būt dota šādā veidā:

$$K = K_1 \times K_2,$$

kur K_1 — *seansa* jeb *vienreizējo* atslēgu kopa (kas tiek lietotas tikai viena seansa laikā), bet K_2 — *strukturālo* atslēgu (šifrēšanas algoritma funkcionālo elementu) vai *ilglaicīgo* atslēgu (atslēgas, kas paredzētas izmantošanai ilgstošā laika periodā) kopa.

Dažos atslēgu izplatīšanas protokolos tiek izmantots cits atslēgu dalījums pēc to funkcionālās nozīmes: seansa atslēgas (kas tiek izmantotas ziņojumu šifrēšanai), *nosūtīšanas* atslēgas (kas tiek izmantotas seansa atslēgu šifrēšanai), *galvenās* atslēgas (kas tiek izmantotas nosūtīšanas atslēgu glabāšanai šifrētā veidā), utt.

Viena no galvenajām atslēgu kopas īpašībām ir tās *kārta*, proti dažādo atslēgu skaits, kas veido atslēgu kopu. Svarīga atslēgu raksturojoša īpašība ir tās *izmērs*, ar ko parasti tiek saprasts vārda garums kaut kādā alfabētā, kas kodē atslēgas vērtību. Atslēgu kopas kārta ir tieši saistīta ar to veidojošo atslēgu izmēru.

Piemēram, ja atslēgu kopa ir $\{0,1\}^n$, tad atslēgu $k \in \{0,1\}^n$ sauc par *bināru* un tās izmērs ir vienāds ar attiecīgā binārā vektora koordinašu skaitu n. Atslēgu kopas kārta šādā gadījumā ir 2^n. Ja par atslēgu tiek izmantota

n–dimensionāla vektora koordinātu substitūcija, tad šādu atslēgu bieži sauc par *komutatoru* un tās izmērs ir vienāds ar substitūcijas pieraksta garumu, proti, n. Atslēgu kopas kārta šādā gadījumā ir vienāda ar $n!$ (Lasītājam atgādinam, ka no algebras redzes viedokļa te mēs darbojamies ar simetrisko grupu $\mathfrak{S}_n$.)

Atslēgas izmēram jābūt pietiekoši lielam, lai šifra atklāšana pārlasot visas atslēgas pieņemamā laikā nebūtu uzbrucēja spēkos. Pie tam jāņem vērā šifējamās informācijas slepenības līmenis, nepieciešamais noslēpuma saglabāšanas ilgums, kā arī uzbrucēja kriptoanalītiķa iespējas, konkrēti pieejamie skaitļošanas resursi. No otras puses, pārlieku liels atslēgas izmērs var negatīvi atsaukties uz atslēgas lietošanas ērtumu, uzticamību un izmaksām. Tāpēc atslēgas izmēra izvēle izstrādājot kriptosistēmu ir kompromiss ar pretējiem faktoriem.

Kriptogrāfijas literatūrā atslēgas garumam tiek doti sekojoši aptuveni novērtējumi. Izmantojot simetriskas kriptosistēmas ar bināru atslēgu, atslēgas garumam jābūt vismaz 128 biti. Kriptosistēmām ar publisko atslēgu šis novērtējums ir vismaz 10 reizes augstāks.

Ja aizsargājamās informācijas drošības līmenis nav īpaši augsts, tad atslēgas garumu var samazināt, bet ne vairāk kā uz pusi.

2.2. Atslēgu kopas varbūtiskais modelis

Kriptoanalītiķis var izmantot atslēgu kopas varbūtisko modeli, lai analizētu kriptosistēmas statistiskās īpašības. Varbūtiskais modelis ir atkarīgs no atslēgu kopas varbūtiskā sadalījuma, kas tiek aplūkota kā elementāru notikumu kopa. Katrai atslēgai $k \in K$ tiek piekārtota atbilstoša varbūtība p_k ar kādu tā varētu tikt izmantota kaut kāda konkrēta vai nejauša teksta šifrēšanai. Pie tam $\sum_{k \in K} p_k = 1$.

Visdabiskākajā un kriptoanalītķu bieži izmantotā modelī tiek pieņemts, ka šifrēšanas atslēgas tiek izvēlētas neatkarīgi no pamattekstiem un tām piemīt labas statistiskas īpašības, proti:

1. pēc kārtējā ekspluatācijas perioda (seansa, diennakts, utt.) katra atslēga tiek izvēlēta nejauši ar vienādām varbūtībām no atslēgu kopas K, proti $p_k = 1/|K|$ katram $k \in K$;

2. nomainot atslēgas, jaunā atslēga tiek izvēlēta neatkarīgi no iepriekšējām atslēgām.

2.3. Atslēgu ģenerēšana

Atslēgas būtiskākais raksturotājs ir tās *nejaušums*. Regularitātes atslēgā vai atslēgu masīvā noved pie nemanāmas atslēgu kopas kārtas samazināšanās, un līdz ar to samazinās šifra kriptogrāfiskā noturība. Piemēram, ja kriptosistēma izmanto 8 baitu bināru atslēgu un atslēgas ģenerēšana pieļauj jebkādu no visām iespējamām atslēgu vērtībām, tad atslēgu kopas kārta ir 2^{64}. Ja, turpretim, atslēga tiek veidota no ASCII simbolu kodiem, tad atslēgu kopas kārta samazinās līdz 2^{40}. Proti, atslēgu pilnajai pārlasei nepieciešami 10 miljonu reižu mazāki resursi nekā pirmajā gadījumā.

Ja par atslēgām tiek izmantoti jēgpilni vārdi un izteikumi, tad tas arī noved pie atslēgu kopas kārtas samazināšanās. Šādu atslēgu pārlasi sauc par *vārdnīcas uzbrukumu*.

Atslēgu kvalitāti nejaušuma nozīmē nosaka atslēgu ģeneratora īpašības. Simetriskām kriptosistēmām par labām atslēgām var uzskatīt nejaušas bitu virknes, kuras iegūtas ar fiziskiem gadījuma skaitļu ģeneratoriem (hardware random number generator), kā arī nejaušas bitu virknes, kas iegūtas ar pseido gadījumskaitļu ģeneratoriem (pseudo random number generator), ja šīs virknes ar statistisku testu palīdzību tiek atzītas par nejaušām.

Atslēgu ģenerēšana asimetriskām kriptosistēmām ir sarežģītāka, jo atslēgām ir jāapmierina noteiktas teorētiskas īpašības, piemēram, jābūt pirmskaitļiem. Par pseido gadījumu skaitļu ģeneratoriem var izmantot bloka šifrus. Piemēram, amerikāņu atslēgu ģenerēšanas standarts ANSI X9.17 tiek realizēts ar kriptogrāfisku algoritmu, kas balstīts uz 3DES shēmas.

2.4. Atslēgu slepenības nodrošināšana

Svarīgākā prasība atslēgām ir to *slepenība*, kas tika ilustrēta 1.4. nodaļā aplūkojot kriptogrāfiskos protokolus. Lai nodrošinātu atslēgu slepenību dažādos tās dzīves cikla posmos tiek lietotas sekojošas tehniskas un organizatoriskas idejas:

1. personu loka, kam ir pieeja atslēgām, ierobežošana;
2. atslēgu izplatīšanas, uzglabāšanas un iznīcināšanas reglamentācija;
3. atslēgu nomaiņas reglamentācija;
4. tiek izmantoti tehniski līdzekļi, lai ierobežu informācijas izplatību par atslēgām no nesankcionētas piekļuves.

Aplūkosim dažus šo ideju realizācijas veidus.

Noslēpuma dalīšanas shēmas

Noslēpuma dalīšana (secret sharing) ir metode ar kuru slepenās informācijas daļas tiek sadalītas starp vairākiem lietotājiem, pie tam personai, kurai ir pieejama kāda no daļām, nav iespējas atjaunot noslēpumu vienkāršāk nekā veicot visu iespējamo noslēpuma vērtību pilno pārlasi.

Noslēpuma dalīšanas princips tiek izmantots kriptogrāfiskiem lietojumiem, tai skaitā atslēgu izplatīšanai.

Pieņemsim, ka noslēpums ir bināra atslēga x ar garumu n. Vienkāršākā šāda noslēpuma dalīšanas shēma starp m lietotājiem ir izdalīt tiem vektorus $x_1, x_2, \ldots, x_m$ no $\{0,1\}^n$ tādus, ka $x = x_1 \oplus x_2 \oplus \ldots \oplus x_m$. Patiešām, ja pietrūkst kaut vienas daļas, nav iespējams noteikt noslēpuma x vērtību. Tikai apvienojot visas daļas iespējams aprēķināt x.

Atzīmēsim, ja lietotājiem tiktu uzticētas x vektora koordinašu vērtības $r_1, r_2, \ldots, r_m$, kur $r_1 + r_2 + \ldots + r_m = n$, to nevarētu saukt par noslēpuma dalīšanu (šādas shēmas dažreiz mēdz saukt par *noslēpuma sadalīšanu*), jo katram lietotājam, lai atjaunotu noslēpumu nepieciešams veikt mazāk par 2^n variantu pārlasi.

Vienkāršākās shēmas nepilnība ir tās neuzticamība. Ja kāda lietotāja daļa tiks pazaudēta, noslēpumu nebūs iespējams atjaunot.

Šinī sakarā ievests *noslēpuma dalīšanas shēmas ar slieksni* (n,t) (secret sharing with threshold (n,t)) jēdziens, kur $1 < t \leq n$. Šādu shēmu definē sekojoši: jebkura t lietotāju grupa var viennozīmīgi atjaunot noslēpumu, bet grupa kurā ir mazāk lietotāju to izdarīt nevar.

Piemērs shēmai ar slieksni (n,t) ir *Šhamira shēma* (Shamir's scheme). Šajā shēmā noslēpums ir iepriekš izvēlēta polinoma $f(x)$ brīvais loceklis a_0:

$$f(x) = \sum_{i=0}^{t-1} a_i x^i.$$

Polinoms $f \in L[x]$, kur L — galīgs lauks ar lielu kārtu. Izvēlamies n dažādus lauka L nenulles elementus $r_1, r_2, \ldots, r_n$. Aprēķinam izvēlētā polinoma $f(x)$ vērtības $f(r_i)$ un izdalam lietotājiem datu fragmentus $(r_i, f(r_i))$ kā noslēpuma daļas, $i \in \overline{1,n}$.

Lai atjaunotu noslēpumu var izmantot Lagranža interpolācijas formulu

$$f(x) = \sum_{i=1}^{t} f(s_i) \cdot \prod_{j \neq i} \frac{x - s_j}{s_i - s_j},$$

te $s_1, \ldots, s_t$ — pa pāriem atšķirīgi lauka L elementi. Ņemot vērā, ka $a_0 = f(0)$ iegūstam:

$$a_0 = \sum_{i=1}^{t} f(s_i) \cdot \prod_{j \neq i} \frac{s_j}{s_i - s_j}.$$

Skaitļi

$$\prod_{j \neq i} \frac{s_j}{s_i - s_j}$$

ir zināmi un nav atkarīgi no polinoma $f(x)$ īpašībām. Ar otrās formulas palīdzību jebkura t lietotāju grupa var atjaunot noslēpumu. Tajā pašā laikā neviena no grupām, kurā būtu mazāk par t lietotājiem, nevarēs atjaunot noslēpumu, jo viennozīmīgai polinoma ar pakāpi $t-1$ locekļu noteikšanai nepieciešamas polinoma vērtības vismaz t lauka punktos.

Šamira shēma ir ērta, jo lietotāju skaitu ir viegli izmainīt, vienkārši jāpievieno lauka L elementiem $r_1, r_2, \ldots, r_n$ jauni elementi.

Ja tiek kompromitētas m noslēpuma daļas, $1 \leq m < t-1$, shēma ar slieksni (n, t) tiek pārveidota par shēmu ar slieksni $(n-m, t-m)$.

Izgudrotas arī smalkākas noslēpuma dalīšanas shēmas, kuras paredz, ka starp lietotājiem var izrādīties uzbrucējs.

Atslēgu izplatīšana

Tradicionālā metode simetrisko kriptosistēmu atslēgu izplatīšanai ir kurjerdienestu izmantošana. Lai ierobežotu nesankcionētas piekļūšanas iespēju nosūtīšanas laikā var tikt izmantots speciāls iepakojums, kas nodrošina atslēgu materiāla veselumu un aizsardzību pret informācijas noplūdi. Šāda metode ir dārga un darbietilpīga.

Izgudroti arī vairāki alternatīvi varianti. Var izmantot vai nu kaut kādu uzticamu aizsargātu sakaru kanālu vai vairākus paralēlus sakaru kanālus un noslēpuma dalīšanas shēmu. Efektīvs līdzeklis ir atslēgu nodošana šifrētā veidā, ja ir nodrošināta pārsūtīšanas atslēgas slepenība.

Ja atslēgas tiek nodotas elektroniskā veidā, tās mēdz kropļoties, kas noved pie tā, ka tās vairs nav izmantojamas. Lai atklātu nosūtītās atslēgas

kropļojumus var izmantot kaut kādu konstanti, ko tad šifrējam ar nododamo atslēgu. Tagad nosūta līdz ar atslēgu arī dažus iegūtās kriptogrammas baitus, kas tad pilda kontrolsummas funkciju. Ja tiek atklāti kropļojumi, atslēgu var nosūtīt atkārtoti.

Atslēgu uzglabāšana

Atslēgu slepenības nodrošināšanai uzglabāšanas laikā var izmantot sekojošas metodes:

1. atslēga tiek glabāta ārpus datora vai šifrējošās iekārtas un lietotājs to katru reizi ievada ar klaviatūras palīdzību, pie tam atbildība par atslēgas slepenības nodrošināšanu un ievades pareizību tiek deliģēta lietotājam;

2. atslēga, kas ir ierakstīta kartē (ar magnētisko joslu, mikroshēmu vai smart-kartē), tiek ievadīta datorā vai šifrētājā ar nolasošas iekārtas palīdzību, pie tam lietotāju skaits, kam ir pieeja atslēgai ir ierobežots un līdz ar to tiek samazināta atslēgas kompromitācijas iespējamība;

3. slepenā atslēga tiek sadalīta divās daļās, viena no kurām tiek glabāta datorā, bet otra kartē, kas izslēdz iespēju kompromitēt atslēgu kompromitējot kādu no tās daļām;

4. atslēga k tiek glabāta datora atmiņā šifrētā veidā un tiek atšifrēta tieši pirms izmantošanas, pie tam atslēgas slepenība ar kuru ir šifrēta k tiek nodrošināta kādā no trim augstāk norādītajiem veidiem.

Atzīmēsim seansu atslēgu ērtību, kas ir saistīta ar to, ka tās nav nepieciešams uzglabāt pēc izmantošanas. Ja seansa atslēga tiek ģenerēta tieši pirms seansa, tad uzglabāšanas problēma atkrīt pilnībā.

Ja kaut kādu iemeslu dēļ atslēga tiek nozaudēta, piemēram, fiziski tiek iznīcināts atslēgas nesējs, tad kriptosistēmas darbaspējas atjaunošanai jāparedz pazaudēto atslēgu atjaunošanas mehānisms. Šim mērķim var izmantot *atslēgu deponēšanu* — to kopijas glabāšanu (vislabāk šifrētā veidā) pie uzticamas personas. Šīs metodes uzticamība palielinās, ja tiek izmantotas vairākas uzticamas personas un atslēgu dalīšanas shēma. Atslēgu deponēšanai var izmantot smart-kartes, kas glabājas pie uzticamās personas.

Atslēgu nomaiņa

Vairākas kriptosistēmas paredz atslēgu nomaiņu reizi dienā. Lai izvairītos no nepieciešamības katru dienu izplatīt atslēgas, dažās kriptosistēmās jaunā atslēga k_{i+1} tiek ģenerēta no iepriekšējās atslēgas k_i, veicot aprēķinus izmantojot vienvirziena funkciju $f : k_{i+1} = f(k_i), i = 1, 2, \ldots$. Šādu procedūru sauc par *atslēgas atjaunošanu*. Atjaunotās atslēgas slepenība ir atkarīga no iepriekšējo atslēgu slepenības.

Ja atslēga tiek kompromitēta, to nekavējoties jānomaina un jāpāriet uz jaunās atslēgas izmantošanu. Ja tiek kompromitēta asimetriskas kriptosistēmas privātā atslēga, tad jāapšauba visu darījumu un vienošanos, kas slēgtas izmantojot šo atslēgu, likumību, jo gan publiskās gan privātās atslēgas pieejamība ļauj uzbrucējam lasīt un parakstīt ziņojumus. Ja kompromitācijas datumu izdevies noskaidrot, tad kompromitācijas kaitējumus iespējams lokalizēt izmantojot laika spiedolus ziņojumos. Lai samazinātu potenciālo atslēgu kompromitācijas kaitējumu var izmantot vairākas atslēgas dažādiem lietojumiem. Piemēram, var sadalīt abonentus grupās un komunikācijai ar katru grupu izmantot dažādus atslēgu komplektus.

Atslēgas "mūža" ilgumam jābūt ierobežotam. Jo ilgāk tiek izmantota atslēga, jo lielāka varbūtība, ka tā tiks kompromitēta un lielāks vilinājums kriptoanalītiķim nodarboties ar tās atklāšanu, jo tas ļaus izlasīt visus ziņojumus, kas šifrēti ar šo atslēgu. It īpaši ja ņem vērā, ka daudzu ziņojumu, kas šifrēti ar šo vienu atslēgu, pieejamība kriptoanalītiķim dod papildus iespējas. Atslēgu iznīcināšanai ir jābūt neatgriezeniskai, lai pilnībā izslēgtu to atjaunošanu. Praksē izmanto arī atslēgu *arhivēšanu* — atslēgu kopiju saglabāšanu, kas kaut kāda laika posma ietvaros var tikt atjaunotas.

2.5. Atslēgu apmaiņas protokoli

Viens no svarīgākajiem kriptogrāfiskajiem uzdevumiem ir atslēgu izplatīšana. Mūsdienīgas metodes izmanto tehniskus sakaru kanālus un ir realizējamas ātrāk un ērtāk, nekā tradicionālās izsūtīšanas metodes.

Populārākie atslēgu izplatīšanas protokolu veidi:

1. atslēgas nodošana abonentam;
2. abonenti kopīgi izstrādā kopēju atslēgu (publiska atslēgu izplatīšana);
3. sākotnēja atslēgu izplatīšana.

Vienlaicīgi ar atslēgu nodošanu bieži rodas vajadzība identificēt pašus abonentus.

Atslēgu izplatīšana izmantojot simetrisko šifrēšanu

Divpusēji protokoli. Pieņemsim, ka abonenti A (Alise) un B (Bucis) sakariem izmanto šifru $E = \{E_k \,|\, k \in K\}$, turklāt pieņemsim, ka viņiem ir pieejama sākotnēji zināma atslēga k_{AB}, kas netiek izmantota šifrēšanai, bet tikai atslēgu informācijas pārsūtīšanai.

Alise nodod Bucim vienreizējo atslēgu k nosūtot kriptogrammu $E_{k_{AB}}(I)$, kas iegūta šifrējot pamattekstu I ar atslēgu k_{AB}, kur

$$I = (k, t, b).$$

Te t — laika spiedols; b — Buča identifikators. Ja netiktu nodots laika spiedols, tad uzbrucējs varētu vēlāk atkārtoti nodot ziņojumu. Savukārt, ja netiktu nodots Buča identifikators b, uzbrucējs varētu pārsūtīt ziņojumu citam adresātam, piemēram abonentam C.

Seansa autentifikācijai var izmantot sekojošu protokolu, kas balstīts uz "pieprasījuma - atbildes" principu.

1. Bucis ģenerē gadījuma skaitli r_B (kas nepieciešams, lai Bucis varētu pārliecināties, ka komunicē tieši ar Alisi, un nosūta to Alisie.

2. Alise ģenerē kriptogrammu $E_{k_{AB}}(I)$, kur

   $$I = (k, r_B, b),$$

   un nosūta to Bucim.

Divpusējai seansa autentifikācijai var izmantot sekojošu protokolu, kas ir iepriekšējā protokola uzlabots variants. Pieņemsim, ka $\varkappa_A$ un $\varkappa_B$ — gadījuma skaitļi (sagataves seansa atslēgas k izveidei), kurus ir ģenerējuši attiecīgi Alise un Bucis.

1. Bucis ģenerē gadījuma skaitli r_B un nosūta to Alisei (vēlāk Alisei Bucim būs jāuzrāda r_B).

2. Alise ģenerē gadījuma skaitli r_A (kuru Bucim būs jāuzrāda Alisei), aprēķina kriptogrammu $E_{k_{AB}}(I)$, kur

   $$I = (\varkappa_A, r_A, r_B, b),$$

un nosūta to Bucim. Kad Bucis atšifrēs kriptogrammu, Alise būs Bucim uzrādījusi r_B.

3. Bucis aprēķina kriptogrammu $E_{k_{AB}}(J)$, kur

$$J = (\varkappa_B, r_A, r_B, a),$$

un nosūta to Alisei (a ir Alises identifikators). Kad Alise atšifrēs kriptogrammu, Bucis būs viņai uzrādījis r_A.

4. Katra no pusēm aprēķina seansa atslēgu k izmantojot kaut kādu funkciju f:

$$k = f(\varkappa_A, \varkappa_B).$$

Atzīmēsim, ka neviena no pusēm iepriekš nezin seansa atslēgu.

Šamira bezatslēgas protokols. Dotajam protokolam nav nepieciešama abiem abonentiem iepriekš zināmas atslēgas izmantošana. Tajā pašā laikā ir nepieciešams, lai visi šifra E attēlojumi būtu komutatīvi, proti, jebkuram atslēgu pārim k_1 un k_2 no atslēgu kopas K un katram pamattekstam x jābūt spēkā vienādībai

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

Šādā gadījumā, lai Alise nodotu slepeno atslēgu k Bucim, var izmantot sekojošu protokolu.

1. Alise ģenerē nejaušu seansa atslēgu k un atslēgu k_A, kas tiks izmantota atslēgas informācijas nodošanai, pēc tam aprēķina kriptogrammu

$$E_{k_A}(k)$$

un nosūta to Bucim.

2. Bucis ģenerē nejaušu atslēgu k_B atslēgas informācijas nodošanai, pēc tam aprēķina kriptogrammu

$$E_{k_B}(E_{k_A}(k))$$

un nosūta to Alisei.

3. Alise atšifrē saņemto kriptogrammu, izmantojot atslēgu k_A, proti, iegūst

$$E_{k_A}^{-1}(E_{k_B}(E_{k_A}(k)) = E_{k_B}(k),$$

un nosūta šo ziņojumu Bucim.

4. Bucis atšifrē saņemto kriptogrammu ar atslēgu k_B, proti, aprēķina seansa atslēgu k.

Taču šis protokols dažos gadījumos ir nenoturīgs pat pret pasīvu uzbrukumu.

Piemērs 2.1. Pieņemsim, ka

$$\begin{aligned} k &= s_1 s_2 \dots s_n, \\ k_A &= a_1 a_2 \dots a_n, \\ k_B &= b_1 b_2 \dots b_n, \end{aligned}$$

kur visi burti s_i, a_i, b_i ir kopas $\{0, 1\}$ elementi. Savukārt

$$\begin{aligned} E_{k_A}(k) &\rightleftharpoons c_1 c_2 \dots c_n, \\ E_{k_B}(E_{k_A}(k)) &\rightleftharpoons d_1 d_2 \dots d_n, \\ E_{k_B}(k) &\rightleftharpoons e_1 e_2 \dots e_n, \end{aligned}$$

kur $c_i \rightleftharpoons a_i + s_i, \quad d_i \rightleftharpoons b_i + a_i + s_i, \quad e_i \rightleftharpoons b_i + s_i,$
visiem indeksiem $i \in \overline{1, n}$ (te saskaitīšana notiek pēc moduļa 2).

Ja nu kriptoanalītiķis (Oskars) iegūst visus trīs vārdus

$$c_1 c_2 \dots c_n, \quad d_1 d_2 \dots d_n, \quad e_1 e_2 \dots e_n,$$

tad Oskars var viegli izskaitļot arī

$$a_1 a_2 \dots a_n, \quad b_1 b_2 \dots b_n, \quad s_1 s_2 \dots s_n,$$

proti,

$$\begin{aligned} c_i + d_i &= b_i, \\ e_i + b_i &= s_i, \\ c_i + s_i &= a_i. \end{aligned}$$

Te saskaitīšana arī notiek pēc moduļa 2.

Šamirs un Omura neatkarīgi parādīja, ka šo protokolu var realizēt izmantojot šāda tipa pārveidojumu:

$$E_{k_A}(k) = k^w \mod p,$$

kur p — liels pirmskaitlis; $p-1$ dalās ar lielu pirmskaitli, un konstanti w nosaka atslēga k_A.

Viens no svarīgiem kriptogrāfisko protokolu elementiem ir *drošības modulis*, kas ir tehniska iekārta, kas atļauj fiziski aizsargāt šifrēšanas algoritmu un atslēgu no nesankcionētas pieejas. Mēģinājums iekļūt iekārtā noved pie neatgriezeniskas shēmas un informācijas iznīcināšanas.

Aplūkosim atslēgu izplatīšanas protokolu, kas izmanto drošības moduli. Pieņemsim, ka simetriskas sistēmas atslēgas veido trīs līmeņu hierarhiju: seansa atslēgas, seansa atslēgu izplatīšanas atslēgas un glabāšanas galvenās atslēgas ar kurām tiek šifrētas izplatīšanas atslēgas. Galvenās atslēgas tiek izsūtītas iepriekš un glabājas fiziski aizsargātā drošības modulī, jo no tām ir atkarīga visas sistēmas drošība.

Nosūtīšanas atslēga k_p glabājas neaizsargātā vietā, taču ir šifrēta (izmantojot galveno atslēgu k_m):

$$E_{k_m}(k_p) = s_p.$$

Sakaru seansa organizēšanai abonents A drošības modulī ievada s_p, kas veic tās atšifrēšanu izmantojot galveno atslēgu:

$$E^{-1}_{k_m}(s_p) = k_p.$$

Pēc tam drošības modulī tiek ģenerēta nejauša seansa atslēga k_c un tā tiek aizšifrēta ar izsūtīšanas atslēgu un pēc tam nosūtīta abonentam B:

$$E_{k_p}(k_c) = s_c.$$

Abonents B analoģiskā veidā izrēķina k_p un ar tā palīdzību atšifrē seansa atslēgu.Tādā veidā drošības modulī atslēgas tiek apstrādātas nešifrētā veidā, bet ārpus tā atslēgas nekad neparādās nešifrētā veidā.

Trīspusējs protokols Protokolā iesaistītās personas ir atslēgu ģenerācijas un izplātīšanas centrs kā starpnieks un abonenti A un B, kuriem ir pieejamas attiecīgi atslēgas k_1 un k_2 sakariem ar AĢIC.

1. Abonents A vēršas pie AĢIC un pieprasa seansa atslēgu, lai sazinātos ar B.

2. AĢIC ģenerē nejaušu seansa atslēgu k un šifrē divas tās kopijas: vienu ar A pieejamo atslēgu k_1, bet otru ar B pieejamo atslēgu k_2:

$$E_{k_1}(k) = s_1; \quad E_{k_2}(k) = s_2.$$

   Bez tam AĢIC šifrē ar atslēgu k_2 identifikatoru p_A, kas apliecina, ka pieprasījums nāk no abonenta A:

$$E_{k_2}(p_A) = s_A.$$

   AĢIC nosūta A šifrētā veidā identifikatoru p_A un abas seansa atslēgu kopijas. Proti nosūta s_1 un informācijas bloku $[s_2, s_A]$.

3. A atšifrē savu seansa atslēgas kopiju:

$$E_{k_2}^{-1}(s_1) = k.$$

4. A nosūta B informācijas bloku $[s_2, s_A]$.

5. B atšifrē savu seansa atslēgas kopiju un A identifikatoru:

$$E_{k_2}^{-1}(s_2) = k; \quad E_{k_2}^{-1}(s_A) = p_A.$$

   B ir saņēmis atslēgu un zin, kas ar viņu vēlas sazināties.

6. A un B var nodibināt slepenus sakarus izmantojot atslēgu k.

Kaut arī šāds protokols ir praktiski lietojams, tam nepieciešama pilnīga AĢIC drošība. Ja aktīvam uzbrucējam izdosies uzpirkt AĢIC, būs kompromitēts viss tīkls, kuru apkalpo dotais AĢIC. Aktīvajam uzbrucējam būs pieejamas visas slepenās atslēgas, kas glabājas pie AĢIC komunikācijai ar abonentiem. Uzbrucējs varēs izlasīt visus ziņojumus, kas jau ir tikuši izsūtīti tīklā un kuri vēl tiks nosūtīti. Lai to izdarītu viņam pietiek kontrolēt sakaru līniju un atšifrēt pārtvertās kriptogrammas. Liela abonentu skaita gadījumā AĢIC var izrādīties pārslogots.

Atslēgu pārvaldīšana sistēmās ar publisko atslēgu

Publiskās atslēgas kriptogrāfija vienkāršo atslēgu pārvaldīšanu, tomēr tai ir savas nepilnības. Ja abonents A pieprasa abonenta B publisko atslēgu, lai varētu ar to sazināties, tad uzbrucējs var pārtvert šo atslēgu un tās vietā nosūtīt savu atslēgu. Pēc tam viņš var komunicēt ar A izliekoties par B. Šāds uzbrucēja darbību modelis ir ieguvis nosaukumu *aktīvs uzbrucējs pa vidu* (man-in-the-middle attack).

Lai novērstu atslēgas viltošanu, tiek lietoti *publisko atslēgu sertifikāti*, kas ir atslēga, kuru ir parakstījusi uzticama persona (centralizētas atslēgu izplatīšanas gadījumā — sertifikācijas centrs). Sertifikāts satur ne tikai atslēgu, bet arī informāciju par tās īpašnieku.

Praksē izmanto arī decentralizētu atslēgu vadību. Šajā gadījumā atslēgas paraksta *galvotāji*, kas ir izvēlēti no līdzvērtīgiem sistēmas lietotājiem. Šādas shēmas priekšrocība ir tā, ka nav jāizmanto sertifikācijas centrs, kam visi uzticas, bet nepilnība — nav garantiju, ka sertifikāta saņēmējs uzticas galvotājiem, kas to ir parakstījuši.

Diffi-Helmana algoritms Lai kopīgi izveidotu atslēgu abonenti A un B izvēlas lielu pirmskaitli p un skaitli g, kas veido lielas kārtas apakšgrupu multiplikatīvajā atlikumu grupā $\mathbb{Z}_p^*$ (ideālā gadījumā g — primitīva sakne pēc moduļa n). Šos skaitļus nav obligāti glabāt slepenībā. Pēc tam tiek izpildīts sekojošs protokols:

1. Abonents A izvēlas lielu veselu gadījuma skaitli x un nosūta B skaitli X:
$$X = g^x \mod p.$$
2. Abonents B izvēlas lielu veselu gadījuma skaitli y un nosūta A skaitli Y:
$$Y = g^y \mod p.$$
3. Abonents A aprēķina vērtību:
$$k = Y^x \mod p.$$
4. Abonents B aprēķina vērtību:
$$k' = X^y \mod p.$$

Acīmredzami, $k = k'$, bet uzbrucējam kopējās atslēgas k aprēķināšanai jāatrod diskrētais logaritms, lai noteiktu x vai y.

Atzīmēsim, ka protokols nav aizsargāts pret aktīva uzbrucēja pa vidu uzbrukumu. Šāds uzbrucējs var pārtvert abonentu ziņojumus un imitēt katram no viņiem otru abonentu.

Protokols "stacija-stacija" Šis protokols ir noturīgs pret uzbrucēju pa vidu. Tiek pieņemts, ka A ir pieejama sertificēta abonenta B publiskā atslēga, bet B ir pieejama sertificēta abonenta A publiskā atslēga. Šos sertifikātus ir parakstījusi uzticamības persona, kas protokolā nepiedalās.

1. Abonents A ģenerē gadījuma skaitli x un nosūta B skaitli X:

$$X = g^x \mod p.$$

2. Abonents B ģenerē gadījuma skaitli y un aprēķina kopējo atslēgu atbilstoši Diffi-Hellmana algoritmam:

$$k = X^y \mod p.$$

Pēc tam B paraksta pāri $(g^x \mod p,\ g^y \mod p)$, šifrē parakstu izmantojot atslēgu k un nosūta A ziņojumu I (kur S_B — paraksts, E_k — šifrēšana):

$$I = [g^y \mod p,\ E_k(S_B(g^x \mod p,\ g^y \mod p))].$$

3. Abonents A arī aprēķina k. Pēc tam atšifrē otru ziņojuma daļu un pārbauda parakstu. Visbeidzot A nosūta B ziņojumu J:

$$J = E_k(S_A(g^x \mod p,\ g^y \mod p)).$$

4. Abonents B atšifrē ziņojumu J un pārbauda abonenta A parakstu.

Sākotnējā atslēgu izplatīšana

Aplūkosim abonentu tīklu ar n abonentiem, kurā katrs abonentu pāris izmanto neatkarīgas atslēgas. Šāda tīkla atslēgu sistēmu apraksta atslēgu matrica K_n:

$$K_n = ||k_{is}||, \quad i, s \in \overline{1, n},$$

kur k_{is} — atslēga i-tā un s-tā abonentu saziņai, $i \neq s$; k_{ii} - tukšs simbols. Abonentam i sakariem ar pārējiem abonentiem jāuzglabā $n-1$ atslēga (matricas K_n i-tais stabiņš), bet vispār tīklā tiek izmantotas $n(n-1)/2$ atslēgas. Līdz ar to pie lieliem n ievērojami sarežģās atslēgu veidošanas, izplatīšanas un nomaiņas jautājumi tīklā.

Lai samazinātu uzglabājamās informācijas apjomu tīklā izmanto atslēgu sākotnējās izplatīšanas shēmas. Šo shēmu būtība ir tāda, ka izplata nevis pašas atslēgas, bet kaut kādu informāciju ar kuras palīdzību dalībnieki var paši aprēķināt nepieciešamo atslēgu saskaņā ar iepriekš atrunātu procedūru.

Kā piemēru aplūkosim *Blūma atslēgu izplatīšanas shēmu*. Pieņemsim, ka L ir galīgs lauks ar lielu kārtu. Fiksēsim n lauka L elementus $r_1, \ldots, r_n$ atbilstoši tīkla abonentiem. Šie skaitļi nav slepeni un tiek glabāti visiem pieejamā serverī. Slepens ir simetrisks (proti, $a_{ij} = a_{ji}$, ja $i \neq j$) polinoms $f(x, y)$ pār lauku L ar kārtu $2m$, $1 \leq m < n$:

$$f(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{m} a_{ij} x_i y_j.$$

Polinoma $f(x, y)$ koeficienti glabājas atslēgu izplatīšanas centrā.

Katrs abonents A kā atslēgu materiālu saņem polinoma

$$f(x, r_A) = a_0^A + a_1^A \cdot x + \ldots + a_m^A \cdot x^m$$

koeficientu komplektu $(a_0^A, a_1^A, \ldots, a_m^A)$.

Abonentu A un B sakariem kā atslēga tiek izmantots $f(r_B, r_A)$, ko var viegli aprēķināt gan A, gan B izmantojot zināmos skaitļus $r_1, \ldots, r_n$. Izmantojot šo shēmu katram abonentam jāuzglabā $m+1$ noslēpuma vērtības $n-1$ vietā, bet kopējais polinoma $f(x, y)$ slepeno koeficientu skaits ir $m \cdot (m+1)/2$, nevis $n \cdot (n-1)/2$.

Protokola atkarība no tīkla arhitektūras

Atslēgu protokoliem ir jābūt veidotiem ņemot vērā sakaru tīkla, kurā tie tiks izmantoti, arhitektūras īpatnības.

Aplūkosim sakaru kanālus, kuros A darbojas kā vadītājs, bet visi pārējie (B, C un pārējie) veic padoto funkcijas. Pieņemsim, ka A izmanto šo tīklu, lai dotu padotajiem uzdevumus, pavēles, noradījumus un tamlīdzīgi. Pie tam A šifrē informāciju ar atslēgu k, kas ir kopīga visam tīklam, un nosūta katram padotajam ziņojuma kopiju. Kriptoanalītiķis, kas kontrolē tīklu un

pārtver kriptogrammas, saņem tik pat daudz materiāla, cik saņemtu, ja kontrolētu tikai A un B sakaru kanālu, jo visi ziņojumi tīklā ir identiski. Ja A sakariem ar n padotajiem izmantotu n atšķirīgas atslēgas, kriptoanalītiķim, kas kontrolē tīklu, dešifrēšanai būtu pieejams vairāk materiāla, nekā vienas atslēgas gadījumā.

Ja tīkls veidots n abonentu sakariem, kas sarunājas "katrs ar katru", tad izmantot vienu un to pašu atslēgu visiem abonentu pāriem ir pats lētākais un vienkāršākais variants no atslēgu ģenerēšanas, izplatīšanas un nomaiņas viedokļa. Tajā pašā laikā tas ir pats vājākais veids no atslēgu sistēmas darbības noturības viedokļa. Ņemam vērā, ja atslēga tiktu kompromitēta tikai vienam abonentam, tā rezultātā tiktu kompromitēts viss tīkls. Tāpēc svarīgākai atslēgu sistēmas raksturotājs ir tās *kompromitācijas noturība*.

Saka, ka atslēgu sistēma, kuras darbību nodrošina N atslēgas *iztur* r *kompromitācijas*, $r < N$, ja r atslēgu kompromitācija neatvieglo pārējo $N-r$ atslēgu noteikšanu.

n abonentu tīkls, kas komunicē pēc principa "katrs ar katru", ir visnoturīgākais, ja dažādu pāru komunikācijai tiek izmantotas dažādas neatkarīgas atslēgas. Šāds tīkls iztur jebkuru $n-2$ abonentu visu atslēgu kompromitāciju.

n abonentu tīkls, kas izmanto Blūma atslēgu izplatīšanas shēmu, iztur jebkuru m abonentu atslēgu kompromitāciju. Lai vienkāršotu atslēgu vadības jautājumus, iespējams organizēt n abonentu sakarus izmantojot hierarhisku tīklu.

Pieņemsim, ka n abonenti sadalīti m zonās (apmēram vienādās), kur katrai zonai izdalīts viens abonents, ko sauc par zonas vadītāju. Zonas iekšpusē abonentu sakari organizēti pēc principa "katrs ar katru", un tādā pašā veidā saistīti visu zonu vadītāji. Dažādu zonu abonenti viens ar otru sazinās caur saviem vadītājiem, kuri šādā gadījumā veic AĢIC funkcijas. Šādā tīklā katrs "ierindas" abonents glabā atslēgas tikai saziņai ar savas zonas abonentiem, proti ar kārtu n/m atslēgas, bet zonas vadītājs glabā vēl arī atslēgas saziņai ar pārējo zonu vadītājiem, proti ar kārtu $m + n/m$ atslēgas. Pie tam sakaru protokols starp dažādu zonu abonentiem sarežģās.

Acīmredzami, šāds tīkls ir kompromis starp tīklu, kur katrs ar katru komunicē tieši, un tīklu, kur katrs ar katru komunicē izmantojot AĢIC.

3. Pamattekstu avoti

Pamattekstu avoti. Pamattekstu raksturlielumi. Determinētie modeļi. Varbūtiskie modeļi.

Kripotsistēmas kriptogrāfisko īpašību izpēte paredz visu tās komponenšu matemātisko modeļu izpēti: šifra attēlojumu, izmantoto protokolu, kā arī atklāto un šifrēto tekstu, kas saistīti ar kriptosistēmas darbību, izpēti.

Aplūkosim pamattekstu matemātiskos modeļus, vai citiem vārdiem, pamattekstu avotu modeļus.

3.1. Pamattekstu raksturlielumi

Gan šifrētais, gan pamatteksts ir simbolu virknes, kas ņemtas no galīgas kopas, ko sauc par *alfabētu*. Alfabēta elementu sauc par *burtu*. Alfabēta simbolu skaitu sauc par *alfabēta apjomu*.

Pieņemsim, ka $\mathcal{A}$ ir alfabēts. Katru kopas

$$\mathcal{A}^+ \rightleftharpoons \bigcup_{n=1}^{\infty} \mathcal{A}^n$$

elementu $u \in \mathcal{A}^+$ sauc par alfabēta $\mathcal{A}$ *netukšu vārdu*. Pieņemsim, ka

$$u = (u_1, u_2, \ldots, u_k), \qquad v = (v_1, v_2, \ldots, v_m)$$

ir alfabēta $\mathcal{A}$ netukši vārdi, tad

$$u\#v \rightleftharpoons (u_1, u_1, \ldots, u_k, v_1, v_2, \ldots, v_m).$$

Šo kopā $\mathcal{A}^+$ definēto operāciju sauc par *konkatenāciju*.

Parasti vārda definīciju vēl papildina ar norunu, ka drīkst lietot arī tā saukto *tukšo vārdu*, kas nesatur nevienu burtu. Šis vārds ir, varētu sacīt, vienkārši tukša vieta. Vienosimies tukšo vietu apzīmēt ar grieķu burtu "lambda": λ.

Saskaņā ar norunu

$$\lambda\#\lambda \rightleftharpoons \lambda, \qquad \forall u \in \mathcal{A}^+ \;\; \lambda\#u \rightleftharpoons u \rightleftharpoons u\#\lambda\,.$$

Kopu

$$\mathcal{A}^* \rightleftharpoons \mathcal{A}^+ \cup \{\lambda\}$$

sauc par *alfabēta $\mathcal{A}$ vārdu kopu.* Kopas $\mathcal{A}^*$ elementus sauc par *vārdiem.* Kā tas tradicionāli pieņemts, ja nerodas pārpratumi, tad konkatenācijas operāciju izlaiž un lieto pierakstu

$$uv \rightleftharpoons u\#v,$$

bez tam

$$u_1u_2\dots u_k \rightleftharpoons (u_1, u_2, \dots, u_k).$$

Ja $a = u_1 = u_2 = \dots = u_n$, tad lieto arī pierakstu $a^n \rightleftharpoons u_1u_2\dots u_n$. Savukārt $a^0 \rightleftharpoons \lambda$.

Pieņemsim, ka $u \in \mathcal{A}^n$, tad skaitli n sauc par vārda u garumu, ko turpmāk apzīmēsim ar $|u|$. Saskaņā ar definīciju pieņemsim, ka $|\lambda| \rightleftharpoons 0$.

Definīcija 3.1. *Vārdu $v \in \mathcal{A}^*$ sauc par vārda $w \in \mathcal{A}^*$ dalītāju jeb apašvārdu, ja eksistē tādi $u \in \mathcal{A}^*$ un $u' \in \mathcal{A}^*$, ka $w = uvu'$. Šai situācijā vārdu u sauc par priedēkli, bet u' — par piedēkli. Ja $u \neq w$, tad u sauc par īstu priedēkli; līdzīgi, ja $u' \neq w$, tad u' sauc par īstu piedēkli.*

Ja vārda v garums $|v| = \nu$, tad vārdu v mādz saukt arī par *ν–grammu* (ja $\nu = 2$, tad v sauc par *bigrammu*; ja $\nu = 3$, tad — par *trigrammu*, utt.)

Definīcija 3.2. *Kopas $\mathcal{A}^*$ patvaļīgu apakškopu L sauc par valodu alfabētā $\mathcal{A}$. Ja kopa L ir galīga, tad valodu L sauc par galīgu valodu; līdzīgi, ja L ir bezgalīga kopa, tad valodu L sauc par bezgalīgu valodu.*

Aplūkosim alfabētu piemērus, kas saistīti ar angļu valodu.

1. $\mathcal{A}_1$ — lielo burtu alfabēts, $|\mathcal{A}_1| = 26$:

$$A, B, C, \dots, X, Y, Z.$$

2. $\mathcal{A}_2$ — lielie un mazie burti, cipari, atstarpe un interpunkcijas zīmes (alfabēta apjoms aptuveni 70):

$$A, B, C, \dots, X, Y, Z,$$
$$a, b, c, \dots, x, y, z,$$
$$0, 1, 2, \dots, 9, \text{atstarpe, komats, punkts, :, ;, ”, ?, !.}$$

3. $\mathbb{Z}_2 \rightleftharpoons \{0, 1\}$. Matemātikā parasti šo apzīmējumu saista ar divelementīgu lauku.

Mūsdienu datortehnoloģijās pamatā izmanto alfabētus, kas atvasināti no kopas $\mathbb{Z}_2$. Tie sakrīt ar bināru n–dimensionālu kortežu kopu $\mathbb{Z}_2^n$. Matemātikā parasti gan šo apzīmējumu saista ar n–dimensionālu vektoru telpu pār lauku $\mathbb{Z}_2$. Atzīmēsim, ka $|\mathbb{Z}_2^n| = 2^n$; lietojumos lielākoties $5 \leq n \leq 8$. Šāda tipa plaši lietots alfabēts ir tā sauktais ASCII kods.

Bieži šifrēšanas procesā tiek veiktas skaitļošanas darbības ar alfabēta simboliem, tāpēc tos ir ērti attēlot kā skaitļus vai bināras virknes. Populārākais šāda tipa alfabēts ir atlikumu gredzens $\mathbb{Z}_m$ pēc moduļa m:

$$\mathbb{Z}_m \rightleftharpoons \{0, 1, 2, \ldots, m-1\}.$$

Papildus mēdz aplūkot arī alfabēta $\mathbb{Z}_m$ atvasinātos alfabētus $\mathbb{Z}_m^k$, kas iegūti kā visu iespējamo izejas alfabēta k-grammu kopums.

Pamattekstu modeļi sadalāmi divās klasēs: determinētie un varbūtiskie.

3.2. Determinētie modeļi

Pamattekstu ziņojumu avoti ir daudzveidīgi. Kā avotu var aplūkot atsevišķu cilvēku vai cilvēku grupu, radiostacijas, telegrāfa vai telefona tīkla punktus, utt. Katru pamatteksta avotu raksturo:

1. viena vai vairākas komunikācijas valodas; alfabētu komplekts;
2. noteikta ģenerēto ziņojumu tematika;
3. ziņojumu biežumu raksturlielumi, u.c.

Piemēram, ziņojumi angļu valodā, kas nodoti izmantojot teletaipu vai rakstīti uz rakstāmmašīnas visticamāk izmanto alfabētu $\mathcal{A}_2$. Privāts korespondents, kas plāno šifrēt savus ziņojumus daudzos gadījumos izvēlēsies lietot alfabētu $\mathcal{A}_1$. Ziņojumus, kas tiek apstrādāti datortīklos, visērtāk attēlot izmantojot alfabētu $\mathbb{Z}_2^n$.

Katrs avots rada tekstus saskaņā ar kaut kādas valodas gramatikas likumiem, kas parādās arī ziņojumu statistiskajos rādītājos. Piemēram, angļu valodas tekstos burtam q vienmēr seko burts u. Vispārīgi runājot jebkuru valodu un jebkuru avotu var raksturot sadalot visu ν-grammu, $\nu = 2, 3, \ldots$ kopu *pieļaujamās* (kas mēdz parādīties kaut kādos tekstos) un *aizliegtās* (kas nekad neparādās).

Visu ν-grammu sadalīšana pieļaujamās un aizliegtās nosaka avota *determinēto modeli*. Šādā modelī pamatteksts tiek aplūkots kā kaut kāda alfabēta

burtu virkne, kas nesatur aizliegtas ν-grammas. Atzīmēsim, ka multigrammu sadalīšana aizliegtās un pieļaujamās ir visai nosacīta. Valodas dinamisma tēļ tās var mainīties. Bez tam, norādītajam dalījumam iespējamas arī individuālas īpašības, kas raksturo konkrēto ziņojumu avotu, proti, tas apraksta ne tikai valodu kopumā, bet arī individuālu avotu.

3.3. Varbūtiskie modeļi

Varbūtiskajos modeļos pamattekstu avots tiek aplūkots kā gadījumvirkņu avots. Pieņemsim, ka avots ģenerē alfabēta $\mathbb{Z}_m$ galīga vai bezgalīga garuma tekstu. Precizēsim, proti, uzskatīsim, ka avots ģenerē galīgu vai bezgalīgu gadījuma skaitļu virkni $x_0, x_1, \ldots, x_{n-1}, \ldots$ kur visi $x_i \in \mathbb{Z}_m$. Nejauša ziņojuma $a_0a_1 \ldots a_{n-1}$ varbūtību definējam kā sekojošas notikumu virknes varbūtību:

$$P(a_0a_1 \ldots a_{n-1}) \rightleftharpoons P\{x_0 = a_0 \wedge x_1 = a_1 \wedge \ldots \wedge x_{n-1} = a_{n-1}\}.$$

Nejaušo ziņojumu kopa veido varbūtību telpu, ja izpildās sekojoši nosacījumi:

1. $P(a_0a_1 \ldots a_{n-1}) \geq 0$ katram nejaušam ziņojumam $a_0a_1 \ldots a_{n-1}$;
2. $$\sum_{(a_0,a_1,\ldots,a_{n-1})} P(a_0a_1 \ldots a_{n-1}) = 1;$$
3. katram nejaušam ziņojumam $a_0a_1 \ldots a_{n-1}$ un katram $s > n$:

   $$P(a_0a_1 \ldots a_{n-1}) = \sum_{(a_n,a_{n+1},\ldots,a_{s-1})} P(a_0a_1 \ldots a_{s-1}),$$

   proti, katra nejauša ziņojuma ar garumu n varbūtība ir visu to ziņojumu virkņu, kas papildinātas līdz garumam s, varbūtību summa.

Teksts, ko rada šāds avots, ir valodas varbūtiskais analogs. Tam piemīt tādas pašas ν-grammu biežuma īpašības kā valodai. Uzdodot noteiktu varbūtību sadalījumu pamattekstu kopā, mēs uzdodam atbilstošu avota modeli.

Aplūkosim biežāk izmantotos avotu varbūtiskos modeļus.

Valoda	Alfabēta burts/izmantošanas biežums %					
Angļu	E / 12,86	T / 9,72	A / 7,96	I / 7,77	N / 7,51	R / 7,03
Latviešu	A / 9,81	I / 7,70	S / 7,31	T / 5,11	E / 5,11	U / 5,02
Spāņu	E / 14,15	A / 12,90	O / 8,84	S / 7,64	R / 7,01	T / 6,95
Itāļu	I / 12,04	E / 11,63	A / 11,12	O / 8,92	N / 7,68	T / 7,07
Vācu	E / 19.18	N / 10,20	I / 8,21	S / 7,07	R / 7,01	T / 5,86
Franču	E / 17,76	S / 8,23	A / 7,86	N / 7,61	T / 7,30	I / 7,23
Krievu	О / 11,0	И / 8,9	Е / 8,3	А / 7,9	Н / 6,9	Т / 6,0

1. tabula: Burtu biežums dažādās valodās

Stacionārs neatkarīgu alfabēta simbolu avots

Šis modelis pieņem, ka ziņojumu varbūtību pilnībā nosaka atsevišķu alfabēta burtu izmantošanas varbūtības nejaušā tekstā:

$$P(a_0, a_1, \ldots, a_{n-1}) = \prod_{i=0}^{n-1} P\{x_i = a_i\},$$

kur visiem $i \in \{0, 1, \ldots, n-1\}$ un katram $a \in \mathbb{Z}_m$ ir spēkā

$$P\{x_i = a\} > 0; \quad \sum_{a \in \mathbb{Z}_m} P\{x_i = a\} = 1.$$

Šāda avota pamatteksts ir neatkarīgu eksperimentu realizācijas virkne polinomiālā varbūtiskā shēmā ar iznākumu skaitu vienādu m. Iznākumu kopa viennozīmīgi atbilst visu alfabētu simbolu kopai.

Tabulā apkopoti visvairāk lietoto burtu biežumi dažām Eiropas valodām (1. tabula). Dotais modelis atļauj sadalīt alfabēta burtus bieži, vidēji un reti izmantoto klasēs. Lai salīdzinātu reti izmantotos burtus un tabulā uzrādītos burtus, norādīsim, ka, piemēram, angļu valodā reti izmantoti ir burti J, Q un Z, bet to biežumi procentos novērtēti attiecīgi kā 0,13, 0,12 un 0,08.

Šo modeli iespējams uzbūvēt katram avotam, izmantojot samērā nelielu materiāla daudzumu, un tas ir ērts praktiskai lietošanai. Piemēram, šo modeli efektīvi izmanto tekstu dešifrēšanai, kas šifrēts ar vienkāršu aizvietošanas šifru.

Tajā pašā laikā dažas šī modeļa īpašības ir pretrunā ar valodu īpašībām. Konkrēti, saskaņā ar šo modeli jebkurai ν-grammai, $\nu > 1$, ir nenulles varbūtība parādīties ziņojumā. Modeļa ierobežotība neļauj to izmantot dešifrēšanā plašai kriptosistēmu klasei.

Stacionārs neatkarīgu bigrammu avots

Šis modelis ir nedaudz sarežģītāks, taču precīzāk atspoguļo valodu īpašības. Šāda avota pamatteksts ir polinomiālas varbūtiskās shēmas neatkarīgu mēģinājumu virknes realizācija ar iespējamo iznākumu skaitu m^2. Rezultātu kopa viennozīmīgi atbilst visu alfabēta bigrammu kopai. Modeli apraksta sekojoša vienādība:

$$P(a_0a_1 \dots a_{2n-1}) = \prod_{i=0}^{n-1} P\{x_{2i} = a_{2i}; x_{2i+1} = a_{2i+1}\},$$

kur visiem $a, b \in \mathbb{Z}_m$ un $i \in \{0, 1, \dots, n-1\}$ ir spēkā

$$P\{x_{21} = a; x_{2i+1} = b\} \geq 0;$$
$$\sum_{a,b} P\{x_{2i} = a; x_{2i+1} = b\} = 1.$$

Lai novērtētu bigrammu varbūtības, izmanto to relatīvo parādīšanās biežumu, ko eksperimentāli nosaka izmantojot lielu teksta materiālu. Alfabēta $\mathbb{Z}_m$ bigrammas iespējams uzrādīt $m \times m$ izmēru matricā :

$$\begin{pmatrix} p_{00} & p_{01} & \cdots & p_{0m-1} \\ p_{10} & p_{11} & \cdots & p_{1m-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m-10} & p_{m-11} & \cdots & p_{m-1m-1} \end{pmatrix},$$

kur p_{is} ir varbūtība, ka nejauša teksta nejaušā pozīcijā atradīsies bigramma (i, s). Kriptoanalītiķis no bigrammu varbūtību matricas var iegūt virkni noderīgu īpatnību, kas piemīt avotam un valodai kopumā. Piemēram, angļu valodā visām bigrammām $(q, \dots)$ piemīt nulles varbūtība, atskaitot bigrammu (q, u), kuras parādīšanās varbūtība ir aptuveni $0,0011$. $(a, \dots)$ veida bigrammām ir nenulles varbūtības, atskaitot bigrammu (a, q).

Šāds modelis atļauj klasificēt visas avota bigrammas pēc to varbūtības parādīties tekstā. Aprakstītais modelis precīzāk par iepriekšējo atspoguļo valodu un avotu īpašības. Konkrēti, saskaņā ar šo modeli, jebkuram ziņojumam, kurā nepāra pozīcijā parādās aizliegtas bigrammas pirmais burts, ir nulles ticamība. Tajā pašā laikā modelis ignorē aizliegtas bigrammas, kurās pirmais burts atrodas pāra pozīcijā, un ignorē arī valodām raksturīgo blakus stāvošu bigrammu savstarpējo atkarību. Piemēram, angļu valodā aizliegtās 4-grammas "*ququ*" varbūtība šinī modelī tiek novērtēta kā $1.21 \cdot 10^{-6}$. Minētās nepilnības ir mazāk raksturīgas nākošajam modelim.

Stacionārs Markova nozīmē savstarpēji atkarīgu burtu avots

Šāda avota pamatteksts ir vienkāršas homogēnas Markova ķēdes ar m stāvokļiem eksperimentu realizācijas virkne. Šo modeli (kā arī atbilstošo Markova ķēdi) nosaka pārejas varbūtību matrica Π un sākotnējo varbūtību sadalījuma vektors π:

$$\Pi \leftharpoondown (p(s/t)), \quad 0 \le t, s < m,$$
$$\pi \leftharpoondown (\pi(0), \pi(1), \ldots, \pi(m-1)),$$

kur $\pi(t)$ — varbūtība simbolam t parādīties nejauša teksta pirmajā pozīcijā. Nejauša ziņojuma $a_0 a_1 \ldots a_{n-1}$ parādīšanās varbūtību izsaka formula

$$P(a_0 a_1 \ldots a_{n-1}) = \pi(a_0) \cdot p(a_1/a_0) \cdot p(a_2/a_1) \cdots p(a_{n-1}/a_{n-2}).$$

Pārejas varbūtības un sākotnējo simbolu sadalījums nejaušā tekstā apmierina sekojošus nosacījumus:

1. $p(s/t), \pi(t) \ge 0$ visiem $t, s \in \mathbb{Z}_m$;
2. $\pi(0) + \pi(1) + \cdots + \pi(m-1) = 1$;
3. $p(0/t) + p(1/t) + \cdots + p(m-1/t) = 1$ visiem $t \in \mathbb{Z}_m$.

Stacionāra simbolu sadalījuma vektoru $w = (w(0), w(1), \ldots, w(m-1))$ nejaušam tekstam nosaka atrisinot vienādojumu sistēmu

$$w(s) = \sum_{t=0}^{m-1} w(t) \cdot p(s/t), \quad s \in \mathbb{Z}_m,$$
$$w(0) + w(1) + \cdots + w(m-1) = 1.$$

Saskaņā ar šo modeli, jebkuram ziņojumam, kurš jebkurā vietā satur aizliegtu bigrammu, ir nulles varbūtība parādīties.

Stacionāro modeļu uzlabošana

Izejot no aplūkotajiem modeļiem var piedāvāt tos uzlabot, palielinot burtu varbūtiskās atkarības dziļumu no vairākiem iepriekšējiem burtiem. Šeit var izdalīt divu veida modeļus:

1. stacionārs neatkarīgu ν-grammu avots;

2. stacionārs savstarpēji atkarīgu ν-grammu avots, $\nu > 2$.

Pirmā veida modeļu nosacījumos jebkurš ziņojums ir eksperimentu virknes realizācija polinomiālā varbūtiskā shēmā ar m^ν iespējamiem iznākumiem. Šie modeļi adekvāti ataino starpsimbolu atkarības katras ν-grammas iekšienē, taču ignorē blakus esošu ν-grammu starpsimbolu atkarības. Jo lielāks ν, jo, no vienas puses, aplūkotie modeļi precīzāk attēlo valodu un avotu īpašības, bet no otras puses, jo tie ir lielāki un darbietilpīgāki izmantošanā.

Otrā veida modeļi spēj ņemt vērā blakus esošu ν-grammu starpsimbolu atkarības, taču ir vēl milzīgāki, nekā pirmā veida modeļi. Ja pirmā veida modeļus apraksta m^ν-izmēra varbūtību vektors, tad otrā veida modeļus — matrica ar izmēru $m^\nu \times m^\nu$ un m^ν-izmēra sākotnējā sadalījuma vektors.

Atbilstoša modeļa izvēlei avota pētīšanai ir kompromisa daba un kriptoanalītiķis to veic ņemot vērā konkrēta šifra īpašības.

Nestacionārie ziņojumu avoti

Nestacionāros modeļos ν-grammu parādīšanās varbūtība ir atkarīga no to vietas ziņojumā. Nestacionāros modeļus var aplūkot kā stacionāro modeļu precizējumu, kurā lielākā vai mazākā mērā ņemta vērā ziņojuma struktūra. Piemēram, ja ziņojuma avots ir premjerministrs, bet adresāts — karalis, tad ar lielu varbūtību visi ziņojumi sāksies ar vārdiem "*Jūsu Augstība*!...", bet beigsies ar atbilstošo parakstu. Tamlīdzīgiem standartiem ir būtiska nozīme kriptogrāfiskajā analīzē. Veiksmīgi izvēlēts nestacionārs avota modelis dažos gadījumos var vienkāršot dešifrēšanu, vienkāršojot uzdevumu līdz dešifrēšanai pēc zināma pamatteksta un attiecīga šifrēta teksta.

Aplūkoto pamattekstu avotu modeļi tiek realizēti statistiski apstrādājot avota ģenerētos tekstus. Šiem mērķiem no pietiekoša daudzuma tekstu "izskatīšanas" tiek izskaitļots: varbūtisko modeļu gadījumā — visu izvēlētajam modelim nepieciešamo varbūtību novērtējumi, bet determinēta modeļa gadījumā — visas aizliegtās ν-grammas. Jo lielāks materiāla daudzums ir apstrādāts, jo precīzāks izveidotais avota modelis un efektīvāka tā pielietojamība praktiskos uzdevumos.

Aplūkotie pamattekstu avoti var tikt izmantoti kriptogrāfiskos lietojumos gan dešifrēšanas algoritmos, gan arī lai atšķirtu pamattekstus no nejaušām simbolu virknēm izmantojot skaitļojamo tehniku. Jāņem vērā, ka otro uzdevumu nevar pilnīgi uzticēt skaitļojamajai tehnikai, jo aizliegtu ν-grammu neesamība tekstā negarantē tā jēgpilnību.

4. Šifru kriptogrāfiskā noturība

Šifru kriptogrāfiskā noturība. Šifru varbūtiskie modeļi. Pilnīgi noturīgi šifri. Šifru praktiskās noturības novērtējumi

Šifrus iespējams klasificēt pēc to kriptogrāfiskās noturības, proti, spējas izturēt kriptoanalītiķa uzbrukumus. Jautājums par šifru noturību ir centrālais šifru kriptoanalīzes jautājums. Vai eksistē nevainojami šifri? Kā jābūt uzbūvētam uzticamam šifram? Kā pareizi lietot šifru informācijas aizsardzībai? Šie un daudzi citi tamlīdzīgi jautājumi ir saistīti ar kriptosistēmu kriptogrāfiskās noturības izpēti.

Aplūkosim dažādas pieejas šifru kriptogrāfiskās noturības novērtēšanai.

4.1. Varbūtiskie šifra modeļi

Balstoties uz šifra attēlojumiem un pamattekstu varbūtiskajiem modeļiem, un atslēgu kopas iespējams izveidot šifra varbūtisko modeli. Ieviesīsim dažus apzīmējumus turpmāk lietotajiem varbūtību sadalījumiem:

P_p — pamattekstu kopā balstīts sadalījums;

P_k — atslēgu kopā balstīts sadalījums;

P_c — šifrēto tekstu kopā balstīts sadalījums;

P_{pk} — sadalījums, kas balstīts pamattekstu un atslēgu pāru kopā;

P_{pc} — pamattekstu un šifrēto tekstu pāros balstīts sadalījums;

$P_{p/c}$ — nosacītais varbūtību sadalījums, kas balstīts pamattekstu kopā (pie nosacījuma, ka šifrētais teksts ir fiksēts).

Pieņemsim, ka a — pamatteksts, z — atslēga, y — šifrētais teksts, $E_a(z)$ — kriptogramma, kas iegūta šifrējot pamattekstu a ar atslēgu z. Parasti pieņem, ka šifrējot atslēga z tiek izvēlēta neatkarīgai no pamatteksta a, tāpēc

$$P_{pk}(a, z) = P_p(a) \cdot P_k(z). \tag{2}$$

Pārejie sadalījumi izsakāmi ar formulām:

$$P_c(y) = \sum_{(a,z):E_a(z)=y} P_p(a) \cdot P_k(z), \tag{3}$$

$$P_{pc}(a/y) = \sum_{z:E_a(z)=y} P_p(a) \cdot P_k(z), \tag{4}$$

$$P_{p/c}(a/y) = P_{pc}(a,y)/P_c(y), \tag{5}$$

kur pēdējā vienādība izriet no nosacītās varbūtības definīcijas un izpildās pie noteikuma, ka $P_c(y) > 0$. Tādā veidā, zinot varbūtību sadalījumu daudziem pamattekstiem un atslēgām, principā iespējams izskaitļot gan šifrēto tekstu kopas varbūtību sadalījumu, gan dažādus kopējus un nosacītus varbūtību sadalījumus.

Izmantojot varbūtisko šifra modeli, Šenons pirmo reizi noformulēja pilnīgi noturīga šifra jēdzienu.

4.2. Pilnīgi noturīgi šifri

Pirmais jautājumu par šifru teorētisko noturību formulējis Klods Šenons: "Cik droša ir kriptosistēma, ja uzbrucēja kriptoanalītiķim pieejams neierobežots laiks un resursi kriptogrammu analīzei?" Ar šo jautājumu ir saistīts vēl viens: "Vai eksistē šifri, kurus kriptoanalītiķis nevarētu uzlauzt, ja tam būtu pieejama pēc izvēles liela kriptogramma un neierobežoti skaitļošanas resursi?" Zināms, ka vēl pirms Šenona ļoti tuvu šo jautājumu atrisinājumam bija nonācis amerikāņu inženieris Džilberts Vernams, kurš strādāja kompānijā *AT&T*, kurš 1917. gadā piedāvāja jaunu telegrāfa ziņojumu šifrēšanas veidu. Vernama šifrēšanas algoritma būtība bija, ka pamatteksta attēlojums binārajā kodā bits pēc bita tika summēts ar atslēgu — nejaušu bināru virkni. Pamatteksta un atslēgas bitu saskaitīšana tika veikta saskaņā ar sekojošiem noteikumiem (summēšana pēc moduļa 2):

$$0 + 0 \rightleftharpoons 0; \quad 0 + 1 \rightleftharpoons 1; \quad 1 + 0 \rightleftharpoons 1; \quad 1 + 1 \rightleftharpoons 0.$$

Vernams nojauta, ka viņa piedāvātajam šifram piemīt labas kriptogrāfiskas īpašības, bet nemācēja to stingri pierādīt.

Vernama šifra labās kriptogrāfiskās īpašības izdevās pamatot K.Šenonam, kurš 1949. gadā nopublicēja teorētiskās kriptogrāfijas pamatnostādnes. Izmantojot šifra varbūtisko modeli, Šenons matemātiski definēja pilnīgi noturīgu šifru un parādīja, ka Vernama šifrs ir pilnīgi noturīgs.

Saskaņā ar Šenonu šifrs ir *pilnīgi noturīgs*, ja pamatteksts un šifrētais teksts ir *statistiski neatkarīgi*, proti, katram pamattekstam a un katrai kriptogrammai y

$$P_p(a) = P_{p/c}(a/y) \tag{6}$$

pie nosacījuma, ka $P_c(y) > 0$.

Citiem vārdiem, izmantojot pilnīgi noturīgu šifru pamattekstu kopas varbūtību sadalījums pēc kriptogrammas y pārtveršanas (aposteriorais varbūtību sadalījums) neatšķiras no pamattekstu kopas varbūtību sadalījuma pirms pārtvertās kriptogrammas y saņemšanas (apriorais varbūtību sadalījums). Ziņojuma pārtveršana, kas šifrēts izmantojot pilnīgi drošu šifru, nesatur kriptoanalītiķim noderīgu informāciju, ja nav pieejama atslēga.

Šifru sauc par *ideāli noturīgu*, ja nav iespējams viennozīmīgi noteikt pamattekstu zinot vienalga cik garu šifrēto tekstu. Acīmredzami, pilnīgi noturīgs šifrs ir ideāli noturīgs. Tā kā visiem pamattekstiem a un šifrētiem tekstiem y tādiem, ka $P_p(a) > 0, P_c(y) > 0$ izpildās vienādība

$$P_p(a) \cdot P_{c/p}(y/a) = P_{pc}(a, y) = P_c(y) \cdot P_{p/c}(a/y),$$

tad zināmiem tekstiem a un y vienādība (6), kas nosaka pilnīgi noturīgu šifru, ir ekvivalenta vienādībai

$$P_{c/p}(y/a) = P_c(y). \tag{7}$$

Šenons pierādīja, ka pilnīgi noturīgi šifri eksistē. Kā piemēru aplūkosim tā saukto *Vernama šifru pēc moduļa* m, kurā pamatteksta, šifrētā teksta un atslēgu simboli ir moduļu gredzena $\mathbb{Z}_m$ vērtības, bet atslēgas un kriptogrammas garums sakrīt ar pamatteksta garumu n.

- Pamatteksta $a = (a_1, a_2, \ldots, a_n)$ n–grammas šifrēšanas operāciju,
- izmantojot atslēgas $z = (z_1, z_2, \ldots, z_n)$ n–grammu, kuras rezultātā
- tiek iegūta šifrētā teksta $y = (y_1, y_2, \ldots, y_n)$ n–gramma,

nosaka vienādojums

$$y_i = a_i + z_i \mod m, \quad i \in \overline{1, n}. \tag{8}$$

Teorēma 4.1. *Vernama šifrs pēc moduļa m ir pilnīgi drošs šifrs, ja atslēga ir izvēlēta pilnīgi nejauši no visu n-grammu kopas $\mathbb{Z}_m^n$ alfabētā $\mathbb{Z}_m$.*

□ Pieņemsim, ka $a, y \in \mathbb{Z}_m^n$ un $P_p(a) > 0$. Ja atslēga z tiek izvēlēta pilnīgi nejauši (visām atslēgām vienāda varbūtība tikt izvēlētām), tad

$$P_k(z) = m^{-n}. \tag{9}$$

Tā kā dotiem tekstiem a un y atbilst viena vienīga atslēga z, kas apmierina vienādojumus (8), tad no (4) un (9) iegūstam:

$$P_{pc}(a, y) = P_p(a) \cdot P_k(z) = P_p(a) \cdot m^{-n}. \tag{10}$$

Tā kā $P_c(y) > 0$, tad ir spēkā Beijesa formula:

$$P_{p/c}(a/y) = \frac{P_p(a) \cdot P_{c/p}(y/a)}{\sum\limits_{b \in \mathbb{Z}_m^n} P_p(b) \cdot P_{c/p}(y/b)}\,.$$

Tagad izmantojam jau pieminēto vienādību

$$P_p(a) \cdot P_{c/p}(y/a) = P_{pc}(a, y)\,:$$

$$\frac{P_p(a) \cdot P_{c/p}(y/a)}{\sum\limits_{b \in \mathbb{Z}_m^n} P_p(b) \cdot P_{c/p}(y/b)} = \frac{P_{pc}(a, y)}{\sum\limits_{b \in \mathbb{Z}_m^n} P_{pc}(b, y)} = \frac{P_p(a) \cdot m^{-n}}{m^{-n} \sum\limits_{b \in \mathbb{Z}_m^n} P_p(b)}\,.$$

Visbeidzot ņemam vērā, ka $\sum\limits_{b \in \mathbb{Z}_m^n} P_p(b) = 1$, tāpēc

$$P_{p/c}(a/y) = \frac{P_p(a) \cdot P_{c/p}(y/a)}{\sum\limits_{b \in \mathbb{Z}_m^n} P_p(b) \cdot P_{c/p}(y/b)} = \frac{P_p(a) \cdot m^{-n}}{m^{-n} \sum\limits_{b \in \mathbb{Z}_m^n} P_p(b)} = P_p(a)\,,$$

kas arī bija jāpierāda. ■

Atzīmēsim dažas svarīgas Vernama šifra pēc moduļa m īpašības.

1. Vienādība (10) nozīmē, ka zinot pamattekstu visas kriptogrammas ir vienādi iespējamas.

2. No vienādības (7) seko, ka arī nezinot pamattekstu visas kriptogrammas ir vienādi iespējamas.

Bez šīm lieliskajām īpašībām jāatzīmē, ka atslēgas garums pilnīgi noturīgos šifros sakrīt ar ziņojuma garumu. Tas nozīmē, ka šādu šifru izmantošana

lielu informācijas apjomu aizsardzībai prasa milzīgus darba ieguldījumus, kas saistīti ar atslēgu veidošanu, izplatīšanu un uzglabāšanu.

Neskatoties uz to pilnīgi noturīgi šifri tomēr ir atraduši praktisku pielietojumu īpaši svarīgu sakaru līniju ar salīdzinoši nelielu pārraidīto datu apjomu aizsardzībā. Vācieši pagājušā gadsimta divdesmitajos gados aprīkoja savas diplomātiskās pārstāvniecības ar pilnīgi noturīgiem Vernama tipa šifriem, bet angļi un amerikāņi sāka izmantot Vernama tipa šifrus otrā pasaules kara laikā. Vairāku valstu spiegi izmantoja šifrbloknotus, kas saturēja atslēgas informāciju saziņai izmantojot Vernama šifru (kā likums pēc moduļa 10). Vernama šifrs pēc moduļa 2 tika izmantots "karstajai līnijai" starp Vašingtonu un Maskavu, bet atslēgu materiāls bija papīra lentas (kas tika izlaistas divos eksemplāros), uz kurām atslēgas zīmes tika uzperforētas.

Eksistē ar šifrbloknota "datorversija", saskaņā ar kuru tiek izmantotas atslēgu virknes, kas ir ierakstītas datora atmiņā. Tiesa šādā variantā parādās problēma, kā aizsargāt atslēgu virknes pret kropļojumiem, aizvietošanas un citiem iespējamiem draudiem.

4.3. Sistemātiska pieeja praktiskai šifru noturības novērtēšanai

Jautājums par praktisko noturību, kuru uzdeva Šenons, formulēts šādi:

— Vai kriptosistēma ir noturīga, ja kriptoanalītiķim ir pieejams ierobežots laiks un resursi pārtverto kriptogrammu analīzei? — Šis jautājums ir cieši saistīts ar kriptosistēmu konstruēšanas problēmu.

No vienas puses, kriptogrāfiskai sistēmai jānodrošina uzticamu informācijas aizsardzību, bet no otras puses, tai jābūt ērtai no tehniskās realizācijas un ekspluatācijas viedokļa. Tā kā kriptosistēmas, kas nodrošina ideālu slepenību, vairumā gadījumu nav pielietojamas, tad jautājums pirmām kārtām attiecas uz kriptosistēmām, kas izmanto ierobežota izmēra atslēgas, un spēj apstrādāt lielus informācijas daudzumus.

Saskaņā ar Šenonu, praktiski noturīgai kriptosistēmai pēc savām īpašībām jābūt tuvai ideālām kriptosistēmām, proti, jābūt sava veida mākslīgai ideāla šifra imitācijai. Šo īpašību izsaka šifru attēlojumu līdzība analoģiskiem nejaušiem attēlojumiem. Piemēram, augsta gammēšanas šifra noturība tiek nodrošināta izmantojot šifrējošu virkni, kas pēc savām īpašībām tuva vienmērīgi sadalītu nejaušu gadījuma lielumu virknei, tāpēc gammēšanas šifra kriptogrāfiskās īpašības nosaka izmantotā gamma ģeneratora īpašības.

Sistemātiska pieeja šifru noturības novērtēšanai paredz kaut kādu jēdziena "noturīgs šifrs" detalizāciju. Šīs detalizācijas rezultātā veidojas virkne matemātiska un tehniska rakstura kritēriju, kurus jāapmierina noturīgai kriptosistēmai. Izstrādājot jaunu pieeju kriptosistēmu analīzei veidojas attiecīgs kriptosistēmas kvalitātes kritērijs, kas papildina agrāko kritēriju sistēmu.

Galvenais skaitliskais kriptogrāfiskās noturības rādītājs ir dešifrēšanas uzdevumu risinājumu *skaitļošanas sarežģītība*. Skaitļošanas sarežģītību nosaka vairāki raksturlielumi. Aplūkosim svarīgākos no tiem.

Pieņemsim, ka kriptoanalītiķim uzdots dešifrēt šifru E izmantojot kaut kādu kriptogrammu kopumu. Pieņemsim, ka A_E — šifram E izmantojamo dešifrēšanas algoritmu klase, kas ir pieejami kriptoanalītiķim. Pie tam kriptoanalītiķis aplūko atslēgu un pamattekstu pāru kopu, ja pamatteksti nav zināmi, vai atslēgu kopu, ja pamatteksti ir zināmi, kā elementāru notikumu varbūtisku telpu W. Apzīmēsim algoritma $\Psi \in A_E$ vidējo realizācijas darbietilpību ar $T(\Psi)$, ko mērīsim kaut kādās nosacītās skaitļošanas operācijās. Pie tam darbietilpība parasti tiecas uz vidējo vērtību, ja aplūko visu kopu W.Viens no galvenajiem šifra E noturības raksturlielumiem ir vidējā dešifrēšanas darbietilpība T_E, ko nosaka izteiksme

$$T_E = \min_{\Psi \in A_E} T(\Psi). \tag{11}$$

Pie tam, svarīgi atzīmēt sekojošo:

1. Eksistē dešifrēšanas algoritmi, kas netiek definēti visā varbūtību telpā W, bet tikai kādā tās daļā. Bez tam, daži dešifrēšanas algoritmi veidoti tā, ka to realizācija noved pie veiksmīga rezultāta (dešifrēšanas uzdevuma risinājuma) ne visā definīcijas apgabalā, bet tikai kaut kādā tā apakškopā. Līdz ar to viens no svarīgākajiem dešifrēšanas algoritma raksturlielumiem ir *lietojamība* $\nu(\Psi)$. Ar šo terminu saprot vidējo informācijas daļu, ko dešifrē algoritms Ψ. Ja algoritma dešifrēšanas lietojamība ir maza, tad no kriptogrāfa viedokļa tas nav bīstams, bet no kriptoanalītiķa viedokļa — neefektīvs. Tādā veidā, iegūstot vērtības T_E novērtējumu (11) ir lietderīgi aplūkot tikai tos algoritmus, kuru lietojamība ir pietiekoši liela. Pie tam, lai noteiktu "labāko" sistēmas E dešifrēšanas algoritmu var izmantot dažādus kritērijus atkarībā no konkrētiem uzdevuma nosacījumiem. Piemēram, par "labāko" var uzskatīt dešifrēšanas algoritmu Ψ, kuram vērtība $T(\Psi)/\nu(\Psi)$ ir vismazākā. Šo lielumu var interpretēt kā vidējo darbietilpību, kas nepieciešama, lai veiksmīgi dešifrētu kriptosistēmu.

2. Dešifrēšanas sarežģītība ir atkarīga no kriptoanalītiķim pieejamo krip-

togrammu kvantitatīvām un kvalitatīvām īpašībām. Kvantitatīvās īpašības nosaka pārtverto kriptogrammu skaits un to garums. Kvalitatīvās īpašības saistītas ar pārtverto kriptogrammu uzticamību (kropļojumu iespējamība, izlaisti gabali, utt.).

Saskaņā ar Šenonu, katram šifram piemīt objektīvs raksturlielums $T_E(n)$ — vidējā (visu kriptogrammu ar garumu n un atslēgu) dešifrēšanas skaitļošanas sarežģītība. Lielums $\lim\limits_{n\to\infty} T_E(n)$ raksturo sistēmas E dešifrēšanas robežas iespējas, ja pieejami neierobežoti šifrēta materiāla daudzumi un kriptoanalītiķis ir absolūti kvalificēts.

Novērtējot šifra noturību, kriptoanalītiķis iegūst robežas noturības augšējos novērtējumus, jo praktiskai dešifrēšanai tiek izmantots ierobežots daudzums šifrēta materiāla un ierobežota tā saucamo *zināmo dešifrēšanas metožu* klase.

3. Svarīgs sistēmas kriptogrāfiskās noturības rādītājs ir tās dešifrēšanas *sarežģītība laikā*. Sarežģītības laikā novērtēšana paredz detalizētāku dešifrēšanas algoritmu realizācijas aplūkošanu, ņemot vērā skaitļojamās tehnikas parametrus, kas tiek izmantota dešifrēšanai. Pie šādiem skaitļojamās tehnikas parametriem pieder tās arhitektūra, ātrdarbība, atmiņas apjoms un struktūra, piekļūšanas ātrums atmiņai, un citi. No tā izriet, ka sistēmas E dešifrēšanas laiks atkarīgs no dešifrēšanas algoritmu klases A_E un kriptoanalītiķa skaitļošanas iespējām. Labākā dešifrēšanas algoritma izvēli sarežģī vēl arī tas, ka dažādām skaitļošanas iekārtām var atbilst dažādi "labākie" dešifrēšanas algoritmi.

Jautājumam par kriptogrāfisko noturību ir vairākas īpatnības raugoties gan no kriptoanalītiķa, gan kriptogrāfa viedokļa. Kriptoanalītiķis uzbrūk kriptosistēmai, izmantojot konkrētus intelektuālus, skaitļošanas un ekonomiskus resursus. Viņa mērķis — veiksmīgi dešifrēt sistēmu.

Kriptogrāfs novērtē kriptosistēmas noturību, imitējot uzbrucēja kriptoanalītiķa uzbrukumu šifram. Lai to izdarītu, kriptogrāfs modelē kriptoanalītiķa darbības, maksimāli augstu novērtējot intelektuālās, skaitļošanas, tehniskās un citas pretinieka iespējas. Kriptogrāfa mērķis — pārliecināties, ka izstrādātajai kriptosistēmai piemīt augsta kriptogrāfiskā noturība.

Tā rezultātā sistemātiska pieeja praktiskai šifra noturībai saistīta ar skaitļošanas darbietilpības novērtēšanu dešifrējot sistēmu no dažādu šifra kvalitātes kritēriju pozīcijām. Izmantojot šifra praktiskās noturības jēdzienu, šifrus var klasificēt pēc noturības parametra vai pēc laika perioda ilguma, kurā šifrs ar augstu uzticamību nodrošina nepieciešamo informācijas aizsardzības

līmeni.

4.4. Citas pieejas šifru praktiskās noturības novērtēšanai

Asimptotiskā noturības analīze

Šo pieeju attīsta algoritmu sarežģītības teorija. Pētot šifru tā noturības novērtējums tiek sasaistīts ar kādu šifra parametru, kas parasti ir atslēgas garums, un tiek veikta asimptotiska noturības novērtējumu analīze. Parasti uzskata, ka kriptosistēmai ir augsta noturība, ja to var izteikt caur atslēgas garumu eksponenciālā veidā, un kriptosistēmai ir zema kriptogrāfiskā noturība, ja noturību var izteikt kā polinomu no atslēgas.

Nepieciešamā šifra materiāla daudzuma novērtējums

Šī pieeja ir balstīta nevis uz dešifrēšanai nepieciešamās skaitļošanas sarežģītību, bet uz vidējā šifrētā materiāla daudzumu, kas kriptoanalītiķim nepieciešams, lai atklātu šifru. Kriptoanalītiķim nepieciešamā šifrētā materiāla daudzuma analīze ir interesanta no tā viedokļa, ka tā ir šifra zemākās noturības novērtējums dešifrēšanas sarežģītības nozīmē.

Šī pieeja tiek pamatā lietota, lai novērtētu plūsmas *randomizētus šifrus*. Šādu šifru uzbūves īpatnība ir tā, ka tie šifrēšanai un dešifrēšanai izmanto neliela izmēra atslēgu, kā arī lielu un visiem pieejamu nejaušu skaitļu virkni (*randomizātoru*). Atslēga nosaka, kādas randomizātora daļas tiek izmantotas šifrēšanai, kamēr kriptoanalītiķim, kurš nezin slepeno atslēgu, nākas analizēt visu randomizātoru. Kā šāda šifra piemēru aplūkosim Diffi šifru. Šī šifra gadījumā randomizātors ir 2^n nejaušu bināru virkņu masīvs, kas sastāv no sanumurētiem kopas $\{0,1\}^n$ elementiem. Atslēga ir binārs vektors ar garumu n. Šifrējot ar atslēgu k binārā pamatteksta secība tiek summēta pa bitiem (kā Vernama šifrā) izmantojot randomizātoru ar numuru k. Tādā veidā, lai dešifrētu ziņojumu, pretiniekam jāizpēta ar kārtu 2^n bitu. Vēlāk Ruppels aizrādīja, ka apmēram tāds pats noturības līmenis tiek sasniegts, ja randomizators satur n nejaušas bināras virknes, bet atslēga k uzdota kā koeficientu kortežs, kas nosaka šifrēšanas secību kā netriviālu randomizātora virkņu lineāru kombināciju.

Izmaksu pieeja

Šī pieeja paredz sistēmas dešifrēšanas izmaksu novērtēšanu. Tā ir īpaši aktuāla, kad kriptosistēmas dešifrēšanai jāizstrādā un jāuzbūvē jauns skaitļošanas kompleks. Izmaksu pieeja ir noderīga salīdzinot materiālos tēriņus sistēmas dešifrēšanai ar informācijas vērtību, kas tiek aizsargāta izmantojot kriptosistēmu. Šādas pieejas realizācijas piemērs ir visai precīzais algoritma *DES* dešifrēšanas izmaksu novērtējums, ko veica Diffijs un Hellmans sakarā ar visu *DES* atslēgu paralelizētas pārlases algoritmu.

Nobeigumā atzīmēsim šifru kriptogrāfiskās noturības novērtējuma dinamisko raksturu. Šos novērtējumus ik pēc kāda laika nepieciešams pārskatīt sakarā ar skaitļojamo līdzekļu attīstību un dešifrēšanas metožu attīstību. Konkrēti Šneiers piemin skaitļojamās tehnikas attīstības "empīrisko likumu", saskaņā ar kuru uzskata, ka kriptoanalītiķa skaitļošanas iespējas dubultojas katros 18 mēnešos.

5. Transpozīciju un substitūciju šifri

Substitūciju šifri. Kardano siets. Cēzara šifrs, Viženera tabula. Hilla šifrs.

Kriptosistēmas var klasificēt pēc dažādām pazīmēm: pēc aizsargājamās informācijas veida (teksts, runa, video), pēc kriptogrāfiskās noturības, pēc informācijas aizsardzības nodrošināšanas principiem (simetriskie, asimetriskie, hibrīda), pēc uzbūves principiem (bloka un plūsmas šifri) un citām. Veidojot šifra attēlojumus no matemātiskā redzes viedokļa tiek izmantoti divi attēlojumu veidi: pamatteksta elementu pārkārtošana un pamatteksta elementu aizvietošana ar kaut kādas kopas elementiem. Šādā nozīmē visu šifru kopu var sadalīt 3 veidu šifros: *transpozīcijas šifri*, *substitūcijas šifri* un *kompozītie šifri*, kas izmanto gan pārkārtošanu, gan aizvietošanu. Aplūkosim nedaudz detalizētāk šīs šifru klases.

5.1. Transpozīcijas šifri

Pieņemsim, ka dots alfabēta $\mathbb{Z}_m$ pamatteksts

$$a_1 a_2 \dots a_n$$

garumā n. Ja šādam ziņojumam pielietojam substitūciju σ, kur

$$\sigma = \begin{pmatrix} 1 & 2 & \dots & n \\ \sigma(1) & \sigma(2) & \dots & \sigma(n) \end{pmatrix} \in \mathfrak{S}_n,$$

tad rezultāts būs kriptogramma

$$a_{\sigma(1)} a_{\sigma(2)} \dots, a_{\sigma(n)}.$$

Šādu pārveidojumu saimi sauc par *transpozīcijas šifru*. Transpozīcija izpaužas kā pamatteksta burtu pārkārtošana, kuras rezultātā to vairs nav iespējams izlasīt. Šāda šifra pārveidojuma atslēga ir izmantotais pārkārtojums σ. Kriptogrammu var atšifrēt, izmantojot apgriezto attēlojumu σ^{-1}.

Aprakstītā transpozīcijas šifra modeļa īpatnība ir tā, ka pamatteksta un atslēgas garumi sakrīt. No praktiskās realizācijas viedokļa tas ir ļoti nepraktiski, tāpēc, kā likums, transpozīcijas šifri izmanto atslēgu ar fiksētu garumu l, pie tam pamatteksts tiek sadalīts $\lceil \frac{n}{l} \rceil$ gabaliņos ar garumu l, bet pēc tam katram gabaliņam pielieto transpozīcijas šifru.

Svarīga transpozīcijas šifru īpatnība no kriptoanalīzes viedokļa ir tā, ka sastopamo simbolu relatīvie biežumi pamattekstā un šifrētajā tekstā sakrīt, jo tie ir invarianti jebkurai transpozīcijai. Otra svarīga īpašība ir izmantojamās atslēgas garuma ierobežotība, kas, it īpaši šifrējot garus tekstus, noved pie tā, ka atslēga tiek izmantota daudzas reizes, kas atvieglo ziņojuma dešifrēšanu. Tāpēc transpozīcijas ir prātīgi izmantot kopā ar substitūcijām kompozīcijas šifros.

Elektroniskos šifros transpozīcijas tiek veiktas ar atmiņas palīdzību dažādu aiztures līniju veidā. Šifrējot ar roku, proti, "pirms mašīnu ēras" šifros, transpozīcijām bija savs pielietojums.

Šifrēšana ar spoles palīdzību

Kopš antīkiem laikiem (11.gs.p.m.ē.) ir pazīstams transpozīcijas šifrs, ko realizē izmantojot *spoli* — koka cilindru, uz kura uztīta siksna tā, ka tā nepārklājas, bet tās malas pieguļ viena otrai. Ziņojumu rakstīja uz uztītās siksnas spoles ass virzienā, bet pēc tam siksnu attina un ar to apjoza ziņnesi. Uz ziņneša jostas ziņojums izskatījās kā nelasāms burtu savākums. Šī šifra atslēga bija spoles diametrs. Lai atšifrētu ziņojumu, ziņnesim noņēma siksnu un uztina to uz vajadzīgā diametra spoles.

Interesanti, ka jau senos laikos tika izgudrots arī oriģināls dešifrēšanas veids — siksnu uztina uz koniskas spoles ar nelielu konusa leņķi. Tā konusa daļa, kurā šifru varēja izlasīt, norādīja uz atšifrēšanai nepieciešamo diametru.

Maršruta šifri

Cits piemērs ir *maršruta transpozīcijas šifri*, kas izmantoja taisnstūra tabulu, kurā pamatteksts tiek pierakstīts pa rindiņām, bet nolasīts tiek citādā kārtībā (pa stabiņiem, pa diagonālēm, ... , proti pa citu maršrutu). Atšifrēšana sastāv no tāda paša izmēra tukša taisnstūra aizpildīšanas ar šifrēto tekstu pa izvēlēto maršrutu un atšifrētā teksta nolasīšanas pa rindiņām. Šādu šifru atslēga ir tabulas izmērs un tabulas burtu nolasīšanas maršruts.

Kardano siets

Visizturīgākais no manuālajiem transpozīcijas šifrēšanas veidiem ar tabulu ir šifrēšana izmantojot *"Kardano sietu"*, kas pazīstams jau vairāk nekā 400 gadu. "Kardano siets" tiek izmantots kā šifrēšanas atslēga, un tas ir

kvadrātiska tabula ar izmēru $2n \times 2n$, kur n — naturāls skaitlis, kurā ceturtā daļa pozīciju (kopumā n^2 pozīciju) izdalītas pamatteksta pierakstam.

Praktiski "Kardano siets" tiek realizēts kā kartona kvadrāts ar izmēriem $2n \times 2n$ rūtiņu, no kurām n^2 ir izgrieztas kā lodziņi. Šie n^2 lodziņi nejaušā veidā vairāk vai mazāk vienmērīgi sadalīti pa kvadrāta laukumu, un izvēlēti tādā veidā, lai pagriežot kvadrātu ap tā ģeometrisko centru par $90°, 180°, 270°$ un $360°$ iespējams noklāt visus $4n^2$ kvadrāta lauciņus.

Lai šifrētu "Kardano siets" tiek uzlikts uz tāda paša izmēra kvadrāta (kas veidots no, piemēram, papīra) ar tukšiem lauciņiem. Pirmajā stāvoklī, kas atbilst leņķim $0°$, papīra kvadrāta lauciņos, kuri redzami "Kardano sieta" "lodziņos", secīgi ieraksta pirmos n^2 pamatteksta simbolus. Pēc tam "Kardano siets" tiek pagriezts $90°$ stāvoklī, un atkal caur lodziņiem tiek ierakstīti nākošie n^2 pamatteksta simboli. Pēc tam "Kardano siets" tiek pagriezts $180°$ stāvoklī un ierakstīti nākošie n^2 ziņojuma simboli, un visbeidzot stāvoklī $270°$ tiek ierakstīti pēdējie n^2 pamatteksta simboli. Ja šifrējamā teksta garums ir lielāks par $4n^2$, tad "Kardano siets" tiek izmantots šifrēšanai vairākas reizes. Šifrētais teksts tiek iegūts noņemot "Kardano sietu" un nolasot simbolus no papīra kvadrāta tādā vai savādākā kārtībā.

Tālāks manuālo transpozīcijas šifru uzlabojums bija dubultās (secīgās) transpozīcijas šifri. Šajos šifros tiek izmantots atslēgu pāris, kur pirmā no tām pārveido pamattekstu par starprezultāta tekstu, bet otra pārveido starprezultātu par šifrēto tekstu.

5.2. Substitūcijas šifri

Pieņemsim, ka dots alfabēta $\mathbb{Z}_{m^k}$ pamatteksts

$$a_1 a_2 \dots a_n$$

garumā n un attēlojumu kortežs

$$\Phi = \langle \phi_1, \phi_2, \dots, \phi_n \rangle,$$

kur

$$\forall i \in \overline{1,n} \quad \phi_i : \mathbb{Z}_{m^k} \to Y\,;$$

te Y — šifrētā teksta alfabēts.

Kriptogrammu y pamattekstam $a_1 a_2 \dots a_n$ aprēķina izmantojot attēlojumu kortežu Φ, kur Φ dotajā gadījumā izpilda atslēgas funkciju:

$$y = \phi_1(a_1)\phi_2(a_2)\dots\phi_n(a_n).$$

Ja ϕ_i — bijektīvi attēlojumi, tad attēlojumu kortežu Φ sauc par *aizvietojošu substitūcijas šifru*. Ja ϕ_i — neviennozīmīgi attēlojumi, kuriem piemīt īpašība

$$\forall i \in \overline{1,n} \quad (a \neq b \Rightarrow \{\phi_i(a)\} \cap \{\phi_i(b)\} = \varnothing),$$

tad kortežu Φ sauc par *neviennozīmīgu substitūcijas šifru*.

Aplūkosim aizvietojošus substitūcijas šifrus. Nezaudējot apskata vispārīgumu var uzskatīt, ka $Y = \mathbb{Z}_{m^k}$ un $\phi_1, \phi_2, \ldots, \phi_n$ — kopas $\mathbb{Z}_{m^k}$ bijekcijas. Ja $\phi_i = \phi$ visiem $i \in \overline{1,n}$, tad substitūcijas šifru sauc par *monoalfabētisku aizvietošanu (parastu substitūciju)* kopā $\mathbb{Z}_{m^k}$. Pretējā gadījumā šifru sauc par *polialfabētisku aizvietošanu (kolonnu aizvietošanas šifrs)*. Ja $k = 1$, tad saka, ka šifrs realizē *simbolu substitūciju*, ja $k = 2$, tad — šifrs realizē *bigrammu substitūciju*, utt.

Tagad pievērsīsim uzmanību dažām pazīstamākām substitūcijas šifru klasēm.

Cēzara šifrs

Cēzara šifrs ir monoalfabētiska aizvietošana, kas alfabēta $\mathbb{Z}_m$ pamattekstu

$$a_1 a_2 \ldots a_n$$

pārveido par kriptogrammu

$$T^j(a_1)T^j(a_2)\ldots T^j(a_n);$$

te

$$T^j(a) \rightleftharpoons a + j \mod m.$$

Cēzars parasti šifrēšanai izmantoja aizvietošanu T^3. Cēzara šifru var salauzt pārlasot visas atslēgas, kuru skaits ir mazāks par m (nulles nobīdi var izslēgt).

Vižinera tabula

Šo šifru raksturo $m \times m$ izmēra tabula, kuras i-tajā rindiņā pierakstīta alfabēta $\mathbb{Z}_m$ nobīdes aizvietošana T^i. Parasti Vižinera tabula izmanto, lai radītu polialfabētisku aizvietošanu. Katra pamatteksta zīme tiek šifrēta ar vienas Vižinera tabulas rindiņas palīdzību. Kārtējās rindiņas izvēle tiek veikta atbilstoši kaut kādam likumam. Tā rezultātā pamattekstu

$$a_1 a_2 \ldots a_n$$

pārveido par kriptogrammu

$$T^{k_1}(a_1)T^{k_2}(a_2)\dots T^{k_n}(a_n);$$

te visi $k_i \in \overline{1,m}$.

Vižinera tabula ir speciālgadījums tabulām, kuras sauc par *latīņu kvadrātiem*.

Definīcija 5.1. *Tabulu*

$$\begin{array}{cccccc} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{array}$$

sauc par m–tās kārtas latīņu kvadrātu, ja

$$\forall i\ \exists \sigma \in \mathfrak{S}_m\ \forall j \quad a_{ij} = \sigma(j)$$

un

$$\forall j\ \exists \tau \in \mathfrak{S}_m\ \forall i \quad a_{ij} = \tau(i)\,.$$

Tātad latīņu kvadrāta katra rindiņa ir rindiņas $1\ 2\ \dots\ m$ pārkārtojums, kā arī katra aile ir ailes

$$\begin{array}{c} 1 \\ 2 \\ \vdots \\ m \end{array}$$

pārkārtojums. Vispārīgā gadījumā kopas $\{1, 2, \dots, m\}$ vietā var būt jebkura m–elementīga kopa $\{a_1, a_2, \dots, a_m\}$, tad latīņu kvadrāta katra rindiņa ir rindiņas $a_1\ a_2\ \dots\ a_m$ pārkārtojums, kā arī katra aile ir ailes

$$\begin{array}{c} a_1 \\ a_2 \\ \vdots \\ a_m \end{array}$$

pārkārtojums..

Visu iespējamo latīņu kvadrātu ar kārtu m pārlase ir sarežģīts kombinatorisks uzdevums. Taču var parādīt metodi, kā uzbūvēt noteiktu latīņu kvadrātu klasi.

1. Katrai alfabēta $\mathbb{Z}_m$ bijekcijai X, $X \cdot T^i$, $\quad i = \ 0, 1, \ldots, m-1$ veido m–tās kārtas latīņu kvadrātu.

2. Ja L ir m–tās kārtas latīņu kvadrāts, tad $X \cdot L \cdot Y$ arī ir m–tās kārtas latīņu kvadrāts, kur X, Y — patvaļīgas m–tās kārtas substitūciju matricas.

Viens no veidiem kā uzlabot Cēzara šifru ir Vižinera tabula uz sajaukta (aizvietota) alfabēta pamata. Šādas sistēmas atslēga ir pāris (k_1, k_2), kur k_1 — lozungs vai multigramma, kas nesatur atkārtojošos alfabēta simbolus un tiek pierakstīta pirmās rindiņas sākuma pozīcijās, $k_2 \in \mathbb{Z}_m$. Pārējās pirmās rindiņas pozīcijas tiek aizpildītas ar alfabēta burtiem, kas neietilpst lozungā, piemēram,

$$lozungsz_7 z_8 \ldots z_{m-1}.$$

Ja ar $\phi(k_1)$ apzīmējam aizvietošanas likumu, kura apakšējā rindiņa sakrīt ar pirmo, piemēram,

$$\phi(k_1) = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \ldots & m-1 \\ l & o & z & u & n & g & s & z_7 & z_8 & \ldots & z_{m-1} \end{pmatrix},$$

tad visa iegūtā latīņu kvadrāta aizvietošanas sistēma izskatās sekojoši:

$$\{T^i \cdot \phi(k_1)\}, \quad i = 0, 1, \ldots, m-1. \tag{12}$$

Atslēgas (k_1, k_2) izvēle nozīmē, ka šifrēšanai tiek izmantots aizvietošanas likums $T^{k_2} \cdot \phi(k_1)$, proti, dotais šifrs ir vienkāršas substitūcijas šifrs.

Galvenā vienkāršas substitūcijas šifru īpatnība, kas nosaka to kriptogrāfisko vājumu, ir tā, ka pamatteksta un šifrētā teksta variāciju rindas sakrīt ar precizitāti līdz kaut kādam aizvietošanas likumam (kas arī ir atslēga!). Šī šifru sistēma tiek uzlauzta salīdzinot pamatteksta un šifrētā teksta variāciju rindas. Pie tam bieži tiek izmantoti papildus apsvērumi, tādi kā, "aizliegtās multigrammas" vai "sagaidāmie vārdi".

Uzlabots uz latīņu kvadrātiem balstītu šifru variants ir šifri ar kolonnu substitūciju, kuros šifrējošās aizvietošanas tiek veiktas saskaņā ar kaut kādu atslēgas vārdu, ko sauc par *lozungu*. Šādas kriptosistēmas atslēga ir pāris (K_1, K_2), kur K_1 un K_2 — alfabēta $\mathbb{Z}_m$ multigrammas. Multigrammas K_1 būtība ir tāda pati kā iepriekšējā piemērā, proti, pilna latīņu kvadrāta aizvietošanas sistēma, kā tas parādīts (12). Ja lozungs $K_2 = b_1 b_2 \ldots b_t$, tad šifrējošo aizvietošanu virknei ir periods t un tā izskatās sekojoši:

$$T^{b_1} \cdot \phi(K_1)\ T^{b_2} \cdot \phi(K_1)\ \ldots\ T^{b_t} \cdot \phi(K_1)\,.$$

Proporcionālās aizvietošanas šifri

Šādu nosaukumu ieguvuši neviennozīmīgas aizvietošanas šifri, kas tika izstrādāti, lai uzlabotu vienkāršās aizvietošanas šifrus. Lai apgrūtinātu uz biežuma analīzi balstītu dešifrēšanu, šajos šifros dažiem pamatteksta burtiem $a \in \mathbb{Z}_m$ tiek piekārtotas vairākas alfabēta Y vērtības šifrētajā tekstā (skaidrs, ka $|Y| > m$). Pie tam pieļaujamo burta a attēlojumu skaits kriptogrammās būs aptuveni proporcionāls burta a izmantošanas biežumam pamattekstos, taču šifrējot burta a attēlojums tiek izvēlēts ar nejaušas procedūras palīdzību no iespējamo attēlojumu kopas. Principiālā pieeja šādu šifrēšanas sistēmu dešifrēšanai ir tāda pati kā vienkāršās substitūcijas šifriem. Taču nepieciešamā materiāla daudzums pieaug.

Bigrammu substitūcija

Lai apgrūtinātu kriptoanalīzi, kas balstīta uz burtu parādīšanās biežumiem tekstā, tika izmantoti substitūcijas šifri, kas realizē bigrammu, trigrammu, utt. šifrēšanu, jo k–grammu parādīšanās biežumi tekstos kā likums ir ievērojami mazāki nekā $(k-1)$–grammu parādīšanās biežumi.

Bigrammu substitūcijas šifra piemērs ir šifrs *Playfair*, kas tika izgudrots 1854. g., kuru briti izmantoja pirmā pasaules kara laikā. Lai šifrētu tekstu, kas pierakstīts alfabētā $\mathbb{Z}_m$, tiek izmantota atslēga, kas ir taisnstūris ar izmēriem $n \times r$, kur $n \cdot r = m$. Tabulas rindās ierakstīts patvaļīgs alfabēta $\mathbb{Z}_m$ aizvietošanas likums. Konkrēti, pirmajās l tabulas pozīcijās var tikt ierakstīts lozungs, kas sastāv no l atšķirīgiem alfabēta burtiem, $m \geq l$, bet pārējās $m-l$ pozīcijās tiek ierakstīti lozungā neizmantotie alfabēta elementi. Iegūto matricu apzīmēsim ar Γ:

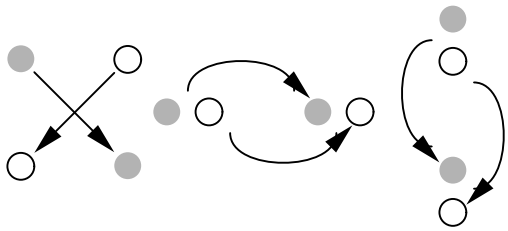
$$\Gamma = (\gamma_{ij}); \quad i = i_0, i_1, \ldots, i_{n-1}; \quad j = j_0, j_1, \ldots, j_{r-1}.$$

Tādā veidā, katram alfabēta $\mathbb{Z}_m$ elementam viennozīmīgi atbilst skaitļu pāris (i, j) — šī elementa koordinātas matricā Γ. Pamatteksts tiek sadalīts bigrammās un vienlaicīgi tiek izmainīts tā, lai modificētajā tekstā nebūtu sastopamas (a, a) veida bigrammas. Piemēram, ja pamattekstā ir sastopama (a, a) veida bigramma, kur $a \neq 0$, tad starp šīs bigrammas burtiem tiek ievietots simbols 0. Ja pamattekstā ir sastopama bigramma $(0, 0)$, tad starp nullēm tiek ievietots cits burts. Proti, modificētā teksta garums var pārsniegt

pamatteksta garumu. Šifrēšanas funkcija ϕ tiek definēta sekojošā veidā:

$$\phi(a,b) = \phi(\gamma_{ij}, \gamma_{st}) = \begin{cases} (\gamma_{it}, \gamma_{sj}), & i \neq s, j \neq t \\ (\gamma_{ij+1}, \gamma_{it+1}), & i = s, j \neq t \\ (\gamma_{i+1t}, \gamma_{s+1t}), & i \neq s, j = t \end{cases} \tag{13}$$

Rindiņu numuri tiek aprēķināti pēc moduļa n, bet stabiņu numuri — pēc moduļa r. Shematiski (2. att.) šifrēšanas funkciju var attēlot ar pāra, kas atbilst pamatteksta bigrammu burtiem (pelēkie riņķīši) matricā Γ, pozīciju pāreju uz pāri, kas atbilst šifrētā teksta burtu bigrammai (tukšie aplīši).



2. zīm.: Šifra *Playfair* šifrēšanas funkcijas.

Dotās trīs shēmas atbilst trim gadījumiem, kas parādīti formulā (13). Šī sistēma, stingri runājot, nav monoalfabēta aizvietošanas sistēma alfabētā $\mathbb{Z}_{m^2}$, jo pamatteksta garums vispārīgajā gadījumā nesaglabājas. Bez tam, pamattekstu un šifrēto tekstu bigrammu variāciju rindas sakrīt ar precizitāti līdz kaut kādam pārkārtojumam. Situāciju priekš kriptoanalītiķa sarežģī tikai tas, ka biežumu vērtību izkliede ir mazāka, nekā simbolu aizvietošanas gadījumā. Salīdzinot ar simbolu substitūciju tas noved pie kriptogrammas garuma palielināšanās, kas nepieciešama kriptosistēmas *Playfair* uzlaušanai.

ν–grammu substitūcija

ν-grammu substitūciju demonstrēsim izmantojot Hilla šifrēšanas metodi. Aplūkosim alfabēta L pamattekstu

$$a_0 a_1 \dots a_{n-1},$$

kur L — lauks. Pieņemsim, ka ν — naturāls skaitlis, pie tam ērtības labad uzskatīsim, ka ν skaitli n dala bez atlikuma. Sadalīsim pamattekstu veselā

skaitā ν–grammu. Šīs ν–grammas uzskatīsim par ν–dimensionālas vetoru telpas L^ν pār lauku L vektoriem:

$$a_0 a_1 \dots a_{n-1} = b_0 b_1 \dots b_{t-1},$$

te $t = n/\nu$; $\quad b_i \in L^\nu, \quad i = 0, 1, \dots, t-1$. Kā atslēga izmantojama $\nu \times \nu$ izmēru nedeģenerēta matrica M pār lauku L. Kriptogrammu $s_0 s_1 \dots s_{t-1}$ aprēķina saskaņā ar formulu:

$$s_0 s_1 \dots s_{t-1} = b_0 M \; b_1 M \; \dots \; b_{t-1} M. \tag{14}$$

Atšifrēšanai šifrētā teksta ν-grammas jāpareizina ar matricu M^{-1}, proti, $\forall i \; b_i = s_i M^{-1}$. No metodes apraksta izriet, ka Hilla sistēma ir monoalfabēta aizvietošana telpā L^k.

Hilla sistēmai ir dažas pozitīvas iezīmes, piemēram, laba atsevišķu simbolu biežuma izlīdzināšanās šifrētajā tekstā. Tomēr šī sistēma neiztur kriptoanalītisku uzbrukumu, lietojot vienādojumu sistēmas risināšanas metodi. Ja kriptoanalītiķim bez šifrētā teksta vēl ir pieejamas dažas pamatteksta ν–grammas, atslēgas elementus var atrast atrisinot lineāro vienādojumu sistēmu.

Hilla šifra uzlabojums ir kolonnu substitūcijas šifrs, kas izmanto matricu krājumu $M^{(r)} = (M_1, M_2, \dots, M_r)$. Uzlabotā šifra atslēga ir matricu krājums $M^{(r)}$ un galīga garuma vārds alfabētā $M^{(r)}$, kura simboli pēc kārtas tiek izmantoti pamatteksta ν-grammu šifrēšanai. Ja pamatteksta garums t (izteikts ν–grammās) ir lielāks par r, tad atslēgas vārds tiek izmantots vairākas reizes, kamēr nav nošifrēta pēdējā pamatteksta ν–gramma. Šāda šifra atslēgu arī iespējams atrast atrisinot lineāro vienādojumu sistēmu, taču tam nepieciešams r reizes vairāk materiāla.

6. Šifrējošās mašīnas

Šifrējošās mašīnas. Šifru matemātiskie modeļi. Mīlija mašīnas. Šifrējošs automāts. Attiecības un operācijas ar šifrējošiem automātiem. Kriptogrāfiskie ģeneratori. Atslēgu un šifrējošo automātu ekvivalence.

Kriptoloģijai, tāpat kā citām matemātikas disciplīnām, ir sakars ar pētāmā priekšmeta matemātiskiem modeļiem, proti, ar kriptosistēmu matemātiskiem modeļiem. Aplūkosim dažus šifra attēlojumu modeļus, kas ir ērti izpētei un analīzei.

6.1. Šifra matemātiskie modeļi

Ar šifra matemātiskiem modeļiem sapratīsim dažādas matemātiskās šifra attēlojumu attēlošanas formas. Šifra attēlojumu uzdošana ar atbilstības tipa tabulām

$$\textit{(“pamatteksts”, “atslēga”)} \rightarrow \textit{“šifrētais teksts”}$$

ir visnotaļ apgrūtinoša un mūsdienīgiem šifriem vienkārši nereāla. Viens no ērtākajiem šifra modeļiem — šifrēšanas un atšifrēšanas algoritmi.

Šifrēšanas (atšifrēšanas) algoritms — secīgas skaitļošanas operācijas, kas jāveic ar ieejas datiem, lai iegūtu kriptogrammas (pamattekstu). Šifrēšanas (atšifrēšanas) algoritma ieejas dati ir pamatteksts (šifrētais teksts) un atslēgas dati. Algoritmiem jābūt korekti uzdotiem visā definīcijas apgabalā. Atšifrēšanas algoritms var no šifrēšanas algoritma atšķirties nenozīmīgi un pat ar to sakrist. Tas ir īpaši raksturīgi simetriskām kriptosistēmām. *Šifru* sauc par *apgriežamu*, ja šifrēšanas un atšifrēšanas algoritmi sakrīt. Šifrēšanas un atšifrēšanas algoritmu realizācija apgriežamiem šifriem atšķiras tikai ar ieejas datiem. Apgriežami šifri ir ērti ar to, ka šifrēšanu un dešifrēšanu realizē viena iekārta (viena programma). Šifra apgriežamība ir specifiskāks nosacījums nekā visu šifra attēlojumu apgriežamība.

Šifrēšanas algoritma izklāsts parasti satur vārdisku aprakstu un formulas, un kā likums arī *šifroshēmu*, proti, shēmu, kas izskaidro šifrēšanas algoritma vai tā bloku darbību. Dažus šifrēšanas (atšifrēšanas) algoritmus izdodas uzdot ar pietiekoši kompaktu formulu, kurā mainīgie atbilst pamatteksta (šifrētā teksta) elementiem un atslēgai, bet funkcijas vērtības, ko uzdod formula, atbilst šifrētajam tekstam (pamattekstam). Šādā gadījumā šifru var adekvāti attēlot ar vienādojumiem, kas sasaista atslēgas, pamatteksta un šifrētā teksta elementus. Šos vienādojumus sauc par *šifrēšanas (atšifrēšanas)*

vienādojumiem. Šifrēšanas vienādojumu piemēri ir vienādojumi (8) un (14), kas apraksta Vernama un Hilla šifrēšanas metodes.

6.2. Simetriska šifra mašīnas modelis

Analizējot mūsdienīgus simetriskus šifrus, kurus realizē ar elektroniskām shēmām un datorprogrammām, ērti aplūkot šifrējošas mašīnas.

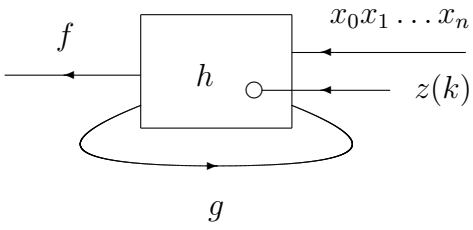
Definīcija 6.1. *Par šifrējošu mašīnu (š.m) sauksim 4 sugu algebru*

$$\breve{\mathfrak{S}} = \langle X, S, Y, K, z, g, h, f \rangle,$$

kur X, S, Y un K — galīgas kopas, ko sauc attiecīgi par pamatteksta alfabētu, š.m stāvokļu kopu, šifrētā teksta alfabētu un š.m atslēgu kopu, bet z, g, h un f ir attēlojumi

$$\begin{aligned} z &: K \to S, \\ g &: S \times K \times X \to K, \\ h &: S \times K \times X \to S, \\ f &: S \times K \times X \to Y, \end{aligned}$$

kurus sauc, attiecīgi, inicializācijas funkcija, atslēgas atjaunošanas funkcija, pārejas funkcija un izejas funkcija. Kopas K elementus sauc par š.m atslēgām.



$$\mathfrak{S} = \langle X, S, Y, K; z, g, h, f \rangle$$

3. zīm.: Šifrējošās mašīnas principiālā darbības shēma.

Š.m strādā diskrētos laika momentos t, kurus sauc par *taktīm*, $t = 1, 2, \ldots$ Pirms mašīnas darbības uzsākšanas tiek izvēlēta atslēgas sākotnējā vērtība $k_1 \in K$ un atkarībā no tās tiek iestādīts mašīnas sākumstāvoklis $s_1 \in S$:

$$s_1 = z(k_1). \tag{15}$$

Taktī t, š.m saņem ieejas simbolu x_t un izdod izejas simbolu y_t, pāriet nākošajā stāvoklī s_{t+1} un atjauno taktī t izmantoto atslēgas vērtību k_t, aprēķinot vērtības y_t, s_{t+1} un k_{t+1} saskaņā ar formulām :

$$y_t = f(s_t, k_t, x_t), \tag{16}$$
$$s_{t+1} = h(s_t, k_t, x_t), \tag{17}$$
$$k_{t+1} = g(s_t, k_t, x_t). \tag{18}$$

Mēs prasam, lai funkcijas z, f, h un g apmierina šādus nosacījumus.

1. Dažas no funkcijām z, f, h un g (bet ne visas) var būt tikai fiktīvi atkarīgas no atslēgas; ja funkcija z ir atkarīga no atslēgas fiktīvi, tad tas nozīmē, ka š.m sākumstāvoklis ir vai nu fiksēts, vai arī tiek izvēlēts nejauši, neatkarīgi no atslēgas un nav slepens.

2. Abas funkcijas f un h var būt fiktīvi atkarīgas no atslēgas tikai tad, ja funkcija z ir būtiski atkarīga no atslēgas.

Ja funkcija $g(s, k, x)$ ir atšķirīga no identiskā kopas K attēlojuma, tad š.m atslēgas vērtība nav konstanta no takts uz takti. Šādu š.m sauksim par *multiatslēgu* š.m. Ja t taksu laikā š.m $\check{\mathfrak{S}}$ ir attēlojusi ieejas vārdu $x_1x_2\dots x_t$ par izejas vārdu $y_1y_2\dots y_t$ izmantojot sākotnējo atslēgas vērtību k_1, tad teiksim, ka t takšu laikā š.m $\check{\mathfrak{S}}$ ir šifrējusi pamattekstu $x_1x_2\dots x_t$ par šifrēto tekstu $y_1y_2\dots y_t$, izmantojot atslēgu k_1.

Visus alfabēta X (alfabēta Y) iespējamos ieejas (izejas) vārdus apzīmēsim ar X^* (Y^*). Tukšo vārdu apzīmēsim ar λ. Ar š.m $\check{\mathfrak{S}}$ darbību saistītos attēlojumus $f^* : S\times K\times X^* \to Y^*$, $h^* : S\times K\times X^* \to S$ un $g^* : S\times K\times X^* \to K$ katram $s \in S$ un katram $k \in K$ definē šādi:

$$f^*(s, k, \lambda) \rightleftharpoons \lambda; \quad h^*(s, k, \lambda) \rightleftharpoons s; \quad g^*(s, k, \lambda) \rightleftharpoons k; \tag{19}$$

bez tam, visiem $x \in X$, $w \in X^*$

$$f^*(s, k, wx) \rightleftharpoons f^*(s, k, w)\#f(h^*(s, k, w), g^*(s, k, w), x); \tag{20}$$
$$h^*(s, k, wx) \rightleftharpoons h(h^*(s, k, w), g^*(s, k, w), x); \tag{21}$$
$$g^*(s, k, wx) \rightleftharpoons g(h^*(s, k, w), g^*(s, k, w), x). \tag{22}$$

No vienādībām (19)–(22) ar matemātisko indukciju var viegli izvest, ka š.m reakcija uz pēc kārtas ievadītiem vārdiem v un w ir tāda pati kā reakcija uz ievadītu vārdu vw.

Teorēma 6.2. *Š.m* $\check{\mathfrak{S}}$ *visiem* $v, w \in X^*$, $k \in K$ *un* $s \in S$ *ir spēkā vienādības:*

$$f^*(s,k,vw) = f^*(s,k,v)\#f^*(h^*(s,k,v), g^*(s,k,v), w);$$
$$h^*(s,k,vw) = h^*(h^*(s,k,v), g^*(s,k,v), w);$$
$$g^*(s,k,vw) = g^*(h^*(s,k,v), g^*(s,k,v), w).$$

Funkciju $f_l : K \times X^l \to Y$, kas attēlo atslēgu sākotnējo vērtību kopu un ieejas vārdu ar garumu l kopu par izejas vārda l-tā simbola vērtību kopu, sauksim par *l-to š.m* $\check{\mathfrak{S}}$ *izejas funkciju*, $l = 1, 2, \ldots$

$$f_l(k, x_1x_2 \ldots x_l) \rightleftharpoons y_l. \tag{23}$$

Ja š.m $\check{\mathfrak{S}}$ atslēgas sākotnējo vērtību kopa nav tukša un ir ierobežota kopā Q, kur $Q \subseteq K$, tad pāri $\langle \check{\mathfrak{S}}, Q\rangle$ sauc par *vāji inicializētu š.m (viš.m).* Mašīnu $\langle \check{\mathfrak{S}}, Q\rangle$ definē sekojoša algebra:

$$\langle \check{\mathfrak{S}}, Q\rangle = \langle X, S, Y, K, Q, z, g, h, f\rangle.$$

Speciālā gadījumā, ja $Q = \{k\}$ — vienelementīga kopa (sākotnējā atslēgas vērtība $k \in K$ ir fiksēta), tad pāri $\langle \check{\mathfrak{S}}, \{k\}\rangle$ sauksim par *inicializētu š.m (iš.m)* un lietosim apzīmējumu:

$$\check{\mathfrak{S}}(k) \rightleftharpoons \langle \check{\mathfrak{S}}, \{k\}\rangle = \langle X, S, Y, K, \{k\}, z, g, h, f\rangle.$$

Parādīsim, ka š.m var modelēt ar Mīlija mašīnu.

Definīcija 6.3. *Par š.m* $\check{\mathfrak{S}}$ *(viš.m* $\langle \check{\mathfrak{S}}, Q\rangle$*, iš.m* $\check{\mathfrak{S}}(k)$*) saistīto Mīlija mašīnu sauksim Mīlija mašīnu*

$$\mathfrak{M} = \langle S \times K, X, Y, \circ, *\rangle,$$

kur

$$(s,k) \circ x \rightleftharpoons (h(s,k,x), g(s,k,x)),$$
$$(s,k) * x \rightleftharpoons f(s,k,x).$$

Par š.m $\check{\mathfrak{S}}$ grafu sauksim ar to saistītās Mīlija mašīnas grafu.

No š.m saistītās Mīlija mašīnas definīcijas, un vienādībām (15) – (18) tieši izriet sekojoša teorēma.

Teorēma 6.4 (par š.m reducējamību uz Mīlija mašīnu)**.** *Viš.m* $\langle \check{\mathfrak{S}}, Q\rangle$*, kur* $Q \subseteq K$*, ir vāji inicializēta Mīlija mašīna* $\langle \mathfrak{M}, \pi(Q)\rangle$*; te* $\mathfrak{M}$ *— ar š.m* $\check{\mathfrak{S}}$ *saistītā Mīlija mašīna, bet* $\pi(Q) \rightleftharpoons \{(z(k), k) \mid k \in Q\}$.

Tādā veidā, š.m $\check{\mathfrak{S}}$ ir Mīlja mašīna ar "paplašinātu" stāvokļu kopu un specifisku sākotnējo stāvokļu kopas ierobežojumu.

6.3. Attiecības un operācijas šifrējošo mašīnu kopā

Definīcija 6.5. *Šifrējošās mašīnas*

$$\check{\mathfrak{S}} = \langle X, S, Y, K, z, g, h, f\rangle,\ \check{\mathfrak{S}}' = \langle X', S', Y', K', z', g', h', f'\rangle$$

sauc par salīdzināmām, ja $X' = X$, $S' = S$ *un* $Y' = Y$. *Salīdzināmas š.m* $\check{\mathfrak{S}}$ *un* $\check{\mathfrak{S}}'$ *sauc par saskaņotām, ja visiem* $s \in S$, $k \in K \cap K'$ *un* $x \in X$ *izpildās*

$$(z(k), h(s,k,x), g(s,k,x), f(s,k,x)) = (z'(k), h'(s,k,x), g'(s,k,x), f'(s,k,x)).$$

Viš.m $\langle \check{\mathfrak{S}}, Q\rangle$ *un* $\langle \check{\mathfrak{S}}', Q'\rangle$ *sauc par salīdzināmām (saskaņotām), ja š.m* $\check{\mathfrak{S}}$ *un* $\check{\mathfrak{S}}'$ *ir salīdzināmas (saskaņotas).*

No iepriekš minētā seko, ka saskaņotas ir šādas mašīnas:

1. visas salīdzināmās š.m $\check{\mathfrak{S}}$ un $\check{\mathfrak{S}}'$, ja $K \cap K' = \varnothing$;
2. visas viš.m $\langle \check{\mathfrak{S}}, Q\rangle$, $Q \subseteq K$, kas atbilst fiksētai š.m $\check{\mathfrak{S}}$.

Š.m $\check{\mathfrak{S}}'$ sauc par š.m $\check{\mathfrak{S}}$ *apakšmašīnu*, ja $X' \subseteq X$, $S' \subseteq S$, $Y' \subseteq Y$, $K' \subseteq K$, funkcija z' ir funkcijas z sašaurinājums kopā K', bet funkcijas g', h', f' ir funkciju g, h, f sašaurinājumi kopā $S' \times K' \times X'$. Viš.m $\langle \check{\mathfrak{S}}', Q'\rangle$ sauc par *viš.m* $\langle \check{\mathfrak{S}}, Q\rangle$ *apakšmašīnu*, ja $\check{\mathfrak{S}}'$ — š.m $\check{\mathfrak{S}}$ apakšmašīna un $Q' \subseteq Q$. Speciālā gadījumā š.m $\check{\mathfrak{S}}$ grafa katra sakarības komponente definē kādu mašīnas $\check{\mathfrak{S}}$ apakšmašīnu $\check{\mathfrak{S}}'$.

Definīcija 6.6. *Attēlojumu četrinieku* $(\phi_1, \phi_2, \phi_3, \phi_4)$

$$\phi_1 : S \to S', \quad \phi_2 : K \to K', \quad \phi_3 : X \to X', \quad \phi_4 : Y \to Y',$$

sauc par š.m $\check{\mathfrak{S}}, \check{\mathfrak{S}}'$ *homomorfismu, ja visiem* $s \in S$, $k \in K$ *un* $x \in X$ *izpildās vienādības:*

$$\begin{aligned} z'(\phi_2(k)) &= \phi_1(z(k)),\\ h'(\phi_1(s), \phi_2(k), \phi_3(x)) &= \phi_1(h(s,k,x)),\\ g'(\phi_1(s), \phi_2(k), \phi_3(x)) &= \phi_2(g(s,k,x)),\\ f'(\phi_1(s), \phi_2(k), \phi_3(x)) &= \phi_4(f(s,k,x)). \end{aligned}$$

Š. m $\check{\mathfrak{S}}, \check{\mathfrak{S}}'$ homomorfismu sauc par *viš.m* $\langle \check{\mathfrak{S}}, Q\rangle$, $\langle \check{\mathfrak{S}}', Q'\rangle$ *homomorfismu*, ja $\phi_2(Q) = Q'$. Ja $\phi_1, \phi_2, \phi_3, \phi_4$ ir bijektīvi attēlojumi, tad š.m $\check{\mathfrak{S}}$ un $\check{\mathfrak{S}}'$ sauc par *izomorfām*. Viš.m $\langle \check{\mathfrak{S}}, Q\rangle$ un $\langle \check{\mathfrak{S}}', Q'\rangle$ sauc par *izomorfām*, ja mašīnas $\check{\mathfrak{S}}$ un $\check{\mathfrak{S}}'$ ir izomorfas un attēlojuma ϕ_2 inducētais attēlojums $Q \to Q'$ ir bijektcija.

Aplūkosim dažas operācijas ar š.m. Definēsim š.m virknes slēgumu. Šai gadījumā pirmās mašīnas izeja tiks aplūkota kā otrās mašīnas ieeja. Par š.m $\check{\mathfrak{S}}_1 = \langle X, S_1, Y, K_1, z_1, g_1, h_1, f_1\rangle$ un $\check{\mathfrak{S}}_2 = \langle Y, S_2, Z, K_2, z_2, g_2, h_2, f_2\rangle$ *1. veida virknes slēgumu* (lieto apzīmējumu $\check{\mathfrak{S}}_1 \to \check{\mathfrak{S}}_2$) sauc š.m

$$\check{\mathfrak{S}} = \langle X, S_1 \times S_2, Z, K_1 \times K_2, z, g, h, f\rangle,$$

kur visiem $s = (s_1, s_2) \in S_1 \times S_2$, $k = (k_1, k_2) \in K_1 \times K_2$ un $x \in X$ izpildās:

$$\begin{gathered} z(k) = (z_1(k_1), z_2(k_2)), \\ h(s,k,x) = (h_1(s_1,k_1,x), h_2(s_2,k_2,f_1(s_1,k_1,x))), \\ g(s,k,x) = (g_1(s_1,k_1,x), g_2(s_2,k_2,f_1(s_1,k_1,x))), \\ f(s,k,x) = f_2(s_2,k_2,f_1(s_1,k_1,x)). \end{gathered}$$

Par viš.m $\langle \check{\mathfrak{S}}_1, Q_1\rangle$ un $\langle \check{\mathfrak{S}}_2, Q_2\rangle$ *1. veida virknes slēgumu* sauksim viš.m $\langle \check{\mathfrak{S}}_1 \to \check{\mathfrak{S}}_2, Q_1 \times Q_2\rangle$.

Š.m. $\check{\mathfrak{S}} = \langle X, S, Y, K, z, g, h, f\rangle$ sauksim par š.m

$$\check{\mathfrak{S}}_1 = \langle X_1, S_1, Y_1, K_1, z_1, g_1, h_1, f_1\rangle \quad \text{un} \quad \check{\mathfrak{S}}_2 = \langle X_2, S_2, Y_2, K_2, z_2, g_2, h_2, f_2\rangle$$

tiešo summu (lieto apzīmējumu $\check{\mathfrak{S}}_1 \times \check{\mathfrak{S}}_2$), ja

$$X = X_1 \times X_2, \quad S = S_1 \times S_2, \quad K = K_1 \times K_2, \quad Y = Y_1 \times Y_2$$

un visiem $s = (s_1, s_2) \in S_1 \times S_2$, $k = (k_1, k_2) \in K_1 \times K_2$ un $x \in X_1 \times X_2$:

$$\begin{gathered} z(k) = (z_1(k_1), z_2(k_2)), \\ h(s,k,x) = (h_1(s_1,k_1,x_1), h_2(s_2,k_2,x_2)), \\ g(s,k,x) = (g_1(s_1,k_1,x_1), g_2(s_2,k_2,x_2)), \\ f(s,k,x) = (f_1(s_1,k_1,x_1), f_2(s_2,k_2,x_2)). \end{gathered}$$

Par viš.m $\langle \check{\mathfrak{S}}_1, Q_1\rangle$ un $\langle \check{\mathfrak{S}}_2, Q_2\rangle$ *tiešo summu* sauksim viš.m

$$\langle \check{\mathfrak{S}}_1 \times \check{\mathfrak{S}}_2, Q_1 \times Q_2\rangle.$$

Par saskaņotu š.m

$$\check{\mathfrak{S}}_1 = \langle X, S_1, Y, K_1, z_1, g_1, h_1, f_1\rangle \quad \text{un} \quad \check{\mathfrak{S}}_2 = \langle X, S_2, Y, K_2, z_2, g_2, h_2, f_2\rangle$$

apvienojumu (lieto apzīmējumu $\check{\mathfrak{S}}_1 \cup \check{\mathfrak{S}}_2$) sauksim š.m

$$\check{\mathfrak{S}} = \langle X, S, Y, K_1 \cup K_2, z, g, h, f\rangle,$$

kur katram $k \in K_i$ un katram $i \in \{1, 2\}$

$$(z(k), h(s,k,x), g(s,k,x), f(s,k,x)) = (z_i(k), h_i(s,k,x), g_i(s,k,x), f_i(s,k,x)).$$

Par saskaņotu viš.m $\langle \check{\mathfrak{S}}_1, Q_1\rangle$ un $\langle \check{\mathfrak{S}}_2, Q_2\rangle$ *apvienojumu* sauc viš.m

$$\langle \check{\mathfrak{S}}_1 \cup \check{\mathfrak{S}}_2, Q_1 \cup Q_2).$$

No tā visa seko, ka katra viš.m $\langle \check{\mathfrak{S}}, Q\rangle$ ir iš.m apvienojums:

$$\langle \check{\mathfrak{S}}, Q\rangle = \bigcup_{k\in Q} \check{\mathfrak{S}}(k). \tag{24}$$

6.4. Monoatslēgas šifrējošās mašīnas

Š.m sauksim par *monoatslēgas* (mš.m), ja $g(s,k,x)$ nav atkarīgs no s un x un realizē identisku kopas K attēlojumu. Mš.m var definēt kā algebru:

$$\check{\mathfrak{S}} = \langle X, S, Y, K, z, h, f\rangle.$$

Tā kā mš.m atslēga neatjaunojas no takts uz takti, tad sākotnējās atslēgas vērtības jēdziens vienkārši sakrīt ar atslēgas jēdzienu. Ja k ir atslēga, tad mš.m darbojas šādi:

$$s_1 = z(k), \tag{25}$$
$$y_t = f(s_t, k, x_t), \tag{26}$$
$$s_{t+1} = h(s_t, k, x_t). \tag{27}$$

No vienādībām (25)–(27) seko, ka monoatslēgas iš.m (miš.m) $\check{\mathfrak{S}}(k)$ ir mš.m ar viena elementa atslēgas kopu:

$$\check{\mathfrak{S}}(k) = \langle X, S, Y, \{k\}, x, h, f\rangle.$$

Mīlija mašīna, kas saistīta ar monoatslēgas viš.m $\langle \check{\mathfrak{S}}, Q\rangle$, ir mašīnas

$$\langle S \times K, X, Y, h'', f''\rangle,$$

kas saistīta ar mš.m $\check{\mathfrak{S}}$, apakšmašīna $\langle S \times Q, X, Y, h', f'\rangle$.

Mš.m $\check{\mathfrak{S}}$ mašīnas grafs ir skaitā $|K|$ miš.m $\check{\mathfrak{S}}(k)$ grafu apvienojums, kur $k \in K$.

Nospiedošais daudzums mūsdienu šifru ir modelējami ar monoatslēgu š.m, kas ir saprotami, ja ņem vērā to realizācijas nosacīto vienkāršību un šifru izveides loģiku (no vienkāršākiem uz sarežģītākiem). Tajā pašā laikā, multiatslēgu š.m piemīt liels potenciāls kriptogrāfisko īpašību uzlabošanai. Konkrēti, to izejas virkņu periodi var būt $|K|$ reizes lielāki nekā monoatslēgu š.m ar tādu pašu stāvokļu skaitu. Bez tam, efektīvi realizējot atslēgu atjaunošanas funkciju g, multiatslēgu š.m realizācija sarežģījas tikai nenozīmīgi.

6.5. Kriptogrāfiskie ģeneratori

Autonomu š.m $\check{\mathfrak{S}}$, kas uzdota kā algebra

$$\check{\mathfrak{S}} = \langle S, Y, K, z, g, h, f\rangle,$$

sauksim par *kriptogrāfisku ģeneratoru* (k.ģ). K.ģ darbību nosaka vienādojums (15) un sekojoši vienādojumi:

$$y_t = f(s_t, k_t), \tag{28}$$

$$s_{t+1} = h(s_t, k_t), \tag{29}$$

$$k_{t+1} = g(s_t, k_t). \tag{30}$$

Ja k.ģ $\check{\mathfrak{S}}$ atslēgu sākotnējo vērtību kopu ierobežo netukša kopa Q, kur $Q \subseteq K$, tad pāri $\langle \check{\mathfrak{S}}, Q\rangle$ sauc par *vāji inicializētu k.ģ* (vik.ģ). Mašīnu $\langle \check{\mathfrak{S}}, Q\rangle$ definē sekojoša algebra:

$$\langle \check{\mathfrak{S}}, Q) = \langle S, Y, K, Q, z, g, h, f\rangle.$$

Konkrēti, ja k.ģ. $\check{\mathfrak{S}}$ atslēgas sākotnējā vērtība $k \in K$ ir fiksēta, tad pāri $\langle \check{\mathfrak{S}}, \{k\}\rangle$ sauc par *inicializētu k.ģ* (ik.ģ). Šai gadījumā lietosim pierakstu

$$\check{\mathfrak{S}}(k) \rightleftharpoons \langle \check{\mathfrak{S}}, \{k\}\rangle \rightleftharpoons \langle S, Y, K, \{k\}, z, g, h, f\rangle.$$

No vienādības (24) seko, ka katrs vik.ģ $\langle \check{\mathfrak{S}}, Q\rangle$ ir ik.ģ apvienojums:

$$\langle \check{\mathfrak{S}}, Q\rangle = \bigcup_{k \in Q} \check{\mathfrak{S}}(k). \tag{31}$$

Mīlija mašīna, kas saistīta ar k.ģ $\check{\mathfrak{S}}$ (vik.ģ $\langle\check{\mathfrak{S}}, Q\rangle$, ik.ģ $\check{\mathfrak{S}}(k)$), ir autonoma mašīna $\mathfrak{M} = \langle S \times K, Y, h', f'\rangle$, kur

$$f'(s,k) = f(s,k),$$
$$h'(s,k) = (h(s,k), g(s,k)).$$

No teorēmas 6.4 seko, ka k.ģ $\check{\mathfrak{S}}$ ir autonoma Mīlija mašīna $\langle\mathfrak{M}, \pi(K)\rangle$, bet vik.ģ ir autonoma vāji inicializēta Mīlija mašīna $\langle\mathfrak{M}, \pi(Q)\rangle$, kur $\mathfrak{M}$ — Mīlija mašīna, kas saistīta ar $\check{\mathfrak{S}}$.

Monoatslēgas k.ģ (mk.ģ) $\mathfrak{Y}$ neatjauno atslēgas vērtības no takts uz takti un to definē kā algebru

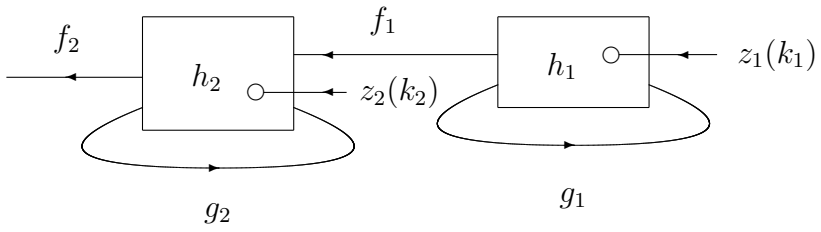
$$\mathfrak{Y} = \langle S, Y, K, z, h, f\rangle.$$

Mk.ģ darbību nosaka vienādība (25) un sekojošas vienādības:

$$y_t = f(s_t, k), \tag{32}$$
$$s_{t+1} = h(s_t, k). \tag{33}$$

Monoatslēgas ik.ģ (mik.ģ) $\mathfrak{Y}(k)$ ir mk.ģ ar viena elementa atslēgas kopu:

$$\mathfrak{Y}(k) = \langle S, Y, \{k\}, z, h, f\rangle.$$



$$\check{\mathfrak{S}}_2 = \langle Y, S_2, Z, K_2; z, g, h, f\rangle \qquad \mathfrak{Y}_1 = \langle S_1, Y, K_2; z_1, g_1, h_1, f_1\rangle$$

4. zīm.: 1. veida virknes slēgums.

Pēc definīcijas k.ģ

$$\mathfrak{Y}_1 = \langle S_1, Y, K_1, z_1, g_1, h_1, f_1\rangle$$

un š.m

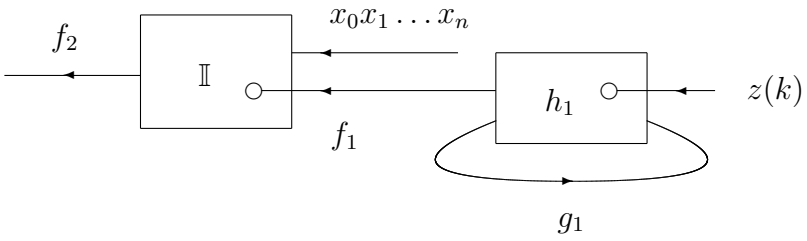
$$\check{\mathfrak{S}}_2 = \langle Y, S_2, Z, K_2, z_2, g_2, h_2, f_2\rangle$$

1. veida virknes slēgums ir k.ğ

$$\mathfrak{Y} = \langle S_1 \times S_2, Z, K_1 \times K_2, z, g, h, f \rangle,$$

kur visiem $s = (s_1, s_2) \in S_1 \times S_2$ un $k = (k_1, k_2) \in K_1 \times K_2$:

$$\begin{aligned} z(k) &= (z_1(k_1), z_2(k_2)), \\ h(s,k) &= (h_1(s_1,k_1), h_2(s_2,k_2,f_1(s_1,k_1))), \\ g(s,k) &= (g_1(s_1,k_1), g_2(s_2,k_2,f_1(s_1,k_1))), \\ f(s,k) &= f_2(s_2,k_2,f_1(s_1,k_1)). \end{aligned}$$



$\mathfrak{M}_2 = \langle W, X, Y; f_2 \rangle$ $\qquad$ $\mathfrak{Y}_1 = \langle S, W, K; z, g_1, h_1, f_1 \rangle$

5. zīm.: $\mathfrak{Y}_1 \Rightarrow \mathfrak{M}_2$

Definēsim otru mašīnu virknes slēguma tipu, kurā pirmās mašīnas izeja nosaka otrās mašīnas pārejas funkciju. Par k.ğ $\mathfrak{Y}_1 = \langle S, W, K, z, g_1, h_1, f_1 \rangle$ un Mīlija mašīnas ar identisku pārejas funkciju $\mathfrak{M}_2 = \langle W, X, Y, f_2 \rangle$ *otrā veida virknes slēgumu* (lieto apzīmējumu $\mathfrak{Y}_1 \Rightarrow \mathfrak{M}_2$) sauksim š.m $\check{\mathfrak{S}} = \langle X, S, Y, K, z, g, h, f \rangle$, kur visiem $s \in S$, $k \in K$ un $x \in X$:

$$\begin{aligned} f(s,k,x) &= f_2(f_1(s,k),x), \\ g(s,k,x) &= g_1(s,k), \\ h(s,k,x) &= h_1(s,k). \end{aligned}$$

Šāda slēguma gadījumā k.ğ $\check{\mathfrak{S}}_1$ sauc par *vadības bloku*, bet Mīlija mašīnu $\mathfrak{M}_2$ — par *šifrējošo š.m* $\check{\mathfrak{S}}$ *bloku*.

Ievērojam, ka 1. veida virknes slēgums ir kriptogrāfisks ğenerators, bet 2. veida virknes slēgums ir šifrējoša mašīna.

6.6. Atslēgu un šifrējošo mašīnu ekvivalence

Viena no svarīgākajām šifrējošo automātu teorijas problēmām, kas ir tieši saistīta ar šifru kriptogrāfiskās noturības novērtēšanu, ir š.m minimizācijas problēma. Aplūkosim dažas definīcijas, kas nepieciešamas šīs problēmas izskatīšanai.

Ar *vāji inicializētas Mīlija mašīnas* $\langle \mathfrak{M}, W\rangle = \langle X, (S, W), Y, h, f\rangle$ *reakciju* sapratīsim atbilstošās Mīlija mašīnas $\mathfrak{M} = \langle S, X, Y, h, f\rangle$ stāvokļu $w \in W$ reakciju kopu. Saskaņā ar definīciju *stāvokļa* $w \in W$ *reakcija* ir attēlojums $f_w(x) \rightleftharpoons f(w, x)$. Tā rezultātā vāji inicializētas Mīlija mašīnas $\langle \mathfrak{M}, W\rangle$ reakcija ir kopa $\{f_w \mid w \in W\}$. Speciālā gadījumā inicializētas Mīlija mašīnas $\mathfrak{M}_w$ reakcija ir atbilstošās Mīlija mašīnas $\mathfrak{M}$ stāvokļa w reakcija.

V.i Mīlija mašīnas

$$\langle \mathfrak{M}, W\rangle = \langle X, (S, W), Y, h, f\rangle \quad \text{un} \quad \langle \mathfrak{M}', W'\rangle = \langle X, (S', W'), Y, h', f'\rangle$$

sauc par *ekvivalentām* (lieto apzīmējumu $\langle \mathfrak{M}, W\rangle \cong \langle \mathfrak{M}', W'\rangle$), ja to reakcijas sakrīt.

Par š.m $\check{\mathfrak{S}}$ *reakciju* sauksim ar to saistītās Mīlija mašīnas stāvokļa $(z(k), k)$ reakciju. Citiem vārdiem, š.m $\check{\mathfrak{S}}(k)$ reakcija ir pamattekstu kopas attēlojums šifrēto tekstu kopā, ko realizē mašīna $\check{\mathfrak{S}}$ izmantojot atslēgu k.

Šifrējošās mašīnas $\check{\mathfrak{S}}$ atslēgas k un q sauc par *ekvivalentām* (lieto apzīmējumu: $k \cong q$), ja š.m $\check{\mathfrak{S}}(k)$ un $\check{\mathfrak{S}}(q)$ reakcijas sakrīt. Līdz ar to, ja $k \cong q$, tad mašīnas $\check{\mathfrak{S}}(k)$ un $\check{\mathfrak{S}}(q)$ vienādu pamattekstu šifrē par vienādu šifrēto tekstu.

Par *š.m* $\check{\mathfrak{S}}$ *reakciju* (*viš.m* $\langle \check{\mathfrak{S}}, Q\rangle$) sauksim visu š.m $\check{\mathfrak{S}}(k)$ reakciju kopu, kad $k \in K$ ($k \in Q$). Divas š.m (viš.m) sauc par *ekvivalentām*, ja to reakcijas sakrīt. No definīcijām seko, ka ekvivalentām š.m (viš.m) ar netukšām stāvokļu un atslēgu kopām jābūt vienādiem ieejas un izejas alfabētiem. Š.m (viš.m) sauc par *reducētām*, ja nevienas divas to atslēgas nav ekvivalentas.

Teorēma 6.7. *Ja reducēta š.m* $\check{\mathfrak{S}}_1$ *ar atslēgu kopu* K_1 *ir ekvivalenta reducētai š.m* $\check{\mathfrak{S}}_2$ *ar atslēgu kopu* K_2*, tad* $|K_1| = |K_2|$.

□ Ja reducētās š.m $\check{\mathfrak{S}}_1$ un $\check{\mathfrak{S}}_2$ ir ekvivalentas, tad katram $k \in K_1$ inicializētajai Mīlija mašīnai $\mathfrak{M}_1(k)$ atradīsies tai ekvivalenta inicializēta Mīlija mašīna $\mathfrak{M}_2(q)$, $q \in K_2$, pie tam tāda būs viena vienīga, jo pretējā gadījumā $\check{\mathfrak{S}}_2$ nebūtu reducēta. No tā seko, ka $|K_1| = |K_2|$. ■

Izmantojot dotās definīcijas aplūkosim dažas sekas no Haffmana-Mīlija teorēmas[1], kas ļaus novērtēt pamattekstu garumu, kas ir pietiekami atslēgu un pašu š.m ekvivalences noteikšanai.

Teorēma 6.8. *Lai noteiktu, vai š.m* $\check{\mathfrak{S}} = \langle X, S, Y, K, z, g, h, f\rangle$ *atslēgas* k *un* q *ir ekvivalentas, pietiek katram pamattekstam* a *ar garumu ne lielāku par* $|K| \cdot |S| - 1$ *pārbaudīt vai sakrīt šifrētie teksti, izmantojot atslēgas* k *un* q*.*

No teorēmas seko, ka š.m atslēgu ekvivalences atpazīšanas problēma ir algoritmiski izšķirama.

□ No teorēmas 6.4 seko, ka, lai noteiktu vai š.m $\check{\mathfrak{S}}$ atslēgas k un q ir ekvivalentas, pietiek noteikt vai ar š.m $\check{\mathfrak{S}}$ saistītās Mīlija mašīnas $\mathfrak{M}$ stāvokļi $(z(k), k)$ un $(z(q), q)$ ir ekvivalenti. Mašīnas $\mathfrak{M}$ stāvokļu skaits ir vienāds ar $|K| \cdot |S|$, tāpēc saskaņā ar Haffmana-Mīlija teorēmu iegūstam, ka š.m $\check{\mathfrak{S}}$ atslēgu k un q ekvivalences noteikšanai pietiek pārbaudīt Mīlija mašīnas $\mathfrak{M}$ stāvokļu $(z(k), k)$ un $(z(q), q)$ reakciju ekvivalenci pie ieejas vārda, kura garums ir vismaz $|K| \cdot |S| - 1$. ■

Teorēma 6.9. *Lai noteiktu, vai š.m*

$$\check{\mathfrak{S}} = \langle X, S, Y, K, z, g, h, f\rangle \quad \text{un} \quad \check{\mathfrak{S}}' = \langle X, S', Y, K', z', g', h', f'\rangle$$

(viš.m $\langle \check{\mathfrak{S}}, Q\rangle$ *un* $\langle \check{\mathfrak{S}}', Q'\rangle$*) ir ekvivalentas, pietiek pārbaudīt, vai to reakcijas sakrīt visiem pamattekstiem ar garumu ne lielāku par* $|K| \cdot |S| + |K'| \cdot |S'| - 1$*.*

No šīs teorēmas seko, ka š.m (viš.m) ekvivalences noteikšanas problēma ir algoritmiski izšķirama.

□ Š.m $\check{\mathfrak{S}}$ un $\check{\mathfrak{S}}'$ (viš.m $\langle \check{\mathfrak{S}}, Q\rangle$ un $\langle \check{\mathfrak{S}}', Q'\rangle$) ekvivalences noteikšanas pārbaude reducējas uz mašīnas $\mathfrak{N} = \mathfrak{M} \cup \mathfrak{M}'$, stāvokļu ekvivalences pārbaudi, kur $\mathfrak{M}$ un $\mathfrak{M}'$ ir Mīlija mašīnas, kas saistītas attiecīgi ar š.m $\check{\mathfrak{S}}$ un $\check{\mathfrak{S}}'$. Tā kā Mīlija mašīnu $\mathfrak{M}$ un $\mathfrak{M}'$ stāvokļu kopas nepārklājas, tad mašīnas $\mathfrak{N}$ stāvokļu skaits ir vienāds ar $|K| \cdot |S| + |K'| \cdot |S'|$, un, saskaņā ar Haffmana-Mīlija teorēmu, lai noteiktu š.m $\check{\mathfrak{S}}$ un $\check{\mathfrak{S}}'$ (viš.m $\langle \check{\mathfrak{S}}, Q\rangle$ un $\langle \check{\mathfrak{S}}', Q'\rangle$) ekvivalenci pietiek pārbaudīt to reakcijas uz ieejas vārdiem ar garumu vismaz $|K| \cdot |S| + |K'| \cdot |S'| - 1$. ■

Teorēma 6.10. *Lai noteiktu vai mš.m* $\check{\mathfrak{S}} = \langle X, S, Y, K, z, h, f\rangle$ *atslēgas* k *un* q *ir ekvivalentas, pietiek pārbaudīt vai katram pamattekstam* a *ar garumu* $2|S| - 1$ *sakrīt šifrētie teksti šifrējot* a*, izmantojot atslēgas* k *un* q*.*

[1]Lai divas Mīlija mašīnas ar n stāvokļiem $(n > 1)$ būtu ekvivalentas, pietiek, ja sakrīt šo stāvokļu reakcijas uz ieejas vārdiem, kuru garums ir vismaz $n - 1$.

□ Lai noteiktu, vai mš.m $\check{\mathfrak{S}}$ atslēgas k un q ir ekvivalentas, pietiek noteikt Mīlija mašīnas $\mathfrak{M}$, kas ir saistīta ar monoatslēgas viš.m

$$\langle \check{\mathfrak{S}}, \{k, q\} \rangle = \check{\mathfrak{S}}(k) \cup \check{\mathfrak{S}}(q),$$

stāvokļu $(z(k), k)$ un $(z(q), q)$ ekvivalenci. Tā kā Mīlija mašīnas $\mathfrak{M}$ stāvokļu skaits ir vienāds ar $2 \cdot |S|$, tad, saskaņā ar Haffmana-Mīlija teorēmu, mš.m $\check{\mathfrak{S}}$ atslēgu k un q ekvivalences noteikšanai, pietiek pārbaudīt vai sakrīt šifrētie teksti, kas iegūti šifrējot pamattekstu ar garumu $2|S| - 1$ ar atslēgām k un q. ■

Teorēma 6.11. *Lai noteiktu, vai mš.m*

$$\check{\mathfrak{S}} = \langle X, S, Y, K, z, h, f \rangle \quad \text{un} \quad \check{\mathfrak{S}}' = \langle X, S', Y, K', z', h', f' \rangle$$

(monoatslēgu viš.m $\langle \check{\mathfrak{S}}, Q \rangle$ un $\langle \check{\mathfrak{S}}', Q' \rangle$) ir ekvivalentas, pietiek pārbaudīt to reakcijas uz visiem pamattekstiem, kas nav garāki par $|S| + |S'| - 1$.

□ Mš.m $\check{\mathfrak{S}}$ un $\check{\mathfrak{S}}'$ (monoatslēgu viš.m $\langle \check{\mathfrak{S}}, Q \rangle$ un $\langle \check{\mathfrak{S}}', Q' \rangle$) ekvivalences pārbaude reducējas uz katras atslēgas $k \in K$ ekvivalences kaut kādai atslēgai $q \in K'$ un katras atslēgas $q \in K'$ ekvivalences kaut kādai atslēgai $k \in K$ pārbaudi. Lai noteiktu, vai divas tādas atslēgas k un q ir ekvivalentas, pietiek aplūkot mašīnas $\mathfrak{M} \cup \mathfrak{M}'$ stāvokļu $(z(k), k)$ un $(z'(q), q)$ ekvivalenci, kur $\mathfrak{M}$ un $\mathfrak{M}'$ ir Mīlija mašīnas, kas saistītas attiecīgi ar miš.m $\check{\mathfrak{S}}(k)$ un $\check{\mathfrak{S}}'(q)$. Tā kā mašīnu $\mathfrak{M}$ un $\mathfrak{M}'$ stāvokļu kopas nešķeļas, tad mašīnas $\mathfrak{M} \cup \mathfrak{M}'$ stāvokļu skaits vienāds ar $|S| + |S'|$, bet tad, saskaņā ar Haffmana-Mīlija teorēmu, lai atpazītu, vai mš.m $\check{\mathfrak{S}}$ un $\check{\mathfrak{S}}'$ (monoatslēgu viš.m $\langle \check{\mathfrak{S}}, Q \rangle$ un $\langle \check{\mathfrak{S}}', Q' \rangle$) ir ekvivalentas, pietiek pārbaudīt to reakciju uz ieejas vārdiem ar garumu $|S| + |S'| - 1$. ■

Teorēma 6.12. *Katrai š.m $\check{\mathfrak{S}}$ (viš.m $\langle \check{\mathfrak{S}}, Q \rangle$) var efektīvi uzkonstruēt tai ekvivalentu reducēto š.m (reducēto viš.m).*

□ Sadalīsim š.m $\check{\mathfrak{S}}$ atslēgu kopu K (viš.m $\langle \check{\mathfrak{S}}, Q \rangle$ atslēgu sākotnējo vērtību kopu Q) ekvivalences klasēs, ko iespējams efektīvi izdarīt saskaņā ar teorēmu 6.8. Reducētā š.m $\mathfrak{R}$ (viš.m $(\mathfrak{R}, Q')$) reprezentējama šādi:

$$\mathfrak{R} = \langle X, S, Y, K', z', h', f' \rangle$$
$$\langle \mathfrak{R}, Q' \rangle = \langle X, S, Y, (K', Q'), z', h', f' \rangle,$$

kur K' — visu š.m $\check{\mathfrak{S}}$ ekvivalento atslēgu klašu pārstāvju sistēma,

$$Q' = K' \cap Q,$$

funkcija z' ir funkcijas z ierobežojums uz kopu K', bet funkcijas h' un f' ir attiecīgi funkciju h un f ierobežojumi uz kopu $S \times K' \times X$. ■

6.7. Ieeju atšķiramība

Š.m nepieciešama īpašība ir spēja *pie jebkuras atslēgas* spēt šifrēt dažādus pamattekstus (sākot no kaut kāda to garuma) par dažādiem šifrētiem tekstiem. Pretējā gadījumā netiek nodrošināts kriptogrammas, kas iegūta šifrējot pamattekstu ar kaut kādu atslēgu, atšifrēšanas viennozīmīgums. Tādēļ ieeju atšķiramības nosacījums š.m gadījumā formulēts daudz stingrāk, nekā Mīlija mašīnu gadījumā.

Lai aplūkotu š.m ieeju atšķiramību ieviesīsim dažus apzīmējums. Alfabēta X pamattekstus v un w sauc par *neatšķiramiem ar iš.m* $\check{\mathfrak{S}}(k)$, ja katram $u \in X^*$

$$f^*(z(k), k, v \cdot u) = f^*(z(k), k, w \cdot u).$$

Pretējā gadījumā alfabēta X pamattekstus v un w sauc par *atšķiramiem ar iš.m* $\check{\mathfrak{S}}(k)$. Piemēram, jebkuri divi dažāda garuma alfabēta X pamatteksti ir atšķirami visām iš.m. Teiksim, ka alfabēta X pamatteksti v un w ir *neatšķirami ar viš.m* $\langle \check{\mathfrak{S}}, Q \rangle$ *(š.m* $\check{\mathfrak{S}}$*)*, ja eksistē kaut kāds $k \in Q$ ($k \in K$), ar kuru šie teksti ir neatšķirami ar iš.m $\check{\mathfrak{S}}(k)$.

Alfabēta X pamattekstus v un w sauc par *atšķiramiem ar viš.m* $\langle \check{\mathfrak{S}}, Q \rangle$ *(š.m* $\check{\mathfrak{S}}$*)*, ja katram $k \in Q$ ($k \in K$), šie teksti ir atšķirami ar iš.m $\check{\mathfrak{S}}(k)$. *Š.m* $\check{\mathfrak{S}}$ *(viš.m* $\langle A_C, Q \rangle$*, iš.m* $\check{\mathfrak{S}}(k)$*) atšķir ieejas*, ja jebkuri divi atšķirīgi pamatteksti ir atšķirami ar šo š.m (viš.m, iš.m).

Teorēma 6.13. *Pamatteksti v un w ir neatšķirami ar iš.m $\check{\mathfrak{S}}(k)$ tad un tikai tad, ja*

$$f^*(z(k), k, v) = f^*(z(k), k, w)$$

un Mīlija mašīnas $\mathfrak{M}$, kas ir saistīta ar š.m $\check{\mathfrak{S}}$, stāvokļi

$$(h^*(z(k), k, v), g^*(z(k), k, v)) \quad \text{un} \quad (h^*(z(k), k, w), g^*(z(k), k, w))$$

ir ekvivalenti.

□ Pieņemsim, ka pamatteksti v un w nav atšķirami ar š.m $\check{\mathfrak{S}}(k)$. Tad, saskaņā ar teorēmu 6.2, visiem $u \in X^*$ izpildās:

$$f^*(z(k),k,v)\cdot f^*(h^*(z(k),k,v),g^*(z(k),k,v),u) = f^*(z(k),k,v\cdot u) =$$
$$= f^*(z(k),k,w\cdot u) = f^*(z(k),k,w)\cdot f^*(h^*(z(k),k,w),g^*(z(k),k,w),u).$$

Salīdzinot vienādības abu pušu sastāvdaļas, iegūstam, ka, pirmkārt, vajadzīgā vienādība ir izpildīta un, otrkārt, norādītie Mīlija mašīnas $\mathfrak{M}$, kas saistīta ar š.m $\check{\mathfrak{S}}$, stāvokļi ir ekvivalenti.

Pretējais apgalvojums izriet no tādiem pašiem spriedumiem pretējā virzienā. ■

Nobeigumā atzīmēsim, ka šinī nodaļā dotā jēdzienu sistēma atļauj pētīt gan esošus, gan perspektīvus simetriskus šifrus kā speciālu Mīlija mašīnu klasi.

7. Plūsmas šifri

Plūsmas šifri. Atšķirības starp plūsmas un bloku šifriem. Sinhronie plūsmas šifri. Pašsinhronizējoši plūsmas šifri. Plūsmas šifru kriptogrāfiskās īpašības.

7.1. Atšķirības starp plūsmas un bloku šifriem

Plūsmas šifri pieder pie substitūcijas šifriem, kas pamattekstu pārveido par šifrēto tekstu simbolu pa simbolam. Plūsmas algoritms šifrē pamatteksta t-to simbolu $x_t \in X$ par alfabēta Y t-to simbolu y_t izmantojot no takts uz takti mainošos attēlojumus $\varphi_t : X \to Y, t = 1, 2, \ldots$

Substitūcijas šifri tradicionāli ir veidoti pēc plūsmas šifrēšanas principa, kur kā šifrējamie simboli tiek izmantoti burti vai bigrammas. Elektroniskajos plūsmas šifros kā simboli visbiežāk figurē biti vai baiti. Kad šifrēšana tiek izmantota liela ātruma datu pārraides sistēmās, visātrākie ir ar speciālām iekārtām realizēti plūsmas šifri.

Atšķirībā no plūsmas šifriem, *bloku šifri*, kas (kā likums) izmanto nemainīgu attēlojumu, apstrādā *informācijas blokus*. Bloku princips piemīt tradicionālajiem transpozīcijas šifriem. Piemēram, maršruta transpozīcijās informācijas bloks sastāv no $m \cdot n$ burtiem, kas tiek ierakstīti tabulā ar izmēru $m \times n$. Tomēr bloka šifra jēdziens parādījās un noformējās tikai XX. gadsimta 70. gadu sākumā, kad radās vajadzība pēc elektroniskiem substitūcijas šifriem datorizētas informācijas apstrādei.

Vajadzība izstrādāt jaunas bloku šifru klases radās galvenokārt sakarā ar diviem apstākļiem: plūsmas šifru aparatūras un skaitļojamo mašīnu nesavietojamības dēļ, un arī tāpēc, ka datorinformācijas šifrēšanas programmas, kas apstrādāja datus pa vienam simbolam, nespēja sasniegt nepieciešamo ātrumu. Tas stimulēja jaunas informācijas šifru — bloku šifru — klases izstrādi, kas spēja šifrēt ātrāk.

Ņemot vērā mūsdienīgo datoru procesoru uzbūvi, programmēšanai visērtākie ir simetriskie bloka šifri ar apstrādājamo bloku izmēru n robežās no 64 līdz 256 bitiem, kur n dalās ar 32.

Atzīmēsim, ka robežšķirtne starp plūsmas un bloka šifriem ir visnotaļ nosacīta. Vēlāk mēs parādīsim simetriskus šifrus, kuriem piemīt gan plūsmas, gan bloku šifru iezīmes.

7.2. Sinhronie plūsmas šifri

Plūsmas šifrus iedala *sinhronajos* (synchronous stream cypher — SSC) un *asinhronajos* jeb *pašsinhronizējošos* (self-synchronizing stream cypher — SSSC). SSC kriptoshēma sastāv no *vadības* un *šifrējošā* bloka. Vadības bloks ģenerē *vadības virkni* (γ_t), kas tiek izmantota šifra attēlojumu φ_{γ_t}, $t = 1, 2, \ldots$ veidošanai. Vadības virkni bieži sauc par *vadības gammu*, bet vadības bloku par *gammas ģeneratoru* (tāpēc, ka tradicionāli vadības virknes locekļi tika apzīmēti ar grieķu burtu γ). Šifrējošais bloks šifrē pamatteksta simbolu x_t par šifrētā teksta simbolu y_t izmantojot attēlojumu φ_{γ_t}, $t = 1, 2, \ldots$

Plūsmas šifra vadības bloku modelē kriptogrāfisks ģenerators

$$\mathfrak{G} = \langle S, \Gamma, K, z, g_1, h_1, f_1 \rangle,$$

bet šifrējošo bloku — Mīlija mašīna ar patstāvīgu atmiņu $\mathfrak{M} = \langle \Gamma, X, Y, f_2 \rangle$ (mašīnas $\mathfrak{M}$ iespējamo stāvokļu kopa Γ sakrīt ar k.ģ $\mathfrak{G}$ izejas alfabētu), un tas ir k.ģ $\mathfrak{G}$ 2. veida virknes slēgums ar Mīlija mašīnu $\mathfrak{M}$:

$$\mathfrak{M}_{SSC} =\!\!\!\!\!\!\rightharpoonup \mathfrak{G} \Rightarrow \mathfrak{M}.$$

Taktī $t = 1, 2, \ldots$ k.ģ $\mathfrak{G}$ ģenerē gammas simbolu γ_t,

$$\gamma_t = f_1(s_t, k_t), \tag{34}$$

kas tiek ierakstīts mašīnas $\mathfrak{M}$ atmiņā. Mašīna $\mathfrak{M}$ simbola γ_t ietekmē šifrē pamatteksta simbolu x_t par šifrētā teksta simbolu y_t, izmantojot attēlojumu φ_{γ_t}:

$$\varphi_{\gamma_t}(x_t) = f_2(\gamma_t, x_t) = y_t. \tag{35}$$

Šifrējošo attēlojumu kopas $\Phi = \{\varphi_\gamma : \gamma \in \Gamma\}$ elementi ir funkcijas $f_2(\gamma, x)$, kas atbilst visām iespējamām mainīgā γ vērtībām.

Turpmāk ierobežosimies ar *substitūcijas* plūsmas šifru aplūkošanu, kādi ir praktiski visi mūsdienīgie plūsmas šifri. Šādiem šifriem ir vienāda apjoma alfabēti X un Y un atšifrēšanas viennozīmīgums tiek nodrošināts tikai tajā gadījumā, kad $f_2(\gamma, x)$ ir mainīga x bijektīva funkcija. Ja $X = Y$, tad katram fiksētam γ funkcija $f_2(\gamma, x)$ ir kopas X substitūcija.

Sakarā ar dabiskiem apsvērumiem, kas saistīti ar šifra realizācijas vienkāršošanu un tā ātrdarbības palielināšanu, veidojot plūsmas šifrus priekšroka tiek dota monoatslēgas k.ģ. Ja vadības bloku modelē mk.ģ

$$\mathfrak{G} = \langle S, \Gamma, K, z, h_1, f_1 \rangle,$$

tad SSC modelē mš.m $\mathfrak{M}_{SSC} = \langle X, S, Y, K, z, h, f \rangle$, kur $\mathfrak{M}_{SSC} = \mathfrak{G} \Rightarrow \mathfrak{M}$ un šifrēšanas vienādojumi izmantojot atslēgu k izskatās šādi:

$$y_t = f_2(f_1(s_t, k), x_t), \quad t = 1, 2, \ldots \tag{36}$$

Savukārt atšifrēšanas vienādojumi, izmantojot atslēgu k, izskatās šādi:

$$x_t = \Psi(f_1(s_t, k), y_t), \quad t = 1, 2, \ldots \tag{37}$$

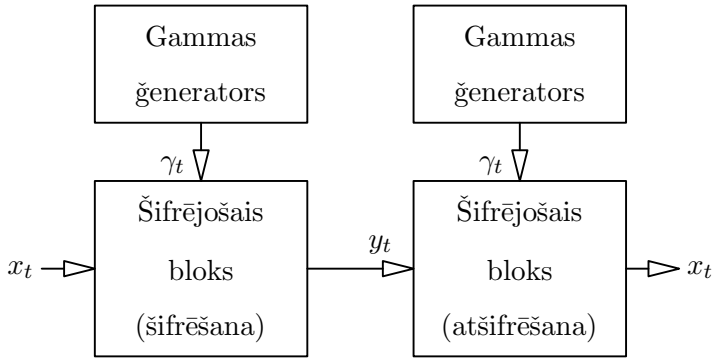
kur katram fiksētam $\gamma \in \Gamma$ attēlojums $\Psi(\gamma, y) : Y \to X$ ir attēlojuma $f_2(\gamma, x) : X \to Y$ inversais attēlojums. Tā kā $f_2(\gamma, x)$ ir mainīgā x bijekcija, tad katram $\gamma \in \Gamma$ šāds inversais attēlojums $\Psi(\gamma, y)$ eksistē.

Monoatslēgas gammas ģeneratori, kas tiek izmantoti SSC, tiek iedalīti divos veidos pēc uzbūves principa. Pirmais veids ir *ģeneratori ar iekšējo atgriezenisko saiti* (internal feedback), kuriem izejas funkcija f_1 nav atkarīga no atslēgas k. Ir arī šo ģeneratoru klases paveids, kuros no atslēgas ir atkarīgs tikai sākumstāvoklis. Otrs veids ir *skaitītāja tipa ģeneratori* (counter mode), kuri atšķiras ar to, ka no atslēgas ir atkarīga tikai izejas funkcija f_1.

Otrās klases ģeneratori atšķirībā no pirmās klases ģeneratoriem atļauj aprēķināt gammas i-to bitu, neaprēķinot visus iepriekšējos bitus. Lai to izdarītu, ģeneratoru jāuzstāda i-tajā iekšējā stāvoklī, un pēc tam var aprēķināt tam atbilstošo gammas i-to bitu. Šo īpašību ir noderīgi izmantot lai nodrošinātu patvaļīgu pieeju datu failiem; tas ļauj atšifrēt atsevišķu datu daļu, neatšifrējot failu pilnībā.

Diskrētu ziņojumu šifrēšanas realizācijas shēma, izmantojot SSC, parādīta zīmējumā 6. Ziņojuma nosūtītājs uzstāda iepriekš norunātu ģeneratora atslēgu k un, aprēķinājis kriptogrammu atbilstoši šifrēšanas vienādojumiem (36), nosūta to saņēmējam. Lai atšifrētu, saņēmējs izmanto identisku gammas ģeneratoru, kurā uzstādīta tā pati atslēga k. Saņēmēja šifrējošais bloks atšifrēšanas režīmā izrēķina pamattekstu no kriptogrammas saskaņā ar atšifrēšanas vienādojumiem (37).

SSC ģenerējamā gamma nav atkarīga no pamatteksta (ģenerators $\mathfrak{G}$ ir autonoms). Tāpēc SSC funkcionē pareizi tik ilgi, kamēr šifrējošās un atšifrējošās iekārtas sakaru līnijas galos darbojas sinhroni, proti, nevar atšifrēt šifrēto zīmi y_j izmantojot gammas simbolu γ_i, $i \neq j$. Šādas nevēlamas novirzes, ko sauc par *desinhronizāciju*, var rasties atšķirīgu aparatūras darbības ātrumu dēļ nosūtošajā un saņemošajā līnijas galā. Tikpat labi desinhronizāciju var izsaukt dažu simbolu pazušana nosūtīšanas laikā. Novirzes var izsaukt nepareizu visa tālākā ziņojuma atšifrēšanu. Ja tā gadās, nosūtītājam un sa-



6. zīm.: Diskrētu ziņojumu šifrēšana izmantojot asinhronu plūsmas šifru.

ņēmējam kaut kādā veidā jāatjauno gammas ģeneratoru sinhronumu, pirms turpināt sakaru seansu.

Parasti sinhronuma atjaunošanas problēmas tiek risinātas vai nu atkārtojot šifrēšanu un reinicializējot atslēgas abiem abonentiem (atkārtota gammas izmantošana ir kā minimums nevēlama, bet dažiem šifriem to vispār nedrīkst darīt!), vai arī sadalot tekstu blokos, kuru sākumi un beigas tiek atzīmēti ar marķieriem — speciāliem "robežsimboliem". Otrajā gadījumā desinhronizācija rada fragmentu nekorektu atšifrēšanu, līdz kamēr lietotājs nesaņem kādu no marķieriem.

Pie SSC pozitīvajām īpašībām jāpieskaita tas, ka tie nepalielina teksta simbolu kropļojumus, kas diezgan bieži parādās noraidot datus pa sakaru kanāliem. Ja nosūtot ziņojumu ir sakropļots simbols x_i vai arī pārraidot pa kanālu ir izkropļots simbols y_i, tad, ja ģeneratori darbojas sinhroni, tas neatsauksies uz pārējo simbolu atšifrēšanu, izņemot i-to. Kā mēs redzēsim vēlāk, šī īpašība nepiemīt SSSC.

SSC aizsargā nosūtāmo ziņojumu pret nesankcionētu papildināšanu un teksta fragmenta izņemšanu, jo šādos gadījumos notiks desinhronizācija un "iejaukšanās" tiks nekavējoties atklāta. Tajā pašā laikā SSC nepilnīgi aizsargā pret ziņojuma fragmenta aizvietošanu ar citu tāda paša garuma fragmentu. Ja uzbrucējam ir zināms pamatteksta fragments, tad viņam nav grūtību aizvietot to ar citu šifrētā teksta fragmentu, kas atšifrējams par viņam vēlamo tekstu.

7.3. Pašsinhronizējoši plūsmas šifri

SSSC arī sastāv no vadības un šifrējošā bloka ar analoģiskiem funkcionāliem uzdevumiem. Tomēr vadības bloka uzbūve ir atšķirīga un bloku savstarpējā mijiedarbība arī ir savādāka. Ja SSSC šifrējošo bloku modelē tāda pati Mīlija mašīna $\mathfrak{M} = \langle \Gamma, X, Y, f_2 \rangle$ kā SSC, tad vadības bloks ir veidots uz neautonomas mš.m

$$\mathfrak{N} = \langle Y, Y^n, \Gamma, K, z, h_1, f_1 \rangle$$

bāzes, kur n — naturāls skaitlis, bet z — kopas Y^n elements, kas nav atkarīgs no atslēgas.

SSSC kopumā modelē mš.m $\mathfrak{M}_{SSSC} \leftharpoondown \langle X, Y^n, Y, K, z, h, f \rangle$ *ar atgriezenisko saiti pēc šifrētā teksta* (output feedback mode), kur katram $x \in X$, $k \in K$ un $y = (y_1, y_2, \ldots, y_n) \in Y^n$

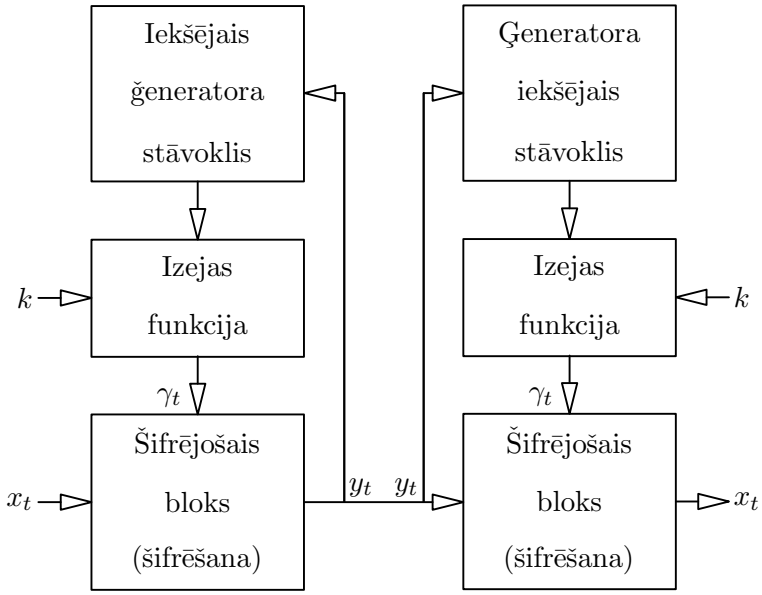
$$f(y, k, x) = f_2(f_1(y, k), x),$$
$$h(y, k, x) = (y_2, \ldots, y_n, f(y, k, x)).$$

Tādā veidā ne tikai mš.m $\mathfrak{N}$ vada mašīnu $\mathfrak{M}$ (vadības mehānisms ir tāds pats kā SSC), bet arī mašīna $\mathfrak{M}$ vada mš.m $\mathfrak{N}$ uz atgriezeniskās saites rēķina, kas izvada šifrētā teksta simbolus ne tikai mašīnas $\mathfrak{M}_{SSSC}$ izejā, bet arī mašīnas $\mathfrak{N}$ ieejā.

Šifrēšanas un atšifrēšanas vienādojumi ir tie paši, kas SSC. Svarīga atšķirība no SSC ir pārejas funkcija, kas ir uzbūvēta tā, ka ģenerētā gamma ir atkarīga no iepriekšējiem šifrētā teksta bitiem:

$$\gamma_{t+1} = f_1(y_{t-n+1}, y_{t-n+2}, \ldots, y_t, k), \quad t \geq n. \tag{38}$$

Diskrētu ziņojumu šifrēšana izmantojot SSSC parādīta zīmējumā 7. Katrs SSSC vadības bloka stāvoklis (izņemot pirmos n stāvokļus) tiek aizpildīts ar n iepriekšējiem šifrētā teksta simboliem. Tāpēc, ja n šifrētā teksta zīmes pēc kārtas nav izkropļotas pārsūtot pa sakaru līniju, tad š.m nosūtošajā un saņemošajā galā tiek uzstādītas vienādā iekšējā stāvoklī, un līdz ar to izstrādā vienādus gammas simbolus. Tādā veidā notiek š.m pašsinhronizācija. Katrs šifrētais ziņojums, kā likums, sākas nevis ar saturīgu tekstu, bet ar nejaušu n simbolu virkni, kas tiek šifrēta, nosūtīta un pēc tam atšifrēta. Un kaut arī šīs virknes atšifrēšana tiek veikta nekorekti ģeneratora sākuma stāvokļu nesakritības dēļ, pēc n sākotnējo simbolu nosūtīšanas ģeneratori ir sinhronizējušies. Lai apgrūtinātu mš.m $\mathfrak{M}_{SSSC}$ kriptoanalīzi izmantojot pirmos



7. zīm.: Diskrētu ziņojumu šifrēšana izmantojot asinhronu plūsmas šifru.

n simbolus, to sākotnējo stāvokli (inicializācijas funkcijas vērtību) vēlams izvēlēties nejauši katram ziņojumam.

SSSC vājā puse ir kļūdu pavairošanās. Viena vienīga kļūda šifrētajā tekstā rada n kļūdas pamattekstā. SSSC ir ievainojami arī ziņojumu imitācijas nozīmē. Uzbrucējs var pierakstīt kaut kādu pārtvertu šifrētā teksta nogriezni un vēlāk to nosūtīt atkārtoti. Pēc vairākām neatbilstībām ziņojuma sākumā (līdz n simboliem) nosūtītais nogrieznis atšifrēsies pareizi un saņēmējs nevarēs noteikt, ka ir saņēmis novecojušu ziņojumu, ja tikai šis ziņojums nesatur laika atzīmi. Tamlīdzīga imitācija nav iespējama tad, ja katram ziņojumam tiek izmantota cita atslēga.

7.4. Gammēšanas šifri

Par *gammēšanas šifriem* sauc aizvietošanas plūsmas šifrus, kuros $X = Y$ un divu dimensiju tabulā uzdotie šifrēšanas attēlojumi $f_2(\gamma, x)$ viedo latīņu kvadrātu alfabētā X.

Ja gammēšanas šifra šifrējošo substitūciju kopa $\Phi = \{\varphi_\gamma \,|\, \gamma \in \Gamma\}$ ir simetriskās grupas kāda apakšgrupa X vai arī ir blakusklase pēc šīs apakš-

grupas, tad šādu gammēšanas šifru sauc par *grupas šifru*.

Ar realizācijas ērtumu izceļas monoatslēgas gammēšanas šifri, kuros $X = Y = \Gamma = \mathbb{Z}_m$ — veselo skaitļu gredzens pēc moduļa m un šifrēšanas operācija f_2 ir viena no sekojošām:

$$f_2(\gamma, x) = (\pm x \pm \gamma) \mod m,$$

proti, saskaitīšana vai atņemšana pēc moduļa m. Šādi šifri ir grupas šifri un tos sauc par *gammēšanas šifriem pēc moduļa*.

Visērtākie no praktiskā lietojuma viedokļa ir gammēšanas šifri pēc moduļa, kuriem atbilst šifrēšanas vienādojumi

$$y_t = x_t \oplus \gamma_t,$$
$$\gamma_t = (\pm\gamma_t - x_t) \mod m.$$

Šādu šifru ērtība ir to apgriežamībā, proti, šifrējošie bloki šifrējot un atšifrējot darbojas identiski.

7.5. Plūsmas šifru kriptogrāfiskās īpašības

Lai nodrošinātu augstu šifrēšanas noturību ir svarīgas gan attiecīgās š.m īpašības, gan arī kriptogrāfiskā protokola uzbūve, kas nosaka šifra izmantošanas kārtību. Aplūkosim š.m un kriptogrāfisko protokolu, kas tiek izmantoti plūsmu šifrēšanas sistēmās, īpatnības.

Atkārtota gammas izmantošana

Svarīga SSC kriptogrāfisko protokolu īpašība ir vienas gammas vairākkārtējas izmantošanas aizliegums dažādu tekstu šifrēšanai. Gammēšanas šifriem ir izslēgta pat atkārtota šifrēšana ar vienu atslēgu.

Ilustrēsim aizlieguma iemeslu ar sinhronu gammēšanas šifru pēc moduļa. Te šifrēšanas vienādojumi ir:

$$y_t = (x_t + \gamma_t) \mod m. \tag{39}$$

Pieņemsim, ka kriptoanalītiķim ir pieejamas divas kriptogrammas

$$y_1^1 y_2^1 \dots y_n^1 \quad \text{un} \quad y_1^2 y_2^2 \dots y_n^2,$$

kas iegūtas no diviem dažādiem pamattekstiem

$$x_1^1 x_2^1 \dots x_n^1 \quad \text{un} \quad x_1^2 x_2^2 \dots x_n^2$$

ar vienu un un to pašu gammu $\gamma_1\gamma_2 \dots \gamma_n$, proti,

$$\begin{aligned} y_t^1 &= (x_t^1 + \gamma_t) \mod m, \\ y_t^2 &= (x_t^2 + \gamma_t) \mod m, \quad t = 1, 2, \end{aligned}$$

Mēģināsim atjaunot pamattekstus. Aplūkosim no gammas neatkarīgo kriptogrammu starpības:

$$y_t^1 - y_t^2 = (x_t^1 - x_t^2) \mod m, \quad t = 1, 2, \dots$$

Līdz ar to uzdevuma atrisināšana reducējas uz divu pamattekstu ar dotu starpību piemeklēšanu. Uzminot viena pamatteksta fragmentu (izmantojot sākotnēji zināmus standartus vai tematiku, kas piemīt dotajam pamattekstu avotam), viegli izskaitļot otra pamatteksta fragmentu. Kad iegūts viens vai vairāki šādi "pieturas punkti" dažādās tekstu pozīcijās, var mēģināt turpināt gan pa kreisi, gan pa labi minēt tekstus, izmantojot valodas gramatiskās saiknes, kā arī saturīgās un loģiskās tekstu saiknes. Iespējamo uzminēto teksta fragmentu turpināšanas variantu meklēšanu var realizēt kā simbolu pārlasi, kas sakārtoti aposterioro varbūtību samazināšanās kārtībā. Pie tam, katram t tiek izmantots iepriekš aprēķināts empīriskais pamattekstu simbolu atšķirību varbūtību sadalījums:

$$P\{x_t^1 = v, x_t^2 = \frac{v-u}{y_t^1 - y_t^2} = u\}, \quad v, u \in X.$$

Šī metode ne vienmēr noved pie pilnīgas veiksmes. Taču, ja pareizi ir atjaunotas tekstu daļas, tad tas atļauj kriptoanalītiķim atjaunot gammu un nodarboties ar atslēgas noskaidrošanu pēc zināmajiem gammas simboliem.

Dotā uzdevuma risinājums ievērojami vienkāršojas, ja pamatteksti atšķiras tikai ar vienu vai dažiem iespraustiem simboliem. Aplūkosim šinī gadījumā izmantoto kriptoanalītisko tekstu atjaunošanas metodi, kura ir ieguvusi nosaukumu *uzbrukums ar simbola ievietošanu* (insertion attack).

Pieņemsim, ka pamatteksts

$$x_1 x_2 x_3 \dots$$

saskaņā ar šifrēšanas vienādojumiem (39) tiek pārveidots par šifrēto tekstu

$$y_1 y_2 y_3 \dots$$

izmantojot gammu

$$\gamma_1 \gamma_2 \gamma_3 \dots .$$

Kriptoanalītiķim ir zināms šifrētais teksts, bet nav zināms pamatteksts un gamma. Pieņemsim arī, ka kriptoanalītiķim ir pieejama vēl viena kriptogramma, kas iegūta šifrējot to pašu pamattekstu, kas izmainīts iespraužot kaut kādā zināmā pozīcijā bitu x', kas šifrēts ar to pašu gammu. Pieņemsim, ka izmainītais pamatteksts ir $x_1 x' x_2 x_3 x_4 \dots$, bet atbilstošais šifrētais teksts ir $y_1 y_2' y_3' y_4' \dots$. No diviem šifrētiem tekstiem iespējams noteikt gan gammu, gan pamattekstu sākot no iespraustā simbola:

$$\begin{aligned} \gamma_2 &= y_2' - x'; \qquad & x_2 &= y_2 - \gamma_2; \\ \gamma_3 &= y_3' - x_2; \qquad & x_3 &= y_3 - \gamma_3; \\ \gamma_4 &= y_4' - x_3; \qquad & x_4 &= y_4 - \gamma_4 \dots \end{aligned}$$

Simbola x' ievietošanas momentu var noteikt salīdzinot izmainīto ar oriģinālo pamattekstu. Ja iespraustā simbola vērtība nav zināma, tad var veikt iespējamo vērtību pārlasi. Lai aizsargātos no uzbrukuma ar simbola ievietošanu pietiek nekad neizmantot vienādus gammas nogriežņus atkārtotai šifrēšanai.

Gammas noplūde sakaru kanālā

Kad tiek izmantots plūsmas šifrs, nevar pieļaut gammas nogriežņa

$$\gamma_j \gamma_{j+1} \dots \gamma_{j+t-1},$$

kur j, t — naturāli skaitļi, noplūdi sakaru kanālā. Šāds notikums var notikt šifrēšanas aparatūras kļūdu gadījumā un tam var būt divu veidu nepatīkamas sekas.

Pirmkārt, aktīvs uzbrucējs var nosūtīt saņēmējam viltus ziņojumu (kura garums nepārsniedz t zīmes), nošifrējot to ar pārtverto gammu. Šī bīstamība ir līdzīga tai, kāda ir lietojot asimetrisku kriptosistēmu, kad publisko atslēgu bāze ir vispārpieejama un nav aizsargāta pret viltus ziņojumu nosūtīšanu.

Otrkārt, kriptoanalītiķis var mēģināt noteikt pēc pārtvertā gammas nogriežņa vadības bloka k atslēgu, kas kā likums ir vienkāršāk, nekā mēģināt noteikt atslēgu izmantojot šifrēto tekstu. Veiksmes gadījumā kriptoanalītiķis iegūs pieeju visai informācijai, kas šifrēta izmantojot atslēgu k.

$\gamma \diagdown y$	y_1	y_2	$\dots$	y_t	$\dots$	y_N
0	y_1	y_2	$\dots$	y_t	$\dots$	y_N
1	$y_1 - 1$	$y_2 - 1$	$\dots$	$y_t - 1$	$\dots$	$y_N - 1$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$r-1$	$y_1 - r + 1$	$y_2 - r + 1$	$\dots$	$y_t - r + 1$	$\dots$	$y_N - r + 1$

2. tabula

Ar nevienmērīgi sadalītu gammu šifrēta teksta atgūšana

Viena no svarīgākajām īpašībām noturīgai šifrēšanai ir gammas, kuru izstrādā vadības bloks, vienmērīgums. Aplūkosim, kā var izmantot gammas nevienmērīgunu, lai uzlauztu šifru. Konkrētības labad pieņemsim, ka dots sinhrons gammēšanas šifrs ar vienādojumiem (39).

Precizēsim šifra varbūtisko modeli. Pieņemsim, ka p_i ir simbola i izmantošanas varbūtība gammā, $i = 0, 1, \dots, m-1$, kur ne visas varbūtības $p_0, p_1, \dots, p_{m-1}$ ir vienādas ar $1/m$. Jāatšifrē kriptogrammu $y_1 y_2 \dots y_N$, kas iegūta šifrējot parastu literāru tekstu izmantojot šo gammēšanas šifru.

Atzīmēsim, ka, lai novērtētu gammas simbolu varbūtības p_i var izmantot divas pieejas: vai nu aprēķināt varbūtības p_i noteikta šifra modeļu kontekstā, izmantojot gammas veidošanas likumu (gammas ģeneratora kriptoshēmu), vai arī izskaitļot tos ņemot vērā šifrēšanas vienādojumus (39), izmantojot simbolu kriptogrammas un pamattekstu simbolu varbūtiskos sadalījumus.

Sākumā aplūkosim vienkāršāku uzdevumu, kad daži simboli vispār gammā neparādās. Pieņemsim, ka

$$p_0 \geq p_1 \geq \dots \geq p_{r-1} > 0, \quad p_r = p_{r+1} = \dots = p_{m-1} = 0, \tag{40}$$

kur $r < m$. Tā rezultātā ($t = 1, 2, \dots, N$):

$$x_t^1 = y_t, x_t^2 = (y_t - 1) \mod m, \dots, x_t^r = (y_t - r + 1) \mod m.$$

Tāpēc pamattekstu var iegūt ar *lasīšanu pa stabiņiem*, proti, piemeklējot t-to simbolu 2.. tabulas t-tajā stabiņā (tabulas elementi tiek aplūkoti pēc moduļa m).

Atzīmēsim, ka simboli katrā stabiņā ir sakārtoti pēc to izmantošanas varbūtībām pamattekstā. Tas atvieglo lasīšanu pa stabiņiem, jo "lasāmais teksts" ar lielāku varbūtību ir atrodams tabulas augšējās rindiņās.

Šī metode ir pieskaitāma *bezatslēgas lasīšanas metodēm*, proti, tā atjauno pamattekstu necenšoties noteikt atslēgu. Lasīšanas no tabulām metodes pielietošana ir jo veiksmīgāka, jo mazāks kolonnu augstums r, kas apraksta uzdevuma risinājuma daudznozīmīgumu. Ja r ir tuvu m, lasīšana pa kolonnām zaudē jēgu, jo tabula satur lielu daudzumu ne tikai jēdzīgu, bet arī savstarpēji pretrunīgu tekstu. Kritiskais kolonnu augstums r ir nosakāms izejot no pamattekstu avota valodas īpašībām un to varbūtiskā sadalījuma (40).

Tagad atgriežamies pie sākotnējā uzdevuma nostādnes, kad nevar izslēgt neviena gammas simbola izmantošanu. Šai gadījumā lasīšanas pa kolonnām metodi ir nedaudz jāizmaina. Lai izveidotu tabulu, jāizslēdz $m - r$ vismazāk varbūtiskos gammas simbolus. Pēc tam var rīkoties tāpat kā iepriekš, bet šādas metodes uzticamība ir mazāka par 1 sakarā ar iespējamu patiesā risinājuma (pamatteksta) zaudēšanu.

Vadības un šifrējošā bloka kriptogrāfisko īpašību novērtēšanas kritēriji

Plūsmas šifra kriptogrāfisko noturību raksturo tā īpašību tuvums ideāla šifra īpašībām. Tāpēc jebkuru plūsmas šifru var aplūkot kā vairāk vai mazāk ideāla šifra imitāciju.

Plūsmas šifra kriptogrāfiskās īpašības nosaka gan šifrējošā, gan vadības bloka īpašības. Atgādināsim, ka šifrējošais bloks tiek uzdots ar attēlojuma $f_2(\gamma, x) : \Gamma \times X \to Y$ palīdzību. Līdzvērtīgi tas nozīmē, ka uzdota kopa $\Phi = \{\varphi_\gamma \,|\, \gamma \in \Gamma\}$. Te $\varphi_\gamma : X \to Y$ attēlojumi, kurus definē vienādojums (35). Kopas Φ attēlojumu izmantošanas kārtību teksta šifrēšanai nosaka vadības bloka gamma.

Vislabākā ideālā šifra imitācija izdodas, ja šifra attēlojumu virkne $(\varphi_{\gamma_t})_{t\in\mathbb{Z}_+}$ imitē neatkarīgu attēlojumu $X \to Y$ virkni. Tas ļauj formulēt divus SSC kriptogrāfiskās noturības nepieciešamos nosacījumus:

1. katram fiksētam $x \in X$ attēlojums $f_2(\gamma, x)$ ir sabalansēts. No kā seko, ka $|Y|$ tāpat kā $|X|$ dala $|\Gamma|$ (gammēšanas šifriem šo nosacījumu apmierina latīņu kvadrāta $f_2(\gamma, x)$ īpašības).

2. Uzskatīsim, ka atslēgas k sākotnējās vērtības izvēle ir neatkarīgs eksperiments, un katrā eksperimentā šis notikums realizējas ar vienu un to pašu varbūtību. Šādā gadījumā vadības gamma (γ_t) vislabāk imitē neatkarīgu, alfabētā Γ vienmērīgi sadalītu gadījuma lielumu virkni.

Šādos apstākļos šifrētais teksts, kuru izstrādā SSC, imitē vienmērīgu neatkarīgu sadalījumu alfabētā Y.

No otrā nosacījuma izriet konkrētas prasības pret SSC vadības gammu.

1. jābūt ar lielu periodu (kas daudzreiz pārsniedz pamatteksta garumu, kas tiek šifrēts ar SSC) un nedrīkst saturēt garus atkārtojošos gabalus, jo pretējā gadījumā tas būtu līdzvērtīgi gammas atkārtotai izmantošanai.

2. Vadības gammai jābūt apveltītai ar dažām statistiskām īpašībām, kas piemīt vienmērīgam neatkarīgam sadalījumam alfabētā Γ. Konkrēti, gammā katras ν–grammas parādīšanās biežumam jābūt tuvu skaitlim $|\Gamma|^{-\nu}$. Te $\nu = 1, 2, \ldots, r_0$, kur r_0 — kāda konstante, kas nosaka SSC uzlaušanas iespēju izmantojot faktu, ka gammas multigrammas nav vienmērīgi sadalītas.

3. Atkarībai starp vadības gammas simboliem ir jābūt tik sarežģītai, lai gammas atjaunošana pēc tās pietiekoši gara nogriežņa būtu darbietilpīgs skaitļošanas uzdevums. Konkrēti, vadības gammai jābūt lielai lineārai sarežģītībai.

4. Vienādojumu sistēmai, kas sasaista nezināmus atslēgas k elementus ar zināmām gammas zīmēm, jābūt sarežģīti risināmai un jāizslēdz praktisku risināšanas algoritmu realizāciju. [2].

Atslēgu k, kuru izmantojot vadības gamma neapmierina vismaz vienu no nosacījumiem 1 – 3, sauc par *vāju SSC atslēgu*. Šajā sakarā nosacījumus 1 – 3 jāpapildina ar atrunu: vāju atslēgu izmantošanai jābūt izslēgtai, vai arī vājo atslēgu daļa nedrīkst pārsniegt informācijas daļu (no šifra lietotāja viedokļa), kuru var pieļaut, ka kriptoanalītiķis dešifrēs.

Šifrējošā bloka un attēlojuma

$$f_1 : Y^n \times K \to \Gamma$$

īpašības nosaka SSSC kriptogrāfiskās īpašības. Nosacījumi, kas nodrošina noturīgu šifrēšanu izmantojot SSSC, kas attiecas uz šifrējošo bloku, ir tādi paši kā SSC. Vadības bloka attēlojumam jāimitē nejaušs attēlojums

$$Y^n \to \Gamma.$$

[2]Nosacījumu 1 – 3 parametrus nosaka ņemot vērā lietotāju prasības attiecībā pret šifru, kā arī rēķinoties ar iespējamā uzbrucēja kriptoanalītiķa iespējām.

Šī nosacījuma detalizācija parasti ir saistīta ar konkrētas š.m klases īpašībām un atbilstošām kriptoanalīzes metodēm. Katrā gadījumā šim attēlojumam ir jābūt sabalansētam, proti $|\Gamma|$ jādala $|Y|^n$ (pretējā gadījumā var uzskatīt, ka k ir vāja SSSC atslēga).

Norādītās īpašības var lielākā vai mazākā mērā nodrošināt piemeklējot konstruktīvus plūsmas šifru kriptoshēmu elementus.

8. Simetriskie bloku šifri

Simetriskie bloku šifri. Iteratīvie šifri. Feisteļa šifri. Cikliskās funkcijas konstrukcija. Vājas atslēgas. Šifrēšanas režīmi.

8.1. Asimetrisku un simetrisku bloka šifru īpašību salīdzinājums

Simetriskos bloka šifros (symetric block cypher — SBC) tipiskais apstrādājamā datu bloka izmērs variē no 64 līdz 256 bitiem un dalās ar 16. Tas saistīts ar informācijas attēlojumu baitos un datoru procesoru specifiku. Tāpēc bez vispārīguma zaudēšanas turpmāk aplūkosim SBC modeļus, kas apstrādā blokus garumā $2n$ biti, kur n dalās ar 8.

Visi asimetriskie šifri izmanto to pašu šifrēšanas principu pa blokiem, taču informācijas pārveidojumiem ir cita daba un bloka izmērs var sasniegt vairākus tūkstošus bitu. Apmēram par kārtu atšķiras arī atslēgu garumi (SBC atslēgas ir īsākas). Šī atšķirība ir izskaidrojama ar to, ka apmēram pie šādas bloku un atslēgu garumu attiecības, tiek sasniegta to kriptogrāfiskās noturības paritāte.

Programmētu simetrisku šifru ātrdarbība ir apmēram 1000 reižu lielāka nekā asimetriskiem šifriem. Taču asimetriskie šifri nozīmīgi paplašina kriptogrāfijas lietojumu sfēru un ļauj izstrādāt virkni ērti realizējamus kriptogrāfiskus protokolus.

8.2. Bloku šifru uzbūves principi

Visus SBC var izmantot vairākos šifrēšanas režīmos. Bāzes režīmu, uz kura pamata ir veidoti visi pārējie režīmi, sauc par *elektronisko kodu grāmatu* (electronic code book — ECB) jeb *vienkāršu substitūciju*. Bloka šifru ECB režīmā var uztvert kā mš.m bez atmiņas (proti, $S = \varnothing$)

$$\mathfrak{M}_{ECB} = \langle X, Y, K, f \rangle, \quad \text{kur} \quad X = Y = \{0,1\}^{2n}$$

un izejas funkcija $f(x,k)$ ir bijektīva pēc ieejas mainīgā x. Šifrēšanas vienādojumi izmantojot atslēgu $k \in K$ izskatās šādi:

$$y_t = f(x_t, k) = E_k(x_t), \qquad t = 1, 2, \ldots \tag{41}$$

kur $E_k(x)$ katram $k \in K$ ir kopas $\{0,1\}^{2n}$ substitūcija. Tā rezultātā ECB realizē vienkāršu substitūciju alfabētā, kura apjoms ir 2^{2n}.

SBC realizācijas novitāte savulaik saistījās ar nepieciešamību izstrādāt kriptogrāfiski noturīgus, ērti realizējamus (gan uz speciālām iekārtām, gan ar programmas līdzekļiem) substitūcijas šifrus lielas kārtas alfabētiem.

Nedaudz vēstures

Pirmais mēģinājums izveidot bloka šifru bija amerikāņu firmas *IBM* izstrādātais šifrs "*Lucifers*". Pamatteksta un šifrētā teksta bloki, kurus apstrādā "*Lucifers*", ir bināri vektori ar garumu 128 biti. Šifrs balstīts uz "sviestmaizes" principa. Tas salikts no vairākiem *slāņiem* — bloku substitūcijas (substitution) un bloku koordinātu transpozīcijas (permutation). Šādas shēmas ir ieguvušas nosaukumu *SP tīkli* (*transpozīcijas un substitūcijas tīkli*). Kriptogrāfiskā SP tīkla ideja ir izveidot sarežģītu pārveidojumu, izmantojot vairāku salīdzinoši vienkāršu un ērti realizējamu pārvietojumu kombināciju.

Realizējot šo kopumā perspektīvo kriptogrāfisko ideju neiztika arī bez trūkumiem: shēma sanāca nedaudz sarežģīta un, līdz ar to, tā bija lēna. Tā šifrēšanas ātrums nepārsniedza 8 kilobaitus sekundē, ja "*Luciferu*" realizēja kā programmu. Realizācija ar speciālu aparatūru šifrēja ar ātrumu 97 kilobaiti sekundē. Pie tam radās bažas par šifra kriptogrāfisko noturību, kuras vēlāk arī apstiprinājās. Tomēr pieredze, kuru bija ieguvuši "*Lucifera*" izstrādātāji jau drīz noderēja.

1973. gadā ASV nacionālais standartu birojs (*NBS*) izsludināja konkursu, kura mērķis bija datu šifrēšanas standarta izstrāde. Uzvarēja firma *IBM*, kas 1974. gadā piedāvāja šifrēšanas algoritmu, kas pazīstams ar nosaukumu *DES — Data Encryption Standart*.

DES algoritms apstrādā 64 bitu datu blokus un izmanto 56 bitu atslēgu. Līdzīgi "*Luciferam*" tas realizē SP tīkla pārveidojumu, kas balstīts uz *iteratīvu* principu, proti, balstīts uz vairāku vienāda tipa pārveidojumu kompozīciju. Turpmāk iteratīvais SP tīkla veidošanas princips tika izmantots nospiedošā vairumā gadījumu radot SBC.

DES algoritms tika publicēts (funkcionējošiem šifriem bezprecedenta gadījums līdz tam) un bija ne tikai vairāku valstu speciālistu aktīvas izpētes priekšmets, bet arī spēcīgs impulss civilās (ne militārās) kriptoloģijas attīstībai. *DES* algoritma izmantošanas apjomi komercsektorā bija iespaidīgi. To sāka izmantot daudzās valstīs tālu aiz ASV robežām. Vienlaicīgi, *DES*

no paša sākuma tika kritizēts par mazo atslēgas garumu, kas noveda pie aizlieguma to izmantot ASV valsts noslēpumu šifrēšanai. Sekojošie 25 gadi pēc algoritma publicēšanas parādīja, ka hipotētiskās *DES* uzlaušanas metodes, kas saistītas ar atslēgu pārlases paralelizāciju, lēnām pārvēršas par realitāti. Turklāt, *DES* algoritma uzbūves īpatnību dēļ, tā ātrdarbība uz mūsdienu procesoriem sāka atpalikt no pieaugušajām vajadzībām.

Kriptogrāfi ne tikai kritizēja esošo *DES*, bet arī vienlaicīgi strādāja pie jaunu bloku šifru izstrādes, kas varētu perspektīvā aizstāt *DES*. Tika izstrādāti dažādi *DES* algoritma shēmas uzlabojumi (ar neatkarīgām cikliskām atslēgām, secīgu šifrēšanu, ar palielinātiem bloku un atslēgu izmēriem ...), kā arī oriģinālas shēmas *NewDES*, *FEAL*, *IDEA* un citas.

1997. gadā *NIST*[3] paziņoja par *AES* (*Advanced Encryption Standart*) — 21. gadsimta kriptogrāfiskās datu aizsardzības standarta izstrādi. Izsludināja trīs posmu starptautisku konkursu ar mērķi radīt mūsdienīgām prasībām atbilstošu bloka šifru ar labām lietošanas kvalitātēm un augstu kriptogrāfisko noturību. Konkursā uzvarēja beļģu (autori J.Daemen un V.Rijmen) izstrādātais algoritms *RIJNDAEL*, kurš tika paziņots par jauno kriptogrāfisko standartu, kas stājās spēkā 2002. gadā.

Iteratīvie bloku šifri

Aplūkosim iteratīva SBC substitūcijas iekārtu E_k, kas apmierina šifrēšanas vienādojumus (41). Pieņemsim, ka $\varphi(x,q)$ ($\delta(x,q), \pi(x,q)$) — mainīgā x bijektīvs attēlojums

$$\begin{aligned} \varphi &: \{0,1\}^{2n} \times Q \to \{0,1\}^{2n} \\ (\delta,\pi) &: \{0,1\}^{2n} \times Q' \to \{0,1\}^{2n}, \end{aligned}$$

kur $x \in \{0,1\}^{2n}$, $q \in Q$ (atbilstoši $q \in Q'$) un Q (atbilstoši Q') — no k atvasinātu atslēgu kopa. Lai realizētu substitūciju E_k izmanto atvasinātās atslēgas q_0, q_{r+1} no Q' un $q_1, q_2, \ldots, q_r$ no Q:

$$q_i = \theta_i(k), \quad i = 0, 1, \ldots, r, r+1, \tag{42}$$

[3]NIST — National Institute for Standards and Technology (līdz 1988. gadam — NBS, National Bureau of Standards), Nacionālais standartu un skaitļojamās tehnikas institūts, kas ir ASV tirdzniecības ministrijas apakšvienība. Saskaņā ar ASV kongresa lēmumu NIST atbild par standartiem, kas attiecas uz datoriem un atbilstošajām saziņas sistēmām.

kur $\{\theta_0, \theta_1, \ldots, \theta_r, \theta_{r+1}\}$ — funkciju saime, kas attēlo atslēgu kopu K kopā $Q' \times Q^r \times Q'$, proti,

$$(\theta_0, \theta_1, \ldots, \theta_r, \theta_{r+1}) : K \to Q' \times Q^r \times Q'.$$

Attēlojuma $\varphi(x, q)$ (atbilstoši $(\delta(x, q), \pi(x, q))$) realizēto substitūciju pie fiksētas vērtības $q \in Q$ (atbilstoši $q \in Q'$) apzīmēsim ar φ_q (atbilstoši (δ_q, π_q)). Substitūcija E_k ir substitūciju reizinājums (pārveidojumi tiek pielietoti no labās uz kreiso pusi):

$$E_k(x) = \pi_{q_{r+1}} \cdot \varphi_{q_r} \cdot \ldots \cdot \varphi_{q_2} \cdot \varphi_{q_1} \cdot \delta_{q_0}(x), \tag{43}$$

kur x — pamatteksta bloks. Atšifrēšanas algoritms ir substitūcijas E_k^{-1} realizācija šifrētā teksta blokam y. No vienādības (43) iegūstam, ka

$$E_k^{-1}(y) = \delta_{q_0}^{-1} \cdot \varphi_{q_1}^{-1} \cdot \varphi_{q_2}^{-1} \cdot \ldots \cdot \varphi_{q_r}^{-1} \cdot \pi_{q_{r+1}}^{-1}(y). \tag{44}$$

Iteratīvam SBC, kuru nosaka vienādojumi (42) un (43), attēlojumu $\varphi(x, q)$ sauc par *cikla funkciju*, attēlojumu $\delta(x, q)$ — *ieejas attēlojumu*, $\pi(x, q)$ — *izejas attēlojumu*. Attēlojumu saimi $\{\theta_0, \theta_1, \ldots, \theta_r, \theta_{r+1}\}$ sauc par iteratīva SBC *atslēgu sarakstu*, bet skaitli r — *šifrēšanas ciklu skaitu*. Stāvokli q_i sauc par šifrēšanas algoritma i-to *cikla atslēgu*, q_0 un q_{r+1} — attiecīgi par *ieejas attēlojuma* un *izejas attēlojuma atslēgām* (pēdējās divas atslēgas nav nepieciešamas, ja bloka šifra ieejas un izejas attēlojumi ir substitūcijas, kas nav atkarīgas no atslēgas).

Šifrēšanas (atšifrēšanas) algoritma daļa, ko definē substitūcijas φ_{q_i} $(\varphi_{q_i}^{-1})$ sauc par i-to *šifrēšanas ciklu* ($r + 1 - i$-to *atšifrēšanas ciklu*), $i = 1, 2, \ldots, r$. Ja $\varphi(x', q_i) = y'$, tad vektoru x' sauc par *ieejas bloku*, bet vektoru y' — par *i-to šifrēšanas izejas bloku*.

Iteratīvo SBC var uzbūvēt tādā veidā, ka tiek nodrošināta tā apgriežamība.

Teorēma 8.1. *Ja katram $q \in Q'$ $\delta_q(x) = \pi_q^{-1}(x)$ un katram $q \in Q$ substitūcija $\varphi_q(x)$ ir involūcija, tad iterācijas SBC ir apgriežams un atšifrēšanas algoritms atšķiras no šifrēšanas algoritma ar to, ka cikla atslēgas tiek izmantotas apgrieztā secībā.*

□ No teorēmas nosacījumiem zināms, ka $\pi_{q_{r+1}}^{-1} = \delta_{q_{r+1}}$, $\delta_{q_0}^{-1} = \pi_{q_0}$ un $\varphi_q(x) = \varphi_q^{-1}(x)$ katram $q \in Q$. Tāpēc vienādība (44) iegūst šādu formu:

$$E_k^{-1}(y) = \pi_{q_0} \cdot \varphi_{q_1} \cdot \varphi_{q_2} \cdot \ldots \cdot \varphi_{q_r} \cdot \delta_{q_{r+1}}(y).$$

Salīdzinot šo vienādojumu ar (43), iegūstam vajadzīgo apgalvojumu. ■

Atzīmēsim, ka teorēmas formulējums ir nedaudz samākslots. No pierādījuma redzams, ka cikla funkcijas atšifrējot kriptotekstu tiešām tiek lietotas apgrieztā secībā, taču $\pi_{q_0} \neq \delta_{q_0}$, bet tas viss attiecas vienīgi uz vienošanos.

Feisteļa šifri

Viens no pirmajiem cikla funkcijas uzbūves veidiem (kas ir realizēts *DES*-algoritmā) ir balstīts uz reģistra nobīdes tipa attēlojuma izmantošanu. Konstrukcija tika atzīta par veiksmīgu un atrada plašu pielietojumu tālākā bloku šifru izstrādē (*FEAL*, *Khufu*, *Khafre*, *LOKI*, *Blowfish*,...). Šādus šifrus sauc par Feisteļa šifriem par godu vienam no *DES* izstrādātājiem.

Feisteļa šifrs — iteratīvs SBC, kura cikla funkcija $\varphi(x, q)$ operē ar ieejas bloka x "pusītēm" (proti, $x = (x_1, x_2) \in \{0,1\}^n \times \{0,1\}^n$) un kurš ir uzrakstāms kā

$$\varphi((x_1, x_2), q) = (x_2, \psi(x_2, q) \oplus x_1). \tag{45}$$

Te $\psi : \{0,1\}^n \times Q \to \{0,1\}^n$; $\oplus$ — bitu saskaitīšana pēc moduļa 2 jeb tā sauktā XOR operācija. Funkciju $\psi(x_2, q)$ sauksim par Feisteļa šifra *sarežģījuma funkciju*.

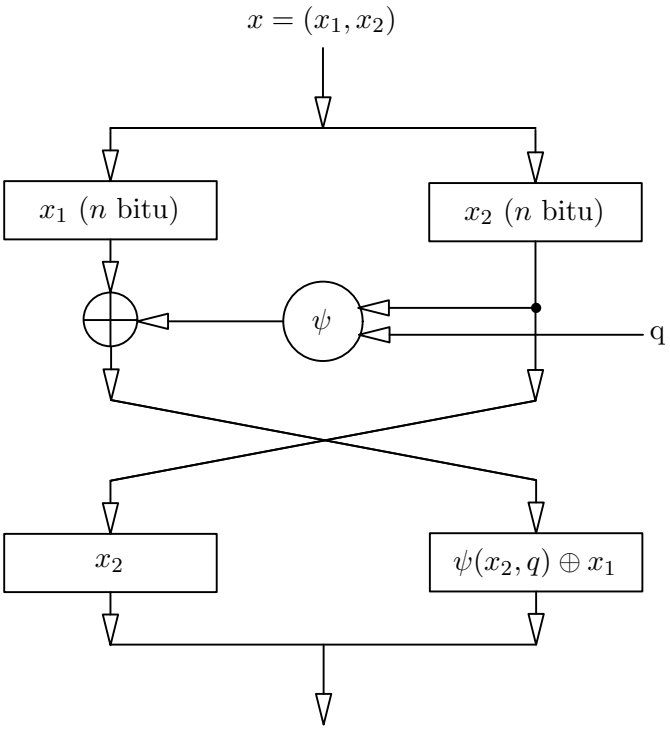
Feisteļa šifrēšanas shēmas varianti atšķiras ar sarežģījuma funkcijas ψ konstrukciju, ieejas un izejas attēlojumiem un atslēgas saraksta īpašībām. Feisteļa shēmas cikla funkcijas $\varphi(x, q)$ realizācijas shēma dota zīmējumā 8.

Feisteļa šifra SP tīkla realizācija ir vienkārša salīdzinot ar vispārīga veida iteratīva SBC SP tīkla realizāciju, kas ir acīmredzami no attēlojumu $\varphi(x, q)$ un $\psi(x_2, q)$ definīcijas, kas ir atbilstošo SP tīklu pamatelementi. Taču Feisteļa šifra SP tīklā ieejas bloku samaisīšana notiek lēnāk, tā kā maisīti tiek tikai vienas bloka puses biti, bet pārējie tiek tikai nobīdīti.

Teorēma 8.2. *Ja katram* $q \in Q'$

$$\pi_q(x) = \delta_q^{-1} \cdot T^n(x),$$

kur T *— bināra vektora kreisās cikliskās nobīdes substitūcija, tad Feisteļa šifrs ir apgriežams un atšifrēšanas algoritms atšķiras no šifrēšanas algoritma ar to, ka cikla atslēgas tiek izmantotas apgrieztā kārtībā.*



8. zīm.: Cikla funkcijas $\varphi(x, q)$ realizācijas shēma

□ E_k un E_k^{-1} teorēmas nosacījumos ir definēti ar formulām:

$$E_k(x) = \delta_{q_{r+1}}^{-1} \cdot T^n \cdot \varphi_{q_r} \cdot \ldots \cdot \varphi_{q_2} \cdot \varphi_{q_1} \cdot \delta_{q_0}(x), \tag{46}$$

$$E_k^{-1}(y) = \delta_{q_0}^{-1} \cdot \varphi_{q_1}^{-1} \cdot \varphi_{q_2}^{-1} \cdot \ldots \cdot \varphi_{q_r}^{-1} \cdot T^n \cdot \delta_{q_{r+1}}(y). \tag{47}$$

No vienādojuma (45) seko, ka katram $q \in Q$

$$\begin{aligned} T^n \cdot \varphi_q \cdot T^n(x_2, \psi(x_2, q) \oplus x_1) &= T^n \cdot \varphi_q(\psi(x_2, q) \oplus x_1, x_2) \\ &= T^n(x_2, \psi(x_2, q) \oplus \psi(x_2, q) \oplus x_1) \\ &= T^n(x_2, x_1) = (x_1, x_2). \end{aligned}$$

Tātad

$$\varphi_q^{-1} = T^n \cdot \varphi_q \cdot T^n. \tag{48}$$

Veiksim substitūcijas sakarā ar (48) vienādojumā (47). Ņemot vērā, ka $T^{2n} = \mathbb{I}$, iegūstam vienādojumu:

$$E_k^{-1}(y) = \delta_{q_0}^{-1} \cdot T^n \cdot \varphi_{q_1} \cdot \varphi_{q_2} \cdot \ldots \cdot \varphi_{q_r} \cdot \delta_{q_{r+1}}(y).$$

Salīdzinot to ar (47), redzam, ka teorēma ir pierādīta. ■

Cikla funkcijas uzbūve

Pazīstamāko bloku šifru atslēgas ir bināri vektori, tāpēc var uzskatīt, ka $Q = \{0,1\}^m$ un aplūkot SBC cikla funkciju $\varphi(x,q)$ kā attēlojumu

$$\varphi : \{0,1\}^{2n+m} \to \{0,1\}^{2n},$$

bet Feisteļa šifra sarežģīšanas funkciju $\psi(x_2,q)$ kā attēlojumu

$$\{0,1\}^{n+m} \to \{0,1\}^n.$$

Bloka šifru pētījumi ir ļāvuši izvirzīt virkni nosacījumu, kurus jāapmierina cikla funkcijai $\varphi(x,q)$ (Feisteļa šifra gadījumā — sarežģīšanas funkcijai $\psi(x_2,q)$), lai tā būtu ar labām kriptogrāfiskām īpašībām. Vairāku nosacījumu vienlaicīga izpilde tiek panākta realizējot šīs funkcijas kā noteikta skaita "elementāru" attēlojumu, kurus sauc par cikla funkcijas $\varphi(x,q)$ *slāņiem*, kompozīciju. Pie tam, katrs slānis nodrošina kādu vai kādas no nepieciešamajām īpašībām, bet to kompozīcija nodrošina visas nepieciešamās īpašības.

Cikla funkcijas (Feisteļa šifra sarežģīšanas funkcijas) konstruktīvajiem slāņiem ir sekojoši funkcionāli uzdevumi:

1. atvasināto atslēgu jaukšana;
2. ieejas bloku jaukšana;
3. sarežģītas nelineāras atkarības starp atslēgas, ieejas un izejas bloku simboliem realizēšana.

Cikla funkcijai jāapmierina virkne nosacījumu.

1. Katram $q \in Q$ cikla funkcijas $\varphi(x,q)$ apakšfunkcijai φ_q jābūt substitūcijai, kas izriet no šifra pārveidojumu apgriežamības prasības. Ja tā ir, tad cikla funkcija $\varphi(x,q)$ ir sabalansēta. Analoģiskai īpašībai jāpiemīt visiem cikla funkcijas slāņiem.

Feisteļa šifra sarežģīšanas funkcija $\psi(x_2, q)$ un tās slāņi principā var neapmierināt šo prasību, jo cikla pārveidojuma apgriežamību nodrošina *XOR* operācijas izmantošana, kā parādīts iepriekš. Tomēr atslēgas noteikšanas kriptoanalītiskās metodes, kas izmanto labus cikla funkcijas afīnos tuvinājumus, stimulē izstrādātāju izvēlēties sabalansētas sarežģīšanas funkcijas.

2. No vienādības (43) seko, ka šifrējošais attēlojums $E_k(x)$ ir afīns attiecībā pret ieejas bloka un cikla atslēgām, ja funkcijas $\delta(x, q)$, $\varphi(x, q)$ un $\pi(x, q)$ ir afīnas. Šinī gadījumā atslēgu nav sarežģīti noteikt pēc pamatteksta un šifrēta teksta atrisinot lineāru vienādojumu sistēmu. Tāpēc cikla funkcijai jābūt nelineārai, pie tam tās koordinātu funkcijām nevajadzētu pieļaut labus afīnus tuvinājumus, jo tādu esamība palielina atslēgas noteikšanas risku.

   Tādā veidā, attēlojuma $\varphi(x, q)$ (vai $\psi(x_2, q)$) koordinātu funkcijām jābūt stipri nelineārām un tām jāpiemīt imunitātei pret korelācijām. Šīs īpašības pamatā nodrošina ar nelineāru substitūcijas slāņa konstrukciju, kaut arī nelinearitāte var tikt pastiprināta arī jaucot atslēgas (skat., piemēram, *IDEA* algoritmu), izmantojot saskaitīšanas vai reizināšanas pēc moduļa operācijas.

   Lieliem n nelineāras substitūcijas uzdevums vienkāršojas, ja nelineārais slānis (apzīmēsim šo pārveidojumu ar $s(x^1, \ldots, x^{2n})$, kur

$$(x^1, \ldots, x^{2n}) \in \{0, 1\}^{2n}$$

   tiek uzdots ar nelineāru pārveidojumu komplektu, kur katrs no tiem apstrādā tikai daļu no ieejas komplekta. Ja $2n$ dalās bez atlikuma ar d, tad

$$s(x^1, \ldots, x^{2n}) = (s_1(x^1, \ldots, x^u), \ldots, s_d(x^{2n-u+1}, \ldots, x^{2n})),$$

   kur $u = 2n/d$, un $s_1, s_2, \ldots, s_d$ — nelineāri kopas $\{0, 1\}^{2n/d}$ pārveidojumi, kas ieguvuši nosaukumu s-box (otrs variants — s–bloki — ir neveiksmīgāks, jo rada zināmu jēdzienu sajukumu ar informācijas blokiem, kurus apstrādā šifrs).

   Nelineārā s-box slānī izmantotais skaitlis d tiek izvēlēts kā kompromiss starp nelineārā slāņa realizāciju un s-box kriptogrāfiskajām kvalitātēm. S-box izveidei iespējami vairāki varianti. Piemēram, katrā *DES* algoritma ciklā datu bloka izmērs no sākuma palielinās no 32 līdz 48 bitiem,

kas atļauj sajaukt 48 bitu cikla atslēgu, bet pēc tam ar nelineāru s-box palīdzību atkal samazinās līdz 32 bitiem.

3. Cikla funkcijas sajaucošajiem slāņiem jārealizē saistība starp s-box ienākošajiem un izejošiem bitiem tādā veidā, lai:

   (a) katrs s-box apmierinātu lavīnas efekta jeb izplatīšanas kritērijus, un, tas nozīmē, būtu pilnīgs ieejas komplektu pārveidojums;

   (b) nākošajā ciklā katra s-box ieejas biti būtu atkarīgi no vairāku iepriekšējā cikla s-box izejām.

   Ja pirmais nosacījums ir izpildīts (ko panāk ar noteiktu s-box konstrukciju), tad viena bita izmaiņa jebkura s-box ieejā noved pie vairāku tā izejas bitu (vidēji ne mazāk kā divu) izmaiņām.

   Otrais nosacījums tiek nodrošināts izmantojot slāni, kas pārvieto starprezultātu bloku koordinātas. Vienkāršākajā gadījumā koordinātas tiek cikliski nobīdītas.

   Abu nosacījumu izpildīšanās atļauj kaut kādam s sasniegt labas attēlojuma $\varphi^{(s)} = \varphi_{q_s}, \ldots, \varphi_{q_1}$ kriptogrāfiskās īpašības, kas tiek realizēts pēc cikla attēlojuma s iterācijām: $\varphi^{(s)}$ apmierina lavīnas tipa efekta kritērijus, konkrēti tie ir pilnīgi. No tā seko, ka r-cikliska SBC šifrējošajam attēlojumam pie $r \geq s$ piemīt tādas pašas pozitīvas kriptogrāfiskās īpašības, proti, katrs šifrētā teksta bits ir būtiski atkarīgs no visiem atslēgas un pamatteksta bitiem. Tas nozīmīgi apgrūtina atslēgas noteikšanu no pamattekstiem un šifrētajiem tekstiem lietojot "skaldi un uzlauz" tipa metodēm, tajā skaitā arī izmantojot speciāli izvēlētus pamatteksta blokus. Piemēram, 16 ciklu *DES* algoritmā pārveidojumi ir pilnīgi pēc pieciem cikliem, bet 32 ciklu algoritmā ГОСТ — pēc trim cikliem.

4. Pie efektīvākajām SBC kriptoanalīzes metodēm ir pieskaitāmi diferenciālās kriptoanalīzes metode un lineārās kriptoanalīzes metode, kas tiek aktīvi attīstītas pēdējos pārdesmit gadus.

   Diferenciālās kriptoanalīzes metodes būtība ir korelāciju starp pamattekstu atšķirībām (*XOR*-summām) un atbilstošo šifrēto tekstu atšķirībām meklēšana un izmantošana ar mērķi noteikt cikla atslēgu daļu.

Lineārās kriptoanalīzes metodes būtība ir lineāru sakarību starp atslēgas, pamatteksta un šifrēto tekstu simboliem noteikšanā, kuras nejauši izvēlētiem atbilstošu bloku pāriem parādās ar varbūtību lielāku par 1/2. Ja šādas sakarības tiek atrastas, tās var tikt izmantotas, lai atmestu nederīgas SBC atslēgu vērtības izmantojot statistiskās metodes. Šajā sakarā ir svarīgi, lai cikla funkcijai piemistu īpašības, kas apgrūtina norādīto metožu pielietošanu. Proti, labai cikla funkcijai nejauši izvēlētiem ieejas un izejas bloku pāriem:

- ir minimāla korelācija starp pamattekstu atšķirībām un atbilstošo šifrēto tekstu atšķirībām;
- katrai netriviālai lineārai sakarībai starp atslēgas, pamatteksta un šifrētā teksta simboliem iespēja parādīties ir nenozīmīgi novirzīta no varbūtības 1/2.

Lai pretotos diferenciālās kriptoanalīzes un lineārās kriptoanalīzes metodēm, svarīgas ir visu cikla funkcijas slāņu īpašības.

Ieejas un izejas attēlojumi

Ieejas un izejas attēlojumu funkcionālais uzdevums kā likums ir nodrošināt šifra apgriežamību un sarežģīt šifrējošo pārveidojumu. Dažās sākotnējās iteratīvo bloka šifru izstrādēs, attēlojumi $\delta(x,q)$ un $\pi(x,q)$ nav atkarīgi no atslēgas un ir pat kopas $\{0,1\}^{2n}$ identiskais attēlojums $\mathbb{I}$, proti, tie praktiski netiek izmantoti E_k kompozīcijā. Piemēram, *DES* algoritmā ieejas attēlojums ir ieejas substitūcija δ, kas nav atkarīga no atslēgas, un tāpat arī izejas attēlojums π, te $\pi = \delta^{-1} \cdot T^n$.

Algoritmā *NewDES* abi pārveidojumi vienādi ar $\mathbb{I}$. Algoritmā ГОСТ pārveidojums $\delta = \mathbb{I}$, pārveidojums $\pi = T^n$.

Vispār bloka šifru ieejas un izejas attēlojumu atslēgas ir bināri vektori, bieži $Q = \{0,1\}^{2n}$, proti, $\delta(x,q)$ un $\pi(x,q)$ var aplūkot kā attēlojumus

$$\{0,1\}^{4n} \to \{0,1\}^{2n}.$$

Lai apgrūtinātu diferenciālās un lineārās kriptoanalīzes metožu pielietošanu iteratīviem SBC, kā ieejas un izejas attēlojumi tiek izmantotas attiecīgi operācijas $XOR(x,q_0)$ un $XOR(x',q_{r+1})$, kur x — pamatteksta bloks, x' — datu bloks pēc šifrēšanas r–tā cikla. Šīs operācijas ieguvušas nosaukumu

balināšana (whitening). Iteratīvu bloka šifru, kas izmanto šīs operācijas, sauc par *šifru ar balināšanu*, bet atslēgas q_0 un q_{r+1} — par *balināšanas atslēgām*.

Apgriežama iteratīva šifra sarežģīšana izmantojot balināšanas operāciju neiespaido tā apgriežamību, jo šādā gadījumā katram $q \in Q'$

$$\delta_q(x) = \delta_q^{-1}(x) = x \oplus q = \pi_q(x) = \pi_q^{-1}(x).$$

Balināšana uzlabo šifra kriptogrāfiskās īpašības, jo šifra ar balināšanu kriptoanalīze ir līdzvērtīga šifra bez balināšanas kriptoanalīzei, kuram ieejas un izejas bloki nav zināmi un ir atbilstoši $x \oplus q_0$ un $y \oplus q_{r+1}$. Šifra realizācija ar balināšanu sarežģījas nenozīmīgi.

Atslēgu saraksta uzbūve

Neveiksmīgi izveidots bloka šifra atslēgu saraksts var nozīmīgi pavājināt šifrēšanas algoritma kriptogrāfiskās īpašības. Aplūkosim piemēru.

Teiksim, ka atslēgas k j-tais bits tiek izmantots šifrēšanas i-tajā ciklā, ja šis bits ir nozīmīgs attēlojuma θ_i, $i \in \{1, 2, \ldots, r\}$ mainīgais. Ja puse no atslēgas k bitiem tiek izmantota tikai šifrēšanas pirmajos l ciklos, bet pārējie biti tikai pēdējos šifrēšanas $r - l$ ciklos, tad atslēgas k noteikšanai no pamatteksta un šifrētā teksta blokiem x un y var lietot "*satikšanās pusceļā*" metodi, ko citādi mēdz saukt par *saskaņošanas metodi*.

Metodes būtība ir tāda, ka katrs vienas atslēgas bitu puses komplekta variants tiek izmantots, lai realizētu pirmos bloka x šifrēšanas l ciklus un to rezultāti tiek ierakstīti atmiņas adresē, kas atbilst starpstāvokļa blokam pēc l cikliem. Katrs otras bitu puses variants tiek realizēts pirmo bloka y $r - l$ bloku atšifrēšanai, pēc tam abi bitu pusīšu komplekti tiek apvienoti, ja sakrīt starpstāvokļu bloki, kas iegūti izmantojot sadalītās šifrēšanas un atšifrēšanas procedūras. Izmantojot atmiņu ar kārtu $\sqrt{|K|} \cdot c$ bitu (c — konstante) saskaņošanas metode samazina atslēgas noteikšanas darbietilpību salīdzinot ar pilno pārlasi apmēram $\sqrt{|K|}$ reizes. Sakarā ar to dubultās šifrēšanas noturība ar neatkarīgām atslēgām ir vienāda ar vienreizējās šifrēšanas noturību, ja ignorē problēmas, kas saistītas ar nepieciešamo atmiņas daudzumu. Tieši tāpēc dubultā šifrēšana izmantojot *DES* algoritmu ar neatkarīgām atslēgām tiek uzskatīta par nelietderīgu.

Līdz ar to viens no nosacījumiem, kas jāapmierina atslēgu sarakstam ir nozīmīga vairāku cikla atslēgu (ne mazāk par divām) atkarība no katra atslēgas k bita. Vēl vairāk, katrā šifrēšanas algoritma "šķērsgriezumā" pa l un

$r-l$ cikliem, $l \in \{1, 2, \ldots, r-1\}$, vismaz vienas "šķērsgriezuma" puses cikla atslēgu kopumam jābūt funkcijai no visiem atslēgas k bitiem.

Nav mazsvarīga arī šifrēšanas ciklu skaita r izvēle. Pie saprātīgi uzbūvētas cikla funkcijas diferenciālās un lineārās kriptoanalīzes efektivitāte samazinās palielinot šifrēšanas ciklu skaitu r. Izejot no tā, tiek izvēlēta r vērtība, kas ir kompromiss starp šifra kriptogrāfisko noturību un šifrēšanas ātrumu.

Cikla atslēgu aprēķināšanas darbietilpīgums (kā arī atslēgu q_0 un q_{r+1} aprēķināšanas darbietilpīgums) nav īpaši svarīgs šifrējot un atšifrējot, jo visas atvasinātās atslēgas no atslēgas k tiek aprēķinātas vienreiz, bet pēc tam tiek ierakstītas atmiņā, lai tās varētu izmantot atkārtoti apstrādājot visu ziņojumu (failu). Turpretim, atvasināto atslēgu aprēķināšanas darbietilpīgums ir svarīgs izstrādājot dešifrēšanas algoritmu, kas saistīts ar šifra atslēgu pārlasi un izriet no daudzkārtēja atvasināto atslēgu pārrēķina.

8.3. Iteratīva šifra vājas atslēgas

Vājas atslēgas jēdziens, kas pirmo reizi tika ieviests veicot *DES* algoritma kriptoanalīzi, saistīts ar atslēgu saraksta īpašībām un dabiskā veidā vispārinās uz iteratīviem SBC. Iteratīva SBC šifrējošā pārveidojuma E_k kriptogrāfiskās īpašības ir jo labākas, jo labāk tas imitē nejaušu pārveidojumu. Tāpēc tiek uzskatīts, ka cikla atslēgu komplektam $q_1, q_2, \ldots, q_r$ jāimitē nejaušu izlasi no kopas Q. Ņemot vērā, ka r ir daudz mazāks par $|Q|$, iegūstam, ka cikla atslēgām $q_1, q_2, \ldots, q_r$ jābūt pa pāriem atšķirīgām. Ar šo īpašību ir saistīts vājas atslēgas jēdziens.

Iteratīva SBC ar r cikliem atslēgu k sauksim par *μ-vāju*, ja atbilstošais cikla atslēgu komplekts $q_1, q_2, \ldots, q_r$ satur μ atšķirīgus elementus, $1 \leq \mu < r$. Par *vāju atslēgu* sauc 1-vāju atslēgu. Nākošā teorēma un tās sekas parāda, cik bīstama var būt vājas atslēgas izmantošana.

Teorēma 8.3. *Pieņemsim, ka k — ir vāja iteratīva SBC ar r cikliem atslēga, un $d \leq r$, kur d — pārveidojuma φ_d kārta. Tādā gadījumā dotais SBC atslēgas k izmantošanas gadījumā ir τ–ciklisks, kur τ — atlikums dalot r ar d.*

□ Vājas atslēgas k gadījumā, kas ģenerē r cikla atslēgas, kas vienādas ar q, iteratīvā SBC pārveidojums E_k saskaņā ar vienādojumu (43) izskatās šādi:

$$E_k(x) = \pi_{q_{r+1}} \cdot (\varphi_q)^r \cdot \delta_{q_0}(x).$$

Tā kā $(\varphi_q)^d = \mathbb{I}$ un $r = d \cdot m + \tau$, tad $(\varphi_q)^r = (\varphi_q)^\tau$. No tā un pēdējās vienādības iegūstam, ka E_k realizē τ-ciklisku iteratīvu SBC. ■

Sekas 8.4. *Ja teorēmas 8.3 nosacījumos r dalās bez atlikuma ar d un pārveidojumi δ_{q_0} un $\pi_{q_{r+1}}$ ir savstarpēji apgriežami, tad šifrējošais pārveidojums E_k ir identisks attēlojums.*

Atzīmēsim, ka šifros ar balināšanu pārveidojumi δ_{q_0} un $\pi_{q_{r+1}}$ ir savstarpēji apgriežami, ja $q_0 = q_{r+1}$.

Teorēma 8.5. *Pieņemsim, ka k ir vāja $2r$–cikliska iteratīva Feisteļa šifra atslēga, $q_0 = q_{r+1}$ un $\pi_{q_0} = \delta_{q_0}^{-1} \cdot T^n(x)$. Tādā gadījumā šifrējošais pārveidojums E_k ir involūcija, kurai ir 2^n nekustīgu elementu.*

□ No teorēmas 8.2 pierādījuma seko, ka vājas atslēgas k gadījumā attēlojumi E_k un E_k^{-1} atšķiras tikai ar pretēju atvasināto atslēgu izmantošanas secību. Pieņemsim, ka vājā atslēga k ģenerē $2r$ cikla atslēgas, kas vienādas ar q. No šejienes, ja $q_0 = q_{r+1}$, no (46) visiem $x \in \{0,1\}^{2n}$ iegūstam:

$$E_k(x) = \delta_{q_0}^{-1} \cdot T^n \cdot (\varphi_q)^{2r} \cdot \delta_{q_0}(x) = E_k^{-1}(x). \tag{49}$$

Tas nozīmē, ka atkārtota šifrēšana ar atslēgu k ir tas pats kas atšifrēšana. Līdz ar to E_k ir involūcija. Saskaitīsim E_k vienības ciklu skaitu. Pieņemsim, ka $\delta_{q_0}(a_1, a_2) = (x_1, x_2)$, kur $a_1, a_2, x_1, x_2 \in \{0,1\}^n$. No (49) seko, ka (a_1, a_2) ir nemainīgs pārveidojuma E_k elements tad un tikai tad, ja (x_1, x_2) ir nemainīgs pārveidojuma $T^n \cdot (\varphi_q)^{2r}$ elements, vai arī, līdzvērtīgi,

$$T^n \cdot (\varphi_q)^r(x_1, x_2) = (\varphi_q)^{-r}(x_1, x_2).$$

Izmantojot (48) iegūstam:

$$(\varphi_q)^r(x_1, x_2) = (\varphi_q)^r \cdot T^n(x_1, x_2). \tag{50}$$

No tā seko, ka E_k vienības ciklu skaits ir vienāds ar bloku (x_1, x_2) skaitu, kas apmierina (50). Tā kā $(\varphi_q)^r$ ir substitūcija, tad šis nosacījums ir līdzvērtīgs tam, ka $(x_1, x_2) = T^n(x_1, x_2) = (x_2, x_1)$ jeb nosacījumam $x_1 = x_2$. Bloku skaits ar vienādām pusēm ir vienāds ar 2^n. ■

Viegli aprēķināt vājo atslēgu skaitu tādiem šifriem kā *DES* un ГОСТ. *DES* algoritma cikla atslēgu ģenerēšanas algoritma būtība ir 24 katra no 28 bitu reģistru koordināšu nolasīšana, kuras tiek aizpildītas cikliski nobīdoties par vienu līdz diviem soļiem. No tā var izvest, ka visas cikla atslēgas ir

vienādas, ja sākotnējais abu reģistru aizpildījums nobīdot nemainās, proti, sastāv tikai no nullēm vai tikai no vieniniekiem. No tā seko, ka *DES* algoritma atslēgu kopā ir 4 atšķirīgas vājas atslēgas.

Noskaidrots, ka *DES* algoritma atslēgu kopā ir sastopamas arī 2^m–vājas atslēgas, kur $m = 1, 2, 3$. Konkrēti, ir 12 dažādas 2-vājas atslēgas (kas parasti tiek sauktas par *pusvājām*), kas veido 6 atslēgu pārus, kuru gadījumā šifrs realizē 6 savstarpēji apgriežamus pārveidojumus.

ГОСТ vājo atslēgu skaits ir vienāds ar bināru matricu (izmērs 8×32) ar vienādām rindiņām skaitu, proti 2^{32}. Vājo atslēgu īpašības neizdevās izmantot, lai samazinātu *DES* un ГОСТ algoritmu noturības novērtējumu, jo vājo atslēgu daļa ir pārlieku maza, un varbūtība, ka šifrējot ar vāju atslēgu tiek izmantots šifrējošā pārveidojuma vienības cikls ir niecīga.

8.4. Šifrēšanas režīmi

Par šifrēšanas režīmiem sauc dažādus datu apstrādes algoritmus, kas uzbūvēti uz bāzes režīma ECB pamata. Šo algoritmu kriptogrāfiskā noturību galvenokārt nosaka bāzes režīma noturība. Taču dažādu šifrēšanas režīmu īpatnības atļauj izmantot bloka šifru dažādu kriptografisko uzdevumu risināšanai.

ECB (vienkārša substitūcija)

Vienkāršās substitūcijas režīmā atslēgai k atbilst 2^{2n} pakāpes aizvietošana E_k, atbilstoši kurai katrs pamatteksta bloks tiek aizvietots ar šifrētā teksta bloku. Tāpēc ECB režīmam piemīt sekojošas īpatnības:

1. Atsevišķu bloku aizvietošana vai pārvietošana šifrētajā tekstā neietekmē atšifrēšanas pareizību.

2. Ar vienādu atslēgu šifrēti vienādi pamatteksta bloki rada vienādus šifrētā teksta blokus.

3. Lai ar šifrējošajām aizvietošanām labi sajauktu informāciju, kas ir kopīga bloka šifru īpašība, katrs no izejas y $2n$ bitiem var tikt izkropļots ar varbūtību 1/2 ja izkropļots ir kaut vai viens nejauši izvēlēts ieejas bloka x bits. Viena kļūda blokā x vidēji rada n kļūdas šifrētajā blokā y. Uz nākošajiem šifrētajiem blokiem kļūda neizplatās. Ja izkropļoti tiek izejas bloka y biti, atbilstošais bloks x tiks atšifrēts nekorekti,

bet pārējie bloki atšifrēsies pareizi. Taču ja šifrētā teksta bits ir nejauši pazaudēts vai pievienots, tad notikušās nobīdes dēļ nepareizi tiks atšifrēts viss sekojošais teksts. Lai lokalizētu nobīdes sekas, jāparedz bloku robežu kontroles līdzekļi.

Pirmās divas īpatnības atļauj aktīvam uzbrucējam, kas kontrolē sakaru līniju, kuru aizsargā šifrs ECB režīmā, novērot atsevišķu bloku un ziņojumu parādīšanās biežumus. Noteiktos apstākļos viņš varēs ģenerēt ziņojumus, nezinot ne atslēgu, ne šifrēšanas algoritmu, pat ja ziņojumi satur laika atzīmi. Šo nozīmīgo nepilnību dēļ ECB režīms netiek izmantots garu ziņojumu šifrēšanai. Šajā režīmā šifrē tikai īsus palīgrakstura ziņojumus: paroles, seansa atslēgas un tamlīdzīgus.

Lielākās ziņojumu daļas garums nedalās ar $2n$. Tāpēc, šifrējot pēdējo nepilno informācijas bloku rodas uzdevums, kā korekti izvēlēties tā šifrēšanas algoritmu. Šī problēma tiek risināta ar dažādiem *bloku papildināšanas* paņēmieniem. Vienkāršākā metode ir papildināt nepilnu m-bitu (m-baitu) pamatteksta bloku ar $2n - m - 8$ bitu garu ($(2n/8) - m - 1$ baitu garu) noteikta veida rindiņu, piemēram visu nuļļu rindiņu, un vienu baitu, kurā norādīts skaitlis $2n - m$ ($2n/8 - m$), kas norāda pēdējā bloka garuma deficītu bitos (baitos). Papildinātais bloks tiek šifrēts parastajā veidā. Ja $2n - m < 8$, tad ziņojuma garumu jāpalielina par vienu bloku.

Citu metodi sauc par *šifrētā teksta nolaupīšanu*. Definēsim attēlojumus

$$v^m : \{0,1\}^r \to \{0,1\}^m \quad \text{un} \quad w^m : \{0,1\}^r \to \{0,1\}^{r-m},$$

kur $1 \leq m < r$ un katram $(x_1, x_2, \ldots, x_r) \in \{0,1\}^r$:

$$v^m(x_1, x_2, \ldots, x_r) = (x_1, x_2, \ldots, x_m),$$
$$w^m(x_1, x_2, \ldots, x_r) = (x_{1+m}, x_{2+m}, \ldots, x_r).$$

Pieņemsim, ka x_t un y_t ir nepilnie m-bitu pamatteksta un šifrētā teksta bloki, bet x_{t-1} un y_{t-1} ir iepriekšējie pilnie pamatteksta un šifrētā teksta bloki. Šādā gadījumā šifrēšanas algoritms tiek izmantots sekojošā veidā:

$$y_t = v^m(E_k(x_{t-1})), \quad y_{t-1} = E_k(x_t, w^m(E_k(x_{t-1}))).$$

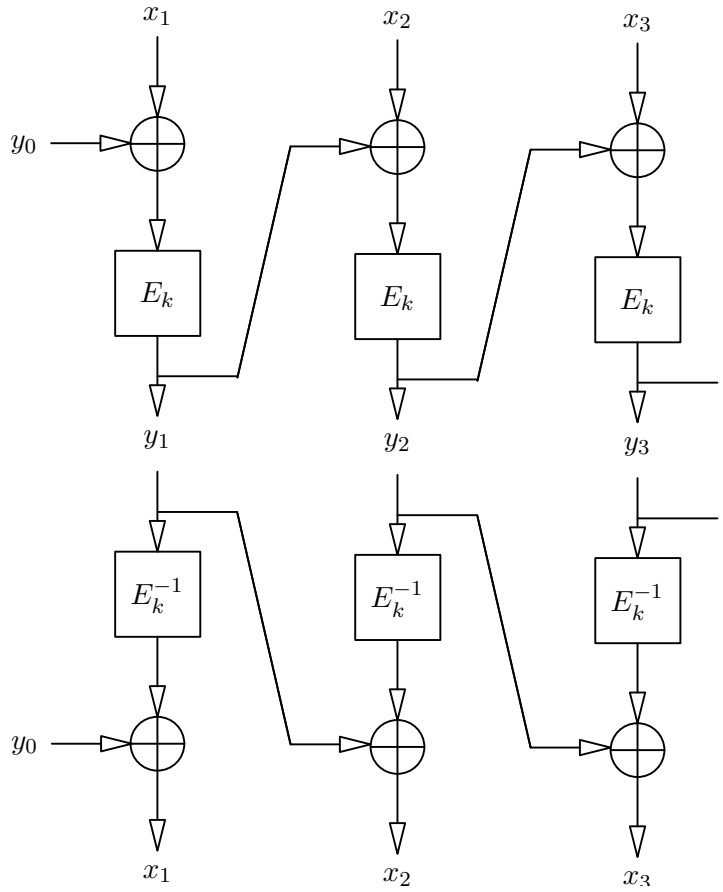
Atšifrējot vispirms tiek aprēķināts bloks $\omega = E_k^{-1}(y_{t-1})$, bet pēc tam tiek noteikti $x_t = v^m(\omega)$ un $x_{t-1} = E_k^{-1}(y_t, w^m(\omega))$.

CBC (šifrētā teksta bloku saķēdēšana)

Bloka šifri CBC režīmā aprakstāmi ar šifrēšanas vienādojumiem

$$y_t = E_k(x_t \oplus y_{t-1}), \quad t = 1, 2, \ldots, \tag{51}$$

kur y_0 ir nejaušs $\{0,1\}^{2n}$ vektors, ko sauc par *inicializācijas vektoru* (*sākotnējo vektoru*), kas tiek ģenerēts pirms katra ziņojuma. 9. attēlā dotas CBC režīma šifrēšanas (augšējā daļa) un atšifrēšanas (apakšējā daļā) blokshēmas. Sākuma



9. zīm.: CBC režīma šifrēšanas un atšifrēšanas shēmas.

vektoru var nodot pa sakaru līniju gan atklātā gan šifrētā veidā (izmantojot ECB režīmu). Svarīgi izvairīties no vienādu inicializācijas vektoru izmantošanas dažādiem ziņojumiem, kas šifrēti ar vienu un to pašu atslēgu.

Tas apgrūtina uzbrukumu šifrētajam tekstam, kas balstīts uz standarta blokiem ziņojuma sākumā. Par inicializācijas vektoru var izmantot nejaušu bitu virkni vai arī laika atzīmi.

Viena bita izkropļojums blokā x_t noved pie tā, ka ir izkropļota vidēji puse bitu visos šifrētā teksta blokos, sākot ar y_t. Atšifrēšanai tas nav nozīmīgi, jo atšifrētais teksts satur tikai šo vienu kļūdu.

Bloka y_t i-tā bita izkropļojums (šādi kropļojumi iespējami dēļ trokšņiem sakaru līnijā vai arī glabāšanas iekārtu bojājumu dēļ) noved pie apmēram puses bloka x_t bitu un bloka x_{t+1} i-tā bita izkropļošanās. Nākošie bloki atšifrējas korekti (pašatjaunojas). Tomēr CBC ir pilnīgi nenoturīgs pret sinhronizācijas kļūdām. Bloku papildināšanu var veikt tāpat kā ECB režīmā, bet ja nepieciešams, lai sakristu ziņojuma un kriptogrammas garumi, var izmantot sekojošas metodes. Nepilna m-bitu bloka x_t gadījumā atbilstošo šifrētā teksta bloku y_t aprēķina šādi:

$$y_t = x_t \oplus v^m(E_k(y_{t-1})).$$

Otra iespēja ir *šifrētā teksta nolaupīšanas* variants. Pieņemsim, ka a ir $(2n-m)$-bitu rindiņa, kas sastāv no nullēm, un $\omega = E_k(x_{t-1} \oplus y_{t-2})$. Šādā gadījumā šifrēšana notiek sekojošā veidā:

$$y_t = v^m(\omega), \quad y_{t-1} = E_k((x_t, a) \oplus \omega).$$

Atšifrējot sākumā tiek atšifrēts datu bloks $\omega' = E_k^{-1}(y_{t-1}) \oplus (y_t, a)$, bet pēc tam var noteikt

$$x_t = v^m(\omega'), \quad x_{t-1} = E_k^{-1}(y_t, w^m(\omega')) \oplus y_{t-2}.$$

CFB (šifrējošā teksta atgriezeniskā saite)

Dažās praktiskās situācijās ienākošās plūsmas simbolus jāvar šifrēt nesagaidot, kamēr izveidojas vesels datu bloks. Šādos gadījumos ir ērts CFB režīms, kurā pamatteksta un šifrētā teksta bloku garums ir m bitu, kur m — režīma parametrs, $1 \leq m < 2n$. Apzīmēsim šādu šifrēšanas režīmu ar CFB-m. Sakarā ar to, ka informācija bieži tiek attēloti baitos, parametru m kā likums izvēlas vienādu ar astoņi.

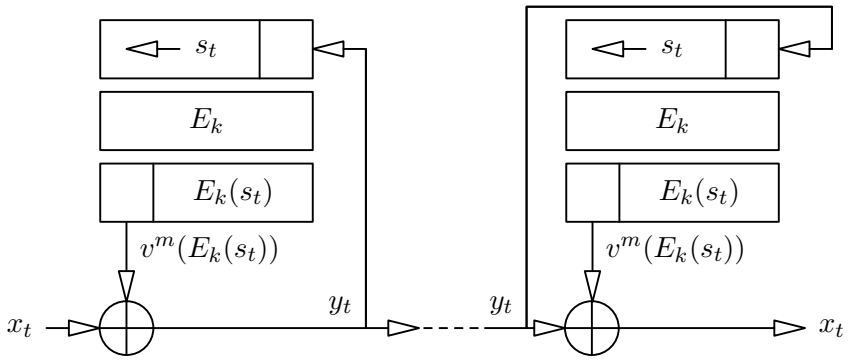
Bloka šifru CFB-m režīmā modelē mš.m $A^m_{CFB} = \langle X, S, Y, K, z, h, f_m \rangle$, kur $X = Y = \{0,1\}^m$, $S = \{0,1\}^{2n}$, $z = s_1$ — no atslēgas k neatkarīgs

iniciālais vektors, bet funkcijas f_m un h izsakāmas kā

$$f_m(s_t, k, x_t) = y_t = v^m(E_k(s_t)) \oplus x_t, \tag{52}$$

$$h(s_t, k, x_t) = s_{t+1} = (w^m(s_t), y_t), \quad t = 1, 2, \ldots \tag{53}$$

Attēlā 10. parādītas šifrēšanas (kreisajā pusē) un atšifrēšanas (labajā pusē) shēmas CFB-m režīmā.



10. zīm.: CFB-m režīma šifrēšanas-atšifrēšanas shēma

Abās procedūrās bāzes režīms tiek izmantots tikai šifrēšanai (aizvietošana E_k^{-1} netiek izmantots). Tāpat kā CBC režīmā sākuma vektoru var nosūtīt pa sakaru līniju atklātā veidā. Tomēr jāizslēdz tā atkārtotas izmantošanas iespēja, šifrējot dažādus ziņojumus ar vienu un to pašu atslēgu.

Viens izkropļots bits blokā x_t noved pie viena bita izkropļošanās y_t, un vidēji pusē bitu visos šifrētā teksta blokos sākot ar y_{t+1}, taču atšifrējot iegūst pamattekstu ar to pašu vienu kļūdu.

Ja tiek izkropļots bloka y_t i-tais bits, tas noved pie i-tā bita izkropļošanās blokā x_t. Pēc tam kļūda nonāk stāvokļu reģistrā un sakropļo vidēji pusi bitu katrā no nākošajiem l blokiem, kur $[2n/m] \leq l \leq]2n/m[$. Tālākie bloki tiks atšifrēti korekti.

CFB režīms patstāvīgi atjaunojas pēc sinhronizācijas kļūdas, tāpat kā SSSC.

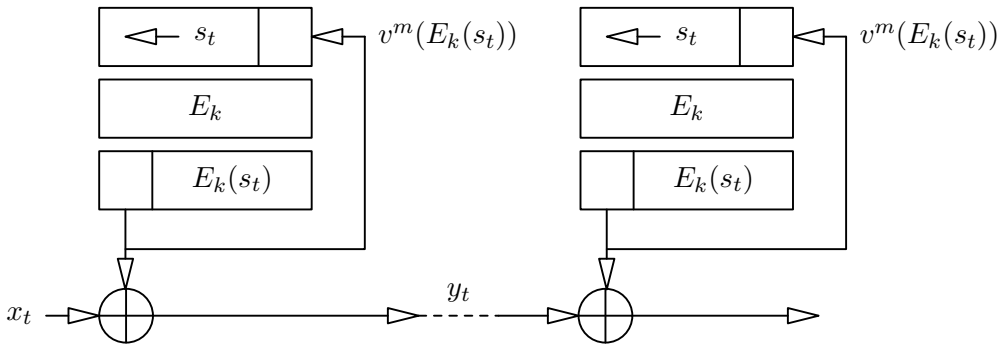
OFB (izejas atgriezeniskā saite, gammēšana)

Bloka šifru gammēšanas režīmā var aplūkot kā sinhronu gammēšanas šifru, kas apstrādā m-bitu pamatteksta un šifrētā teksta blokus (apzīmēsim

šādu režīmu ar OFB-m). Šo šifru modelē mš.m $A^m_{OFB} = \langle X, S, Y, K, z, h, f_m \rangle$, kur $X = Y = \{0,1\}^m$, $S = \{0,1\}^{2n}$, $z = s_1$ — no atslēgas k neatkarīgs inicializācijas vektors, izeju funkciju uzdod vienādojums (52), bet pāreju funkcija nav atkarīga no ieejas un uzrakstāma sekojoši:

$$h(s_t, k) = s_{t+1} = (w^m(s_t), v^m(E_k(s_t))), \quad t = 1, 2, \ldots \tag{54}$$

Attēlā 11. parādītas šifrēšanas (kreisajā pusē) un atšifrēšanas (labajā pusē) shēmas OFB-m režīmā. Abās procedūrās bāzes režīms tiek izmantots tikai šifrēšanai.



11. zīm.: OFB-m režīma šifrēšanas-atšifrēšanas shēma

Inicializācijas vektoru var nodot pa sakaru līniju atklātā veidā, taču jāizslēdz atkārtota tā izmantošana dažādiem ziņojumiem, kas šifrēti ar vienu un to pašu atslēgu.

Izmantojot OFB režīmu svarīgi saglabāt sinhronizāciju. Lai to nodrošinātu, jāparedz sinhronizācijas kontrolēšanas un, gadījumam, ja tā pazūd, sinhronizācijas atjaunošanas metodes.

OFB režīmā kļūdas neizplatās, kas ir pozitīvi, ja tiek nodoti šifrēti skaņas vai video signāli.

Dažu SBC izmantošanas OFB režīmā iespējas ir ierobežotas dēļ ģenerētās gammas samērā īsajiem periodiem. Piemēram, *DES* algoritmam gammas periods ar lielu varbūtību nepārsniedz 2^{32}.

Citi šifrēšanas režīmi

Citu šifrēšanas režīmu izstrādi stimulē centieni novērst dažas četru pamatrežīmu nepilnības.

BC — bloku saķēdēšana. Šo režīmu definē sekojošas vienādības:

$$y_t = E_k(x_t \oplus y_{t-1} \oplus \ldots \oplus y_1 \oplus y_0), t = 1, 2, \ldots,$$

kur y_0 — inicializācijas vektors. Galvenā BC režīma nepilnība ir tā, ka viena vienīga kļūda šifrētajā tekstā nozīmē, ka nekorekti tiek atšifrēti visi sekojošie šifrētā teksta bloki.

PCBC — šifrētā teksta bloku saķēdēšana ar kļūdas izplatīšanos. Režīmu definē sekojošas vienādības:

$$y_t = E_k(x_t \oplus y_{t-1} \oplus x_{t-1}), 1, 2, \ldots$$

Šo režīmu izmanto *Kerberos 4* protokols, lai ar vienu operāciju veiktu gan šifrēšanu, gan ziņojuma veseluma kontroli. PCBC režīmā viena vienīga kļūda šifrētajā tekstā noved pie visu nākošo šifrēto tekstu nepareizas atšifrēšanas, kas tiek izmantots ziņojuma veseluma kontrolei. Tomēr, ja veseluma pārbaude aptver tikai teksta noslēdzošo posmu, var palikt nepamanītas šifrēto bloku maiņa vietām. Šī aizdomīgā īpašība piespieda izstrādātājiem atteikties no šī režīma par labu CBC režīmam nākošajā *Kerberos* protokola versijā.

OFBNLF — nelineāra atgriezeniskā saite ar ieeju. Šis režīms manto dažas OFB un ECB režīmu īpašības. Darbību OFBNLF režīmā modelē multiatslēgu š.m, kura atslēga tiek mainīta katrā blokā:

$$y_t = E_{k_t}(x_t); \quad k_t = E_k(k_{t-1}); \quad t = 1, 2, \ldots$$

Viena kļūda šifrētajā tekstā izplatās tikai uz vienu pamatteksta bloku, tomēr nepieciešams uzturēt sinhronizāciju.

Informācijas apstrādes ātrumu nosaka ne tikai bāzes algoritma šifrēšanas ātrums, bet arī tekošās atslēgas atjaunošanas ātrums.

Sadale

Dažos gadījumos rodas vajadzība paātrināt datu plūsmas šifrēšanu vairākas reizes. To iespējams panākt sadalot datu plūsmu un izmantojot vairākus procesorus. Tiešā veidā to var izdarīt tikai ECB režīmā, bet citus režīmus vajag nedaudz izmainīt izmantojot sadales paņēmienu.

Aplūkosim sadales būtību tādos režīmos kā CBC, CFB, OFB. Lai paātrinātu šifrēšanu apmēram m reizes, pamatteksta bloku virkne

$$X_{\rightarrow} = \{x_1, x_2, \ldots, x_t, \ldots\}$$

tiek sadalīta m apakšvirknēs $X_{\rightarrow,1}, X_{\rightarrow,2}, \ldots, X_{\rightarrow,m}$, kur katram $i = 1, 2, \ldots, m$

$$X_{\rightarrow,i} = \{x_i, x_{i+m}, \ldots, x_{i+t\cdot m}, \ldots\}.$$

Pēc tam katru no m apakšvirknēm šifrē ar atslēgu k un unikālu inicializācijas vektoru.

8.5. Simetrisko bloka šifru uzlabošana

Kriptoanalītiķiem neizdevās izstrādāt praktiski pielietojamu *DES* algoritma dešifrēšanas metodi, kas būtu labāka par pilno atslēgu pārlases metodi. Tomēr, *DES* algoritma atslēgas īsais garums neatļāva to aplūkot kā uzticamu informācijas aizsardzības līdzekli. Tas stimulēja kriptogrāfus nodarboties ar jaunu bloka šifru ar garāku atslēgu izstrādi, kas kā bāzes elementu izmantotu *DES* algoritmu.

Viens no veidiem ir daudzkārtēja šifrēšana izmantojot bāzes algoritmu. Šo metodi var izmantot ar jebkuru SBC, tomēr tā pielietošana noved pie attiecīgo reižu šifrēšanas ātruma samazināšanās (vai arī nepieciešams vairāk aparatūras). Bez tam, svarīgi, lai šifrējošo aizvietošanu kopa nebūtu grupa (*DES* gadījumā tas ir pierādīts), jo pretējā gadījumā daudzkārtēja šifrēšana reducējas uz vienreizēju.

Vienkāršākā daudzkārtējās šifrēšanas shēma ir *divkāršā šifrēšana* izmantojot divas šifrējošas aizvietošanas ar neatkarīgām atslēgām:

$$y_t = E_{k_2}(E_{k_1}(x_t)), \quad t = 1, 2, \ldots$$

Šo shēmu noraidīja jau no paša sākuma, jo atslēgas iespējams noteikt salīdzinot pamattekstu ar šifrēto tekstu, izmantojot saskaņošanas metodi. Metodes darbietilpība ir ar kārtu $|K|$ (kas atbilst bāzes algoritma atslēgu pilnajai pārlasei), pie tam nepieciešama atmiņa ar kārtu $|K|$.

Cita divkāršās šifrēšanas metode, ko sauc par *Deivisa-Praisa metodi*, ir uzbūvēta uz CBC šifrēšanas režīma idejām:

$$y_t = E_{k_2}(x_t \oplus E_{k_1}(y_{t-1})), \quad t = 1, 2, \ldots$$

"Satikšanās pusceļā" metode atļauj noteikt atslēgas arī šinī gadījumā, pie tam metodes sarežģītības raksturlielumi ir aptuveni tādi paši kā iepriekšējā gadījumā.

Noturīgākas shēmas izmanto trīskāršu šifrēšanu. Tačmena trīskāršās šifrēšanas shēmu ar divām neatkarīgām atslēgām k_1 un k_2 sauc par *EDE režīmu* (šifrēšana-atšifrēšana-šifrēšana):

$$y_t = E_{k_1}(E_{k_2}^{-1}(E_{k_1}(x_t)), \quad t = 1, 2, \ldots$$

Ja atslēgas ir vienādas, tad šī shēma atbilst vienreizējai šifrēšanai, kas atļauj šīs shēmas realizēt ar vienu mikroshēmu. Neskatoties uz atslēgu mīšanos, kas izslēdz standarta saskaņošanas metodi, Merkls un Hellmans izstrādāja oriģinālu laika un atmiņas saskaņošanas metodi, kam nepieciešams izpildīt ar kārtu $|K|$ operācijas, ja pieejama atmiņa ar kārtu $|K|$ un kaut kāds daudzums izvēlētu pamatteksta bloku.

Visuzticamākā trīskāršās šifrēšanas shēma ir shēma ar trim neatkarīgām atslēgām:

$$y_t = E_{k_3}(E_{k_2}^{-1}(E_{k_1}(x_t)), \quad t = 1, 2, \ldots$$

Laika un atmiņas saskaņošanas metodei šai shēmai piemīt darbietilpība $|K|^2$ un tai nepieciešama atmiņa ar kārtu $|K|$.

Ir arī *trīskāršās šifrēšanas ar minimālu atslēgu shēma — TEMK*, kurā trīskāršā šifrēšana tiek izmantota divos veidos: kā datu šifrēšanas shēma un kā inicializācijas funkcija, kas no divām neatkarīgām atslēgas k_1 un k_2 aprēķina trīs atvasinātās šifrēšanas atslēgas q_1, q_2 un q_3:

$$q_i = E_{k_1}(E_{k_2}^{-1}(E_{k_1}(x_i)), \quad i = 1, 2, 3,$$

kur x_1, x_2 un x_3 — bloki, kas nav slepeni. Atslēgas k_1 un k_2 nosaka ar pilno pārlasi. Proti shēmas uzlaušanas darbietilpībai ir kārta $|K|^2$, bet atmiņa nav nepieciešama.

Aplūkotās vairākkārtējās šifrēšanas shēmas var tikt savietotas ar dažādiem šifrēšanas režīmiem. Pie citām vairākkārtējās šifrēšanas metodēm pieskaitāma *dubultās gammēšanas* shēma ar šifrēšanas vienādojumiem

$$y_t = x_t \oplus \gamma_t^{(1)} \oplus \gamma_t^{(2)}, \quad t = 1, 2, \ldots,$$

kur gammas $\{\gamma_t^{(1)}\}$ un $\{\gamma_t^{(2)}\}$ ģenerē izmantojot neatkarīgas atslēgas k_1 un k_2 sekojošā veidā:

$$\gamma_t^{(i)} = E_{k_i}(\gamma_{t-1}^{(i)} \oplus \xi(t)), \quad i = 1, 2,$$

bet $\xi(t)$ ir $2n$-bitu vektors, kas sakrīt ar skaitļa t bināro reprezentāciju.

Ir arī vairākkārtējas šifrēšanas shēmas, kas saistītas ar izmantoto atslēgu un apstrādāto bloku izmēru palielināšanu, un shēmas, kas apvieno vairāku bāzes algoritmu izmantošanu.

9. Pseidogadījumvirkņu ğenerēšana

Pseidogadījumvirkņu ğenerēšana. Lineāras rekurentas virknes. Virknes lineārā sarežğītība. Statistiskās prasības. Virkņu statistiskā testēšana

9.1. Pieejas virkņu analīzei

Pseidogadījumvirkņu, kas tiek izmantotas šifrēšanai, novērtēšanas kritēriji ir ārkārtīgi daudzveidīgi. Katru no šifrējošo virkņu analīzes pieejām var pieskaitīt vienai no divām grupām.

Pirmā grupa saistīta ar likumsakarību meklēšanu, kas atļautu atjaunot šifrējošo virkni zinot nosacīti nelielu nogriezni. Pie tam pamatprasības reducējas uz to, lai pseidogadījumvirknē nebūtu sastopamas relatīvi vienkāršas starpsimbolu atkarības.

Otra kritēriju grupa saistīta ar virkņu statistisko īpašību novērtēšanu: vai pētāmajā virknē ir sastopams kaut kāds biežumu disbalanss, kas analītiķim atļautu pieņemt nākošā bita vērtību ar varbūtību lielāku nekā nejaušās izvēles gadījumā. Pie tam pamatprasības pret šifrējošo virkni var reducēt uz to, ka pseidogadījumvirknei piemīt tādas pašas īpašības kādas piemistu nejaušai virknei. Šie nosacījumi konkrēti nozīmē, ka biežumiem, ar kādiem tiek sastapti gan atsevišķi simboli, gan s-grammas, būtu jābūt vienmērīgi sadalītiem.

Abas pseidogadījumvirkņu analīzes grupas veido sistemātisku pieeju plūsmas šifru izstrādei. Tikai dažām atsevišķām pseidogadījumvirkņu klasēm izdodas analītiski pierādīt dažas svarīgas īpašības. Lai pamatotu daudzas citas īpašības tiek izmantoti statistiskie testi. Sistemātiski pieejot gadījumvirkņu analīzei tiek izmantoti pazīstami testi un izstrādāti jauni testi, ņemot vērā pētāmā objekta īpatnības. Ja analīzes gaitā kaut kādā gadījumvirkņu klasē tiek novērota jauna vājība, tiek izstrādāts jauns tests, kurš papildina izmantoto zināmo testu komplektu.

9.2. Lineāras rekurentas virknes

Pseidogadījumvirkņu, kas tiek izmantotas kriptogrāfiskiem lietojumiem, ğenerēšana balstīta uz kaut kādas galīgas kopas X pārveidojumu daudzkārtēju iterāciju realizācijas. Viens no kriptogrāfisko shēmu bāzes elementiem ir lineārie nobīdes reğistri (LNR).

Aplūkosim koordinātu virkņu, kas tiek ģenerētas ar LNR, īpašības. Lauka L elementu virkni (x_i), $i = 1, 2, \ldots$ sauc par *lineāru rekurentu virkni* (LRV), ar kārtu $n > 0$, ja eksistē konstantes $a_0, a_1, \ldots, a_{n-1} \in L$ tādas, ka katram $i \geq 0$

$$x_{i+n} = \sum_{j=0}^{n-1} a_j \cdot x_{i+j}. \tag{55}$$

Vienādību (55) sauc par *rekursijas likumu*, polinomu pār lauku L

$$F(\lambda) = \lambda^n - a_{n-1} \cdot \lambda^{n-1} - a_{n-2} \cdot \lambda^{n-2} - \ldots - a_1 \cdot \lambda - a_0 \tag{56}$$

sauc par LRV *raksturīgo polinomu*, bet vektoru $(x_0, x_1, \ldots, x_{n-1})$ par LRV *sākuma vektoru*. Dotais LRV sakrīt ar j-to vektoru virknes $j = 1, 2, \ldots, n$, koordinātu virkni, kuru ģenerē LNR ar raksturīgo polinomu (56). Tāpēc LRV periods sakrīt ar attiecīgā LNR pārveidojumu periodu. LRV, kura maksimālā perioda kārta ir n, apzīmēsim ar LRVmax-n.

Funkciju $l_i(x_1, x_2, \ldots, x_n)$, kas attēlo LNR sākuma stāvokļu kopu ar garumu n par i-to tā ģenerētā LRV locekli, sauksim par *LNR i-to izejas funkciju*, $i = 1, 2, \ldots$

Apgalvojums 9.1. *LNRmax-n izejas funkciju pār lauku L ar kārtu k virkne ir tīri periodiska ar periodu $k^n - 1$ un sastāv no visām lineārajām n mainīgo funkcijām, kas atšķirīgas no nulles.*

□ No LRVmax-n periodiskuma seko, ka LNRmax-n izejas funkciju virkne ir periodiska ar periodu $k^n - 1$. Katra no LNR ieejas funkcijām ir lineāra sakarā ar LNR pārveidojumu linearitāti un neatkārtojas perioda laikā, jo pretējā gadījumā, ja i-tā un j-tā izejas funkcijas sakristu, $1 \leq i < j \leq k^n - 1$, periods t būtu $j - i$, kas ir mazāk nekā $k^n - 1$. Tas nozīmē, ka visas no nulles atšķirīgās lineārās funkcijas $x_1, x_2, \ldots, x_n$ parādās periodā vienu reizi. ■

Atzīmēsim svarīgas LRVmax-n statistiskās īpašības.

Apgalvojums 9.2. *LRVmax-n pār lauku L ar kārtu k katra nenulles s-gramma parādās k^{n-s} reizes, bet tukšā s-gramma parādās $k^{n-s} - 1$ reizes, $1 \leq s \leq n$.*

□ Periodisks LRVmax-n nogriezni (x_i), $i = 0, 1, 2, \ldots, k^n - 2$, var stādīties priekšā kā virkni $(\,(x_i, x_{i+1}, \ldots, x_{i+n-1})\,)$, ko veido visi telpas $L^{(n)}$ pār lauku L nenulles vektori. Tāpēc katras LRVmax-n periodā sastopamās s-grammas

biežums ir vienāds ar telpas $L^{(n)}$ nenulles vektoru skaitu, kuros s pēc kārtas ņemtas koordinātas (piemēram, s pirmās koordinātes) sakrīt ar uzdoto s-grammu. ■

Līdz ar to, LRVmax-n piemīt labas statistiskās īpašības. Tomēr, LRV rekursijas likums nozīmē, ka LRV piemīt samērā vienkāršas starpsimbolu sakarības. Šīs sakarības atļauj, piemēram, no neliela LRV fragmenta, atrisinot lineāru vienādojumu sistēmu, noteikt sākuma vektoru. Tāpēc kriptogrāfiskās shēmās, kurās tiek izmantots LRV, paredzētas metodes, kas sarežģī ģenerētās virknes.

9.3. Virknes lineārā sarežģītība

Pieņemsim, ka L^m m–dimensionāla vektoru telpa pār lauku L un

$$X_{\rightarrow} = (x_i), \quad i = 0, 1, 2, \ldots$$

ir telpas L^m elementu virkne.

Definīcija 9.3. *Nenulles polinomu*

$$P(X) = X^n - a_{n-1}X^{n-1} - a_{n-2}X^{n-2} - \ldots - a_1X - a_0$$

pār lauku L sauc par par virknes $X_{\rightarrow}$ anulatoru, ja

$$\forall j \geq n \quad x_j - a_{n-1}x_{j-1} - a_{n-2}x_{j-2} - \ldots - a_1x_{j-n+1} - a_0x_{j-n} = \mathbf{0}\,.$$

Virknes $X_{\rightarrow}$ anulatoru veidotās kopas apzīmēšanai lietosim pierakstu $\mathrm{Ann}(X_{\rightarrow})$.

Sekas 9.4. *Ja $f(X) \in \mathrm{Ann}(X_{\rightarrow})$ un $g(X) \in L[X]$ ir nenulles polinoms, tad $f(X)g(X) \in \mathrm{Ann}(X_{\rightarrow})$.*

Sekas 9.5. *Ja $f_1(X), f_2(X), \ldots, f_k(X) \in \mathrm{Ann}(X_{\rightarrow})$, tad jebkura šo polinomu netriviāla lineāra kombinācija arī ir anulators.*

Polinomu $m_{X_{\rightarrow}} \in \mathrm{Ann}(X_{\rightarrow})$ sauc par *virknes $X_{\rightarrow}$ minimālo polinomu*, ja

$$\forall P \in \mathrm{Ann}(X_{\rightarrow}) \quad \deg P \geq \deg m_{X_{\rightarrow}}\,.$$

Skaitli $\deg m_{X_{\rightarrow}}$ sauc par *virknes $X_{\rightarrow}$ lineāro sarežģītību.*

9.4. Statistiskās prasības pret virknēm

Vienu no pirmajiem trīs virkņu statistisko īpašību prasību formulējumiem sniedza S.Golombs. Šīs īpašības tika noformulētas binārām virknēm un tās kriptogrāfijā pazīstamas kā *Golomba postulāti*. Pieņemsim, ka $X_{\rightarrow} = (x_0, x_1, \ldots, x_{T-1})$ — bināra tīri periodiska virkne ar periodu T. Ar n_1 un n_0 apzīmēsim "vieninieku" un "nuļļu" skaitu:

$$n_1 = x_0 + x_1 + \cdots + x_{T-1}, \quad n_0 = T - n_1,$$

bet ar n_1^s un n_0^s — attiecīgi s-grammu skaitu, kas veidotas attiecīgi no "vieniniekiem" un "nullēm", $s \geq 1$. Ar $n_1(d)$ un $n_0(d)$ apzīmēsim attiecīgi "vieninieku" un "nuļļu" skaitu virknē $X_{\rightarrow}^d$:

$$X_{\rightarrow}^d = (x_i \oplus x_{i+d}), \quad d \in \{0, 1, \ldots, T-1\}, \quad i = 0, 1, 2, \ldots,$$

kur indeksi tiek aplūkoti pēc moduļa T. Par virknes $X_{\rightarrow}$ *autokorelācijas funkciju* sauc argumenta d funkciju (apzīmē ar $c_{X_{\rightarrow}}(d)$):

$$c_{X_{\rightarrow}}(d) = (n_1(d) - n_0(d))/T.$$

Golumba postulāti ir šādi:

1. $|n_1^1 - n_0^1| \leq 1$.
2. $n_1 \cdot 2^{-s} = n_1^s = n_0^s = n_0 \cdot 2^{-s}, \quad s = 1, 2, \ldots, [\log_2 T]$.
3. Funkcija $c_{X_{\rightarrow}}(d)$ var pieņemt tikai divas vērtības.

Trešais likums formulē virknes $X_{\rightarrow}$ simbolu neatkarības nosacījumu. Vienlaicīgi tas ir arī zināms virknes $X_{\rightarrow}$ un tās kopijas, kas sākas citā cikla punktā atšķiramības nosacījums. Virknes, kas apmierina trīs Golomba postulātus, sauc par *pseidotroksni* jeb *PT-virknēm*. Atzīmēsim, ka LRV ar maksimālu periodu ir pseidotrokšņi. Tas nozīmē, ka postulāti tika formulēti izejot no labu statistisko īpašību saglabāšanas nosacījuma, kas piemīt LRV ar maksimālu periodu.

Jāpiebilst, ka LKG ir mazāk pievilcīgs kriptogrāfiski pielietojamu pseidogadījumvirkņu ģenerēšanai, jo tā ģenerētajām virknēm piemīt samērā izteiktas starpsimbolu sakarības. Tomēr dažos uzdevumos tiek izmantotas LKG virknes dēļ to realizācijas ērtības. Piemēram, ja pilna cikla LKG, ko definē formula

$$\varphi(x) = (a \cdot x + b) \mod 2^r,$$

izmanto kā kopas $\{0,1\}^r$ pseidogadījumskaitļu ģeneratoru, tad LKG parametrus vēlams izvēlēties sekojošā veidā:

1. $r = n - 1$, kur n — vārda garums datorā;
2. $a \equiv 1 \mod 4$ un ir nenozīmīgi lielāks par $2^{[r/2]}$;
3. b — nepāra skaitlis.

Golomba likumi negarantē augstu pseidogadījumvirknes kvalitāti, bet drīzāk ir tai nepieciešami. Laika gaitā noformulētas arī citas prasības. Proti, ieviests jēdziens par (kaut kādā kopā) *vienmērīgi sadalītu gadījuma virkni* (VSGV). VSGV (kopā X ar apjomu k) — tā ir gadījuma lielumu virkne $(\zeta_1, \zeta_2, \dots, \zeta_t, \dots)$ ar vērtībām no kopas X, kas definēta kaut kādā varbūtību telpā un apmierina divus nosacījumus:

1. katram n un patvaļīgām indeksu vērtībām $1 \le t_1 < \cdots < t_n$ gadījuma lielumi $\zeta_{t_1}, \dots, \zeta_{t_n}$ ir kopumā neatkarīgi;
2. katram naturālam t gadījuma lielums ζ_t vienmērīgi sadalīts kopā X, proti, $P(\zeta_t = x) = 1/k$, katram $x \in X$.

Ja izpildās šie divi pamatnosacījumi, tad izpildās arī virkne citu VSGV īpašību. Aplūkosim dažas no tām.

1. Katram n un patvaļīgām indeksu $1 \le t_1 < \cdots < t_n$ vērtībām n-izmēra gadījuma lielums $(\zeta_{t_1}, \dots, \zeta_{t_n})$ ir vienmērīgs kopā $X^{(n)}$.
2. Katrai naturālās virknes apakšvirknei $1 \le t_1 < \cdots < t_n < \cdots$ atbilstošā virknes (ζ_t) apakšvirkne $\zeta_{t_1}, \dots, \zeta_{t_n}, \dots$ arī ir VSGV.
3. Ja X — aditīva grupa un (η_t) — patvaļīga gadījuma virkne vai patvaļīga gadījuma virkne pār X, kas nav atkarīga no (ζ_t), tad gadījuma virkne (y_t), kur $y_t = \zeta_t + \eta_t$, arī ir VSGV.
4. Katram naturālam t vērtības ζ_t noteikšana izmantojot vērtības $\zeta_1, \zeta_2, \dots, \zeta_{t-1}$ nav iespējama.

VSGV ģenerators ir iekārta, kas atļauj iegūt patvaļīga garuma VSGV realizāciju. Šīs realizācijas elementus sauc par *gadījuma skaitļiem*.

Izšķir 3 VSGV ģeneratoru tipus: tabulas, fiziskais un programmētais.

Tabulas VSGV ģenerators ir nejaušu skaitļu tabula, kas iegūta eksperimentāli izvēloties skaitļus no vienmērīga sadalījuma. Tabulas ģeneratoru galvenie trūkumi ir:

1. tabulas apjomu ierobežojumi;
2. liels datora operatīvās atmiņas patēriņš, kas nepieciešams lai glabātu nejaušo skaitļu tabulu;
3. nepieciešamība aizsargāt tabulas masīvus.

Fiziskais VSGV ģenerators ir elektroniska iekārta, kuras izejas signāls pēc savas dabas ir nejaušs process. Fizisko ģeneratoru trūkumi ir:

1. iepriekš iegūtas realizācijas neatkārtojamība;
2. elektroniskās iekārtas darbības nestabilitāte (traucējumi, kļūdas, režīma nobīdes un citas nejaušības), kas rada vajadzību kontrolēt katru realizāciju.

Programmētais VSGV ģenerators ir datorprogramma, kas imitē VSGV. Tā galvenais trūkums — realizētajai virknei piemītošās sakarības. Programmēta ģeneratora kvalitāte ir tiešā veidā saistīta ar šo sakarību noskaidrošanas sarežģītību. Lai noskaidrotu šīs sakarības analizētajām virknēm tiek pielietots plašs dažādu statistisko testu klāsts.

9.5. Virkņu statistiskā testēšana

Pēdējās desmitgadēs izstrādāts liels daudzums virkņu "nejaušuma" analīzes testu. Testēšanas būtība parasti reducējas uz tā sauktās "nulles hipotēzes" pārbaudi attiecībā uz pētāmo virkni, saskaņā ar kuru virkne ar garumu N iegūta izmantojot N Bernulli shēmas mēģinājumus ar "vieninieka" parādīšanās varbūtību $1/2$.

Statistisku testu T binārai virknei ar garumu N var aplūkot kā Būla funkciju (b.f)

$$T : \{0,1\}^N \to \{1,0\} = \{\text{"pieņemt"}, \text{"noraidīt"}\},$$

kura sadala bināro virkņu ar garumu N kopu $\{0,1\}^N$ kopās $V_{N,0}$ ("ne-nejaušu" virkņu) un $V_{N,1} = \{0,1\}^N \setminus V_{N,0}$ (nejaušu virkņu):

$$V_{N,j} = \{s^N \in \{0,1\}^N : T(s^N) = j\}, \quad j \in \{0,1\},$$

kur $s^N = (s_1, s_2, \ldots, s_N)$. Varbūtību ρ, ka nejauši izvēlēta virkne ar garumu N tiks noraidīta izmantojot testu, izsaka vienādība

$$\rho = |V_{N,0}| \cdot 2^{-N}.$$

Kā likums, reālos testos ρ ir neliels, $\rho \leq 0,01$.

Virknēm ar lielu garumu N testa realizācija nosakot precīzu b.f T vērtību prasa darbietilpīgus aprēķinus. Tāpēc statistisko testu parasti realizē uzdodot efektīvi izrēķināmu testa funkciju (statistiku) f_T, kura attēlo $\{0,1\}^N$ reālo skaitļu kopā.

Apzīmēsim N neatkarīgu un vienādi sadalītu bināru lielumu virkni ar R^N un aplūkosim varbūtisko gadījuma lieluma $f_T(R^N)$, kas pieņem reālas vērtības, sadalījumu. Dotai ρ vērtībai funkcijai f_T tiek uzdoti augšējie un apakšējie sliekšņi t_1 un t_2:

$$P\{f_T(R^N) \leq t_1\} + P\{f_T(R^N) \geq t_2\} = \rho.$$

Parasti sliekšņi tiek izvēlēti tā, lai

$$P\{f_T(R^N) \leq t_1\} \approx P\{f_T(R^N) \geq t_2\} \approx \rho/2.$$

"Ne-nejaušo" virkņu kopu $V_{N,0}$ ar apjomu $\rho \cdot 2^N$ uzdot attiecība

$$V_{N,0} = \{s^N \in \{0,1\}^N : f_T(s^N) \leq t_1 \vee f_T(s^N) \geq t_2\}.$$

Parasti funkciju f_T izvēlas tādā veidā, lai lieluma $f_T(R^N)$ sadalījums būtu pietiekoši tuvs labi pazīstamam "etalona" sadalījumam. Visbiežāk šāds "etalona" sadalījums ir vai nu normālais sadalījums vai arī *hi-kvadrāta* sadalījums ar kaut kādu brīvības pakāpju skaitu. Tā kā šādiem varbūtiskajiem sadalījumiem ir sastādītas precīzas skaitliskas tabulas, tas atvieglo sliekšņu t_1 un t_2 noteikšanu dotām ρ un N vērtībām.

Parasti normālo sadalījumu iegūst, ja summē lielu daudzumu neatkarīgus un vienādi sadalītus gadījuma lielumus. *Hi-kvadrāta* sadalījumu ar d brīvības pakāpēm iegūst, ja tiek summēti d neatkarīgu un normāli sadalītu (ar matemātisko cerību 0 un dispersiju 1) gadījuma lielumu kvadrāti. Aplūkosim dažus populārākos statistiskos testus.

Biežumu tests

Biežuma testa statistika $f_b(s^N)$ uzrakstāma šādi:

$$f_b(s^N) = \frac{2}{\sqrt{N}} \cdot \left(\sum_{i=1}^{N} s_i - \frac{N}{2} \right).$$

Vieninieku skaits nejaušā virknē $R^N = (R_1, R_2, \ldots, R_N)$ ir sadalīts atbilstoši binomiālajam sadalījumam, kuru pie $N \geq 30$ labi aproksimē normālais sadalījums ar matemātisko cerību 0 un dispersiju 1. Pieņemamas kritiskās vērtības ir $t_1 = -t_2 \approx 2,5 \div 3,0$.

Autokorelāciju tests

Par virknes s^N *autokorelācijas testu ar aizturi* τ sauc biežuma testu virknei

$$s_\tau^N = (s_1 \oplus s_{1+\tau}, s_2 \oplus s_{2+\tau}, \ldots, s_N \oplus s_{N+\tau}).$$

Šo testu izmanto lai noteiktu korelācijas starp virknes s^N bitiem attālumā τ.

Virkņu tests

Realizējot virkņu testu ar parametru L virkni s^N sadala N/L nogriežņos ar garumu L (piemēram baitos, $L = 8$) un tiek noteikts biežums n_i, ar kādu virknē s^N parādās binārais skaitļa i attēlojums, $0 \le i \le 2^L - 1$. Virkņu testa $f_n(s^N)$ uzrakstāms sekojoši:

$$f_n(s^N) = \frac{L \cdot 2^L}{N} \cdot \sum_{i=0}^{2^L-1} \left(n_i - \frac{N}{L \cdot 2^L} \right)^2 .$$

Loceklis $N/(L \cdot 2^L)$ — ir vidējā lieluma n_i vērtība un vienlaicīgi dalītājs, kas normē kvadrātā nepaceltos summas locekļus ar vidējo vērtību 0 un nodrošina, ka dispersija ir vienāda ar viens.

Varbūtisko statistikas $f_n(s^N)$ sadalījumu lielām N vērtībām aproksimē *hi-kvadrāts* ar $2^L - 1$ brīvības pakāpēm. Šo sadalījumu rekomendēts izmantot, ja vidējā lielumu n_i vērtība nav mazāka par pieci. No tā seko, ka dotai parametra vērtībai L testējamās virknes garumam jābūt vismaz $5L \cdot 2^L$ bitu.

Sēriju tests

Virknes $X_\rightarrow$ "vieninieku" ("nuļļu") r-grammu $(x_{i+1}, x_{i+2}, \ldots, x_{i+r})$ sauksim par *1-sēriju (0-sēriju) ar garumu* r, $r \ge 1$, ja $x_i = x_{i+r+1} = 0$ ($x_i = x_{i+r+1} = 1$), $i \in \{0, 1, \ldots, T-1\}$, kur indeksi tiek aplūkoti pēc moduļa T. Realizējot sēriju testu virknei s^N tiek noteikts 1-sēriju un 0-sēriju skaits ar garumu r (ko apzīmē, attiecīgi, ar $z_{1,r}$ un $z_{0,r}$), $1 \le r \le L$, (piemēram, $L = 15$). Sēriju testa statistika $f_s(s^N)$ uzrakstāma šādi:

$$f_s(s^N) = \sum_{r=1}^{L} \frac{(z_{1,r} - N/2^{r+2})^2}{N/2^{r+2}} + \sum_{r=1}^{L} \frac{(z_{0,r} - N/2^{r+2})^2}{N/2^{r+2}}.$$

Varbūtiskais šīs statistikas sadalījums pie lieliem N ir labi aproksimējams ar *hi-kvadrāta* sadalījumu ar $2L$ brīvības pakāpēm. No citiem testiem ir vērts

r	*Virkņu tests*	*Universālais tests*	*Atkārtošanās tests*
14	$1,15 \cdot 10^6$	$2,32 \cdot 10^8$	14336
16	$5,24 \cdot 10^6$	$1,06 \cdot 10^9$	32768
20	$1,05 \cdot 10^8$	$2,12 \cdot 10^{10}$	$1,64 \cdot 10^5$
24	$2,01 \cdot 10^9$	$4,07 \cdot 10^{11}$	$7,86 \cdot 10^5$
28	$3,76 \cdot 10^{10}$	$7,59 \cdot 10^{12}$	$3,67 \cdot 10^6$
32	$6,87 \cdot 10^{11}$	$1,39 \cdot 10^{14}$	$1,68 \cdot 10^7$

3. tabula: Minimālie bināro virkņu garumi r-grammu analīzei (bitos)

pieminēt *universālo testu*, kas operē ar virknes s^N blakus esošu vienādu r-grammu attālumiem, un *atkārtošanās testu*, kas mēra atkārtojošos r-grammu skaitu. 3. tabulā doti minimālie virkņu garumi, kas nepieciešami dažādo testu pielietošanai. Praksē nulles hipotēzes pieņemšanu vai noraidīšanu balsta uz vairāku neatkarīgu testu pielietošanas rezultātiem. Gadījumos, kad neatkarīgie testi noved pie atšķirīgiem secinājumiem, testu rezultāti tiek kombinēti izmantojot statistikas, kas ņem vērā visu izmantoto testu rezultātu kopumu. Ja tiek kombinēts neliels testu daudzums, tiek izmantota Fišera-Pīrsona statistika, kas tiek salīdzināta ar *hi-kvadrātu*. Ja tiek kombinēts pietiekoši liels testu daudzums, tad ir rekomendējams izmantot Kolmogorova-Smirnova testu.

Testējot tā vai cita pseidogadījumvirkņu ģeneratora īpašības tiek pētīts liels daudzums ar to ģenerētu pseidogadījumvirkņu, un tiek novērtēts, kāda to daļa neiztur izmantotos testus. Ģenerators tiek uzskatīts par derīgu, ja virkņu, kas atzītas par ne-nejaušām, daļa ir salīdzināma ar analoģisku nejauša ģeneratora raksturlielumu.

No kriptogrāfisko lietojumu viedokļa, pat virkne, kas ģenerēta izmantojot pilnīgi nejaušu avotu, var izrādīties pilnīgi nepiemērota šifrēšanai. Piemēram, virkne, kurā nomācošā daudzumā sastopamas nulles. Tāpēc ģeneratora konstrukcijai jāgarantē, ka acīmredzami vājas virknes neparādās kriptogrāfiskas iekārtas izejā.

10. Kriptogrāfiskie ğeneratori

Kriptogrāfiskie ğeneratori. Filtrējošie ğeneratori. Kombinētie ğeneratori. Korelācijas uzbrukums. Korteža ğenerēšana ar nevienmērīgu soli. Ģeneratori ar atmiņu.

Kriptogrāfiskos ğeneratorus (k.ğ) izmanto atslēgu un plūsmas šifru vadības virkņu veidošanai. K.ğ $A_G = (S, Y, K, z, g, h, f)$ kvalitāti nosaka tā izejas virkņu kriptogrāfiskās īpašības. Iepazīsimies ar ğeneratoru kriptoshēmu izveides galvenajiem paņēmieniem.

10.1. Kriptoshēmu bāzes elementi

Daudzu ğeneratoru kriptoshēmu bāzes elements ir LNR ar maksimālu periodu, kura izejas virknei (LRV) ir liels periods un labas statistiskās īpašības. No otras puses LRV nepiemīt nelineāras īpašības (piemēram, tām ir neliela lineārā sarežğītība), kas neatļauj aplūkot LNR kā galīgu kvalitatīva k.ğ shēmu. Tāpēc LNR tiek izmantoti kopā ar dažādām funkcionālām shēmām un atmiņas elementiem. Atmiņas elementu uzdevums — ienest kriptoshēmā zināmas nelineāras īpašības, pie tam nezaudējot LRV labās īpašības. Šāda ğeneratora gammu var aplūkot kā zināmu LRV uzlabojumu.

LKG priekšrocība ir virkņu aprēķināšanas ātrums, izmantojot modernus skaitļotājus. Taču šifru veidošanai tiem nav atradies patstāvīgs pielietojums dēļ to "prognozējamības", proti, LKG visus parametrus iespējams atjaunot izmantojot nedaudzus virknes locekļus, pat ja tie nav pilnībā zināmi. Tādas pašas priekšrocības un trūkumi zināmā mērā piemīt arī polinomiālajiem kongruences ğeneratoriem.

10.2. Filtrējošie ğeneratori

Viens no vienkāršākajiem veidiem kā sarežğīt virknes pār galīgu kopu X ir to n-grammu attēlošana par citu virkni, izmantojot n mainīgo funkciju.

Par *filtrējošo shēmu* pār X ar *filtru (filtrējošo funkciju)* $f(x_1, x_2, \ldots, x_n)$ (apzīmēsim — f.s $f(x_1, x_2, \ldots, x_n)$) sauksim Mūra automātu

$$A = \langle X, X^n, X, h, f \rangle,$$

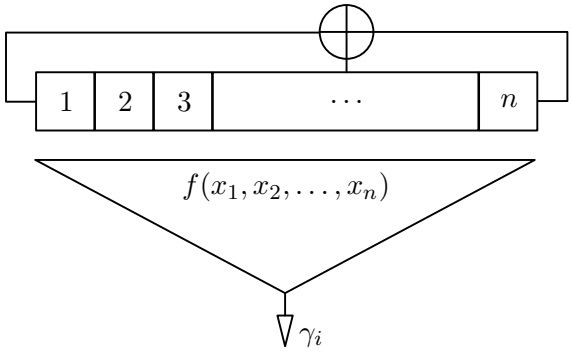
kur katram $s = (s_1, s_2, \ldots, s_n) \in X^n$

$$h(s, x) = (s_2, s_3, \ldots, s_n, x). \tag{57}$$

Izmantojot f.s $f(x_1, x_2, \ldots, x_n)$, virkne (x_i) tiek attēlota par virkni

$$(f(x_i, x_{1+i}, \ldots, x_{n+i})), \quad i = 1, 2, \ldots$$

Pieņemsim, ka L ir galīgs lauks. Par *filtrējošo ģeneratoru* (f.ģ) pār lauku L sauksim autonomu automātu $A_{FG} = \langle L^n, L, h, f \rangle$, kur h — LNR pārveidojums pār lauku L ar garumu n. F.ģ izejas gammu var aplūkot kā f.s $f(x_1, x_2, \ldots, x_n)$ attēlojuma rezultātu, kas pielietots LRV ar kārtu n. F.ģ kriptoshēma parādīta 12. attēlā.



12. zīm.: Filtrējošais ģenerators

F.ģ gammēšanas vienādojumi ir šādi:

$$\gamma_i = f(h^{i-1}(x)), \quad i = 1, 2, \ldots, \tag{58}$$

kur x — sākotnējais LNR stāvoklis.

F.ģ sauc par *nelineāru*, ja tā izeju funkcija $f(x_1, x_2, \ldots, x_n)$ ir nelineāra. F.ģ atslēgas elementi var būt LNR sākumstāvoklis, tā raksturīgais polinoms, kā arī funkcija $f(x_1, x_2, \ldots, x_n)$.

F.ģ piemīt labākas īpašības, ja tiek izmantots LNR ar maksimālu periodu un sabalansēta funkcija $f(x_1, x_2, \ldots, x_n)$. Tas garantē, ka gammas periods ir vienāds ar $k^n - 1$, kur k — lauka P kārta.

Gammas nelineārās īpašības nodrošina nelineārās funkcijas

$$f(x_1, x_2, \ldots, x_n)$$

izvēle. Apzīmēsim funkcijas $f(x_1, x_2, \ldots, x_n)$ nelinearitātes pakāpi ar d. Tādā gadījumā f.ģ gammas lineārā sarežģītība $\Lambda(\gamma)$ nav lielāka par dažādu n

mainīgo ar rangu no 0 līdz d konjunkciju. Aprēķinot šo skaitli jāņem vērā, ka katrs mainīgais konjunkcijā ieiet kā reizinātājs ar pakāpi ne lielāku par r, kur $r = \min(d, k-1)$. Binārajā gadījumā ($k = 2$) $\Lambda(\gamma) \leq s(n,d)$, kur

$$s(n,d) = \sum_{i=0}^{d} \binom{n}{i}$$

Novērtējumu no apakšas iegūšana ir sarežģītāka un kriptoloģijai svarīgāka, kas prasa ņemt vērā dziļākas funkcijas $f(x_1, x_2, \ldots, x_n)$ īpašības. Piemēram, dažām Būla bent-funkcijām pierādīts, ka n, kas dalās ar 4,

$$\Lambda(\gamma) \geq \binom{n/2}{n/4} \cdot 2^{n/4}.$$

Ir izdevies izvest apakšējo novērtējumu LNR ar maksimālu periodu un noteiktu Būla filtrējošo funkciju gadījumā. Pieņemsim, ka homogēns polinoms $f^d(x_1, \ldots, x_n)$ ar pakāpi d, kas ir Žegalkina funkcijas $f(x_1, x_2, \ldots, x_n)$ polinoma daļa, ir uzrakstāms šādi:

$$f^d(x_1, \ldots, x_n) = \bigoplus_{j=1}^{N} \alpha_j \cdot x_j \cdot x_{j+r} \cdot \ldots \cdot x_{j+(d-1)r},$$

kur $(r, 2^n - 1) = 1$ un $N = \left[\frac{n}{(d-1)\cdot r}\right]$. Tādā gadījumā izpildās nevienādība

$$\Lambda(\gamma) \geq \binom{n}{d} - N + 1.$$

Līdz ar to, lai iegūtu f.ğ gammu ar augstu lineāro sarežģītību jāizmanto funkciju $f(x_1, x_2, \ldots, x_n)$ ar pietiekoši lielu nelinearitātes pakāpi.Var parādīt, ka filtrējošo ģeneratoru daļa, kas veidoti uz LNR ar maksimālu periodu bāzes, kuras garums ir n, kuru lineārā sarežģītība $\Lambda(\gamma)$ sasniedz maksimālo $s(n,d)$ vērtību, tiecas uz viens.

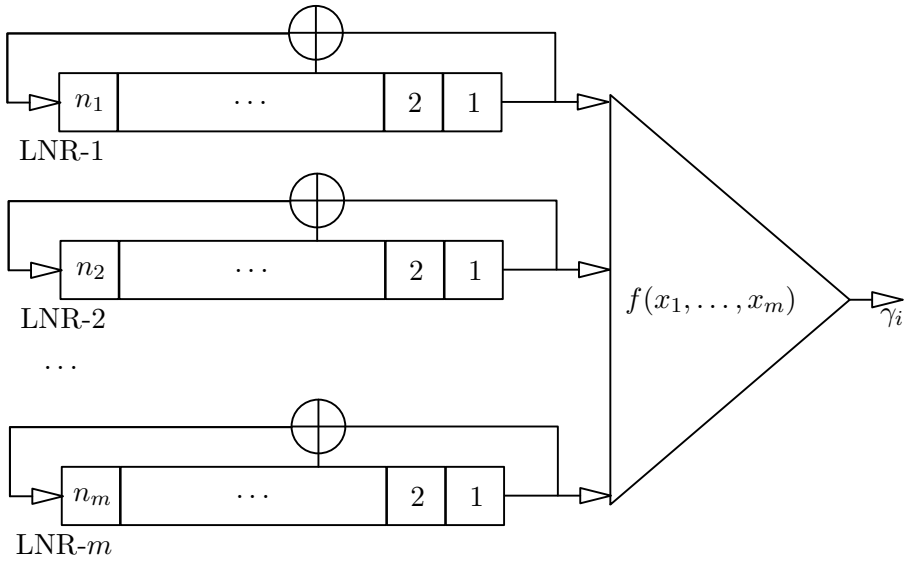
10.3. Kombinēts ģeneratori

Aplūkosim *kombinētu ģeneratoru* (komb.ğ) pār lauku L, kurš ir filtrējošā ģeneratora uzlabojums. Komb.ğ ir veidots par pamatu ņemot m LNR pār lauku L (apzīmēsim tos ar LNR-1,...,LNR-m) un funkciju $f(x_1, x_2, \ldots, x_m)$,

kuru sauc par *kombinējošo funkciju*, $m > 1$, kuras ieejā ienāk LNR simboli, kurus izstrādā lineārie reģistri. Komb.ğ attēlots 13. attēlā, kur ar n_j apzīmēts LNR-j garums, $j = 1, 2, \ldots, m$. Apzīmēsim $n = n_1 + \cdots + n_m$ un bez apraksta vispārīguma zaudēšanas pieņemsim, ka $n_1 \leq n_2 \leq \cdots \leq n_m$. Komb.ğ gammēšanas vienādojumi izskatās sekojoši:

$$\gamma_i = f(\nu(\varphi_1^{i-1}(x^{(1)})), \nu(\varphi_2^{i-1}(x^{(2)})), \ldots, \nu(\varphi_m^{i-1}(x^{(m)})), \quad i = 1, 2, \ldots, \quad (59)$$

kur $x^{(j)}$ — sākotnējais LNR-j stāvoklis, φ_j — LNR-j stāvokļu kopas pārveidojumi, $j = 1, 2, \ldots, m$, un katram vektoram $x = (x_1, \ldots, x_s) \in L^s$ definēsim $\nu(x) = x_1$. Vienādības (59) labās puses funkcija (apzīmēsim to ar $\Psi_i(x^{(1)}, x^{(2)}, \ldots, x^{(m)})$) ir dotā komb.ğ i-tā izejas funkcija, $i = 1, 2, \ldots$



13. zīm.: Kombinēts ğenerators

Komb.ğ sauc par *nelineāru*, ja funkcija $f(x_1, x_2, \ldots, x_m)$ ir nelineāra. Komb.ğ atslēgas elementi var būt visu LNR sākumstāvokļi, to raksturīgie polinomi, kā arī funkcija $f(x_1, x_2, \ldots, x_m)$. Visu LNR sākumstāvokļi veido ğeneratora sākumstāvokli. Ja visu LNR sākumstāvokļi ir atšķirīgi no nulles stāvokļa, tad atbilstošo ğeneratora sākumstāvokli sauksim par *nesingulāru*.

Novērtēsim ğeneratora gammas lineāro sarežģītību. Katram kanoniskajam polinomam $f(x_1, x_2, \ldots, x_m)$ pār galīgu lauku L izvēlēsimies atbilstošu

polinomu $f_z(x_1, x_2, \ldots, x_m)$ pār veselo skaitļu gredzenu $\mathbb{Z}$, kas iegūts no $f(x_1, x_2, \ldots, x_m)$ aizstājot visus nenulles koeficientus ar 1 un lauka L operācijas ar gredzena $\mathbb{Z}$ operācijām.

Teorēma 10.1. *Ja $f(x_1, x_2, \ldots, x_m)$ ir kanonisks kombinējošās funkcijas polinoms, tad visiem komb.ğ sākumstāvokļiem gammas lineārā sarežģītība $\Lambda(\gamma)$ nepārsniedz $f_z(n_1, n_2, \ldots, n_m)$.*

□ $\Lambda(\gamma)$ ir telpas F, kuru veido visu komb.ğ izejas funkciju kopa, dimensija. Novērtēsim telpas F bāzes, ko veido mainīgo konjunkcija (tā kā ne visām funkcijām ir polinomiāls attēlojums), apjomu.

No (59) seko, ka i-tā izejas funkcijas izskatās sekojoši:

$$\Psi_i(x^{(1)}, x^{(2)}, \ldots, x^{(m)}) = f(l_1^{(i)}(x^{(1)}), l_2^{(i)}(x^{(2)}), \ldots, l_m^{(i)}(x^{(m)})), \qquad (60)$$

kur $l_1^{(i)}(x^{(1)}), l_2^{(i)}(x^{(2)}), \ldots, l_m^{(i)}(x^{(m)})$ ir dažādu neatkarīgu mainīgo lineāras funkcijas, proti, ja $i \neq j$ un

$$\begin{aligned} x^{(i)} &= (x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}), \\ x^{(j)} &= (x_1^{(j)}, x_2^{(j)}, \ldots, x_{n_j}^{(j)}), \end{aligned}$$

tad

$$\{x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}\} \cap \{x_1^{(j)}, x_2^{(j)}, \ldots, x_{n_j}^{(j)}\} = \emptyset.$$

Līdz ar to, pēc tam, kad i-tās izejas funkcijas polinoms ir novests līdz kanoniskajam veidam, iegūstam ne vairāk kā $f_z(n_1, n_2, \ldots, n_m)$ konjunkciju summu. No tā seko, ka dažādu konjunkciju skaits, kuru lineārā kombinācija sakrīt ar i-to izejas funkciju nepārsniedz $f_z(n_1, n_2, \ldots, n_m)$.

Atzīmēsim, ka spriedums nav atkarīgs no sākotnējā komb.ğ stāvokļa un skaitļa i. ■

Lai nodrošinātu labākās komb.ğ kriptogrāfiskās īpašības, kombinējošai funkcijai $f(x_1, x_2, \ldots, x_m)$ jābūt nozīmīgi atkarīgai no m mainīgajiem un jāizmanto nesingulārus sākuma stāvokļus, jo pretējā gadījumā dotais komb.ğ tiek vienkāršots par komb.ğ ar mazāku LNR skaitu. Uzskatīsim, ka turpmāk šie nosacījumi vienmēr ir izpildīti.

Teorēmā 10.1 iegūto $\Lambda(\gamma)$ novērtējumu pie dažiem nosacījumiem var aizvietot ar vienādību

$$\Lambda(\gamma) = f_z(n_1, n_2, \ldots, n_m).$$

Piemēram, šī vienādība izpildās, ja L — vienkāršs lauks, bet LNR raksturīgie polinomi ir primitīvi un to pakāpes ir savstarpēji pirmskaitļi.

Dažiem LNR un funkcijām $f(x_1, x_2, \ldots, x_m)$ iespējams nodrošināt lielu komb.ğ gammas periodu un labas statistiskās īpašības. Var parādīt, ja LNR-1, LNR-2, ..., LNR-m pie kaut kāda sākumstāvokļa ğenerē virknes ar periodiem $t_1, t_2, \ldots, t_m$, bet funkcija $f(x_1, x_2, \ldots, x_m)$ ir bijektīva pēc visiem mainīgajiem, tad gammas periods $T(\gamma)$ nav mazāks par

$$\mathrm{MKD}(t_1, t_2, \ldots, t_m)/\mathrm{LKD}(t_1, t_2, \ldots, t_m).$$

Aplūkosim tuvāk gadījumu, kad $L = GF(2)$.

Teorēma 10.2. *Pieņemsim, ka visi komb.ğ LNR ir ar maksimālu periodu, un to garumi ir savstarpēji pirmskaitļi. Tad gammas periods $T(\gamma)$ ir:*

$$T(\gamma) = \prod_{j=1}^{m} (2^{n_j} - 1), \tag{61}$$

bet vieninieku skaits N_1 periodā apmierina nevienādības

$$\|f\| \cdot 2^{n-m} \cdot (1 - 2^{1-n_1}) < N_1 \leq \|f\| \cdot 2^{n-m}. \tag{62}$$

□ Saskaņā ar teorēmas nosacījumiem, LNR garumi $n_1, n_2, \ldots, n_m$ ir savstarpēji pirmskaitļi, tāpēc nesingulāram ğeneratora sākumstāvoklim

$$(x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

LNR izejas virkņu periodi ir attiecīgi $2^{n_1} - 1, 2^{n_2} - 1, \ldots, 2^{n_m} - 1$ un arī ir savstarpēji pirmskaitļi. Bet līdz ar to m–dimensionālo bināro vektoru kopas virknes

$$\left((l_1^{(i)}(x^{(1)}), l_2^{(i)}(x^{(2)}), \ldots, l_m^{(i)}(x^{(m)})) \right), \quad i = 1, 2, \ldots$$

kas ir kombinējošās funkcijas $f(x_1, x_2, \ldots, x_m)$ ieeja, periods ir vienāds ar t_m:

$$t_m = \prod_{j=1}^{m} (2^{n_j} - 1).$$

Tā kā gammas periods $T(\gamma)$ dala t_m, tad

$$T(\gamma) = d_1 \cdot d_2 \cdot \ldots \cdot d_m,$$

kur d_j dalās bez atlikuma ar $2^{n_j} - 1$, $j = 1, 2, \ldots, m$.

Pierādīsim, ka visiem $j = 1, 2, \ldots, m$ $d_j = 2^{n_j} - 1$. Pieņemsim pretējo, konkrēti, ka $d_m < 2^{n_m} - 1$. Pie tam $d_m > 1$, jo pretējā gadījumā $T(\gamma) = d_1 \cdot \ldots \cdot d_{m-1}$ un dala bez atlikuma t_{m-1}, kas nozīmētu, ka $f(x_1, x_2, \ldots, x_m)$ ir nebūtiski atkarīgs no x_m.

Aplūkosim virkni $(\gamma_{\tau+i\cdot r})$ — regulāru iztvērumu ar soli r no ğeneratora gammas, sākot ar τ-to simbolu un tai atbilstošo izejas funkciju virkni

$$(\Psi_{\tau+i\cdot r}(x^{(1)}, x^{(2)}, \ldots, x^{(m)})), \quad i = 1, 2, \ldots$$

kur $r = t_{m-1}$ un $\tau \in \{1, 2, \ldots, r\}$.

Tā kā skaitlis $r \cdot d_m$ dalās ar $T(\gamma)$, tad $r \cdot d_m$ ir arī gammas periods (iespējams, ka ne mazākais). No tā seko, ka periodisks virknes $(\gamma_{\tau+i\cdot r})$ nogrieznis, kura garums ir $2^{n_m} - 1$, satur q_1 vieninieku un q_0 nuļļu, kur q_1 un q_0 ir lielāki par nulli (jo pretējā gadījumā $f(x_1, x_2, \ldots, x_m)$ būtu fiktīvi atkarīga no x_m) un dalās ar h_m, kur $h_m = (2^{n_m} - 1)/d_m$. Atzīmēsim, ka $h_m > 1$.

No otras puses, tā kā r dalās ar LNR-1,..., LNR-m periodiem, tad no (60) izriet, ka $(\, \Psi_{\tau+i\cdot r}(x^{(1)}, x^{(2)}, \ldots, x^{(m)})\,)$ ir izskatā

$$(\alpha_1, \alpha_2, \ldots, \alpha_{m-1}, l_m^{\tau+i\cdot r}(x^{(m)})), \quad i = 1, 2, \ldots$$

kur $f(\alpha_1, \ldots, \alpha_{m-1}, x_m)$ ir funkcijas $f(x_1, x_2, \ldots, x_m)$ apakšfunkcija un mainīgo $(x_1, x_2, \ldots, x_{m-1})$ korteža fiksāciju $(\alpha_1, \alpha_2, \ldots, \alpha_{m-1})$ viennozīmīgi nosaka skaitlis τ. Apakšfunkcija $f(\alpha_1, \ldots, \alpha_{m-1}, x_m)$ ir viena mainīgā funkcija un tāpēc tā ir afīna, pie tam atradīsies tāds $\tau \in \{1, 2, \ldots, r\}$ un atbilstošais kortežs $(\alpha_1, \alpha_2, \ldots, \alpha_{m-1})$ tāds, ka funkcija $f(\alpha_1, \ldots, \alpha_{m-1}, x_m)$ būs atšķirīga no konstantes, citādi $f(\alpha_1, \ldots, \alpha_{m-1}, x_m)$ būs nenozīmīgi atkarīga no x_m.

Aplūkosim šādas korteža $(\alpha_1, \alpha_2, \ldots, \alpha_{m-1})$ gadījumā virkni

$$(\, f(\alpha_1, \ldots, \alpha_{m-1}, l_m^{\tau+i\cdot r}(x^{(m)}))\,), \quad i = 1, 2, \ldots, 2^{n_m} - 1.$$

Ņemot vērā, ka skaitļi r un $2^{n_m} - 1$ ir savstarpēji pirmskaitļi, tad virknes $(\, l_m^{\tau+i\cdot r}(x^{(m)})\,)$ periodiskais nogrieznis pie nenulles sākuma stāvokļa $x^{(m)}$ ir periodiska LRV nogriežņa, kuru izstrādā LNR-m, pārveidojums. No tā seko, ka ja dotajam kortežam $(\alpha_1, \alpha_2, \ldots, \alpha_{m-1})$ virkne $(\, f(\alpha_1, \ldots, \alpha_{m-1}, l_m^{\tau+i\cdot r}(x^{(m)}))\,)$ satur vai nu 2^{n_m-1} vieninieku un $2^{n_m-1} - 1$ nuļļu, vai otrādi. Katrā gadījumā abi skaitļi vienlaicīgi nedalās ar h_m. Esam ieguvuši pretrunu. Līdz ar to vienādība (61) ir pierādīta.

Pieņemsim, ka vektora $(\alpha_1, \alpha_2, \ldots, \alpha_{m-1})$ parādīšanās biežums virknē

$$(\, (l_1^{(i)}(x^{(1)}), l_2^{(i)}(x^{(2)}), \ldots, l_m^{(i)}(x^{(m)}))\,), \quad i = 1, 2, \ldots, t_m$$

ir $N(\alpha_1, \alpha_2, \ldots, \alpha_m)$. Ņemot vērā, ka visi LNR periodi ir savstarpēji pirmskaitļi, kā arī nulļu un vieninieku skaitu to pārejās, iegūstam:

$$N(\alpha_1, \alpha_2, \ldots, \alpha_\tau) = \prod_{j=1}^{m} (2^{n_j - 1} + \alpha_j - 1).$$

Tāpēc visiem $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ izpildās nevienādības:

$$2^{n-m}(1 - 2^{-n_1} - 2^{-n_2} - \cdots - 2^{-n_m}) < N(\alpha_1, \alpha_2, \ldots, \alpha_\tau) \leq 2^{n-m}, \qquad (63)$$

un vienādība labajā pusē tiek sasniegta tikai kad $(\alpha_1, \alpha_2, \ldots, \alpha_m) = (1, \ldots, 1)$. Tā kā LNR garumi ir atšķirīgi un sakārtoti, tad

$$2^{-n_1} + 2^{-n_2} + \cdots + 2^{-n_m} < 2^{1-n_1}.$$

Līdz ar to novērtējumi (63) ir uzrakstāmi kā:

$$2^{n-m}(1 - 2^{1-n_1}) < N(\alpha_1, \alpha_2, \ldots, \alpha_\tau) \leq 2^{n-m}. \qquad (64)$$

Vieninieku skaits N_1 gammas periodā ir vienāds ar:

$$N_1 = \sum_{(\alpha_1, \ldots, \alpha_m) \in \{0,1\}^m : f(x_1, x_2, \ldots, x_m) = 1} N(\alpha_1, \alpha_2, \ldots, \alpha_m).$$

No šī un nevienādībām (64) iegūstam (62). ■

Sekas 10.3. *Ja $f(x_1, x_2, \ldots, x_m)$ ir vienādi varbūtiskas, tad*

$$2^{n-1} - 2^{n-n_1} < N_1 < 2^{n-1}.$$

□ Atzīmēsim, ka (62) labā nevienādība izpildās tikai priekš

$$f(x_1, x_2, \ldots, x_m) = x_1 \cdot x_2 \cdot \ldots \cdot x_m.$$

Aizvietojot (62) $\|f\| = 2^{m-1}$ vērtību un labo nevienādību ar stingru nevienādību, iegūstam nepieciešamos novērtējumus. ■

10.4. Korelācijas uzbrukumi

Neskatoties uz atzīmētajām filtrējošo un kombinēto ģeneratoru spēcīgajām īpašībām, pastāv risks, kas saistīts ar korelāciju starp ģeneratora gammu un dažu LNR izejas virknēm. Šādā gadījumā ģeneratora atslēgas noteikšanu var realizēt pa posmiem (izmantojot metodi "skaldi un uzlauz").

Pirmajā posmā, izmantojot ģeneratora gammu, statistiski var noteikt dažu LNR sākuma stāvokļus, bet tālākajos posmos atjaunot pārējos atslēgas elementus, ņemot vērā, ka tagad jau jādarbojas ar vienkāršotu kriptosistēmu. Kā piemēru aplūkosim *Geffes ģeneratoru*, kas izmanto trīs LNR kombināciju. Šajā ģeneratorā LNR-1 un LNR-2 ir ģenerējoši, bet LNR-3 ir vadības ģenerators. Kombinējošā funkcija $f(x_1, x_2, x_3)$ ir uzrakstāma sekojoši:

$$f(x_1, x_2, x_3) = x_1 \cdot x_3 \oplus x_2 \cdot (x_3 \oplus x_1).$$

Ja visiem LNR ir maksimāli periodi un to garumi ir savstarpēji pirmskaitļi n_1, n_2, n_3, tad gammas periods ir vienāds ar LNR periodu reizinājumu, bet ģeneratora gammas lineārā sarežģītība ir vienāda ar $n_1 \cdot n_3 + n_2 \cdot n_3 + n_2$.

Tajā pašā laikā funkcijai $f(x_1, x_2, x_3)$ ir atrodami labi lineāri tuvinājumi, proti, $f(x_1, x_2, x_3)$ sakrīt ar funkciju x_1 (kā arī ar funkciju x_2) 3/4 gadījumu no visiem tabulas kortežiem. Tas nozīmē, ka ģeneratora gamma sakrīt ar LNR-1 izeju apmērām 75% simbolu. Līdz ar to var pārbaudīt LNR-1 sākotnējos stāvokļus un statistiski atmest "nepareizās" vērtības, ģenerējot LNR-1 izejas simbolus un novērojot biežumus ar kādiem šie simboli sakrīt ar atbilstošajiem ģeneratora gammas simboliem. Ar nepareizām vērtībām sakritīs apmēram 50% simbolu.

Aprēķināts, lai atmestu vienu "nepareizu vērtību" pietiek izmantot apmēram 15 simbolu salīdzināšanas. Tālākās atslēgas uzlaušanas darbietilpība ir nenozīmīga.

Cits pazīstams kombinētā ģeneratora piemērs ir *sliekšņa ģenerators*, kas izmanto nepāra skaitu m LNR. Kombinējošā funkcija $f(x_1, x_2, \ldots, x_m)$, kuru sauc par *mažorēšanas funkciju*, pieņem vērtību 1 tad un tikai tad, kad $\|(x_1, x_2, \ldots, x_m)\| > m/2$. Pie attiecīgiem LNR sliekšņa ģeneratora gammas periods ir LNR periodu reizinājums, bet gammas ģeneratora lineārā sarežģītība ir $f_z(n_1, n_2, \ldots, n_m)$ (kad $m = 3$, lineārā sarežģītība ir $n_1 \cdot n_2 + n_1 \cdot n_3 + n_2 \cdot n_3$).

Šim ģeneratoram arī piemīt vājās vietas. Mažorēšanas funkcijai atrodami labi afīnie tuvinājumi, piemēram, ja $m = 3$, tad katrs mainīgais sakrīt ar $f(x_1, x_2, x_3)$ 0,689 daļā no visām tabulas kortežiem. Korelāciju uzbrukums,

kas izmanto ar kārtu 30 gammas simbolu, atļauj secīgi atsijāt nepareizas visu LNR simbolu vērtības. Palielinot skaitli m, kas tiek izmantots LNR shēmā, korelāciju uzbrukuma sarežģītība palielinās.

Vislabākais pretlīdzeklis pret korelācijas uzbrukumiem ir kombinācijas funkciju ar augstu korelācijas imunitāti izmantošana. Tajā pašā laikā ir parādīts, ka eksistē saistība starp kombinējošās funkcijas korelācijas imunitāti un gammas lineāro sarežģītību, ko nosaka viena raksturlieluma vājināšanās pastiprinoties otram.

10.5. Korteža ģenerēšana ar nevienmērīgu soli

Ja informācijas plūsma kādā reģistrā ir atkarīga no cita reģistra ģenerētās virknes, tad sagaidāms, ka ģenerētais vārds varētu būt sarežģītāks. Šāda tipa ģenerātorus sauc par *nevienmērīgiem ģeneratoriem.*

Nevienmērīgumu var panākt virknē saslēdzot mašīnas. Te katru iepriekšējo mašīnu sauc par *vadības iekārtu* attiecībā pret tai sekojošo. Pēdējo mašīnu sauc par *ģenerējošo.* Ja $m > 2$, tad virknes slēgumu sauc par kaskādi, bet m — kaskāžu skaitu.

Nevienmērīgumu var panākt arī, ja lieto vairākas savstarpēji saistītas vadības mašīnas.

Kā piemēru apskatīsim, tā saukto $\delta - \tau$ soļu ģeneratoru. Šis ģenerators ir divu LNR virknes slēgums. Ja LNR–1 i–tajā taktī ir stāvoklī

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{in}),$$

bet LNR–2 atbilstoši — stāvoklī

$$y_i = (y_{im}, \ldots, y_{i2}, y_{i1}),$$

tad $\gamma_i = y_{\sigma(i)1}$, kur

$$\sigma(i) = \sum_{j=1}^{i} [\delta(f(x_j) \oplus 1) + \tau f(x_j)].$$

10.6. Ģeneratori ar atmiņu

Lai apgrūtinātu korelācijas uzbrukumu, izmanto ģeneratorus ar papildus atmiņu. Kā piemēru apskatīsim Maklarena — Marsaljī ģeneratoru.

Pieņemsim, ka mūsu rīcībā ir skaitā m atmiņas šūnu, kas sanumurētas ar skaitļiem no 0 līdz $m-1$. Šajās šūnās var ierakstīt kopas X elementus; kopas X apjoms $|X|$ ir fiksēts skaitlis n.

Definēsim Mīlija mašīnu

$$\mathfrak{M} \rightleftharpoons \langle X^m, X \times \mathbb{Z}_m^2, X, \circ, * \rangle .$$

Pieņemsim, ka

$$q = (s_0, s_1, \ldots, s_{m-1}) \in X^m$$

un

$$a = (x, u, v) \in X \times \mathbb{Z}_m^2,$$

tad

$$q \circ a \rightleftharpoons (s'_0, s'_1, \ldots, s'_{m-1}),$$

kur

$$s'_i \rightleftharpoons \begin{cases} x, & \text{ja } i = u; \\ s_i, & \text{ja } i \neq u. \end{cases}$$

Savukārt $q * a \rightleftharpoons s_v$.

Tā rezultātā, ja fiksēts sākuma stāvoklis $q \in X^m$ un dotas virknes (u_i), (v_i), tad, laižot virkni (x_i) caur mašīnu $\mathfrak{M}$, tās izejā iegūst virknes (x_i) sarežģījumu (y_i). Vispārīgā gadījumā mēs nonākam pie problēmas:

— Cik efektīvi Mīlija mašīna spēj sarežģīt virkni (x_i)?

Vienkāršākajā gadījumā pieņem, ka (x_i), (u_i) un (v_i) ir periodiskas virknes.

Bibliogrāfija

[1] Douglas R. Stinson. (1995) *Cryptography: theory and practice.* CRC Press, — 434 p.

[2] В. М. Фомичев. (2003) *Дискретная математика и криптология. Курс лекций.* Под общей редакцией доктора физико-математических наук Н. Д. Подуфалова. — Москва «ДИАЛОГ-МИФИ», — 400 с.