Saugos metodų taikymas programavime (3 lab. darbas)

1 Darbo tikslas

Laboratorinio darbo tikslas – susipažinti su pagrindinių kriptografijos metodų praktiniu naudojimu programose. Praktiškai išbandyti simetrinio ir asimetrinio šifravimo bei maišos funkcijų ir pranešimų autentifikavimo metodų panaudojimą įvairioms duomenų konfidencialumo ir vientisumo problemoms spręsti. Išsiaiškinti kaip priklauso metodų veikimas ir gaunami rezultatai keičiant pradinius metodų parametrus ir nežymiai modifikuojant duomenis.

2 Naudojamos priemonės

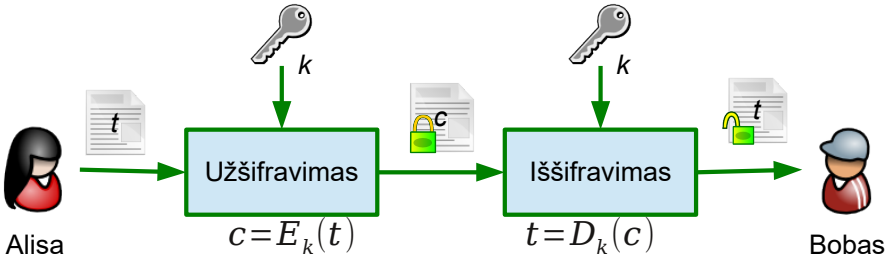
1. Integruota programavimo aplinka NetBeans;
2. Standartinės Java kriptografijos bibliotekos ir Java kriptografijos architektūra (JCA, JCE);
3. Bouncy Castle JCA kriptografijos biblioteka.

3 Teorinė dalis

Šiame laboratoriniame darbe praktiškai naudosite simetrinio ir asimetrinio šifravimo metodus, maišos funkcijas ir pranešimų autentifikavimo metodus. Trumpas įvadas į visus šiuos kriptografinius metodus pateiktas šiame skyrelyje. Plačiau apie naudojamus metodus galite sužinoti aprašymo pabaigoje pateiktoje literatūroje, per studijų modulio paskaitas ir oficialiuose naudojamų metodų ir standartų aprašymuose.

3.1 Simetriniai blokiniai šifrai

Simetrinėse kriptosistemose duomenų užšifravimui ir iššifravimui naudojamas tas pats raktas, kurį turi tiek siuntėjas, tiek gavėjas. Šis raktas yra vadinamas slaptuoju raktu. Bendra simetrinio šifro schema pateikta 3.1 pav. Simetrinio šifravimo metodai dažnai realizuojami taip, kad dirbtų su tam tikro fiksuoto ilgio pradinių duomenų blokais, tokie šifravimo metodai vadinami blokiniai šifrais. Java kriptografijos bibliotekų palaikomų blokinių simetrinių šifrų sąrašas, jų naudojamų raktų ilgiai ir apdorojamo bloko dydžiai pateikti 1 lent.

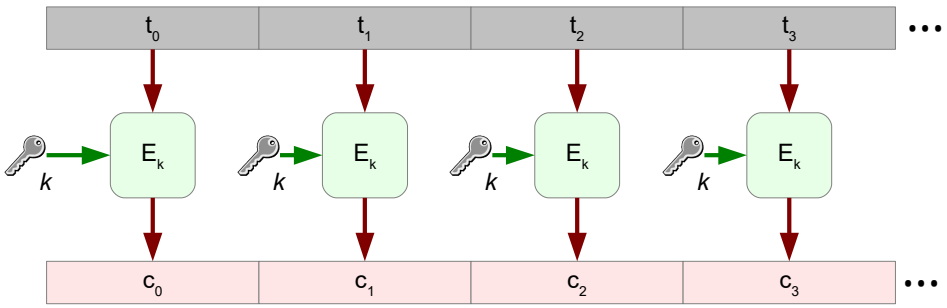


3.1 pav. Simetrinio šifro schema

Realiose situacijose dažniausiai tenka apdoroti ne vieną mažą duomenų bloką, o didelius duomenų masyvus. Egzistuoja visa eilė metodų, kurie leidžia nuosekliai apdoroti didelį kiekį duomenų naudojant blokinius šifrus. Šie duomenų pateikimo blokiniam šifrui algoritmai vadinami režimais. Pats paprasčiausias blokinio šifro režimas vadinamas ECB (angl. Electronic Code Book), jo schema parodyta 3.2 pav. Šiuo atveju pradinė tekstograma suskaidoma į blokus ir kiekvienas blokas šifruojamas nepriklausomai. Paskutinis tekstogramos blokas papildomas iki pilno kriptosistemos bloko panaudojant specialų užpildą (populiariausių užpildų sąrašą galite rasti 3 lent.).

1 lent. Simetrinių blokinio šifravimo metodų sąrašas ir pagrindiniai parametrai (BC realizacija)

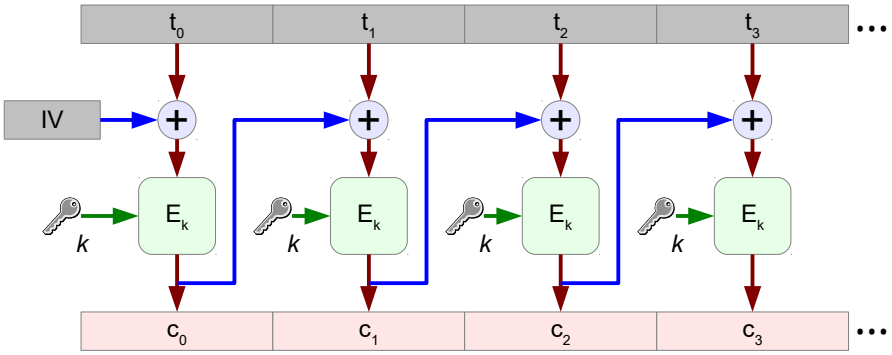
Pavadinimas	Rakto ilgis bitais	Bloko dydis
AES	128, 192, 256 (192)	128 bit
AESWrap	128, 192, 256 (192)	128 bit
Blowfish	0 ... 448 (448)	64 bit
Camellia	128, 192, 256	128 bit
CamelliaWrap	128, 192, 256	128 bit
CAST5	0 ... 128(128)	64 bit
CAST6	0 ... 256(256)	128 bit
DES	64	64 bit
DESede	128 ir 192	64 bit
GOST28147	256	64 bit
IDEA	128 (128)	64 bit
Noekeon	128(128)	128 bit
RC2	0 ... 1024 (128)	64 bit
RC5	0 ... 128 (128)	64 bit
RC5-64	0 ... 256 (256)	128 bit
RC6	0 ... 256 (128)	128 bit
Rijndael	0 ... 256 (192)	128 bit
SEED	128(128)	128 bit
SEEDWrap	128(128)	128 bit
Serpent	128, 192, 256 (256)	128 bit
SkipjackEngine	0 .. 128 (80)	64 bit
TEA	128 (128)	64 bit
Twofish	128, 192, 256 (256)	128 bit
XTEA	128 (128)	64 bit



3.2 pav. Blokinio šifro ECB režimas (užšifravimas)

Nors ECB režimas yra labai paprastai realizuojamas, jis turi visą eilę trūkumų. Tokiu režimu užšifruoti duomenys nėra atsparūs pakartojimo atakoms (angl. replay attack), kai viena kokia nors (labai svarbi) užšifruoto pranešimo dalis pakeičiama kita, kuri yra palankesnė piktavaliui. Naudojant šį režimą ir naudojant tą patį raktą, kiekvienas tekstogramos blokas atitiks tą patį šifrogramos bloką ir piktavalis, net nežinodamas rakto, gali sudaryti tokių atitikčių bazę ir jomis pasinaudoti. Be to, tokiu režimu užšifruotus duomenis kartais galima tiesiog „pamatyti“, kas atsitinka šifruojant paveikslėlius ar kitą vizualią informaciją.

Svarbiems duomenims šifruoti geriau tinka režimai, kurie užtikrina, jog pakeitus bent vieną bitą pranešimo pradžioje, pakinta ir visas pranešimas. Be to geras režimas turi užtikrinti, jog šifruojant tą patį bloką tuo pačiu raktu kelis kartus iš eilės gautume vis kitą šifrogramą. Tokiomis savybėmis pasižymi CBC blokinių šifrų režimas, kurio schema pateikta 3.3 pav. Šiuo atveju naudojamas atsitiktinis režimo inicializavimo vektorius IV, kurį būtina žinoti, norint pilnai iššifruoti pranešimą.



3.3 pav. Blokinio šifro CBC režimas (užšifravimas)

Vienas iš svarbiausių simetrinio šifravimo metodų trūkumų – nėra patikimo ir saugaus būdo apsikeisti slaptuoju raktu, kurį turi žinoti tiek gavėjas tiek siuntėjas. Ši problema dar labiau pasunkėja, kai duomenimis reikia keisti ne tarp dviejų, bet tarp daugiau suinteresuotų šalių

3.2 Maišos funkcijos

Simetriniai šifravimo metodai gerai tinka norint užtikrinti perduodamų pranešimų konfidencialumą. Kita ne ką mažiau aktuali problema – pranešimų vientisumas. Šiai problemai spręsti naudojamos maišos funkcijos. Maišos funkcija (angl. hash function) arba vienakryptė maišos funkcija (angl. one-way hash function), tai kriptografinė transformacija, kuris iš bet kokio ilgio pranešimo suformuoja fiksuoto ilgio duomenų bloką, kuris vadinamas santrauka (angl. digest) arba maišos rezultatu (angl. hash value). Maišos funkcijos parenkamos taip, kad dėl menkiausių duomenų pakitimų visiškai pakinta ir maišos rezultatas. Maišos funkcijos yra vienakryptės, todėl žinant maišos rezultatą neįmanoma atkurti pradinių duomenų. Iš kitos pusės kiekvienas pranešimas duoda tik jam vienam būdingą maišos rezultatą, o du skirtingi pranešimai visada duos skirtingus maišos rezultatus.

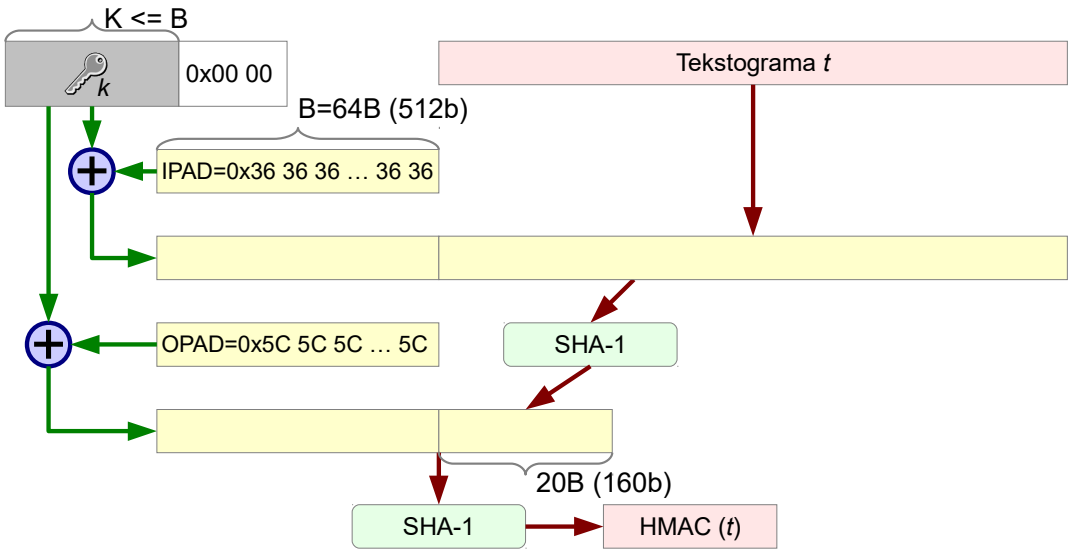
Norint pranešimo vientisumo tikrinimui panaudoti maišos funkciją kartu su pranešimu reikia perduoti ir jo maišos rezultatą. Gavėjas gali lengvai apskaičiuoti gauto pranešimo maišos rezultatą ir jį sulyginti su tuo, kuris atsiųstas kartu su pranešimu. Jei abu maišos rezultatai sutampa – pranešimas nesugadintas. Populiarių maišos funkcijų pavadinimai ir jų generuojamų maišos rezultatų ilgiai pateikti 5 lent.

3.3 Pranešimų autentifikavimo metodai

Maišos funkcijų naudojimas ne visada užtikrina pranešimo vientisumą. Jei piktavalis perimtų pranešimą, jis gali ne tik modifikuoti pranešimo turinį, bet ir iš naujo suskaičiuoti maišos rezultatą. Tokiu atveju gavėjas nežinos, kad pranešimas buvo piktavališkai pakeistas.

Norint išvengti tokios situacijos galima panaudoti pranešimų autentifikavimo kodus (angl. Message Authentification Code, MAC). Tai maišos funkcijos, kurios priklauso nuo slaptojo parametro. Naudojant MAC galima užtikrinti tiek duomenų vientisumą tiek autentiškumą (neišsiginamumą). Kadangi MAC sudaryme naudojamas ir slaptasis raktas, tai pranešimo vientisumą gali patikrinti tik gavėjas turintis tą patį slaptąjį raktą. Piktavalis, perėmęs ir modifikavęs pranešimą, negali apskaičiuoti naujos MAC reikšmės, nes nežino slaptojo rakto.

Kaip pranešimų autentifikavimo kodus galima panaudoti įprastus simetrinio šifravimo metodus (nes jie priklauso nuo slapto rakto). Kitas dažnai praktikoje naudojamas sprendimas – panaudoti santraukos funkcijas. Tokie metodai vadinami santraukos pranešimų autetifikavimo kodais (angl. Hash Message Authentification Code, HMAC). Tokio metodo apibendrinta schema pateikta 3.4 pav. HMAC realizuoti naudojamos populiariausios, kriptografiškai saugios maišos funkcijos: SHA-1, MD5, SHA-256. Atitinkami HMAC metodai vadinami: HMAC-SHA1, HMAC-MD5 ir HMAC-SHA256.



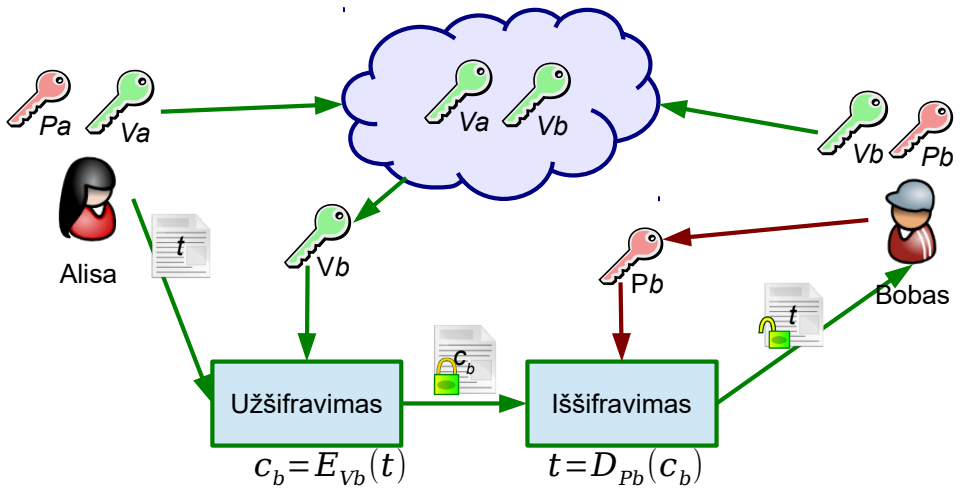
3.4 pav. Maišos funkcijas naudojančių pranešimų autentifikavimo metodų (HMAC) schema.

3.4 Asimetrinės kriptosistemos

Simetrinių kriptosistemų trūkumas tas, jog abi bendraujančios pusės turi žinoti ir naudoti tą patį slaptąjį raktą. Kaip perduoti šį raktą neapsaugotu Internetu taip, kad niekas jo negalėtų perimti? Šiam tikslui galima panaudoti asimetrinę kriptosistemą.

Apibendrinta asimetrinės kriptosistemos schema parodyta 3.5 pav. Asimetrinėje kriptografijoje kiekviena šalis naudoja du raktus – viešąjį ir privatųjį. Duomenis užšifruotus viešuoju raktu galima iššifruoti tik naudojant atitinkamą privatųjį raktą. Žinant tik viešąjį raktą duomenų iššifruoti neįmanoma, be to neįmanoma sužinoti ir atitinkamo privataus rakto.

Viešasis raktas gali būti dalinamas laisvai visoms bendraujančioms šalims arba publikuojamas viešame informaciniame šaltinyje, bet susijęs privatusis raktas turi likti privatus. Asimetrinės kriptosistemos nėra tokios efektyvios greitaveikos ir saugos požiūriu kaip simetrinės, todėl jos naudojamos tik labai mažų duomenų kiekių šifravimui. Dažniausiai, asimetrinė kriptografija naudojama susigeneruotai arba saugiau pasikeisti simetrinės kriptografijos slaptuoju raktu.



3.5 pav. Asimetrinės kriptosistemos schema

Šiuo metu dažniausiai naudojami asimetrinės kriptografijos metodai skirti duomenų šifravimui yra RSA ir ElGamalio.

3.4.1 RSA kriptosistema

RSA asimetrinio šifravimo metodą 1977 m. metais pasiūlė Ron Rivest, Adi Shamir ir Leonard Adleman, dabar tai viena iš populiariausių asimetrinių kriptosistemų. RSA sauga remiasi faktorizavimo uždavinio sudėtingumu. Įrodyta, jog 110 dešimtainių skaitmenų

skaičiams faktorizavimo uždavinį galima išspręsti naudojant šiuolaikines (2008 m. duomenimis) priemones. Pradedant nuo 250 skaitmenų faktorizavimo uždavinys yra sunkiai išsprendžiamas (naudojant tokio ilgio modulį RSA kriptosistemą galima laikyti saugia, 2012 m. įvertinimas). 1024 bitų dvejetainis skaičius turi daugiau nei 300 dešimtainių skaitmenų, todėl RSA metode rekomenduojama naudoti 1024 ir daugiau bitų modulį.

RSA raktų generavimas:

❶ Imkime du didelius, maždaug vienodo dydžio, pirminius skaičius p ir q. Rekomenduojama, jog jie būtų nemažiau kaip 512 bitų ilgio.

❷ Apskaičiuokime $n=pq$, n turi gautis apie 1024 bitų ilgio.

❸ Apskaičiuokime skaičių $m=(p-1)(q-1)$, m dar vadinamas Oilerio funkcijos reikšme.

❹ Imkime sveikąjį skaičių e, $1<e<m$. Skaičiai e ir m turi būti reliatyviai pirminiai, $DBD(e,m)=1$.

❺ Apskaičiuokime e atvirkštinį moduliu m, $e' \equiv d \pmod m$; $ed \equiv 1 \pmod m$.

RSA kriptosistemos viešasis raktas yra skaičių pora (e,n), o privatusis raktas - (d,n).

Tekstogramos užšifravimas ir iššifravimas:

❶ Tekstograma atvaizduojama į sveikąjį skaičių t, $0<t<n$. Norint nusiųsti šifrogramą, reikia iš anksto turėti gavėjo viešąjį raktą (e,n).

❷ Užšifravimas (atlieka siuntėjas): $c=t^e \pmod n$. Šifrograma c nusiunčiama gavėjui.

❸ Gavėjas turi atitinkamą privatųjį raktą (d,n), todėl gali iššifruoti tekstogramą: $t=c^d \pmod n$.

3.4.2 ElGamalio kriptosistema

Egipto kriptografas Taher ElGamal 1985 m. pasiūlė savo asimetrinę kriptosistemą, kuri dabar plačiai žinoma jo vardu. Šios kriptosistemos sauga remiasi diskrečiojo logaritmo uždavinio sudėtingumu. Norint, jog šis uždavinys būtų neišsprendžiamas, būtina naudoti modulį, kurio ilgis yra analogiškas RSA modulio ilgiui.

ElGamalio raktų generavimas (čia nepateikiamos papildomos sąlygos keliamos kai kuriems metodo parametrams):

❶ Imkime didelį pirminį skaičių p (pagal ilgį ekvivalentų RSA moduliui).

❷ Pasirinkime kitą specifinį skaičių g, multiplikacinės grupės $\mathbb{Z}_p$ generatorių.

❸ Pasirinkime atsitiktinį skaičių x, $(2 \le x \le p-2)$.

❹ Apskaičiuokime skaičių y, $y=g^x \pmod p$.

ElGamalio kriptosistemos viešasis raktas: y, privatusis raktas: x. Parametrai p ir g gali būti bendri visiems kriptosistemos vartotojams, jie vadinami sisteminiais parametrais.

Tekstogramos užšifravimas:

❶ Tekstograma atvaizduojama į sveikąjį skaičių t, $0<t<p$. Pasirenkamas atsitiktinis skaičius k, $(1<k<p-1)$, jis turi būti iš naujo generuojamas kiekvienam šifravimui. Siuntėjas turi iš anksto žinoti gavėjo viešąjį raktą y ir sisteminius parametrus p ir g.

❷ Apskaičiuojamos dvi reikšmės: $r \equiv g^k \pmod p$ ir $s \equiv t \cdot y^k \pmod p$.

❸ Šifrograma yra skaičių pora $c=(r,s)$, ji yra du kartus ilgesnė už tekstogramą.

Tekstogramos iššifravimas:

❶ Gavėjas žino savo privatųjį raktą x, iš siuntėjo gauna šifrogramą (skaičiaus k jis nežino): $c=(r,s)=(g^k \pmod p,\ t \cdot y^k \pmod p)$.

❷ Gavėjas apskaičiuoja: $r^x \equiv (g^k)^x \pmod p$. Be to, bendru atveju galioja tokie lyginiai: $r^x \equiv (g^k)^x \equiv (g^x)^k \equiv y^k \pmod p$.

❸ Pradinė tekstograma t randama taip: $s \cdot (r^x)' \equiv t \cdot y^k \cdot (y^k)' \pmod p \equiv t \pmod p$.

ElGamalio kriptosistemoje labai svarbu kiekvieną pranešimą šifruoti naudojant skirtingą atsitiktinį skaičių k, priešingu atveju metodas tampa neatsparus žinomos tekstogramos atakai. Dėl naujo atsitiktinio skaičiaus k generavimo, tą pačią tekstogramą atitinka kelios skirtingos šifrogramos.

4 Praktinė dalis

Atlikdami šį laboratorinį darbą susipažinsite su Java kriptografijos architektūra (JCA) ir išmoksite savo reikmėms pasinaudoti standartiniais duomenų šifravimo ir vientisumo patikrinimo metodais.

Toliau šiame skyrelyje yra pateikiamos užduotys, jas išsprendžiantys Java programos kodo fragmentai, šių fragmentų paaiškinimai bei papildoma informacija tiesiogiai susijusi su kiekvieno tipo metodais ir jų naudojimu.

Pastaba. Pateikti programų fragmentai, bei visos pagalbinės funkcijos yra patalpintos laboratorinių darbų kompiuteryje, direktorijoje `~/uzd/java/` ir modulio Moodle aplinkoje.

4.1 Simetrinių kriptosistemų naudojimas

Tarkime, yra duota 32 baitų ilgio šifrograma: `DFD1AD8FED3D091F 79D85F1A0E8F1F61 D98C1A5003FDEE0B 615FC394D8FE1C54`. Žinome, jog ši šifrograma gauta naudojant `serpent` simetrinio blokinio šifravimo metodą. Pradinė tekstograma buvo papildyta iki pilno bloko naudojant `TBCPadding` metodą, o šifravimas buvo vykdomas `CBC` režimu. Be to žinome slaptąjį raktą (`6665566666655666 3331133333311333`) ir inicializavimo vektorių (`0706050403020100 08090A0B0C0D0E0F`). Reikia parašyti programą, kuri iššifruotų šią šifrogramą ir atspausdintų tekstogramą. Tokios programos išeities tekstas pateiktas 4.1 pav., o jos veikimo rezultatas 4.2 pav.

```
1     public static void doDecryptSerpent() throws Exception
2     {
3         byte[]  input = new byte[] {
4                 (byte) 0xDF, (byte) 0xD1, (byte) 0xAD, (byte) 0x8F, (byte) 0xED, 0x3D, 0x09, 0x1F,
5                 (byte) 0x79, (byte) 0xD8, 0x5F, 0x1A, 0x0E, (byte) 0x8F, 0x1F, 0x61,
6                 (byte) 0xD9, (byte) 0x8C, 0x1A, 0x50, 0x03, (byte) 0xFD, (byte) 0xEE, 0x0B,
7                 0x61, 0x5F, (byte) 0xC3, (byte) 0x94, (byte) 0xD8, (byte) 0xFE, 0x1C, 0x54};
8         byte[]  keyBytes = new byte[] {
9                 0x66, 0x65, 0x56, 0x66, 0x66, 0x65, 0x56, 0x66,
10                0x33, 0x31, 0x13, 0x33, 0x33, 0x31, 0x13, 0x33};
11        byte[]   ivBytes = new byte[] {
12                0x07, 0x06, 0x05, 0x04, 0x03, 0x02, 0x01, 0x00,
13                0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f};
14
15        System.out.println("Duotoji sifrograma : " + toHex(input));
16        SecretKeySpec   key = new SecretKeySpec(keyBytes, 0, 16, "serpent");
17        // IV turi buti lygiai tiek baitu, koks yra bloko ilgis
18        IvParameterSpec ivSpec = new IvParameterSpec(ivBytes, 0, 16);
19        Cipher          cipher = Cipher.getInstance("serpent/CBC/TBCPadding", "BC");
20
21        cipher.init(Cipher.DECRYPT_MODE, key, ivSpec);
22        byte[] plainText = new byte[cipher.getOutputSize(input.length)];
23
24        int ptLength = cipher.update(input, 0, input.length, plainText, 0);
25        ptLength += cipher.doFinal(plainText, ptLength);
26        System.out.println("Serpent issifruota tekstograma : " + toHex(plainText, ptLength) + " bytes:
   " + ptLength);
27        byte[] raktas = key.getEncoded();
28        System.out.println("Naudotas raktas : " + toHex(raktas));
29        System.out.println("Naudotas IV : " + toHex(ivSpec.getIV()));
30
31        //Patikrinimas
32        cipher.init(Cipher.ENCRYPT_MODE, key, ivSpec);
33        byte[] cipherText = new byte[cipher.getOutputSize(ptLength)];
34
35        int ctLength = cipher.update(plainText, 0, ptLength, cipherText, 0);
36        ctLength += cipher.doFinal(cipherText, ctLength);
37
38        System.out.println("Vel užšifruotas pranešimas : " + toHex(cipherText, ctLength) + " bytes: "
   + ctLength);
39
40    }
```

4.1 pav. Tekstogramos iššifravimo naudojant serpent metodą programa

```
run:
Duotoji sifrograma : DFD1AD8FED3D091F 79D85F1A0E8F1F61 D98C1A5003FDEE0B 615FC394D8FE1C54
Serpent issifruota tekstograma : BABA000000000000 0102030400000000 BABA bytes: 18
Naudotas raktas : 6665566666655666 3331133333311333
Naudotas IV : 0706050403020100 08090A0B0C0D0E0F
Vėl užšifruotas pranešimas: DFD1AD8FED3D091F 79D85F1A0E8F1F61 D98C1A5003FDEE0B 615FC394D8FE1C54 bytes: 32
```

4.2 pav. Tekstogramos iššifravimo naudojant serpent metodą rezultatas

Pateiktame programos fragmente 3-13 eilutėse statiškai aprašomi pradiniai duomenys, sukuriami reikiami rakto, inicializavimo vektoriaus ir šifravimo klasių objektai (16-20 eil.) ir atliekamas duomenų iššifravimas (24-26 eil.). Likusi programos dalis atspausdina pradinius duomenis bei atlieka gautos tekstogramos užšifravimą su tais pačiais parametrais (31-37 eil.). Kaip ir galima buvo tikėtis, iš naujo užšifruotas pranešimas generuoja lygiai tokią pačią šifrogramą (4.2 pav.). Suprantama, jei pakeistume inicializavimo vektoriaus reikšmę arba raktą, tai gautume visai kitokius rezultatus. Patikrinkite tai darydami laboratorinį darbą.

Toliau, 2, 3 ir 4 lentelėse pateikti standartinių (Sun) ir Bouncy Castle (BC) simetrinio blokinio šifravimo metodų pavadinimai, veikimo režimai ir galimi bloko užpildai. Lentelių viršutinėje dalyje išvardinti standartiniai algoritmai kuriuos realizuoja tiek Sun tie BC bibliotekos. Apatinėje dalyje išvardinti algoritmai kuriuos papildomai palaiko BC biblioteka. Iš šių trijų lentelių parinkus po vieną (suderinamą) reikšmę gaunamas pilnas šifravimo metodo aprašas, kurį reikią nurodyti kuriant šifravimo objektą (19 eil. programos išeities tekste, 4.1 pav.), pavyzdžiui, `AES/CBC/NoPadding`, `DES/ECB/PKCS5Padding` arba `IDEA/CBC/ISO10126Padding`.

2 lent. Sun ir BC bibliotekų palaikomi blokinio šifravimo metodai

Algoritmas	**Aprašas**
AES	Advanced Encryption Standard, pagal NIST, FIPS 197. 128b. Blokas, raktai: 128, 192 ir 256 b.
AESWrap	The AES key wrapping algorithm pagal RFC 3394.
ARCFOUR	Srautinis šifras, suderinamas su RC4
Blowfish	Blowfish blokinis šifras, autorius Bruce Schneier.
DES	Digital Encryption Standard pagal FIPS PUB 46-3.
DESede	Triple DES (dar žinomas kaip DES-EDE, 3DES ar Triple-DES).
ECIES	Elliptic Curve Integrated Encryption Scheme
RC2	Kintamo rakto ilgo šifras sukurtas Rono Rivesto
RC4	Kintamo rakto ilgo srautinis šifras sukurtas Rono Rivesto
RC5	Kintamo rakto ilgo šifras sukurtas Rono Rivesto
BC	
Camellia	
CAST5	
CAST6	
GOST28147	
IDEA	
RC6	
SEED	
Serpent	
Skipjack	
TEA	
Twofish	

3 lent. Sun ir BC bibliotekų palaikomi blokinio šifravimo metodų režimai

Režimas	Aprašas
NONE	Jokio režimo
CBC	Cipher Block Chaining Mode, pagal FIPS PUB 81.
CFB, CFBx	Cipher Feedback Mode, pagal FIPS PUB 81. Leidžia apdoroti duomenis porcijomis mažesnėmis už bloką (Čia x - kiek bitų mažiausiai gali apdoroti, 8 – apdoroja po vieną baitą)
CTR	Supaprastintas OFB
CTS	Cipher Text Stealing
ECB	Electronic Codebook Mode, pagal FIPS PUB 81.
OFB, OFBx	Output Feedback Mode, as defined in FIPS PUB 81. Leidžia apdoroti duomenis porcijomis mažesnėmis už bloką
PCBC	Propagating Cipher Block Chaining, apibrėžtas Kerberos V4.
BC	
SIC	dar žinomas kaip CTR
OpenPGPCFB	Naudojamas Open PGP
CTS	dar žinomas kaip CBC/WithCTS
GOFB	
CCM	
EAX	

4 lent. Sun ir BC bibliotekų palaikomi blokinio šifravimo metodų užpildai

Algoritmas	Aprašas
NoPadding	nenaudoti užpildo
PKCS5Padding	Užpildo schema apibrėžta PKCS #5, naudojama su 64b blokiniais šifrais
SSL3Padding	SSL 3.0 naudojama užpildo schema
BC	
PKCS7Padding	Užpildo schema apibrėžta PKCS #7, naudojama su didesniais nei 64b blokiniais šifrais
ISO10126-2Padding	
ISO10126Padding	
ISO7816-4Padding	
X9.23Padding	
WithCTS	
TBCPadding	
ZeroBytePadding	Užpildoma nuliais iki pilno bloko

1 lentelėje pateikta bendra informacija apie blokinius simetrinius šifrus. Ji gali būti naudinga praktiškai parenkant reikiamą bloko dydį ar inicializavimo vektoriaus ilgį duotajam šifravimo metodui. Rakto ilgio reikšmė skliausteliuose reiškia, kad tokio ilgio raktas naudojamas, jei nenurodyta kitaip.

4.2 Maišos funkcijų naudojimas

Tarkime, yra duota tekstograma (11 baitų ilgio masyvas): `BAD0ACE000050607 08090A` ir jos maišos rezultatas `2E1ADC2ADAF89305 61CD1F955E17D214 D6564A8D33EF7819`. Žinome jog šis maišos rezultatas buvo gautas naudojant `tiger` maišos funkciją. Reikia parašyti programą, kuri patikrintų ar pradinis tekstas nebuvo pakeistas po to kai buvo apskaičiuota jo santrauka.

Tokios programos išeities tekstas pateiktas 4.3 pav., o jos veikimo rezultatas 4.4 pav.

```
1       public static void doTigerHashCheck() throws Exception
2       {
3           boolean ok = false;
4           byte[]  inputBytes = new byte[] {
5                   (byte) 0xBA, (byte) 0xD0, (byte) 0xAC, (byte) 0xE0, 0x00, 0x05, 0x06, 0x07,
6                   0x08, 0x09, 0x0A};
7           byte[]  hashBytes = new byte[] {
8                   0x2E, 0x1A, (byte) 0xDC, 0x2A, (byte) 0xDA, (byte) 0xF8, (byte) 0x93, 0x05,
9                   0x61, (byte) 0xCD, 0x1F, (byte) 0x95, 0x5E, 0x17, (byte) 0xD2, 0x14,
10                  (byte) 0xD6, 0x56, 0x4A, (byte) 0x8D, 0x33, (byte) 0xEF, 0x78, 0x19};
11
12          System.out.println("Tekstograma : " + toHex(inputBytes));
13          System.out.println("Tiger santrauka : " + toHex(hashBytes));
14
15          MessageDigest   hash = MessageDigest.getInstance("tiger", "BC");
16
17          hash.update(inputBytes, 0, inputBytes.length);
18          byte[] inputHash = new byte[hash.getDigestLength()];
19          inputHash = hash.digest();
20
21          System.out.println("Apskaiciuota santrauka : " + toHex(inputHash));
22
23          ok = MessageDigest.isEqual(inputHash, hashBytes);
24          System.out.println("Tekstograma nepakeista? : " + ok);
25      }
```

4.3 pav. Tekstogramos vientisumo tikrinimas naudojant tiger maišos funkciją

```
run:
Tekstograma : BAD0ACE000050607 08090A
Tiger santrauka : 2E1ADC2ADAF89305 61CD1F955E17D214 D6564A8D33EF7819
Apskaiciuota santrauka : 2E1ADC2ADAF89305 61CD1F955E17D214 D6564A8D33EF7819
Tekstograma nepakeista? : true
```

4.4 pav. Tekstogramos vientisumo tikrinimo naudojant tiger maišos funkciją rezultatas

Šiame programos fragmente 4-11 eilutėse statiškai aprašomi pradiniai duomenų masyvai, 12-13 eilutėse atspausdinami pradiniai duomenys naudojant pagalbines funkcijas `toHex`. 15 eilutėje sukuriamas reikiamas `MessageDigest` klasės objektas, kurio pagalba 17-20 eilutėse apskaičiuojamas pateiktų duomenų maišos rezultatas. 23 eilutėje, panaudojant specialų `MessageDigest` metodą `isEqual`, sulyginami du maišos rezultatai ir patikrinama ar pradinis pranešimas nebuvo pakeistas.

Laboratorinio darbo metu patikrinkite kaip keičiasi santrauka ir galutinis patikrinimo rezultatas nežymiai keičiant tekstogramą ar pačią santrauką

5 lentelėje pateikti standartinių (Sun) ir Bouncy Castle (BC) maišos algoritmų pavadinimai ir jų santraukos ilgiai. Darydami laboratorinį darbą atkreipkite dėmesį į šiuos parametrus. Jei užduotyje pateikta santrauka neatitinka reikalavimų, tai nebūtina rašyti programos, norint suprasti, kad santrauka neteisinga.

5 lent. Maišos algoritmai ir jų parametrai

Algoritmas	Santraukos ilgis bitais	Aprašymas
MD2	128	MD2 maišos algoritmas, pagal RFC 1319.
MD5	128	MD5maišos algoritmas pagal RFC 1321.
SHA-1	160	Maišos algoritmai pagal FIPS PUB 180-2.
SHA-256	256	
SHA-384	384	
SHA-512	512	
BC		
GOST3411	256	
MD4	128	

Algoritmas	Santraukos ilgis bitais	Aprašymas
RipeMD128	128	Bazinis RipeMD
RipeMD160	160	Pagerintas RipeMD
RipeMD256	256	Išplėsta RipeMD128 metodo versija
RipeMD320	320	Išplėsta RipeMD160 metodo versija
SHA-224	256	Pagal FIPS 180-2
Tiger	192	
Whirlpool	512	

4.3 Pranešimų autentifikavimo metodų naudojimas

Tarkime, yra duota tekstograma (13 baitų ilgio skaičių masyvas): `BABCE00000010203 0405060708` ir jos autentifikavimo funkcijos (HMAC) rezultatas `A8047F7323FAAF00 7ACC628FDEC63CB7 69733FB4`. Žinome, jog šis rezultatas buvo gautas naudojant `Hmac-SHA1` metodą, o naudotas slaptasis raktas yra `5172333435363738 393A3B3C`. Reikia parašyti programą, kuri patikrintų ar pradinis tekstas nebuvo pakeistas po to kai buvo apskaičiuota jo autentifikavimo funkcijos reikšmė.

Sudarytos programos išeities tekstas pateiktas 4.5 pav., o jos veikimo rezultatas 4.6 pav.

```
1  public static void doSHA1HMACCheck() throws Exception
2      {
3          boolean ok = false;
4          byte[]  inputBytes = new byte[] {
5                  (byte) 0xBA, (byte) 0xBC, (byte) 0xE0, 0x00, 0x00, 0x01, 0x02, 0x03,
6                  0x04, 0x05, 0x06, 0x07, 0x08};
7          byte[]  macKeyBytes = new byte[] { 0x51, 0x72, 0x33, 0x34, 0x35, 0x36, 0x37,
8                  0x38, 0x39, 0x3A, 0x3B, 0x3C};
9          byte[]  hmacBytes = new byte[] {
10                 (byte) 0xA8, 0x04, 0x7F, 0x73, 0x23, (byte) 0xFA, (byte) 0xAF, 0x00,
11                 0x7A, (byte) 0xCC, 0x62, (byte) 0x8F, (byte) 0xDE, (byte) 0xC6, 0x3C, (byte)
   0xB7,
12                 0x69, 0x73, 0x3F, (byte) 0xB4};
13
14         Mac hMac = Mac.getInstance("Hmac-SHA1", "BC");
15         Key hMacKey = new SecretKeySpec(macKeyBytes,"Hmac-SHA1");
16
17         System.out.println("Tekstograma : " + toHex(inputBytes));
18         System.out.println("Slaptas raktas : " + toHex(macKeyBytes));
19         System.out.println("Pateiktas hmac : " + toHex(hmacBytes));
20
21         hMac.init(hMacKey);
22         hMac.update(inputBytes, 0, inputBytes.length);
23
24         byte[] inputMac = new byte[hMac.getMacLength()];
25         inputMac = hMac.doFinal();
26
27         System.out.println("Apskaiciuotas hmac : " + toHex(inputMac) + " ilgis " +
   hMac.getMacLength());
28
29         ok = MessageDigest.isEqual(inputMac, hmacBytes);
30         System.out.println("Pranesimas nesuklastotas : " + ok);
31     }
```

4.5 pav. Tekstogramos vientisumo tikrinimas naudojant Hmac-SHA1 funkciją

```
run:
Tekstograma : BABCE00000010203 0405060708
Slaptas raktas : 5172333435363738 393A3B3C
Pateiktas hmac : A8047F7323FAAF00 7ACC628FDEC63CB7 69733FB4
Apskaiciuotas hmac : A8047F7323FAAF00 7ACC628FDEC63CB7 69733FB4 ilgis 20
Pranesimas nesuklastotas : true
```

4.6 pav. Tekstogramos vientisumo tikrinimo naudojant Hmac-SHA1 funkciją rezultatai

Pateiktame programos fragmente 4-12 eilutėse statiškai aprašomi pradiniai duomenų masyvai, 14-15 eilutėse sukuriami MAC tikrinimui būtini `Mac` ir `Key` klasių objektai. 17-20

eilutėse, naudojant pagalbines funkcijas `toHex`, atspausdinami pradiniai duomenys. 21-25 eilutėse apskaičiuojamas pateiktų duomenų MAC funkcijos rezultatas. 29 eilutėje, panaudojant specialų `MessageDigest` metodą `isEqual`, sulyginami du MAC rezultatai ir patikrinama ar pradinis pranešimas nebuvo pakeistas.

Atlikdami šią užduotį patikrinkite kaip keičiasi MAC reikšmė ir galutinis patikrinimo rezultatas nežymiai keičiant tekstogramą ir naudojamą raktą. Patikrinkite ar galima naudoti raktą kuris yra trumpesnis ar ilgesnis už pateiktą užduotyje.

6 lentelėje pateikti standartinių (Sun) ir Bouncy Castle (BC) pranešimų autentifikavimo metodų pavadinimai ir jų generuojamų santraukų ilgiai. Darydami laboratorinį darbą atkreipkite dėmesį į šiuos parametrus. Jei jūsų užduotyje pateikta santrauka neatitinka ilgio reikalavimų, tai galima teigti, kad pranešimas tikrai buvo suklastotas, nes patikrinti jo autentiškumo neįmanoma.

6 lent. Maišos algoritmai ir jų parametrai

Algoritmas	Santraukos ilgis bitais	Komentaras
HmacMD5	128	HMAC-MD5 pagal RFC 2104
HmacSHA1	160	HmacSHA* algoritmai pagal RFC 2104
HmacSHA256	256	
HmacSHA384	384	
HmacSHA512	512	
PBEWith<mac>		Slaptažodžius naudojantys MAC pagal PKCS #5 v 2.0. pvz., PBEWithHmacSHA1.
BC		
VMPC-MAC	128	
HMac-MD2	128	
HMac-MD4	128	
HMac-RipeMD128	128	
HMac-RipeMD160	160	
HMac-SHA224	224	
HMac-Tiger	192	

4.4 Asimetrinių kriptosistemų naudojimas

Tarkime, kad yra gauta 64 baitų ilgio šifrograma: `4C22AD8CBDA81732 96B746A2EF4ED714 1B206004A3627F68 B9CA50397F45D842 1E5E3E3172DEA839 AFA4B90ED40385DC 0F3E847322C0B941 00207BCA64AAA6BF`. Žinome jog ši šifrograma sudaryta naudojant `ElGamal` asimetrinio šifravimo metodą. Pradinė tekstograma buvo papildyta iki reikiamo ilgio nuliais (naudotas `NoPadding` metodas). Be to, žinome slaptąjį raktą x (`0BDBE219B628E8C3 7C2723ECBD7B9E27 402DA552386A05C5 4C44EEAE438E370A`) ir naudotus sisteminius ElGamal metodo parametrus – modulį p (`00EBFCB7E2CB29A9 C9EF551690E0A276 B643A78B9B54F1C0 DF26A7F778F219A1 DF`) ir generatorių g (`65EFF5CCAAC2B7E1 32335DECB7A7BC21 B9AFC7FF42259535 5BA83141C7910A9A`). Papildomai žinome ir viešąjį raktą y (`00E54C26F99C6213 5DA0DC788C20C54D A2836C93D80E26DF 0E350B353D286D9A 7C`). Reikia parašyti programą, kuri iššifruotų šią šifrogramą ir atspausdintų tekstogramą.

Sudarytos programos išeities tekstas pateiktas 4.7 pav., o jos veikimo rezultatas 4.8 pav.

```
1      public static void doElGamalDecrypt() throws Exception
2      {
3          BigInteger g256 = new BigInteger(
4              "65EFF5CCAAC2B7E132335DECB7A7BC21B9AFC7FF422595355BA83141C7910A9A", 16);
5          BigInteger p256 = new BigInteger(
6              "00EBFCB7E2CB29A9C9EF551690E0A276B643A78B9B54F1C0DF26A7F778F219A1DF", 16);
7          ElGamalParameterSpec  egSpec = new ElGamalParameterSpec(p256, g256);
8
9          BigInteger       ct = new BigInteger
   ("4C22AD8CBDA8173296B746A2EF4ED7141B206004A3627F68B9CA50397F45D8421E5E3E3172DEA839AFA4B90ED40385DC0F3E
```

```
   847322C0B94100207BCA64AAA6BF", 16);
10         byte[]           inputBytes = ct.toByteArray();
11         Cipher             cipher = Cipher.getInstance("ElGamal/None/NoPadding", "BC");
12         SecureRandom     random = new SecureRandom();
13
14         KeyFactory      keyFactory = KeyFactory.getInstance("ElGamal", "BC");
15         ElGamalPublicKeySpec pubKeySpec = new ElGamalPublicKeySpec(
16              new BigInteger("00E54C26F99C62135DA0DC788C20C54DA2836C93D80E26DF0E350B353D286D9A7C", 16),
17              egSpec);
18         ElGamalPrivateKeySpec privKeySpec = new ElGamalPrivateKeySpec(
19              new BigInteger("0BDBE219B628E8C37C2723ECBD7B9E27402DA552386A05C54C44EEAE438E370A", 16),
20              egSpec);
21         ElGamalPublicKey pubEG = (ElGamalPublicKey)keyFactory.generatePublic(pubKeySpec);
22         ElGamalPrivateKey privEG = (ElGamalPrivateKey)keyFactory.generatePrivate(privKeySpec);
23
24         System.out.println("Duotoji sifrograma : " + toHex(inputBytes));
25         cipher.init(Cipher.DECRYPT_MODE, privEG);
26         byte[] plainText = cipher.doFinal(inputBytes, 0, inputBytes.length);
27
28         System.out.println("Issifruota tekstograma : " + toHex(plainText));
29
30         //patikrinimas
31         cipher.init(Cipher.ENCRYPT_MODE, pubEG, random);
32         byte[] cipherText = cipher.doFinal(plainText);
33
34         System.out.println("Vel uzsifruotas : " + toHex(cipherText));
35
36         System.out.println("EG viesas Y : " + toHex(pubEG.getY().toByteArray()));
37         System.out.println("EG privatus X : " + toHex(privEG.getX().toByteArray()));
38         System.out.println("EG generatorius G : " +
   toHex(privEG.getParameters().getG().toByteArray()));
39         System.out.println("EG modulis P : " + toHex(pubEG.getParameters().getP().toByteArray()));
40     }
```

4.7 pav. Šifrogramos iššifravimas naudojant ElGamalio kriptosistemą

```
run:
Duotoji šifrograma : 4C22AD8CBDA81732 96B746A2EF4ED714 1B206004A3627F68 B9CA50397F45D842
1E5E3E3172DEA839 AFA4B90ED40385DC 0F3E847322C0B941 00207BCA64AAA6BF
Iššifruota tekstograma : ABBA2122
Vėl užšifruotas : 6C829A35AA99AD7E 6644F71CA9C9A33C C0652936A04C44B7 44A24B1C888C1076 C682A4E3A337581C
78A9232A62F49EC7 55B5B1883B3D5F9A E410723318321D09
EG viešasis Y : 00E54C26F99C6213 5DA0DC788C20C54D A2836C93D80E26DF 0E350B353D286D9A 7C
EG privatusis X : 0BDBE219B628E8C3 7C2723ECBD7B9E27 402DA552386A05C5 4C44EEAE438E370A
EG generatorius G : 65EFF5CCAAC2B7E1 32335DECB7A7BC21 B9AFC7FF42259535 5BA83141C7910A9A
EG modulis P : 00EBFCB7E2CB29A9 C9EF551690E0A276 B643A78B9B54F1C0 DF26A7F778F219A1 DF
```

4.8 pav. Šifrogramos iššifravimo naudojant ElGamalio kriptosistemą rezultatai

Išnagrinėjus programos tekstą matome, jog 3-6 eilutėse statiškai apsirašomi du `BigInteger` klasės skaičiai, kurie 7 eil. panaudojami ElGamalio metodo sisteminių parametrų (`ElGamalParameterSpec`) nustatymui. 9-10 eil. panaudojant `BigInteger` klasės kintamąjį apsirašomas pradinių duomenų masyvas. Šį masyvą buvo galima apsirašyti ir įprastai, tarpinis kintamasis `ct` panaudotas tik dėl to, kad būtų lengviau nukopijuotą didelį šešioliktainį skaičių paversti baitų masyvu. 11-14 eil sukuriami šifravimui reikiamų klasių objektai, 15-22 eilutėse nustatomi ElGamalio metodo raktai. 24-28 eil. Atliekamas tekstogramos iššifravimas ir pradinių duomenų bei rezultatų spausdinimas. 31-34 eil. panaudojant tuos pačius raktus ir sisteminius parametrus gautas pradinis pranešimas vėl užšifruojamas. Kaip matome iš 4.8 pav. pateiktų rezultatų, vėl užšifravus tą pačią tekstogramą, gaunama kitokia šifrograma. Ruošdamiesi darbo gynimui pasirenkite atsakyti kodėl taip yra. 36-69 eilutėse atspausdinamos visų ElGamalio metodo parametrų reikšmės, tai naudinga ieškant pradinių duomenų apsirašymo klaidų.

Šiame skyrelyje pateiktas tik ElGamalio metodo naudojimo pavyzdys. Kitas asimetrinis šifravimo metodas, RSA, veikia labai panašiai. Skiriasi tik jo naudojami parametrai ir raktai. Daugiau informacijos apie RSA raktų apsirašymą galite rasti atitinkamos modulio paskaitos skaidrėse.

Toliau, 7 ir 8 lentelėse pateikti standartinių (Sun) ir Bouncy Castle (BC) bibliotekų palaikomi asimetrinio blokinio šifravimo metodų pavadinimai ir galimi užpildai. Lentelių viršutinėje dalyje išvardinti standartiniai algoritmai kuriuos realizuoja tiek Sun tie BC bibliotekos. Apatinėje dalyje išvardinti algoritmai kuriuos papildomai palaiko BC biblioteka. Pilnas konkretaus šifravimo metodo pavadinimas (kaip ir simetrinių šifrų atveju) sudaro-

mas iš trijų dalių: algoritmo pavadinimo, režimo ir užpildo. Šis pilnas pavadinimas turi būti nurodytas kuriant atitinkamą šifravimo objektą (11 eil. 4.7 pav.). Visos trys dalys skiriamos pasviruoju brūkšniu, pvz., `RSA/NONE/OAEPWithSHA1AndMGF1Padding`, `RSA/NONE/ISO9796-1Padding` arba `ELGAMAL/ECB/PKCS1PADDING`. Reikia pažymėti, jog tik ElGamalio metodas realizuotas BC bibliotekoje palaiko ECB režimą. Bendru atveju asimetriniams šiframs joks režimas nenaudojamas (nurodoma `NONE`).

7 lent. Asimetrinės kriptosistemos

Algoritmas	**Rakto ilgis**	**Aprašas**
RSA	Bet koks, kartotinis 8 bitams, pakankamo ilgio šifruojamiems duomenims ir užpildui (tik BC, 2048)	RSA algoritmas pagal PKCS #1
BC		
ElGamal	Bet koks, kartotinis 8 bitams, pakankamo ilgio šifruojamiems duomenims ir užpildui (1024)	Tinka tik * pažymėti užpildai

8 lent. Asimetrinėse kriptosistemose naudojami užpildai.

Užpildas	**Aprašas**
NoPadding*	Užpildo nuliais
OAEPPadding OAEPWith<digest>And<mgf>Padding	Optimal Asymmetric Encryption Padding, apibrėžtas PKCS #1 pvz., OAEPWith**MD5**And**MGF1**Padding arba OAEPWith**SHA-512**And**MGF1**Padding
PKCS1Padding*	Užpildo schema apibrėžta PKCS #1
BC	
ISO9796-1Padding	Standarte ISO9796-1 aprašytas užpildas

5 Literatūra

[1] Eligijus Sakalauskas, et. al. Kriptografinės sistemos: mokomoji knyga. Kauno technologijos universitetas. Vitae Litera, 2008, 166 p.

[2] Eligijus Sakalauskas, et. al. Kriptografijos teorija: mokomoji knyga. Kauno technologijos universitetas. Vitae Litera, 2008, 150 p.

[3] Bruce Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition. John Wiley & Sons, Inc., 1996, 784 p.

[4] Man Young Rhee. Internet Security. John Wiley & Sons Ltd., 2003, 405 p.

[5] Mark Stamp, Richard M. Low. Applied Cryptanalysis: Breaking Ciphers in the Real World. Wiley-IEEE Press, 2007, 402 p.

[6] David Hook. Beginning Cryptography With Java. Wrox, 2005, 484 p.

[7] Oracle Technology. Java ™ Cryptography Architecture (JCA) Reference Guide for Java™ Platform Standard Edition 6. Prieiga per Internetą: http://docs.oracle.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html

[8] Legion of the Bouncy Castle Java cryptography APIs. Prieiga per Internetą: http://www.bouncycastle.org/java.html

[9] Legion of the Bouncy Castle Specifications. Prieiga per Internetą: http://www.bouncycastle.org/specifications.html

6 Laboratorinio darbo užduotis

Laboratorinis darbas gali būti atliekamas naudojant laboratorijos kompiuterį arba bet kokį kitą kompiuterį kuriame įdiegta Java programavimo aplinka (pvz., NetBeans) ir Bouncy Castle JCE kriptografijos biblioteka. Kiekvienas studentas iš dėstytojo turi gauti varianto numerį, pagal kurį 8 skyrelio lentelėse gali susirasti savo užduoties individualius parametrus. Šiame darbo aprašyme pateiktų programų fragmentus galite pasiimti iš Moodle sistemos (kopijuoti programos fragmentų iš šio aprašymo nereikia) ir panaudoti savo darbe. Jums reikia tik modifikuoti pateiktą programą pagal savo užduotį.

1. Pasileiskite Java programavimo aplinką (pvz., NetBeans), susikurkite naują projektą. Iš Moodle sistemos pasiimkite saugos laboratorinio darbo projekto šabloną (failas `Main.java`), jo pagrindinę dalį ir pagalbines funkcijas įkelkite į savo programą. Paleiskite programą ir šablone pateiktais metodais patikrinkite ♦ ar įdiegta BC kriptografinė biblioteka ir ♦ ar nėra apriboti šifravimo raktų dydžiai.

2. Pagal 4.1 skyrelį sukurkite programą, kuri ♦ užšifruotų pateiktą tekstogramą naudojant nurodytą blokinį šifravimo metodą (parametrai 9 lentelėje). Tegul programa, naudodama tuos pačius pradinius duomenis ir iššifruoja gautą šifrogramą. Patikrinkite ar gaunama tokia pati tekstograma. Gautus programos veikimo rezultatus su savo komentarais pateikite darbo ataskaitoje.

♦ Modifikuokite vieną baitą tekstogramos pradžioje, patikrinkite kiek skiriasi gauta šifrograma nuo pradinės.

♦ Prailginkite arba sutrumpinkite tekstogramą vienu baitu. Ar ji užsišifruoja? Kodėl?

♦ Modifikuokite vieną baitą pradinės šifrogramos pradžioje, patikrinkite ar modifikuota šifrograma išsišifruoja. Kiek skiriasi gauta tekstograma nuo pradinės? Darbo aprašyme pateikite gautus rezultatus ir pakomentuokite kodėl taip yra. Į ataskaitą įdėkite sukurtą programos fragmentą.

3. Pagal 4.1 skyrelį sukurkite programą, kuri ♦ iššifruotų pateiktą šifrogramą naudojant nurodytą blokinį šifravimo metodą (parametrai 10 lentelėje). Tegul programa, naudodama tuos pačius pradinius duomenis, vėl užšifruoja gautą tekstogramą. Patikrinkite ar gaunama tokia pati šifrograma. Gautus programos veikimo rezultatus pateikite darbo ataskaitoje.

♦ Modifikuokite vieną baitą tekstogramos pabaigoje, patikrinkite kiek skiriasi gauta šifrograma nuo pradinės.

♦ Modifikuokite vieną baitą šifrogramos pradžioje, patikrinkite ar modifikuota šifrograma išsišifruoja. Kodėl? Kiek skiriasi gauta tekstograma nuo pradinės? Kodėl taip yra?

♦ Modifikuokite vieną IV baitą. Patikrinkite kiek skiriasi gauta tekstograma. Kodėl? Darbo aprašyme pateikite visus gautus rezultatus ir savo pastabas bei atsakymus. Į ataskaitą įdėkite sukurtą programos fragmentą.

4. Pagal 4.2 skyrelį sukurkite programą, kuri ♦ patikrintų pateiktų tekstogramų vientisumą naudojant nurodytus maišos algoritmus (individualios užduoties parametrai 11 lentelėje). Patikrinkite ar užduotyje pateiktos tekstogramos nebuvo pakeistos, tikrinimui būtina naudoti metodą `MessageDigest.isEqual()`.

♦ Pradinėse tekstogramose (tiktai tose, kurios nebuvo pakeistos) pakeiskite po vieną baitą. Patikrinkite kiek pakito santraukos. Kodėl? Darbo aprašyme pateikite visus gautus rezultatus ir savo atsakymus. Į ataskaitą įdėkite **vienai maišos funkcijai** sukurtą programos fragmentą.

5. Pagal 4.3 skyrelį sukurkite programą, kuri ♦ patikrintų pateiktų tekstogramų vientisumą naudojant nurodytus pranešimų autentifikavimo metodus (parametrai 12 lentelėje). Patikrinkite ar visos tekstogramos nebuvo suklastotos, tikrinimui būtina naudoti metodą `MessageDigest.isEqual()`. Gautus programos veikimo rezultatus pateikite darbo ataskaitoje.

♦ Pabandykite sutrumpinti arba pailginti pateiktą raktą, ar metodas apskritai veikia? Kodėl? Darbo aprašyme pateikite visus gautus rezultatus ir savo atsakymus. Į ataskaitą įdėkite vienai MAC funkcijai sukurtą programos fragmentą.

6. Pagal 4.4 skyrelį sukurkite programą, kuri ♦ iššifruotų pateiktą šifrogramą naudodama nurodytą metodą, jo parametrus ir raktus (parametrai pateikti 13 ir 14 lentelėse). Tegul programa iš naujo užšifruoja gautą tekstogramą naudodama tuos pačius raktus ir

parametrus. Patikrinkite ar gaunama ta pati šifrograma. Kodėl taip yra? Gautus programos veikimo rezultatus pateikite darbo ataskaitoje.

♦ Nežymiai modifikuokite šifrogramą. Ją iššifruokite ir vėl užšifruokite. Kiek pakito galutinis rezultatas, kodėl? Kodėl ElGamalio kriptosistemoje naudojama `SecureRandom` klasė? Kam ji ten reikalinga, jei raktai negeneruojami (jie pateikti užduotyje)? Darbo aprašyme pateikite visus gautus rezultatus ir savo pastabas bei atsakymus. Į ataskaitą įdėkite sukurtą programos fragmentą.

7 Kontroliniai klausimai

1. Pagrindiniai kriptografijos uždaviniai (konfidencialumas, neišsiginamumas, vientisumas).
2. Kas yra blokiniai simetriniai šifrai? Kokios jų savybės? Kam jie naudojami?
3. Nubrėžkite ir paaiškinkite blokinio šifro panaudojimo schemą.
4. Kas yra blokinių šifrų režimai (ECB ir CBC) ir kokios yra jų ypatybės?
5. Nubrėžkite ir paaiškinkite blokinio šifro ECB (CBC) režime naudojimo schemą (užtenka užšifravimo dalies).
6. Kokias unikalias savybes turi srautiniai (angl. stream) šifrai.
7. Kuo skiriasi blokiniai ir srautiniai šifrai? Ar galima blokinį šifrą panaudoti ten kur reikalingas srautinis? O atvirkščiai?
8. Kas yra maišos funkcijos? Kam naudojamos maišos funkcijos?
9. Kas yra pranešimų autentifikavimo metodai (MAC)? Kam jie naudojami?
10. Kaip galima „sukonstruoti“ pranešimų autentifikavimo metodą? Kokius kitus kriptografijos metodus tam galima panaudoti?
11. Kuo skiriasi simetriniai ir asimetriniai šifrai? Kokiems kriptografijos uždaviniams spręsti geriau tinka asimetriniai šifrai? Kokiems nelabai tinka? Kodėl?
12. Kokias žinote asimetrines kriptosistemas (būtina žinoti bent tris), paaiškinkite jų esmę?
13. Nubrėžkite ir paaiškinkite apibendrintą asimetrinių šifrų naudojimo schemą.
14. Kokios yra asimetrinių šifrų savybės, raktai?
15. Paaiškinkite RSA kriptosistemos esmę, nubrėžkite schemą.
16. Kodėl RSA užpildas dedamas į tekstogramos pradžią, o AES – į pabaigą?
17. Kuo skiriasi JCA klasės `SecretKey`, `PublicKey` ir `PrivateKey`?
18. Kodėl Sun (Oracle) rekomenduoja kartu su programa neteikti kriptografinių bibliotekų? Kodėl bibliotekos pasirinkimą reiktų deleguoti programos naudotojui?
19. Kodėl kriptografiniuose taikymuose nenaudotina Java klasė Random? Ką rekomenduotumėte naudoti?
20. Kodėl kriptografiniuose taikymuose naudotinas metodas `MessageDigest.isEqual`, o ne `Arrays.equals`? Kokia yra laiko atakos (angl. timing attack) esmė?

8 Individualių užduočių parametrai

8.1 Simetrinių kriptosistemų naudojimas

9 lent. Individualių užduočių parametrai.
Užšifruokite tekstogramą naudodami pateiktus raktus ir metodus

Var. nr.	Meto-das	Reži-mas	Užpildas	Raktas	Tekstograma
1	AES	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617 2021222324252627	035816224D55DEC6 173EFE204B47B054 E063DCF424590D9F 2B5B087F1968AB24
2	DESede	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	18E8ABB73E3EB33F 853423CC08D403AC 853423CC08D403AC AB17F3E091D4C07A
3	camellia	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617	19E055E0DBDE4D32 D25CF19666895ABF 936AE30E306AD346 3A145DF32EF35162
4	blowfish	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617 2021222324252627	1C83E4B142E9A508 666F47EE87520E07 666F47EE87520E07 EE824E00D4A5F072
5	twofish	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	1E37EF00212DA96A 7341836121AE4976 8A19207B201300B4 7FCB9EE52AEA9FD6
6	DESede	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	204C542574404A14 853423CC08D403AC 853423CC08D403AC 04C319DDD8ECC110
7	DESede	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617	26AD9AF6DECD675C 31B3EE88B63E75F0 31B3EE88B63E75F0 A6D2DBE9D6B99D6C
8	serpent	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	33109C93DA9C5C93 3521B159AAF335FB DB6670109C3D6264 43EDCF4940F723E6
9	seed	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	3AC79F5614CEFC02 62BBFCA709CE9BAB 913649A2F5AC7F51 995BCE433F75165D
10	Blowfish	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	3B7AEB92B7A9EC03 74101F0CDAAE7570 74101F0CDAAE7570 1E0D0B0A18A0F4C9
11	AES	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617	427640D3C5F89135 E801E3B686068634 2B770FB61B352AB8 AEC5D2F7BA7AE796
12	tea	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	556EF3A4BB12B353 493E07ECE7716EDB 493E07ECE7716EDB 9618C3122F4250E0
13	DES	ECB	NoPadding	0001020304050607	58888D0BC37320D8 3789CD7F52B6F886 3789CD7F52B6F886 8596B9F375A971FB
14	twofish	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617 2021222324252627	5EFC061A93125CDF 76ADAA6DE6FC0AD5 AC28CA4F13856780 1CFD69BEA9B4EFE2
15	serpent	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617	78DDD621A23E70C3 103079A296C883D6 926B07EEB5C1F3CF FDC78BDC42A469B6
16	seed	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	6D3A99686664032B E15EF8F536115ACF 151C4E291733B58B 680D68C74E7A284C
17	AES	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	719AEAA97C5A673B 5C4B61E822F5E5F5 3280868F660CA282 2488E8BDCA6AC6EB

Var. nr.	Meto-das	Reži-mas	Užpildas	Raktas	Tekstograma
18	tea	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	723BB107755F928D 493E07ECE7716EDB 493E07ECE7716EDB 29620F694EF8C6E8
19	AES	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	9320C89B0E240258 C93D30F75B815270 65E86D8F2ABFD639 95ECB943EC270318
20	CAST6	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	B90E23EFB4584D56 6D7C90C64B0E84E2 437BB4031574D7F8 A9B085D858ACE3C0
21	CAST6	ECB	NoPadding	0001020304050607 08	B940B2EAB8AFEF99 D8C3A451BA3A7B4A A40CE3937CCBE8A3 D5D3B0F6C74F854C
22	serpent	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617 2021222324252627	3F2E9C1A20F2F231 376DB643247F2D37 C2C9386D2805DA5D C1C1AB612B01FAA5
23	CAST6	ECB	NoPadding	0001020304050607	BB57AEF28A560A84 4029BEA47EE8492D 40A773F48D30325D 8236FBA15E0B8D91
24	idea	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	BC7EE56FFBC0B096 988CE490B00FCD63 988CE490B00FCD63 BDE4DBB04FD9DB76
25	CAST5	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	BE246CCD6851088E 3330D2230A7B741A 3330D2230A7B741A 2C0C56EC02F846A5
26	CAST5	ECB	NoPadding	0001020304050607	D375D1E39C432C16 81EFAFA98DE025A9 81EFAFA98DE025A9 B084D0AF06679F7A
27	DES	ECB	NoPadding	0001020304050607	D39FFF545A4023B5 3789CD7F52B6F886 3789CD7F52B6F886 962DD7689021E7E1
28	camellia	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F	D49CFC9CAD5832F6 E4B5BB5FEF19168D A24038C6A97C4355 3D4D2F49B855D348
29	CAST5	ECB	NoPadding	000102030405	DB09986BEB27F1C8 20D009051C52439E 20D009051C52439E C48A3C7C10186315
30	blowfish	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617	DE60C0F68F18FD90 1B5094C80A482E23 1B5094C80A482E23 B1B88F746EA1472A
31	twofish	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617 2021222324252627	E383AA80931C42ED 3DBDE8D7F00BF561 07344B041BB0DCFC 7F305E4508743FCF
32	camellia	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617 2021222324252627	E5BEE5079D553EE9 DBE644048A8775B4 5E0B5B23D634F3E3 4C4F419E691C5805
33	Blowfish	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617	F24CF06FBFE0BC7C 1B5094C80A482E23 1B5094C80A482E23 E973106579073BA7
34	twofish	ECB	NoPadding	0001020304050607 08090A0B0C0D0E0F 1011121314151617	F8CC872C7BE46486 29137F1D79C5CFEC E3A261BAF9BC4D3F 5CE5D8FADE699132

10 lent. Individualių užduočių parametrai.
Iššifruokite šifrogramą naudodami pateiktus raktus ir metodus

Var. nr.	Metodas	Režimas	Užpildas	Raktas	IV	Šifrograma
1	TEA	CBC	TBCPadding	6665566666655666 3331133333311333	0706050403020100	0BFB0E46DA1A19D5 B8E283386FB492F3 574A4C3D4DA0FB82
2	xTEA	CBC	TBCPadding	6665566666655666 3331133333311333	0706050403020100	17262444CD22B4C4 5E6712F3D8AA4183 7EF65D0DBC593635
3	AES	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	1A70EC4BB8C98D63 A5B1A7BC631B04C1 CDAD3C77DB2566C8 5E7F8F55A8634261
4	AES	CBC	ZeroBytePadding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	1A70EC4BB8C98D63 A5B1A7BC631B04C1 F624DE98B1C73244 3E171CED5BE85C32
5	TEA	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100	1A8F94EA4AC2D508 836D9184553C3AC3 78111FB1E8302960
6	twofish	CBC	ZeroBytePadding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	215F00831FA0DA6C 726AE19D68058CF3 1B0910067EC282DE B9FA0100C09A4629
7	twofish	CBC	ISO10126Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	215F00831FA0DA6C 726AE19D68058CF3 30EFDB1168408BB0 13E85850665E5CBB
8	DESede	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100	2B86540F89BC7F01 376C3C3D5A08B04D E1C1E5B9C988D12B
9	Blowfish	CBC	TBCPadding	6665566666655666 3331133333311333	0706050403020100	376A3178C4FDA55E 30CF9462699D9E77 DFF10654ECDC1925
10	serpent	CBC	ISO10126Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	74367FCF34F45435 32B220D023399C9C B80CBE89BC7B62EF ED535CD8345F158C
11	Blowfish	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100	45A469BA661E6BF7 2E6089A69F37E721 A41BD7A956C361D0
12	CAST6	CBC	ZeroBytePadding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	4B09801F8F47940A D0CBC1E2CA223838 D4A10639AC40917A 490E600F5C8E963E
13	AES	CBC	TBCPadding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	4E4A5BA03B3730BA 04489A44D337C413 84F332F6AC6FE95E 4A41FBA4527F7D0C
14	Camellia	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	4F72FCA72E78E4B5 224D5D0A14182C1A 96A7CD82DAC9E3DB CFB66AB34A1FE013
15	twofish	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	5418E53392692DEE 3D592C4DE21B8F38 D2F054434C0D89EF C3DD97B34B229424
16	seed	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	5CADE7A3D99AAEC2 A016DA4A307C5DDA 3DF5C1A1D1A0EF26 DA91D7E94D7292B7
17	xTEA	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100	6294DF99EB4F2429 42FCCC8291FB9CC4 63788C13122A1D80
18	Skipjack	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100	78B55766B83A2158 2E060691D96B7B30 59F4E783A2B045C7
19	IDEA	CBC	ISO10126Padding	6665566666655666 3331133333311333	0706050403020100	91C1793C8B131869 B50A70335F644B5B 823C9D22F506F99D

Var. nr.	Metodas	Režimas	Užpildas	Raktas	IV	Šifrograma
20	IDEA	CBC	ZeroBytePadding	6665566666655666 3331133333311333	0706050403020100	91C1793C8B131869 B50A70335F644B5B 94E07177C968FF50
21	IDEA	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100	91C1793C8B131869 B50A70335F644B5B B6210991B5E2F4FD
22	seed	CBC	ZeroBytePadding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	9450B148342A5BE4 4DC676355E5A9823 55E1CEFE11EC636E C9549B157031F20A
23	seed	CBC	ISO10126Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	A128D916C0A90429 5BE39EB0B26BB959 CC1A3974CBBB9D9C C6F24443E9145C08
24	Camellia	CBC	ZeroBytePadding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	A9CA02BECC9D88B8 25368135143E30A3 A04F321DF554B59F C4F73B68069E6625
25	serpent	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	B4BE532D2ECF7929 697C2B362CDF9207 EE3D22791D4FBF4B 2BA8C15A7EAB9CDD
26	DESede	CBC	TBCPadding	6665566666655666 3331133333311333	0706050403020100	C6D319DD173FDD93 CDFB95184DC62F1B B282B31C8B16597A
27	Skipjack	CBC	TBCPadding	6665566666655666 3331133333311333	0706050403020100	CD6CE3CBE22A17E7 A13F41C138F4ABBD C12B9B9D57833793
28	CAST5	CBC	TBCPadding	6665566666655666 3331133333311333	0706050403020100	D5176220C46FAD86 4536469510EF8A82 B687B023019D2BDA
29	serpent	CBC	ZeroBytePadding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	614231DB612587F3 F04FD87058183EE0 8560CE97928601B2 7BCC29C2C0585240
30	AES	CBC	ZeroBytePadding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	EAC49CCD0198E472 CD6496C3C2DC4564 CF5A157D8C332164 83E059A4CC083C11
31	CAST6	CBC	ISO10126Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	EB61627B52349F7C E17EAB1918768985 DF98C24DC1736140 9631814EC7AB3B84
32	CAST6	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	ED9B85233E1A889C D1217A00E896510C ECBD14FA09FF296D E72B13819B8A6B0C
33	Camellia	CBC	ISO10126Padding	6665566666655666 3331133333311333	0706050403020100 08090A0B0C0D0E0F	F07557E3B0E9FF85 D05AB55D33749788 FC71A243DFE23616 C85B5525BB843597
34	CAST5	CBC	PKCS7Padding	6665566666655666 3331133333311333	0706050403020100	FF198D70B7292B94 7EA65494C6380934 F34FBD30851ECA51

8.2 Pranešimų vientisumo tikrinimas

11 lent. Individualių užduočių parametrai.
Naudodami pateiktus parametrus patikrinkite tekstogramos vientisumą

Var.	Metodas	Tekstograma	Santrauka
1	MD4	BABE000004050607 08090A0B0C0D	1657674F1D008C56 428C8B6D9DB199DC
1	RipeMD160	ABBA000004050607 08090A0B0C0D	138C4F74479575B7 AE8B23F175A35D04 E4E77AE3
1	GOST3411	DAF0020304050607 08090A90	2EB90EFCDCC91F26 42303229AFB70867 9F70095AB1A1D2F7 ED5B5E0A8B99BA48

Var.	Metodas	Tekstograma	Santrauka
2	GOST3411	DAF0020304050607 08090A	2EB90EFCDCC91F26 42303229AFB70867 9F70095AB1A1D2F7 ED5B5E0A8B99BA48
2	RipeMD128	BABCE00004050607 08090A0B0C0D10	2432D31571AFF71A BC49761BB17089
2	SHA-1	DAF0020304050607 08090A0B0C0D00	13CF2D5D2C8F3C48 CCEA7F54874B8F04 6572BB550136699C
3	RipeMD160	ABBA000004050607 08090A0B00	3ABC593D58191E06 011FCADA888E1C21 339D1B9039E919F7 F75223DF85A90AA0
3	RipeMD128	BABCE00004050607 08090A0B0C0D	2432D31571AFF71A BC49761BB17089EC
3	SHA-1	E0F0000004050607 08090A0B0C	6181F2DDF285E563 9BBC74E67A9AC34E 08622F6B
4	Tiger	E0F0000004050607 08090A0B00	6181F2DDF285E563 9BBC74E67A9AC34E 08622F6B
4	RipeMD256	ABBA000004050607 08090A0B00	3ABC593D58191E06 011FCADA888E1C21 339D1B9039E919F7 F75223DF85A90AA0
4	RipeMD128	DAF0020304050607 08090A	5C6FDE70D4C34C32 D89923A5240B1B2E
5	RipeMD128	DAF0020304050607 08090A00	5C6FDE70D4C34C32 D89923A5240B1B2E
5	RipeMD160	ABBA000004050607 08090A0B0C00	62869986F2168B75 BC1C9EB23ACEB36C F507142B678A5E23
5	RipeMD256	FADE000004050607 08090A0B0C	5510511B6DDECC39 F507142B678A5E23 6572BB550136699C 52F9D14A3114C0EF
6	RipeMD256	ABBA000004050607 08090A0B0C00	3ABC593D58191E06 011FCADA888E1C21
6	RipeMD256	FADE000004050607 08090A0B0C70	4510511B6DDECC39 F507142B678A5E23 6572BB550136699C 52F9D14A3114C0EF
6	RipeMD128	BABCE00004050607 08090A0B0C0D0E0F	689B4645E799B342 E52143007CBF4ABC
7	RipeMD160	ABBA000004050607 08090A0B0C10	62869986F2168B75 BC1C9EB23ACEB36C 40D09EF0
7	GOST3411	DAF0020304050607 08090A0B0C0D00	707094E03296DE4D 5AD462EB3DEA4DD6
7	RipeMD160	ABBA000004050607 08090A0B0C	62869986F2168B75 BC1C9EB23ACEB36C 40D09EF0
8	SHA-256	BAD0000004050607 08090A0B0C0D00	7D9BEBAAC4EA7660 E221A53B4403A193 5FD1136F
8	MD2	DAF0020304050607 08090A0B0C0D0E	707094E03296DE4D 5AD462EB3DEA4DD6
8	SHA-1	E0F0000004050607 08090A0B0C0D0E	7600FE33F45C6752 9F81AD171022BE84 24F602B9
9	MD2	DAF0020304050607 08090A0B0C0D00	707094E03296DE4D 5AD462EB3DEA4DD6 2E463EB37E9493DE
9	GOST3411	DAF0020304050607 08090A00	7EB90EFCDCC91F26 42303229AFB70867 9F70095AB1A1D2F7 ED5B5E0A8B99BA48
9	SHA-1	DAF0020304050607 08090A0B0C0D0E	A3CF2D5D2C8F3C48 CCEA7F54874B8F04 5FD1136F
10	MD2	DAF0020304050607 08090A0B0C0D0E00	707094E03296DE4D 5AD462EB3DEA4DD6
10	SHA-1	E0F0000004050607 08090A0B0C00	AC01CD5FAD323433 2E463EB37E9493DE 9F81AD171022BE84
10	SHA-256	BAD0000004050607 08090A0B0C	8ACD36BEA958C66B 2D69BB18025C6F6D BBF4DAA0CA816F85 073DE45C50FB95BE

Var.	Metodas	Tekstograma	Santrauka
11	RipeMD128	BABCE00004050607 08090A0B0C0D00	12C9967ED8C8453D F46CED7238FA67
11	SHA-1	E0F0000004050607 08090A0B0C0D	CC01CD5FAD323433 2E463EB37E9493DE C093979B
11	RipeMD256	ABBA000004050607 08090A0B0C0D	9ABC593D58191E06 011FCADA888E1C21 339D1B9039E919F7 F75223DF85A90AA0
12	MD4	DAF0020304050607 08090A0B0C0D0E00	15AB0BC7FCE39F6D D13810FE85FF5C58 2E8D65A5A6BEC9B8
12	Tiger	BAD0ACE004050607 08090A0B0C0D90	1D840257D657B778 2E8D65A5A6BEC9B8 3C705331701B8C1A
12	GOST3411	BABE000004050607 08090A0B	CBB5863DFD4E5FA2 F839EDD26A334691 33A9BEAAD97D9F4E D5102FB9AA299A1B
13	MD5	DAF0020304050607 08090A0B0C0D00	1BC3F9D1BECC121D 7624F6CF29CA6488 2E8D65A5A6BEC9B8
13	GOST3411	BABE000004050607 08090A0B0C0D0E00	B65BB73316C61ABD 4272C8B3C62CA7AE 8C81DA928CA1338E 196FD7A3F9FD7D93
13	Tiger	FACE000004050607 08090A0B0C0D0E	283FB88C4524C020 A5957B416C16FD49 1A4CBA8909583C3A
14	MD4	BABE000004050607 08090A00	22C579B89E12ABC5 1F404948F803B19F
14	Tiger	BAD0ACE004050607 08090A0B0C0D00	4D840257D657B778 2E8D65A5A6BEC9B8 3C705331701B8C1A
14	GOST3411	BABE000004050607 08090A0B0C0D0E	B65BB73316C61ABD 4272C8B3C62CA7AE 8C81DA928CA1338E 196FD7A3F9FD7D93
15	SHA-256	BAD0000004050607 08090A0B0C0D00	9D9BEBAAC4EA7660 E221A53B4403A193 9449B493BC84FCDB
15	MD4	BABE000004050607 08090A	22C579B89E12ABC5 1F404948F803B19F
15	Tiger	BAD0ACE004050607 08090A	5B3C0A40BE386987 7E5802DF767A0A71 EFFBAC62A51A11CB
16	MD5	BABCE00004050607 08090A0B0C0D	4119F924ACD0A299 C33E66A3183D70FC
16	Tiger	BAD0ACE004050607 08090A0B0C00	8D840257D657B778 2E8D65A5A6BEC9B8 3C705331701B
16	RipeMD320	ABBA000004050607 08090A0B0C0D	25399BCEC86662AA 1379862A91CB79E7 D50C1050CCEC2726 C9B086F44735B134 FB44BB8BA99B326D
17	Tiger	FACE000004050607 08090A0B0C0D00	983FB88C4524C020 A5957B416C16FD49 1A4CBA8909583C
17	RipeMD320	ABBA000004050607 08090A0B0C0D50	85399BCEC86662AA 1379862A91CB79E7 D50C1050CCEC2726 C9B086F44735B134 FB44BB8BA99B326D
17	RipeMD128	BABCE00004050607 08090A0B0C0D0E	82C9967ED8C8453D F46CED7238FA67D3
18	RipeMD256	FADE000004050607 08090A0B00	38229BB3726F9EBE 912B35C2BE280291 44A58CF69C11EF91 E38496DBB21E0BB4 35622BE5E01FCAEE
18	MD2	BABE000004050607 08090A0B0C0D0E	84679B41B8D2219E FA6873FDC366A834
18	Tiger	BAD0ACE004050607 08090A0B0C0D0E	BD906F235EA85D94 0C5BBAE9DE8C43FE 7D7BE29705E7EF75
19	MD4	DAF0020304050607 08090A0B0C0D0E0F	95AB0BC7FCE39F6D D13810FE85FF5C58
19	SHA-224	FADE000004050607 08090A0B0C0D0E	1E2E76EE296E8713 225B33DFEBABB5B9 435296A64FDC6EAC 2D37708C
19	RipeMD320	FADE000004050607 08090A0B00	48229BB3726F9EBE 912B35C2BE280291 44A58CF69C11EF91 E38496DBB21E0BB4 35622BE5E01FCAEE

Var.	Metodas	Tekstograma	Santrauka
20	RipeMD320	FADE000004050607 08090A0B0C	38229BB3726F9EBE 912B35C2BE280291 44A58CF69C11EF91 E38496DBB21E0BB4 35622BE5E01FCAEE
20	MD4	DAF0020304050607 08090A0B0C0D0E00	95AB0BC7FCE39F6D D13810FE85FF5C58
20	SHA-224	E0F0000004050607 08090A0B00	2073B9DF11D54C33 FE338A2F198A8F5D 735C6BB875D1939F 74AF0F12
21	MD2	BABE000004050607 08090A0B0C00	B2214F2F88B97DB2 6F6A25AC8B65BD7E
21	RipeMD320	ABBA000004050607 08090A0B0C0D0E00	5344CCA819D6EDF5 FB3817EEB4798BD1 E1142CCA8DA5DB5E 104A6D4C826261A2 857FD266A0B47518
21	SHA-224	E0F0000004050607 08090A0B0C	2073B9DF11D54C33 FE338A2F198A8F5D 735C6BB875D1939F 74AF0F12
22	MD2	BABE000004050607 08090A0B0C	B2214F2F88B97DB2 6F6A25AC8B65BD7E
22	SHA-224	E0F0000004050607 08090A0B0C0D0E	36AA7C1C507AA321 17B77FC2C5B6B440 100C17363ABB4729 3A4DE0F6
22	SHA-384	FACE000004050607 08090A0B0C0D0E	5DAAE24B54F0C953 F7D8350D7A1787D6 58493812147F243B 21D951DA9DB8CE3A 546C28EF98F474EC 541CF29A7A17204E
23	SHA-224	FADE000004050607 08090A0B0C0D00	7E2E76EE296E8713 225B33DFEBABB5B9 435296A64FDC6EAC 2D3770
23	MD5	DAF0020304050607 08090A0B0C0D0E	BBC3F9D1BECC121D 7624F6CF29CA6488
23	RipeMD320	ABBA000004050607 08090A0B0C0D0E	968CE1EF5108C8F3 A4470581F2A710B0 E71672D781995A9B E8EF37E2CFA55E59 F2D4152A1CB7B0DA
24	SHA-224	E0F0000004050607 08090A0B0C80	9073B9DF11D54C33 FE338A2F198A8F5D 735C6BB875D1939F 74AF0F12
24	SHA-384	FACE000004050607 08090A0B0C0D00	1DAAE24B54F0C953 F7D8350D7A1787D6 58493812147F243B 21D951DA9DB8CE3A 546C28EF98F474EC 541CF29A7A1720
24	MD5	BABCE00004050607 08090A0B0C0D0E0F 10	BDBE8BA56981DF78 CC3F49A3EB8A33EF
25	MD5	DAF0020304050607 08090A0B0C0D0E30	CBC3F9D1BECC121D 7624F6CF29CA6488
25	SHA-224	E0F0000004050607 08090A0B0C0D	98975EB9D222DE80 76ECFFABF20E12F1 75BAF1B1E2BFD66D 884054FC
25	SHA-384	BAD0000004050607 08090A0B0C0D0E	23A6CD9BA7502C95 57113E63BB8F66CB 378938811A5CF815 466F091D57FB5E0D 5EB2C758C05FE424 91E92F56274319ED
26	SHA-256	FACE000004050607 08090A0B0C0D00	128175CBA2492C6C 236A5EA8075940FB F8C04E1150837CD5 AA7293ED17432B41
26	Tiger	BAD0000004050607 08090A0B0C0D00	23A6CD9BA7502C95 57113E63BB8F66CB 378938811A5CF815 466F091D57FB5E0D 5EB2C758C05FE424 91E92F56274319ED
26	MD2	BABE000004050607 08090A0B0C0D0E0F	D4F6B76B66233CD0 702CD4D954CFE1D0
27	MD5	BABCE00004050607 08090A0B0C0D0E00	DACA23D1167E50A0 32145958D4B9937D
27	RipeMD256	ABBA000004050607 08090A0B0C	6216123E77523A6F C7736E48318AE558 91221DDBFE57C237 F4D25E603C689547
27	SHA-384	BAD0000004050607 08090A0B0C0D0E0F 10	446BF73CBF92A796 EFB9557F72873E74 051A951516FDC6B5 9FAEEDE9E4ED4627
28	RipeMD320	ABBA000004050607 08090A0B0C0D0E0F	7344CCA819D6EDF5 FB3817EEB4798BD1 E1142CCA8DA5DB5E 104A6D4C826261A2 857FD266A0B47518

Var.	Metodas	Tekstograma	Santrauka
28	MD4	BABE000004050607 08090A0B0C	F45BB5B16B3D90C6 87B4589529E67562
28	GOST3411	BABE000004050607 08090A0B0C0D	6ADF9941C97CC5B2 C23AC0583E59A16F 79586104E93B9559 3F1009FC407C6204
29	SHA-224	FACE000004050607 08090A0B0C0D00	B28175CBA2492C6C 236A5EA8075940FB F8C04E1150837CD5 AA7293ED17432B
29	SHA-384	BAD0000004050607 08090A0B0C0D0E00	5DFA18860857EC25 6ADA2DB0E2E6438E DF251E67F868AE09 D4DCFC018230C384 6D94890868749BC4 6A045D6BAA01A4ED
29	MD5	BABCE00004050607 08090A0B0C0D0E	FACA23D1167E50A0 32145958D4B9937D
30	SHA-384	BAD0000004050607 08090A0B0C0D0E0F 70	946BF73CBF92A796 EFB9557F72873E74 051A951516FDC6B5 9FAEEDE9E4ED4627 34D0C48BFF4B05D8 C8AC0DF911713DC7
30	RipeMD160	DAF0020304050607 08090A0B0C	06A34725B29CB5E6 9E881256F076D8AE 29E8A7FC
30	SHA-256	FACE000004050607 08090A0B0C0D0E	B28175CBA2492C6C 236A5EA8075940FB F8C04E1150837CD5 AA7293ED17432B41
31	RipeMD160	DAF0020304050607 08090A0B0C50	06A34725B29CB5E6 9E881256F076D8AE 29E8A7FC
31	RipeMD256	ABBA000004050607 08090A0B	0CBE67ED9B0E41D3 160FA36743F00003 7866EF14241F84C4 22D5247234B1F065
31	SHA-384	BAD0000004050607 08090A0B0C0D0E0F	9DFA18860857EC25 6ADA2DB0E2E6438E DF251E67F868AE09 D4DCFC018230C384 6D94890868749BC4 6A045D6BAA01A4ED
32	RipeMD160	ABBA000004050607 08090A0B0C0D0E	0796BCC9CD883C79 5507180DEB06B626 210B40F6
32	SHA-256	BAD0000004050607 08090A0B0C0D0E	1D9BEBAAC4EA7660 E221A53B4403A193 9449B493BC84FCDB ED0EA0EE5AD4B0A0
32	Whirlpool	FACE000004050607 08090A0B0C0D00	7D0B0529EBF2D8F7 395C73771A6E0172 27F576374FCA0D37 2B6D5C202949050F B897A8DE7142A170 F96166C90688014A 98837A7E1A7AAF19 8C57C72A73EB255C
33	Whirlpool	FACE000004050607 08090A0B0C0D0E	3D0B0529EBF2D8F7 395C73771A6E0172 27F576374FCA0D37 2B6D5C202949050F B897A8DE7142A170 F96166C90688014A 98837A7E1A7AAF19 8C57C72A73EB255C
33	SHA-1	E0F0000004050607 08090A0B0C50	1181F2DDF285E563 9BBC74E67A9AC34E 08622F6B
33	SHA-256	BAD0000004050607 08090A0B0C0D0E0F	26706E11292C1A2A 6629C1EC417E813E BBD182335E81721C 637D8197C4DFC158

12 lent. Individualių užduočių parametrai.
Naudodami pateiktus parametrus patikrinkite ar tekstogramos nebuvo suklastotos

Var.	Metodas	Tekstograma	Raktas	MAC
1	HmacSHA1	BABCE00000010203 0405060708090A00	5172333435363738	098C3E536A4E3579 EB4254F10F5A4D84 6950D8AA
1	HMac-RipeMD128	FADE000000010203 040506	717233343536	0C7F638EB778C2F3 72AB520385F8FAAE

Var.	Metodas	Tekstograma	Raktas	MAC
1	HMac-Tiger	BAD0000000010203 040506070809	3132333435363738 393A3B3C	AF7D0DC4DA5E8E47 436768F68FB095CF 2530FCCF4683F147
2	HMacMD5	DAD0000000010203 040506070800	313233343536	212C7FCAF96BE289 FCB718C0F3435E35
2	HMac-Tiger	BAD0000000010203 040506	3132333435363738 393A	AFB4B0C41D6FE4AD 37F2154B53994ECC 0E89E9AF2AF1E7D7
2	HMac-MD4	BABE000000010203 040506	7172333435363738 393A3B3C	E95D9855ADE25FCC 8F416664ABC0A78D
3	HMac-RipeMD128	FADE000000010203 0405060708090A0B 0C	7172333435363738 393A3B3C	174F3F9F06A79BC1 64DEC6B5611D1AFE
3	HMacMD5	DAD0000000010203 040500	313233343536	2276E51B63AD57F0 2448A895BF04340E
3	HMac-Tiger	BAD0000000010203 040506	313233343536	BD982F2433464235 07BEFF9BFB52D421 29F1208E42FE4CB8
4	HMac-RipeMD160	DAFF000000010203 040500	7132333435363738 393A3B3C	25858E196265AD6E 87F6BF1C0EE73710 D5B3A157110
4	HMac-Tiger	BAD0000000010203 0405060708090A0B 0C	313233343536	BE85418E6B57E5E3 57CF76E5AC1DFD56 6E323AE8F81B5E30
4	HMac-MD2	BABA000000010203 040506070809	517233343536	EE65C1944F310F9E 0C1F18BDA36E3448
5	HMacMD5	DAD0000000010203 040506070809	313233343536	212C7FCAF96BE289 FCB718C0F3435E35
5	HMac-Tiger	BAD0000000010203 0405060708090A0B 00	3132333435363738 393A3B3C	37CEB1FC3A3F5885 9C53B7E409085989 32993AD19086BAFD
5	HMac-Tiger	BAD0000000010203 040506	3132333435363738 393A3B3C	D44EA4382C9DA884 DFB1B1ECF81386F1 1614AA803F679FC7
6	HmacSHA384	FACEB00000010203 040500	517233343536	7873A686EC9882EF 0A082706E57D335B FEE295AB46B8F0CB 3693B1BCB6586238 6E9D01522E4E8BB3 19EF5BF8889C6D5E
6	HMac-Tiger	BAD0000000010203 040506070809	3132333435363738 393A	E0C8BB0533275B5C E01E057F4A721187 75F0A88579489B5A
6	HMac-MD4	BABE000000010203 0405060708090A0B 0C	7172333435363738 393A3B3C	F1D03B0252C828EE 97438C91C7C6C049
7	HmacSHA512	FACE000000010203 040506	517233343536	55162FF896BF36CB 8FED32CC23A68A42 578DFE25694F894B 982C646452BF15D6 679AC480D3DD3F91 F269FF2A945CFA04 25BCE16A7C3D2BC4 2C305DDA714CFFB5
7	HMac-MD2	BABA000000010203 0405060708090A0B 00	517233343536	78F37E497409772E 42453CBF7208DBD6
7	HMac-MD2	BABA000000010203 0405060708090A0B 0C	5172333435363738	F3DBA2377307BD8D C597A510556E394B
8	HMac-RipeMD128	FADE000000010203 040506	7172333435363738 393A3B3C	37D933226A4715D4 36D8D92D2F8E648D
8	HmacSHA512	FACE000000010203 0405060708090A0B 0C	5172333435363738 393A3B3C	40F1B54226D588FE D13AE274281E5951 EFD0AD44527E6697 72E95F01B208C2E9 86FA0793DE3FE727 34FEA92208DAF5A9 CD8E2700AE18A79B 50657D42003CFDE1
8	HMac-SHA224	BAD0000000010203 0405060708090A0B 00	3132333435363738 393	832AC8ED591B70F4 0C492B1E7897CD96 D2CA3754D3940726 51F66DA8
9	HMac-RipeMD160	DAFF000000010203 0405060708090A0B 0C	7132333435363738 393A3B3C	1BF5C3B3B307436B 1DC9E049397AAECF C6755CCD

Var.	Metodas	Tekstograma	Raktas	MAC
9	HmacSHA512	FACE000000010203 0405060708090A0B 0C	517233343536	37C1E977ED603005 3DFF6F40A43F7ED8 B394F671E6FD87F7 778FB0F95144E376 122C540C6C2AFC79 DF0A6F1D4F986C12 DD573D0FF5356CAE D46FF3BA82023636
9	HMac-RipeMD160	DAFF000000010203 040500	713233343536	86079C13DCA42948 973DDD7A90D4A50F 30F1AFA5
10	HmacSHA1	BABCE00000010203 040506070809	5172333435363738 393A	301136085E659D96 A573227A427C088C A98E43FD
10	HMacMD5	DAD0000000010203 040506	3132333435363738	4F85B8FF1B27F796 085F0BEAF0162C1B
10	HMac-MD4	BABE000000010203 0405060708090A0B 00	7172333435363738 393A	87BB8AF2FE6C22BA C1D17DA818F432
11	HmacSHA1	BABCE00000010203 0405060708090A0B 0C	5172333435363738 393A3B3C	33DB42A0ADC3FD49 AD03259A7AF59854 15CF071F
11	HMac-RipeMD128	FADE000000010203 040506	7172333435363738	56A6EA943DFF0B39 4A376DB0689C1C61
11	HMac-SHA224	BAD0000000010203 0405060708090A0B 00	31323334353	9EDFF059BAD0EB3F 1DB21DE468C6AD92 8C62D7D2D4C68B5B 245E
12	HmacSHA1	BABCE00000010203 040506070809	517233343536	4B9C7B568C444AA6 9E513D325F9A590B 61F6F57A
12	HMac-RipeMD128	FADE000000010203 040506	7172333435363738 393A	642ECB834C485909 B51EB0197DFF5312
12	HMac-Tiger	BAD0000000010203 040506070800	3132333435363738 393A3B3C	AF7D0DC4DA5E8E47 436768F68FB095CF 2530FCCF4683F147
13	HMac-RipeMD160	DAFF000000010203 040506	7132333435363738 393A	4DE98604A9BAC749 1BE081C3248E513E 0D9A4B91
13	HMac-MD2	BABA000000010203 0405060708090A0B 0C	5172333435363738 393A	67A68375CEC380D7 D4F9FE3AB87FD723
13	HMac-RipeMD160	DAFF000000010203 040506070800	7132333435363738	BD646C8B16256B4A 080DDB196A636769 350A803B
14	HMac-RipeMD160	DAFF000000010203 040506070809	7132333435363738 393A3B3C	5A78848418E35661 777EC55633295218 A5DDE3A7
14	HmacSHA384	FACEB00000010203 040500	5172333435363738 393A3B3C	BEAA9F707CF7EA56 0507727C6E30E382 6F2603BDB4F66FC3 DB77CF4854F44249 42E3017A23AAC384 9A63669CAD4FF6A2
14	HmacSHA384	FACEB00000010203 040506070809	5172333435363738	D6F06C16AEA72A1B D5EEFE8B687137B7 91056BB43BE19A31 5A1E940FFE3017C4 7334CA28EDA0933C 0490D52D30465928
15	HMac-RipeMD160	DAFF000000010203 0405060708090A0B 0C	713233343536373	5F362D10FCF027EA 85213A0968E932AB 29B8A708
15	HmacSHA384	FACEB00000010203 040506	5172333435363738 393A3B3C	BEAA9F707CF7EA56 0507727C6E30E382 6F2603BDB4F66FC3 DB77CF4854F44249 42E3017A23AAC384 9A63669CAD4FF6A2
15	HMac-MD4	BABE000000010203 0405060708090A0B 00	717233343536	C32D60D84FEE25F9 CED88E185741F6FC
16	HmacSHA1	BABCE00000010203 040506070809	5172333435363738 393A3B3C	6332945834FCB38D 6F24CB558FBC5E24 A810799A
16	HMac-MD2	BABA000000010203 0405060708090A0B 0C	517233343536	78F37E497409772E 42453CBF7208DBD6
16	HmacSHA512	FACE000000010203 040506070800	5172333435363738	D1CC001ACEB0CD1E 00AD61F9E130FABF EB3D10C0D7068AC9 1C0308CC10A4393C 5D9ABFB7762A0012 C15B23D1C2316712 D02DA19ECC7BFA34 26A1533D62FBEE7F
17	HMac-RipeMD160	DAFF000000010203 040506070809	7132333435363738 393A	7AE7819F782E54B7 4FC53C55C6B372AF 5A1064FF
17	HmacSHA384	FACEB00000010203 040506070809	5172333435363738 393A	A828DE429941828E 1837FF67003DAFFC 8AD64C82BFFFB3FB 0C85E654596CB60D 8B0B47906D1AEC18 4AAEC73AED5ADE4F

Var.	Metodas	Tekstograma	Raktas	MAC
17	HmacSHA1	BABCE00000010203 040500	517233343536	D634AE8023C5957E D5178A0DD37EAD81 A7A77B76
18	HmacSHA1	BABCE00000010203 040506070809	5172333435363738	7DDB1FE836942636 BEB63C43B7A20F64 F7486336
18	HMac-RipeMD128	FADE000000010203 0405060708090A0B 0C	7172333435363738	7F6F5FD8A8444766 A68DDE27560591F1
18	HMac-SHA224	BAD0000000010203 040506070800	3132333435363738 393A	D7D5D3B73B1A556B 75AEC6BD4E65DA68 CB0151C1ECD57177 053
19	HmacSHA1	BABCE00000010203 040506	5172333435363738 393A3B3C	85DBEA5BCF02E117 23CC99233A33EA94 AD1C336A
19	HMac-MD2	BABA000000010203 040506	5172333435363738 393A3B3C	8D785B4ED71DE07B D2266145D1E6CBE1
19	HMac-SHA224	BAD0000000010203 040506070800	3132333435363738 393A	D7D5D3B73B1A556B 75AEC6BD4E65DA68 CB0151C1ECD57177 0532F7EF
20	HmacSHA384	FACEB00000010203 0405060708090A0B 0C	5172333435363738	67A6B1907AD2BF54 5E1E55BF9235CA1C EF435A744DD3B9C6 1822D05EF5432B84 6FF6D21D9FD16E2F ACCBE6ECAC2E332E
20	HMac-RipeMD160	DAFF000000010203 040506070809	713233343536	8AF8A33D0DE4B8B0 F6B528D78E10B127 79F6899E
20	HMac-MD2	BABA000000010203 040500	5172333435363738 393A	DE67674E6AA493B5 1F5A9A9FCEEF7822
21	HmacSHA384	FACEB00000010203 0405060708090A0B 0C	517233343536	4BEA5C48EE25980B 43270A09508695BA CE884EA5FEB36254 1266A046223E06D1 079CE7C22B36BD4A 0F28FACC7BE69D04
21	HMacMD5	DAD0000000010203 0405060708090A0B 0C	3132333435363738	961291860AC81E0D 3A982DD6AF0CC2D6
21	HMac-MD2	BABA000000010203 040506070800	5172333435363738 39	E1120C261DB3E6B8 F8D6F11AD3F6B85B
22	HMac-MD4	BABE000000010203 0405060708090A0B 0C	7172333435363738	997CB3C6A82AFEB9 1FBE283695758A9F
22	HMac-RipeMD160	DAFF000000010203 0405060708090A0B 0C	7132333435363738 393A	A7013BE43546C6AA 4F9B21FECD9E26F7 F2BAD11C
22	HmacSHA512	FACE000000010203 040506070800	5172333435363738 393A	E889198A20A6FD04 8ED5A14C9C4C5F9A 5BCB00DA1D92DEA5 B5BA84CF4E20D70B 50C1EFF2BEE749F4 EE7533441764F507 386FFE49C250AC40 2976D55F7E7D5942
23	HmacSHA384	FACEB00000010203 0405060708090A0B 0C	5172333435363738 393A	33D1F6CFA301CB3F 9C887CAA784F4220 21071A695ECD208A 17CB9F32120759BE C440488CF959DF6E DF05D623872E010F
23	HMac-RipeMD160	DAFF000000010203 040506070809	7132333435363738	BD646C8B16256B4A 080DDB196A636769 350A803B
23	HMac-RipeMD128	FADE000000010203 0405060708090A0B 00	717233343536	ED10436CDC1CA3A9 2292978EA3C51BC6
24	HMac-MD4	BABE000000010203 040506	7172333435363738 393A	A191330DC03C5729 E32799B78A12253E
24	HmacSHA1	BABCE00000010203 040506	517233343536	D634AE8023C5957E D5178A0DD37EAD81 A7A77B76
24	HMac-RipeMD128	FADE000000010203 040506070800	717233343536	EF6AB640B0845F4C B3C25D246B039824
25	HMac-SHA224	BAD0000000010203 040506070809	3132333435363738	CD167D856FB82C8A BE45D3764C61DC90 55E07AFC94CF9A22 5FEB57D5
25	HmacSHA1	BABCE00000010203 040506	5172333435363738 393A	DC357299C80D941C BDD8B4B2A3300910 899C4A83
25	HMac-RipeMD128	FADE000000010203 040506070800	717233343536	EF6AB640B0845F4C B3C25D246B039824
26	HMac-MD4	BABE000000010203 040506070809	7172333435363738 393A	C1E8C0BE51E744F0 28A80EBBB1E236FF
26	HMac-SHA224	BAD0000000010203 040506070809	313233343536	CADEB283E8B60821 FAC88DB8474B4BA1 CE4994A00027B7DB E77283B0
26	HMac-RipeMD160	DAFF000000010203 040500	7132333435363738	F2FA9AB80E7B6638 2FAA062A3A49654C EFA7F13D

Var.	Metodas	Tekstograma	Raktas	MAC
27	HMac-SHA224	BAD0000000010203 040506	3132333435363738 393A3B3C	C05BC084CF8A5E6B 2650E953D6683EB3 F38B7A841D46A1E0 6476A0E2
27	HMac-RipeMD160	DAFF000000010203 040506	7132333435363738	F2FA9AB80E7B6638 2FAA062A3A49654C EFA7F13D
27	HMac-MD4	BABE000000010203 040506070800	71723334353	FA4F9AE9B2C2FCB3 D3C590E44F63FDB6
28	HMac-Tiger	BAD0000000010203 0405060708090A0B 0C	3132333435363738 393A3B3C	37CEB1FC3A3F5885 9C53B7E409085989 32993AD19086BAFD
28	HMac-SHA224	BAD0000000010203 0405060708090A0B 0C	3132333435363738 393A3B3C	A7B338B0B1DB4F79 59502466CF871EB2 11DDA6271F119F0C 34262B49
28	HMac-MD4	BABE000000010203 040506070800	717233343536	FA4F9AE9B2C2FCB3 D3C590E44F63FDB6
29	HMac-RipeMD128	FADE000000010203 0405060708090A00	7172333435363738 393A	034C67F735673CB7 E8CA43E863255ED7
29	HMac-Tiger	BAD0000000010203 0405060708090A0B 0C	3132333435363738 393A	56DB6F627E420D95 753C33933E1F9FAE 5F49931323CF4383
29	HMac-SHA224	BAD0000000010203 040506	3132333435363738	83E26DA2C0AB7E5E AD7445363646F620 40AA7F6F6B880476 4CB573E4
30	HMac-MD2	BABA000000010203 0405060708090A0B 00	5172333435363738 393A3B3C	071E88A8892D4ECC 9A5AEA245C22034B
30	HMac-Tiger	BAD0000000010203 040506	3132333435363738	64541E8920F0B2CB D56A59AA2A598915 6D3C68878B02C4D7
30	HMac-SHA224	BAD0000000010203 0405060708090A0B 0C	3132333435363738 393A	832AC8ED591B70F4 0C492B1E7897CD96 D2CA3754D3940726 51F66DA8
31	HMac-RipeMD160	DAFF000000010203 0405060708090A0B 00	713233343536	072C0B70A6AD3FA4 D3D57FFDF812D65E E697DA7
31	HMac-SHA224	BAD0000000010203 040506	313233343536	346B3011DB4A366F 0CD38B490ADDCCC5 FDED0EA8DB2A270B 3572A7E3
31	HMac-Tiger	BAD0000000010203 0405060708090A0B 0C	3132333435363738	6A8F6EE26F82E721 5E1424EF0CF027A2 6DF3AF6C9FA9CCCC
32	HmacSHA1	BABCE00000010203 0405060708090A0B 00	517233343536	0800CCEE3D056119 66C6F63AC125ECEB 8029E649
32	HMac-SHA224	BAD0000000010203 0405060708090A0B 0C	3132333435363738	1FCE50577170264D 6461F80000F32520 41394D45EADC017F FDE59E84
32	HMac-Tiger	BAD0000000010203 040506070809	3132333435363738	72BB064C69A63416 BE68FCB002C13C1A 5FA5CD216C366E8A
33	HMac-MD2	BABA000000010203 04050600	517233343536373	091B11FDB0FE3ED3 07A2E7371CFCDBAE
33	HMac-Tiger	BAD0000000010203 040506070809	313233343536	A3563C857729AF5D 67CBB13DE04F5AA9 593D31029BF87C36
33	HMacMD5	DAD0000000010203 040506	3132333435363738 393A	E22EC41B246056AE 516F7056956E8929

8.3 Asimetrinių kriptosistemų naudojimas

13 lent. Individualių užduočių parametrai.
Naudodami pateiktus RSA parametrus iššifruokite šifrogramas

Var.	Rak.	Užpildas	Vieš.	Modulis	Šifrograma	Privatusis raktas
1	256	PKCS1Padding	010001	00B3446AF443CD84 13C155114359C501 DF6616282F89F3B1 78CFB62B689E899E 03	0554AE129B4788E8 03F5E5F6D01D6002 439C432D47A6D97A 298536D615309039	3D4224F641712A30 0201CABB6422B127 8E7008C9D6D3AFA6 3A67D919CED15719

Var.	Rak.	Užpildas	Vieš.	Modulis	Šifrograma	Privatusis raktas
3	256	PKCS1Padding	03	00C57C7AAC6CC49C 1EB0AB8CDDF0A3C2 2183BA532D876983 3549830B958BC013 01	0595ABCECABFF0E7 72B50C766FA843C8 FAEED53E591EF7E8 52ABA74E9C7389E7	0083A851C8488312 BF20725DE94B17D6 BFD668AC48A3D977 0F1B51C8B7B57935 E3
5	256	PKCS1Padding	03	00C57C7AAC6CC49C 1EB0AB8CDDF0A3C2 2183BA532D876983 3549830B958BC013 01	0D5A33488BDA4543 F0B79DE959AE0E78 DCFA3BFD5EA2D8FE FB204BCAA7FCB0F8	0083A851C8488312 BF20725DE94B17D6 BFD668AC48A3D977 0F1B51C8B7B57935 E3
7	256	ISO9796-1Padding	03	008548412EB82519 30B57BBFA3274094 D5AA6B8FC6180ACB 50B914E7A79276B8 15	0DDA354FF7D26017 BE8E72AEB8CBCA9A 16CDC74DA3B9E926 059AC659D8C738A1	58DAD61F256E10CB 23A7D5176F806338 1F092C9DD1F2EAFB C4A8BFB63F8DC3EB
9	256	ISO9796-1Padding	03	008548412EB82519 30B57BBFA3274094 D5AA6B8FC6180ACB 50B914E7A79276B8 15	101ACB7636ECF610 D1B88511FF545D1B E1BDC02A9E59C9B5 E250409952BA5F00	58DAD61F256E10CB 23A7D5176F806338 1F092C9DD1F2EAFB C4A8BFB63F8DC3EB
11	368	ISO9796-1Padding	010001	0092BA50B38C5C5F 91D9203B4420459B DA0C42662926FDC1 F0C18DB6B81A1DA2 BA95184939B1CD59 99F260E2252165	14D09710A0366325 019E9C6BB89D3962 E49A1304CD383EC2 1E74EF0EBFA48435 FF92AAD268E2A376 9B026BFCD588	05F802546B7F4E7B F6633346003214B6 F80CB36F3520D975 04D5E70FFAA502C4 4F6D665AD85855AE 274A99C59BB9
13	512	OAEPWithSHA1AndMGF1Padding	010001	00F4044E3AF2E724 A27E778BB695F8C1 0647B95821B878DF 5268EE87C4BC8553 3B392A3BF42AB6F7 1F3CEA4F5A61A661 E82093350E285411 13F7FF722A3672A9 3F	16769A30619EFF60 562DA5F10C511654 868D901BC60AE6C8 7252D4969D38E85D C271C50B98507314 8A5CD7B2A113346D CF4AB2DCD583F944 BBA7C05C0D6577A4	00DE20953E2023BD 3B9638289C7B04C8 617925054F1CE81B 129FA6933CCA07EB EC745972940C7940 38EA3E9BC4C0D79B EF3EBE318F072361 332889D0F7681DB1 21
15	256	ISO9796-1Padding	03	008548412EB82519 30B57BBFA3274094 D5AA6B8FC6180ACB 50B914E7A79276B8 15	177B800C091435B1 0B61DF25FB7FE056 1BCD4B3EEBF90BAC CBDEB2B55EA85A55	58DAD61F256E10CB 23A7D5176F806338 1F092C9DD1F2EAFB C4A8BFB63F8DC3EB
17	256	PKCS1Padding	010001	00B3446AF443CD84 13C155114359C501 DF6616282F89F3B1 78CFB62B689E899E 03	1F0E15B0D491DB7B 6C8F66883E809CE1 7F8CC510C314E320 2D0811455E335DA7	3D4224F641712A30 0201CABB6422B127 8E7008C9D6D3AFA6 3A67D919CED15719
19	256	PKCS1Padding	03	00C57C7AAC6CC49C 1EB0AB8CDDF0A3C2 2183BA532D876983 3549830B958BC013 01	26EFF9F30AF410EB 036DEBFC07F44236 82400C091E782E3D 17A1048F09B2367C	0083A851C8488312 BF20725DE94B17D6 BFD668AC48A3D977 0F1B51C8B7B57935 E3
21	368	OAEPWithMD5AndMGF1Padding	03	00A17DE2DBF5C2C6 8AF8EA044547670D 3C8E2F4F9693804B 7EC6FBCA5F09D0AB F85A73DB4DFC8FFF 0EADF9B040375D	2D9487AA09DBE708 084A97CC466A7CB2 CAE2E00100D11FF8 C79CF0568858F91E 2A5CFC8FD3007F53 0FC12695870F	6BA941E7F92C845C A5F1582E2F9A08D3 0974DFB9B7AADBEF 595662C4E29DD76F DEDF48F5CBBD4ED2 E961355F2013
23	256	ISO9796-1Padding	03	008548412EB82519 30B57BBFA3274094 D5AA6B8FC6180ACB 50B914E7A79276B8 15	30F1D86A417214EB 0ED5657093D2DA87 42C808CF675BBE15 9288442B704D9F9D	58DAD61F256E10CB 23A7D5176F806338 1F092C9DD1F2EAFB C4A8BFB63F8DC3EB
25	512	OAEPWithSHA1AndMGF1Padding	03	00B8BC77DC4EE55C E2BA2B3030CBABFE B3B67B8D0FC43AB0 B21C0BC44B4D17E5 0D136185EB33BF22 28D6C33F9D9A8966 85391A140EA5F75B 9F2D483A1525EA64 C5	3502745A31E8EEFC 52F0244BEA1BF16C 718E4F10E57E3EBB 7199A9D69457F6C8 6F953EECB4A13933 2E6A3808792961BE B8C2AE0AC3213374 F18FB86B4997144A	7B284FE83498E897 26C77575DD1D5477 CEFD08B52D7C75CC 12B282DCDE0FEE07 9507FAA1EB920210 5D0A439DE197B609 7194FBA32F5272FC FA73FC4CED16822B
27	512	OAEPWithSHA1AndMGF1Padding	03	00B8BC77DC4EE55C E2BA2B3030CBABFE B3B67B8D0FC43AB0 B21C0BC44B4D17E5 0D136185EB33BF22 28D6C33F9D9A8966 85391A140EA5F75B 9F2D483A1525EA64 C5	36216F625805D8C7 B2A9202D1AEA8957 F9D031FC89D0A340 E79BF1E6783783C9 AEE4E8E556FE9047 3709192F518CBE42 A9464447AC6502D1 CBAC3FDF1CD3F677	7B284FE83498E897 26C77575DD1D5477 CEFD08B52D7C75CC 12B282DCDE0FEE07 9507FAA1EB920210 5D0A439DE197B609 7194FBA32F5272FC FA73FC4CED16822B
29	368	OAEPWithMD5AndMGF1Padding	010001	0095DFC6FE031214 817A6BCF45A57946 FA27661B3C9FA900 DBCA76CCF90293F0 D4B0FAA401FD93D5 75F746FFDCB4C3	363208331B49E007 CF9D8DA96FFDFA19 325158EF37EF101A BC94F63CF7936DA5 28752CF686339091 146A418973E0	008B789FFA2A695C E3D1451AB239774E F1E45044F079B40F 41DC09D1A56971D2 C6A31CCAFB56A80D 58B9EB69A8F341

Var.	Rak.	Užpildas	Vieš.	Modulis	Šifrograma	Privatusis raktas
31	512	OAEPWithSHA1AndMGF1Padding	03	00B8BC77DC4EE55C E2BA2B3030CBABFE B3B67B8D0FC43AB0 B21C0BC44B4D17E5 0D136185EB33BF22 28D6C33F9D9A8966 85391A140EA5F75B 9F2D483A1525EA64 C5	382F7F012625F723 DE21D1C02614B5F6 7A28FEC6975E74A5 B268B9E0E9CCCB9C 6D929F88BC2C4648 4FA5C531514A619C A630BB01B60BCA55 AA3EAB6031F5B0C7	7B284FE83498E897 26C77575DD1D5477 CEFD08B52D7C75CC 12B282DCDE0FEE07 9507FAA1EB920210 5D0A439DE197B609 7194FBA32F5272FC FA73FC4CED16822B
33	368	OAEPWithMD5AndMGF1Padding	010001	0095DFC6FE031214 817A6BCF45A57946 FA27661B3C9FA900 DBCA76CCF90293F0 D4B0FAA401FD93D5 75F746FFDCB4C3	3966AB208B65B6DE BE5E9682A444AB81 F7046A7C5E70958E 0D04B9DF3BE4955F 92B1F381D2CE0241 75706290BB8B	008B789FFA2A695C E3D1451AB239774E F1E45044F079B40F 41DC09D1A56971D2 C6A31CCAFB56A80D 58B9EB69A8F341
35	256	PKCS1Padding	03	00C57C7AAC6CC49C 1EB0AB8CDDF0A3C2 2183BA532D876983 3549830B958BC013 01	3AE46D7CF5B01119 492AE220E454D773 7472A88545D4FA5A D69790737C867134	0083A851C8488312 BF20725DE94B17D6 BFD668AC48A3D977 0F1B51C8B7B57935 E3
37	368	OAEPWithMD5AndMGF1Padding	03	00A17DE2DBF5C2C6 8AF8EA044547670D 3C8E2F4F9693804B 7EC6FBCA5F09D0AB F85A73DB4DFC8FFF 0EADF9B040375D	3B2C3C5C59649E69 6C47CA14BB65BDE9 86523EE0203ED4BB 557F53B93102EF39 7EFB97B396001701 7A9DEAC65B46	6BA941E7F92C845C A5F1582E2F9A08D3 0974DFB9B7AADBEF 595662C4E29DD76F DEDF48F5CBBD4ED2 E961355F2013
39	368	OAEPWithMD5AndMGF1Padding	03	00A17DE2DBF5C2C6 8AF8EA044547670D 3C8E2F4F9693804B 7EC6FBCA5F09D0AB F85A73DB4DFC8FFF 0EADF9B040375D	3E9E580BFC8A6292 78BF2C3A641E5E6C D2EC4D4FA70414FB 0152FD5E46B13890 8BFA864D7518D985 A5B0D1993821	6BA941E7F92C845C A5F1582E2F9A08D3 0974DFB9B7AADBEF 595662C4E29DD76F DEDF48F5CBBD4ED2 E961355F2013

14 lent. Individualių užduočių parametrai.
Naudodami pateiktus ElGamalio kriptosistemos parametrus iššifruokite šifrogramas. Visų raktų ilgis 256b.

Var.	Užpildas	Privatus X	Viešas Y	Generatorius G	Modulis P	Šifrograma
2	PKCS1Padding	6BA1A8CEB02B1478 D59C15305A6979E2 B38871912356CAB5 8D63B263738A1BCC	3BF48147393A021C 3452679582233033 10FA81EC08ABDFD0 9C922ED91CE53DFE	3F798DAD0EA7742F B9FE821D556B1AF7 B3E1747B4E88C483 DC495CBFE0F4078E	00A791047E786375 ECADB5AC4557AE26 D0475425A1AA6C34 A707214548E62931 C3	0814411BFA2FB620 CC11E40AB1EA0DBB 055F19BB06E3F463 B457D1F281DA34DB 2B6BF5C1C48AB1C8 0A13101E5EC8BB7F FB537A9C235FDAB9 83DC334549D5D76F
4	PKCS1Padding	3A7D8F3C04D11D9E 3B8589B3C593496E 7DA4370D0C9DC6E6 2643ABAAB9986CBB	00E60EA2585CBAA4 279EF4A63B757DDC 4BF5AC525FDAFB7A F6054DF3DCE7F1F7 39	00804CC02FE800EC A09225EDC4B3EFB3 B2F1DCC8E4E22630 F95BD79B94DA1A34 90	00FB94772769A930 34BB7CE92C2B979B 4F034D63CDFE797F 59D0FF3D103A8DE8 63	0C4081EF9DF36D62 ED1C76D5DF1FA6D3 3F1975AAB4868427 C3C552BE571DF45C 6E879456C8113798 1357DAFE9E753CEA 717591A1D0777269 B6FAEDEA4FD8800A
6	PKCS1Padding	1E75407BEAAE22C1 8CD82C5E94A727A2 DCF2D9467A3264D7 04686819E1F0FA3F	717CE2A8B18F4382 B3CA0B29D0AB886C 9D2E546BCC973B29 C29EAD40E5022F31	00C9D167714BDDCC A0FDE7670C118EB3 3D217875085CF07F EDE745E21E5AC8AB C8	00E28DD8921B337F A87120753A18575F 9AB91D179902AFEF 899437B583E30D21 83	138B71DAB93BD102 0F5E691C1AFB2145 85FD25BEF6703687 85148ACA3A440DC0 27BBD1EA89BFE87C D3272B98F9B5ADF9 193059F1272AF6D1 81EEB0ADCAD1D84C
8	PKCS1Padding	72CB604BA7157AC9 A842277EA3C3F903 ADE1A8F6C5AC76A9 2847DBEC163D2A96	16A74F1670C2E9EB 207F330B0CC5DBDA 03B06CA9953BFA56 18EE69047AD22156	495EFB15CE004EB7 8A3067A7E2EA84CA 2AC59281CEDD0B83 1FC408F855999054	008FCF0E241474F6 D638A6BDD79161D3 DEF1CC8C7674DCCE BED4D7A42C5361C4 9B	1446EA77016F8088 0DA21530226E7BA7 412E6857ECD52574 0484D3F59A332075 24D4CB7C63FF6C49 B43EA3D0E039539B 50F6FB5EB0A2F584 565D729B4E6D487B
10	NoPadding	009A7CA929682D26 77444DE19A6B6726 4986F01CA1DADC44 44E2C523AC4202CA CE	2B54B9223CD6C151 103C768C8387D520 23AA4A2CA224994A 96FF119BFA44DEF7	1410A993B957573C 8D6096076EE31599 D7D90DC1FEC2F7BF ED669E2D410446B3	00BC56BEA21B4F55 FF23AD648AAB99A5 3341DF51D82EE2E8 9AF5E9FC0144D541 13	1CB542F3A54E99F4 F89F91BF36F58EF5 A70416009A29260F 1849D74BC0A46359 5356B4C200847C09 2E55FC95FEDAF608 F762151B9A6BF19D 9CF2BB510E49712B
12	PKCS1Padding	6BA1A8CEB02B1478 D59C15305A6979E2 B38871912356CAB5 8D63B263738A1BCC	3BF48147393A021C 3452679582233033 10FA81EC08ABDFD0 9C922ED91CE53DFE	3F798DAD0EA7742F B9FE821D556B1AF7 B3E1747B4E88C483 DC495CBFE0F4078E	00A791047E786375 ECADB5AC4557AE26 D0475425A1AA6C34 A707214548E62931 C3	20F1145FE4009689 123D1631B023DA1D 8DC287C779F7F43C 8CCBCE07384681CB 2E34FC8783FFC8B2 5BBD961270496912 A9E95FF6399BDA57 A0A9696E634DE8B5

Var.	Užpildas	Privatus X	Viešas Y	Generatorius G	Modulis P	Šifrograma
14	NoPadding	009A7CA929682D26 77444DE19A6B6726 4986F01CA1DADC44 44E2C523AC4202CA CE	2B54B9223CD6C151 103C768C8387D520 23AA4A2CA224994A 96FF119BFA44DEF7	1410A993B957573C 8D6096076EE31599 D7D90DC1FEC2F7BF ED669E2D410446B3	00BC56BEA21B4F55 FF23AD648AAB99A5 3341DF51D82EE2E8 9AF5E9FC0144D541 13	249F84E879B28FBC EFC2F40F42154272 AB03A1939CD5EEE5 DC261B07E7ABF464 2A145276E0FA6D37 09D5EC1660387FEA 45DD38B20E7B0598 8596E2BD586F6324
16	PKCS1Padding	6BA1A8CEB02B1478 D59C15305A6979E2 B38871912356CAB5 8D63B263738A1BCC	3BF48147393A021C 3452679582233033 10FA81EC08ABDFD0 9C922ED91CE53DFE	3F798DAD0EA7742F B9FE821D556B1AF7 B3E1747B4E88C483 DC495CBFE0F4078E	00A791047E786375 ECADB5AC4557AE26 D0475425A1AA6C34 A707214548E62931 C3	36D932E3A086C334 FFED0194F381E2E8 30A25D04B95AA0CB 1875AF4B7D159CE5 A60EB2291FA870DA 29F29726ECCC90BB 8601F5A4CC8FE4CB 6884DE39532DE30E
18	PKCS1Padding	72CB604BA7157AC9 A842277EA3C3F903 ADE1A8F6C5AC76A9 2847DBEC163D2A96	16A74F1670C2E9EB 207F330B0CC5DBDA 03B06CA9953BFA56 18EE69047AD22156	495EFB15CE004EB7 8A3067A7E2EA84CA 2AC59281CEDD0B83 1FC408F855999054	008FCF0E241474F6 D638A6BDD79161D3 DEF1CC8C7674DCCE BED4D7A42C5361C4 9B	3FBA3DA14183C971 E3C9347D8C3C71A6 3CF06DBC94520530 A80ECD73B86462EF 4E4E693BAB30BC08 82884FCE55EB3A11 D1D22F47B7EA5694 C85F19724EFD6ED6
20	PKCS1Padding	6BA1A8CEB02B1478 D59C15305A6979E2 B38871912356CAB5 8D63B263738A1BCC	3BF48147393A021C 3452679582233033 10FA81EC08ABDFD0 9C922ED91CE53DFE	3F798DAD0EA7742F B9FE821D556B1AF7 B3E1747B4E88C483 DC495CBFE0F4078E	00A791047E786375 ECADB5AC4557AE26 D0475425A1AA6C34 A707214548E62931 C3	40A9E0BB2754F5D3 8B42232276131B40 C3A41FC8A9C5D84C 577377E0DA1ED619 7FC72A6385DE3D45 90D370BB4C960AD4 CCC33A7B32F7E2CC AD7D22A34D207FB0
22	PKCS1Padding	6BA1A8CEB02B1478 D59C15305A6979E2 B38871912356CAB5 8D63B263738A1BCC	3BF48147393A021C 3452679582233033 10FA81EC08ABDFD0 9C922ED91CE53DFE	3F798DAD0EA7742F B9FE821D556B1AF7 B3E1747B4E88C483 DC495CBFE0F4078E	00A791047E786375 ECADB5AC4557AE26 D0475425A1AA6C34 A707214548E62931 C3	4B90064D17A47BF7 F041949077E05A54 535F040B8E96DEE0 4010BBFB31BDBD97 8EC98E9A33C87AA3 94D382CF718C5A90 AE499633832E7034 0568D171D4EBDAE1
24	NoPadding	0BDBE219B628E8C3 7C2723ECBD7B9E27 402DA552386A05C5 4C44EEAE438E370A	00E54C26F99C6213 5DA0DC788C20C54D A2836C93D80E26DF 0E350B353D286D9A 7C	65EFF5CCAAC2B7E1 32335DECB7A7BC21 B9AFC7FF42259535 5BA83141C7910A9A	00EBFCB7E2CB29A9 C9EF551690E0A276 B643A78B9B54F1C0 DF26A7F778F219A1 DF	4C22AD8CBDA81732 96B746A2EF4ED714 1B206004A3627F68 B9CA50397F45D842 1E5E3E3172DEA839 AFA4B90ED40385DC 0F3E847322C0B941 00207BCA64AAA6BF
26	PKCS1Padding	3A7D8F3C04D11D9E 3B8589B3C593496E 7DA4370D0C9DC6E6 2643ABAAB9986CBB	00E60EA2585CBAA4 279EF4A63B757DDC 4BF5AC525FDAFB7A F6054DF3DCE7F1F7 39	00804CC02FE800EC A09225EDC4B3EFB3 B2F1DCC8E4E22630 F95BD79B94DA1A34 90	00FB94772769A930 34BB7CE92C2B979B 4F034D63CDFE797F 59D0FF3D103A8DE8 63	4D0B0DDA9C8AE608 A5AF404F22258ED2 AACF9189F85C36CC 409E2881413A8729 525BA602D76660D8 B81F0B9CE52E4511 70A2D89F75D10592 CDBE15C9D7178D4E
28	PKCS1Padding	0D64E7A7B0520C63 58D15B7774939C62 C50B537986A2F14E C67AA5242635E951	4C2ABB78D8CE5246 4256F21E8025C0A5 10A6B07F1C23BC1A 0CF632BF1C348AD4	543890C267A83AC9 3AC8CCB7F2CF00C8 60672CAC73E2CB4F 54A19E87A24BA759	008843F84198CAA8 2413DD129C9DD2DE 7353FFBD7DE14226 B8CB037EA888ECCC 87	4E1BBA2D28691C2F A8B88932D590880F 26C939F9EA1FA17B C093F00F032D84E4 370F607F71012C25 8044FCA3213CC26D D738513B9F006049 F6E9A682FE4C9FB3
30	PKCS1Padding	0D64E7A7B0520C63 58D15B7774939C62 C50B537986A2F14E C67AA5242635E951	4C2ABB78D8CE5246 4256F21E8025C0A5 10A6B07F1C23BC1A 0CF632BF1C348AD4	543890C267A83AC9 3AC8CCB7F2CF00C8 60672CAC73E2CB4F 54A19E87A24BA759	008843F84198CAA8 2413DD129C9DD2DE 7353FFBD7DE14226 B8CB037EA888ECCC 87	51D575FC11086DC6 CD32EB83DDC6C570 C97B44097BDFD6EB CFFB63D120510614 49D45562EA240018 B9637FDAC2C3CE2E 6FA2FE09DE27FA61 9E235639821B583C
32	PKCS1Padding	0D64E7A7B0520C63 58D15B7774939C62 C50B537986A2F14E C67AA5242635E951	4C2ABB78D8CE5246 4256F21E8025C0A5 10A6B07F1C23BC1A 0CF632BF1C348AD4	543890C267A83AC9 3AC8CCB7F2CF00C8 60672CAC73E2CB4F 54A19E87A24BA759	008843F84198CAA8 2413DD129C9DD2DE 7353FFBD7DE14226 B8CB037EA888ECCC 87	58C957A4BF32553C 833164FF96D6B8E8 10DC75A15BF24620 56B264738E0285D7 7844778CBC1741C8 80E4FDDF6A910520 1CBB875221442073 5FB0B7692125D088
34	PKCS1Padding	72CB604BA7157AC9 A842277EA3C3F903 ADE1A8F6C5AC76A9 2847DBEC163D2A96	16A74F1670C2E9EB 207F330B0CC5DBDA 03B06CA9953BFA56 18EE69047AD22156	495EFB15CE004EB7 8A3067A7E2EA84CA 2AC59281CEDD0B83 1FC408F855999054	008FCF0E241474F6 D638A6BDD79161D3 DEF1CC8C7674DCCE BED4D7A42C5361C4 9B	5B387D748497A0EE BB04841E03082B78 02A6AB01A3651E71 F9DE6DCFF625617F 5FA35A40C9B5FF3B 3CBE0075235C64E4 70FCD390206A1FC5 E01E394A7EDA06D5

Var.	Užpildas	Privatus X	Viešas Y	Generatorius G	Modulis P	Šifrograma
36	NoPadding	0BDBE219B628E8C3 7C2723ECBD7B9E27 402DA552386A05C5 4C44EEAE438E370A	00E54C26F99C6213 5DA0DC788C20C54D A2836C93D80E26DF 0E350B353D286D9A 7C	65EFF5CCAAC2B7E1 32335DECB7A7BC21 B9AFC7FF42259535 5BA83141C7910A9A	00EBFCB7E2CB29A9 C9EF551690E0A276 B643A78B9B54F1C0 DF26A7F778F219A1 DF	6BB12D5ED3C2F757 D76D10EBE976894A A98947D14E16F9EB 72BF65039B6C59AE 4461D7A627DC6719 F00D97F5DDE144FD D77FD0DDEB671B00 444EC62CD4B8A5EB
38	PKCS1Padding	0D64E7A7B0520C63 58D15B7774939C62 C50B537986A2F14E C67AA5242635E951	4C2ABB78D8CE5246 4256F21E8025C0A5 10A6B07F1C23BC1A 0CF632BF1C348AD4	543890C267A83AC9 3AC8CCB7F2CF00C8 60672CAC73E2CB4F 54A19E87A24BA759	008843F84198CAA8 2413DD129C9DD2DE 7353FFBD7DE14226 B8CB037EA888ECCC 87	6F53B0B8C2916EFB F3668C23B98E6257 23135D4E7BE398ED 0847DE0EC126C02C 56C78DBFED07E51D 4F2F4B23384EB74F 4FA135A143DC1DB9 7D1649640CF1FB56