

Kodutöö

1. variant

- 1.1. Realiseerida javas teksti krüpteerimist kodeerimisviisi Atbash abil. Sisestatakse tekst. Krüpteerimise käigus asendatakse märk, mille positsioon tähestikus on i , selle märgiga, mille positsioon tähestikus on $n-i+1$, kus n on märkide arv tähesikus. Seejärel väljastatakse šifreeritud/dešifreeritud tekst. Krüpteerida selle abil tekst (≈ 4000 märke). Kasutades märkide statistikat <http://www.eki.ee/corpus/stats1.html> näidata, kuidas on võimalik dekrüpteerida saadut teksti, kasutades selleks teksti statistilisi omadusi.
- 1.2. Olgu funktsiooniks f DES krüpteerimine (javax.crypto). Realiseerida javas plokkšifri **dešifreerimise** OFB (Output Feedback). Sisestatakse krüpteeritud tekst (pikkusega $3 \cdot 64$ bitti), teisendatakse see kahendkoodi, tulemus jagatakse 64-bitisteks plokkideks. Vastavalt krüptoteksti suurusele läbitakse raunde, kuni kogu tekst on dešifreeritud. Dešifreeritud teksti väljastatakse nii kahendkoodis kui ka ASCII kujul.
- 1.3. Seletada (kirjalikult) samm-sammult algoritmi RSA põhimõtteid oma nime/perekonnanime kaheksa esimese järjest paikneva tähe šifreerimise näitel.

2. variant

- 2.1. Realiseerida javas teksti krüpteerimist ja dekrüpteerimist Caesari šifri abil. Sisestatakse tekst. Šifreerimise käigus asendatakse iga täht teise tähega, mis paikneb tähestikus asendatavast tähest n koha võrra paremal, kus n on fikseeritud. $E_n(x) = (x + n) \bmod 26$, $D_n(x) = (x - n) \bmod 26$. Seejärel väljastatakse šifreeritud/ dešifreeritud tekst.
- 2.2. Realiseerida javas programm, mis jaguneb kaheks osaks. Esimene genereerib avaliku võtme ja privaat- võtme (kasutada java.security.interfaces) ja šifreerib teksti. Teine osa dešifreerib teksti vastava võtmega. Šifreerimiseks kasutada teksti tagurpidi kirjutamist ning XOR operatsiooni vastava võtmega.
- 2.3. Olgu funktsiooniks f DES krüpteerimine (javax.crypto). Realiseerida javas plokkšifri **šifreerimise** OFB (Output Feedback). Sisestatakse krüpteeritud tekst, teisendatakse see kahendkoodi, tulemus jagatakse 64-bitisteks plokkideks. Vastavalt krüptoteksti suurusele läbitakse raunde, kuni kogu tekst on dešifreeritud. Dešifreeritud teksti väljastatakse nii kahendkoodis kui ka ASCII kujul.
- 2.4. Seletada (kirjalikult) samm-sammult algoritmi MD5 põhimõtteid oma nime/perekonnanime kaheksa esimese järjest paikneva tähe šifreerimise näitel.

3. variant

- 3.1. Realiseerida java teksti krüpteerimist ja dekrüpteerimist „Affine“ šifri abil (Affine cipher). Sisestatakse tekst. Iga tähe positsioon tähestikus on märgitud vastava numbriga. Krüpteeritud ja dekrüpteeritud tähe saab leida valemite abil $E(x) = (ax + b) \pmod m$, $D(x) = a^{-1}(x - b) \pmod m$, kus $1 = aa^{-1} \pmod m$. Seejärel väljastatakse šifreeritud/ dešifreeritud tekst.
- 3.2. Olgu funktsiooniks f DES krüpteerimine (javax.crypto). Realiseerida java plokkšifri **dešifreerimise** CFB (k-bit Cipher Feedback Mode). Sisestatakse krüpteeritud tekst, teisendatakse see kahendkoodi, tulemus jagatakse 64-bitisteks plokkideks. Vastavalt krüptoteksti suurusele läbitakse raunde, kuni kogu tekst on dešifreeritud. Dešifreeritud teksti väljastatakse nii kahendkoodis kui ka ASCII kujul.
- 3.3. Seletada (kirjalikult) samm-sammult algoritmi RC6 põhimõtteid oma nime/perekonnanime kaheksa esimese järjest paikneva tähe šifreerimise näitel.

4. Variant

- 4.1. Realiseerida java teksti krüpteerimist ja dekrüpteerimist „maagiliste ruutude“ abil. Kasutajal palutakse sisestada maagiline ruut (väljastada ekraanile tabelina), seejärel sisestatakse tekst. Tähed paigutatakse vastavatesse lahtritesse (arvude suurenemise järgi) ning väljastatakse ridade kaupa. Analoogselt teostada dešifreerimist.
- 4.2. Valida neli meetodit pseudojuhuarvude genereerimiseks. Kirjeldada nende meetodite matemaatilisi põhialuseid. Genereerida nelja meetodi abil jada pikkusega 100 arvu. Analüüsida tõenäosusjaotusi, näidata, millise meetodi abil on jada kõige juhuslikum.
- 4.3. Olgu funktsiooniks f DES krüpteerimine (javax.crypto). Realiseerida java plokkšifri **šifreerimise** CFB (k-bit Cipher Feedback Mode). Sisestatakse krüpteeritud tekst, teisendatakse see kahendkoodi, tulemus jagatakse 64-bitisteks plokkideks. Vastavalt krüptoteksti suurusele läbitakse raunde, kuni kogu tekst on dešifreeritud. Dešifreeritud teksti väljastatakse nii kahendkoodis kui ka ASCII kujul.
- 4.4. Seletada (kirjalikult) samm-sammult algoritmi Twofish põhimõtteid oma nime/perekonnanime kaheksa esimese järjest paikneva tähe šifreerimise näitel.

5. variant

- 5.1. Realiseerida java teksti krüpteerimine järgmise valemi abil:
$$y_n = \sum_{i=1}^k m_{n+i} \pmod{26}$$
. Sisestatakse tekst. Seejärel teisendatakse sisestatud tekst ASCII koodi ning arvutatakse iga järgmise šifreeritud tähe (y_n) ASCII kood, lähtudes eelmiste tähtede koodidest (m_{n+1} – on täht avatekstis). Väljastatakse tulemused nii tekstina kui ka ASCII kujul.

- 5.2. Olgu funktsiooniks f DES krüpteerimine (javax.crypto). Realiseerida javas plokkšifri **dešifreerimise** PCBC (Propagating cipher-block chaining). Sisestatakse krüpteeritud tekst, teisendatakse see kahendkoodi, tulemus jagatakse 64-bitisteks plokkideks. Vastavalt krüptoteksti suurusele läbitakse raunde, kuni kogu tekst on dešifreeritud. Dešifreeritud teksti väljastatakse nii kahendkoodis kui ka ASCII kujul.
- 5.3. Seletada (kirjalikult) samm-sammult algoritmi Serpent põhimõtteid oma nime/perekonnanime kaheksa esimese järjest paikneva tähe šifreerimise näitel.

6. variant

- 6.1. Realiseerida javas teksti krüpteerimist ja dekrüpteerimist valikulise ümberpaigutusega võtme alusel. Kasutajalt küsitakse tabeli suurust. Seejärel teisendatakse sisestatud tekst kahendkoodi. Saadud kahendkood kirjutatakse tabelisse tulpade kaupa. Tulbad paigutatakse ümber võtme alusel. Seejärel väljastatakse andmed nii kahendkoodis kui ka ASCII kujul.
- 6.2. Olgu funktsiooniks f DES krüpteerimine (javax.crypto). Realiseerida javas plokkšifri **šifreerimise** PCBC (Propagating cipher-block chaining). Sisestatakse krüpteeritud tekst, teisendatakse see kahendkoodi, tulemus jagatakse 64-bitisteks plokkideks. Vastavalt krüptoteksti suurusele läbitakse raunde, kuni kogu tekst on dešifreeritud. Dešifreeritud teksti väljastatakse nii kahendkoodis kui ka ASCII kujul.
- 6.3. Seletada (kirjalikult) samm-sammult algoritmi MARS põhimõtteid oma nime/perekonnanime kaheksa esimese järjest paikneva tähe šifreerimise näitel.

7. Variant

- 7.1. Realiseerida javas teksti krüpteerimist ja dekrüpteerimist Vigenère šifri abil. Sisestatakse tekst. Tabelis esimesele reale kirjutatakse tähestik (26 tähte), siis iga järgmisele reale tähestik, mis on nihutatud mõne sammu võrra. Kokku saadakse 26 erinevat šifrit. Šifreerimise etappidel valitakse erinevad "read" vastavalt võtmele. $C_i \equiv (P_i + K_i) \pmod{26}$, $P_i \equiv (C_i - K_i) \pmod{26}$. Seejärel väljastatakse šifreeritud/dešifreeritud tekst.
- 7.2. Kirjutada javas programm, mis jaguneb kaheks osaks. Esimene genereerib avaliku võtme ja privaat võtme (kasutada java.security) ja šifreerib teksti. Teine osa dešifreerib teksti vastava võtmega. Šifreerimiseks kasutada teksti tagurpidi kirjutamist ning XOR operatsiooni vastava võtmega.
- 7.3. Olgu funktsiooniks f DES krüpteerimine (javax.crypto). Realiseerida javas plokkšifri **dešifreerimise** ahelrežiim (Cipher Block Chaining Mode, CBC). Sisestatakse

krüpteeritud tekst, teisendatakse see kahendkoodi, tulemus jagatakse 64-bitisteks plokkideks. Vastavalt krüptoteksti suurusele läbitakse raunde, kuni kogu tekst on dešifreeritud. Dešifreeritud teksti väljastatakse nii kahendkoodis kui ka ASCII kujul.

7.4. Seletada (kirjalikult) samm-sammult algoritmi Blowfish põhimõtteid oma nime/perekonnanime kaheksa esimese järjest paikneva tähe šifreerimise näitel.

8. variant

8.1. Realiseerida java-s teksti krüpteerimist ja dekrüpteerimist lihtsustatud Alberti šifri abil. Sisestatakse tekst. Välisringi alfabeet on kirjutatud tähestiku järgi. Siseringi alfabeet on kirjutatud tagurpidi tähestiku järgi. Esimeses alfabeedis tähistatakse a-täht arvuga 1. Järgmisi tähti nummerdatakse selles alfabeedis kasvavas järjekorras (mod 26). Genereeritakse siseringi alfabeedi esimese tähe jaoks juhuarv (mod 26). See arv tähistab siseringi keeramist ning näitab, mis tähega asendatakse a-täht. Järgnevalt šifreeritakse teksti esimesed n tähte ning peale seda suurendatakse genereeritud arvu ühe võrra (ehk keeratakse sisemist ringi), šifreeritakse järgmist n tähte jne. Väljastatakse šifreeritud/dešifreeritud tekst.

(abiks on <http://scratch.mit.edu/projects/niethammer/324752>)

8.2. Olgu funktsiooniks f DES krüpteerimine (javax.crypto). Realiseerida java-s plokkšifri **šifreerimise** ahelrežiim (Cipher Block Chaining Mode, CBC). Sisestatakse krüpteeritud tekst, teisendatakse see kahendkoodi, tulemus jagatakse 64-bitisteks plokkideks. Vastavalt krüptoteksti suurusele läbitakse raunde, kuni kogu tekst on dešifreeritud. Dešifreeritud teksti väljastatakse nii kahendkoodis kui ka ASCII kujul.

8.3. Seletada (kirjalikult) samm-sammult algoritmi IDEA põhimõtteid oma nime/perekonnanime kaheksa esimese järjest paikneva tähe šifreerimise näitel.