Peatükk 4

Avaliku võtmega süsteemid. RSA

4.1 Eelteadmisi keerukusteooriast

Definitsioon 4.1.1 *Olgu f ja y kaks kujutust. Öeldakse, et $f = \operatorname{poly} g$ kui leidub selline polünoom p, et $f = O(p(g))$.*

Definitsioon 4.1.2 *Tähistagu Σ suvalist **tähestikku** ja Σ^* kõikide sõnade hulka üle tähestiku Σ.*

Definitsioon 4.1.3 *Kujutus $f : \Sigma^* \to \Sigma^*$ on **arvutatav polünomiaalses ajas** $\Leftrightarrow$ leidub Turingi masin M, mis arvutab selle kujutuse ajas* $\operatorname{poly}|x|$.

*Kui f on arvutatav polünomiaalses ajas, öeldakse ka, et ta on **arvutatav ökonoomselt**.*

Definitsioon 4.1.4 *Öeldakse, et funktsioon $f : \Sigma^* \to \Sigma^*$ kuulub keerukusklassi $\mathcal{FP}$, kui ta on polünomiaalses ajas arvutatav.*

*Öeldakse, et f kuulub keerukusklassi $\mathcal{FNP}$, kui suvalise $x \in \Sigma^*$ korral leidub **tõestus** $C(x)$, $|C(x)| = \operatorname{poly}|x|$, nii et leidub polünomiaalajas töötav Turingi masin, mis seab sisendile $(x, C(x))$ vastavusse arvu $f(x)$.*

Põhimõtteliselt on $\mathcal{FP}$ selliste funktsioonide hulk, mille väärtust saab leida polünomiaalses ajas, $\mathcal{FNP}$ aga selliste funktsioonide hulk, millel leidub "lühike" tõestus, mille abil funktsiooni väärtus on arvutatav polünomiaalses ajas.

Vahest rohkem tuntud keerukusklassid $\mathcal{P}$ ja $\mathcal{NP}$ saadakse, kui piirdutakse kujutustega $f : \Sigma^* \to \mathbb{Z}_2$. Selliseid kujutusi nimetatakse ka **otsustusülesanneteks**.

Otsustusülesande f **kaaskujutuseks** nimetatakse kujutust $f'(x) := 1 - f(x)$.

Suvalise otsustusülesannete klassi $\mathcal{K}$ korral olgu $co\mathcal{K}$ klass, kuhu kuuluvad parajasti klassi $\mathcal{K}$ kujutuste kaaskujutused.

4.2 Ühesuunalised kujutused

Laiema publiku ette jõudis krüptoloogia kui teadus alles pärast 1976. aastat, kui Diffie ja Hellmann käisid välja *avaliku võtme* (*public key*) idee. 1978. aastal esitlesid Rivest, Shamir ja Adleman avalikkusele RSA krüptosüsteemi, esimese, mis antud ideed realiseeris.

Salajaste võtmetega süsteemidel oli kõigil üks suur puudus: enne suhtluse algust tuli võtmes kokku leppida ja *see* pidi ikkagi toimuma privaatselt, näiteks õlleklaasi taga kuskil hubases kohvikus. Selline variant ei ole aga eriti mõeldav tänapäeval, tiigrihüppe ajastul: kui Alice elab Honolulus ja Bob Võrus, on seda õlleklaasi suht raske organiseerida. Pealegi, turvalisuse säilitamiseks tuleks näiteks DES'i võtit vahetada kord tunnis. Selle peavad tegema arvutid, inimesest sõltumatult.

Teine pakitsev vajadus oleks oma andmeid signeerida, varustada neid **signatuuriga**, st. peale turvalisuse oleks vaja ka tagada autentsus: Bob tahab olla kindel, et kohtamise määras tõepoolest Alice, mitte Smith&Watson.

Kumbagi probleemi ei saa lahendada sümmeetrilise krüptograafia vahendite abil, kuid üllataval kombel on avaliku võtmega krüptograafia edukas nii ühel kui teisel juhul. Nimelt, anname igale kasutajale kaks erinevat võtit, e (krüptimisvõti) ja d (dekrüptimisvõti), kusjuures krüptimisvõti e on avalik ja teada kõigile, dekrüptimisvõti d on aga privaatne ning teada ainult tema omanikule. Kogu protsess käib nii:

1. Alice genereerib sügavas saladuses võtmepaari $(e_{\mathrm{A}}, d_{\mathrm{A}})$, nii et suvalise avateksti x korral

   (a) $x = d_{\mathrm{A}}(e_{\mathrm{A}}(x))$

   (b) $|e_{\mathrm{A}}(x)| = \mathrm{poly}\,|x|$ ja $|x| = \mathrm{poly}\,|e_{\mathrm{A}}(x)|$

   ning avalikustab võtme e_{A}.

2. Saatmine: otsida avalikust andmebaasist Alice'i Alice'i krüptimisvõtme e_{A} ning saata Alice'ile teade $e_{\mathrm{A}}(x)$.

3. Vastuvõtmine: Alice dekrüptib teate y oma salajase dekrüptimisvõtme d_{A} abil: $x := d_{\mathrm{A}}(y)$. Võtmepaari valiku tõttu on protokoll korrektne.

Märkus 4.2.1 *Juhul kui soovime krüptosüsteemi ka signatuuriskeemina kasutada, tuleb eeldada lisaks, et* $\forall x, x = e_A(d_A(x))$.

Olulised tingimused (edaspidi täpsustame neid):

1. Dekrüptimisvõtme d leidmine (teades krüptimisvõtit e) on arvutuslikult ebaökonoomne.

2. Algne võtmepaari leidmine, krüptimine ning dekrüptimine on kiired.

Funktsioon d_A on triviaalselt keerukusklassist $\mathcal{FNP}$ (võtame $C(x) := d\_A(x)$) ning seega *ei ole olemas tingimusteta turvalisi krüptosüsteeme*, vaid parimal juhul eksisteerivad ainult arvutuslikult turvalised avaliku võtmega krüptosüsteemid.

Turvaliste asümmeetriliste (nii neid veel kutsutakse) krüptosüsteemide eksistentsiks on vaja järgnevate väga spetsiifiliste kujutuste olemasolu:

Definitsioon 4.2.2 *Rahuldagu kujutus $f : \Sigma^* \to \Sigma^*$, järgmisi tingimusi:*

1. *f on üksühene ning leidub selline $k > 0$, et iga $x \in \Sigma^*$ jaoks $|x|^{1/k} \leq |f(x)| \leq |x|^k$ (st et $f(x)$ ja x pikkused on polünomiaalses sõltuvuses).*

2. *$f \in \mathcal{FP}$, st. et f on arvutatav polünomiaalses ajas.*

3. *Suvalise polünomiaalses ajas töötava randomiseeritud Turingi masina T korral tõenäosus $P\left(T(x) \in f^{-1}(f(x))\right)$ on tähtsusetu (*negligible*), st suvalise polünoomi $p(\cdot)$ ja piisavalt suure $|x|$ korral $P\left(T(x) \in f^{-1}(f(x))\right) < \frac{1}{p(|x|)}$.*

*Sellist kujutust nimetatakse **ühesuunaliseks**.*

Kui oleks $\mathcal{NP} = \mathcal{P}$, siis triviaalselt ühesuunalisi kujutusi ei leiduks. Ja vastupidi, kui me leiaksime mõne ühesuunalise kujutuse, kehtiks $\mathcal{NP} \neq \mathcal{P}$ ning me oleksime lahendanud tänapäeva teoreetilise arvutiteaduse kõige pakilisema ülesande. Samas, mittevõrdusest $\mathcal{NP} \neq \mathcal{P}$ ei tulene veel ühesuunaliste kujutuste olemasolu.

Paneme tähele, et ühesuunalistele kujutustele esitatud nõudmised on küllaltki põhjendatud. Nimelt, krüptoloogias jääb väheks sellest, et mingi krüptosüsteem pole lahtimurtav 'halvimal juhul", seda ei tohiks saada lahti murda *enamustel juhtudel*, ehk "kõikjal, väljaarvatud tähtsusetul alamhulgal".

Veelgi enam, Oskar võib kasutada randomiseeritud algoritme (vt. punkti 4.6) ning seega omada suuremat arvutusvõimsust kui polünomiaalajas töötavad deterministlikud Turingi masinad (nagu hiljem näeme, annab randomiseeritus võimsust juurde).

Samuti, praktikas kasutatavad avaliku võtme krüptosüsteemid fikseerivad ära võtme pikkuse, seega võiks Oskar eelnevalt konstrueerida (sealjuures kasutades kasvõi eksponentsiaalset aega) sellise Turingi masina T_n, mis pöörab antud kujutust f ainult antud võtme pikkuse n korral (ent teeb seda see-eest edukalt). Täpsustatud tingimus oleks selline: 'Suvalise polünomiaalses ajas töötava randomiseeritud Turingi masinate pere $\{T_n\}$ korral tõenäosus

$$P\left(T_{|x|}(x) \in f^{-1}(f(x))\right) < \frac{1}{p(|x|)}.$$

Ning lõpuks, f peab olema pööratav salainformatsiooni olemasolul.

Definitsioon 4.2.3 *Ühesuunalist kujutust f, mille korral leidub kujutus $d \in \mathcal{P}$, mis muudab pööramise triviaalseks, nimetatakse **püünisega kujutuseks** (*trapdoor function*).*

Seega, krüptosüsteemi alustalaks peab olema püünisega kujutus.

4.2.1 Ühesuunaliste kujutuste kandidaadid

Järgmist kahte kujutust peetakse ühesuunaliseks:

1. Täisarvude korrutamine.
2. Astendamine algarvulise mooduli järgi.

Vaatleme otsustusülesannet PRIMES (kas x on algarv?) (selle ülesande kaasülesandeks on COMPOSITES: kas x on kordarv?).

Teoreem 4.2.4 ([Pap94], lk. 222) PRIMES *ja* COMPOSITES *kuuluvad keerukusklassi* $\mathcal{NP} \cap co\mathcal{NP}$.

Seega suvalise algarvu p korral leiduvad tõestus $C(p)$, $|C(p)| = \text{poly}\,|p|$, ja deterministlik Turingi masin T, et $T(p, C(p)) = 1 \Leftrightarrow p$ on algarv. Siinjuures kasutatakse järgmist teoreemi:

Teoreem 4.2.5 *Täisarv* $p > 1$ *on algarv parajasti siis kui leidub selline arv* $1 < r < p$*, nii et* $r^{p-1} \equiv 1 \pmod p$*, ent* $r^{\frac{p-1}{q}} \not\equiv 1 \pmod p$ *iga* $q \mid p-1$*,* q *on algarv, jaoks.*

Sertifikaat $C(p)$ defineeritakse rekursiivselt,

$$C(p) := (r;\ q_1, C(q_1), \ldots, q_k, C(q_k)),$$

kus r on sama, mis eelmises teoreemis, ning arvud q_i on arvu $p-1$ kõik algarvulised jagajad.

Esimeseks ühesuunalise kujutuse kandidaadiks on *täisarvuline korrutamine* $f \colon \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$, $f(x, y) := x \cdot y$. Oletame, et p ja q on algarvud, siis

$$f(p, C(p), q, C(q)) := p \cdot q$$

(juhul kui p või q ei ole algarv (see on kontrollitav polünomiaalses ajas, kuna $C(p)$ ja $C(q)$ on ette antud), siis f väljastab midagi mõttetut, näiteks oma sisendandmed) on üksühene ning polünomiaalses ajas arvutatav. Hoolimata aastatepikkusest pingutusest ei ole leitud ökonoomseid faktoriseerimisalgoritme (vt. peatükk 6.2).

Teine kandidaat ühesuunaliste kujutuste perre on *astendamine algarvulise mooduli järgi.* Kujutuse f argumentideks on algarv p, algjuur r modulo p (selline juur sisaldub arvu p sertifikaadis) ning täisarv $x < p$,

$$f(p, C(p), r, x) := (p, C(p), r^x \mod p).$$

Kujutuse f pööramine on teine hästituntud raske arvutuslik probleem — *diskreetse logaritmi probleem* (vt. peatükk 7).

4.3 Eelteadmisi arvuteooriast

Lemma 4.3.1 *Olgu m ja n naturaalarvud, $m = \prod p_i^{k^i}$, $n = \prod p_i^{\ell^i}$, kus p_i on paarikaupa erinevad algarvud ning $k_i, \ell_i \geq 0$. Siis*

1. $m \mid n \Leftrightarrow \forall i, k_i \leq \ell_i$.

2. $\gcd(m,n) = \prod p_i^{u_i}$, *kus* $u_i := \min(k_i, \ell_i)$.

3. $\mathrm{lcm}(m,n) = \prod p_i^{v_i}$, *kus* $v_i := \max(k_i, \ell_i)$.

Proof:

1. Olgu $m \mid n$, st $\exists a \in \mathbb{N}, m \cdot a = n$. Olgu $a = \prod p_i^{j_i}$, kus $j_i \geq 0$. Seega

$$\left(\prod p_i^{k_i}\right)\left(\prod p_i^{j_i}\right) = \left(\prod p_i^{k_i+j_i}\right) = ma = n = \prod p_i^{\ell_i}.$$

Aritmeetika põhiteoreemi põhjal $k_i + j_i = \ell_i$, seega $k_i \leq \ell_i$.

Olgu $k_i \leq \ell_i$. Siis $m \cdot \left(\prod p_i^{\ell_i - k_i}\right) = n$, ehk $m \mid n$.

2. Olgu $d := \prod p_i^{u_i}$. Kuna $u_i = \min(k_i, \ell_i)$, siis 1. põhjal $d \mid m$ ja $d \mid n$. Olgu $c \mid m$ ja $c \mid n$, kus $c = \prod p_i^{j_i}$. Siis 1. põhjal $j_i \leq \min(k_i, \ell_i)$ ja seega $j_i \leq u_i = \min(k_i, \ell_i)$, millest järeldub, et $c \mid d$.

3. Jääb lugejale tõestada.

■

Lemma 4.3.2 *Kui $a \mid bc$ ja $\gcd(a,b) = 1$, siis $a \mid c$.*

Proof: Kuna $\gcd(a,b) = 1$, siis leiduvad sellised täisarvud x ja y, et $ax + by = 1$, seega $axc + byc = c$. Et a jagab selle võrduse vasakut poolt, siis peab ta jagama ka paremat poolt, st $a \mid c$. ■

Kongruentse kujul $ax \equiv b \mod n$ nimetatakse **lineaarkongruentsideks**.

Lemma 4.3.3 *Olgu $d := \gcd(a,n)$, $a' := a/d$, $n' := n/d$. Lineaarkongruentsil $ax \equiv b \mod n$ on lahend parajasti siis, kui $d \mid b$. Sel juhul on kongruentsil d erinevat lahendit $x_0 + in' \mod n$, $0 \leq i \leq d-1$, kus x_0 on selle kongruentsi mingi lahend.*

Proof: Kui x_0 on lahend, siis leidub $k \in \mathbb{Z}$, nii et $b = ax_0 + kn$. Kuna d jagab viimase võrduse paremat poolt, siis ka $d \mid b$.

Olgu $d \mid b$, ning $b' := b/d$. Teoreemi 4.3.1 põhjal leiduvad sellised täisarvud x_0' ja y_0', et $ax_0' + by_0' = d$. Korrutades võrduse mõlemad pooled arvuga b', saame $a(x_0'b') + b(y_0'b') = b$. Võttes $x_0 := x_0'b'$, saame $ax_0 \equiv b \mod n$ ning x_0 on lahend.

Olgu $0 \leq i \leq d-1$. Sel juhul $a(x_0 + kn') \equiv ax_0 + a'dkn' \equiv b + a'kn \equiv b \mod n$ on lahend. Näitame, et need lahendid on erinevad. Tõepoolest, olgu $x_0 + in' = x_0 + jn' \mod n$, st $n \mid (j-i)n'$, mis pole aga võimalik juhul $0 \leq i < j \leq d-1$.

Olgu x_1 mingi suvaline lahend. Kuna $ax_0 \equiv b \mod n$ ja $ax_1 \equiv b \mod n$, siis $a(x_1 - x_0) = 0 \mod n$, st leidub $k \in \mathbb{Z}$, nii et $a(x_1 - x_0) = kn$. Jagades viimase võrduse mõlemaid pooli arvuga d saame, et $a'(x_1 - x_0) = kn'$. Kuna

$\gcd(a', n') = 1$, siis lemma 4.3.2 põhjal $n' \mid x_1 - x_0$, st leidub selline $l \in \mathbb{Z}$, et $x_1 - x_0 = ln'$ ehk $x_1 = x_0 + ln'$. Lemma A1.1 põhjal leiduvad sellised $q, r \in \mathbb{Z}$, et $l = qd + r$, $0 \leq r < d$ ning seega $x_1 = x_0 + (qd + r)n' = x_0 + rn' + qn \equiv x_0 + rn' \mod n$. ∎

Teoreem 4.3.4 (Hiina jääkide teoreem) *Olgu* $a_1, \ldots, a_s, n_1, \ldots, n_s \in \mathbb{Z}$, *kusjuures* $\gcd(n_i, n_j) = 1$, *kui* $i \neq j$. *Siis kongruentside süsteemil*

$$\begin{aligned} x &\equiv a_1 \mod n_1 \\ &\cdots \\ x &\equiv a_s \mod n_s \end{aligned}$$

on ühene lahend mooduli $n := n_1 \cdots n_s$ *järgi.*

Proof: Olgu $m_i := \frac{n}{n_i}$. Kuna $\gcd(n_i, n_j) = 1$, kui $i \neq j$, siis lemma A4.1 põhjal ka $\gcd(m_i, m_j) = 1$. Seega $m_i \in \mathbb{Z}^*_{n_i}$. Olgu $k_i := m_i^{-1} \mod n_i$ ning

$$x := \sum_{j=1}^{s} a_j k_j m_j.$$

Siis $x = \sum_{j=1}^{s} a_j k_j m_j \equiv a_i k_i m_i \equiv a_i \mod n_i$, st x on lahend. Kui ka y oleks lahend, siis $x \equiv a_i \equiv y \mod n_i$ ning kuna $\gcd(n_i, n_j) = 1$, siis lemmast A4.2 saame, et $x \equiv y \mod n$.

∎

Definitsioon 4.3.5 ***Euleri funktsioon*** $\phi(n) := |\{0 < i < n : \gcd(i, n) = 1\}|$ *(seega* $|\mathbb{Z}^*_n| = \phi(n)$*).* ***Carmichael'i funktsioon*** $\lambda(n)$ *on vähim selline* k, *mille korral* $a^k = 1 \mod \phi(n)$, *suvalise* $a \in \mathbb{Z}^*_n$ *korral.*

Lemma 4.3.6 *Kui* $n = pq$, *kus* p *ja* q *on algarvud, siis*

$$\lambda(n) = \operatorname{lcm}(p-1, q-1) = \frac{\phi(n)}{\gcd(p-1, q-1)}.$$

Proof: Algebrast on teada, et $\mathbb{Z}_{pq} \cong \mathbb{Z}_p \times \mathbb{Z}_q$, tõestame võrduse viimases ringis. Olgu k selline, et

$$\forall (a, b) \in \mathbb{Z}_p \times \mathbb{Z}_q, (a, b)^m = (a^m, b^m) = (1 \pmod p, 1 \pmod q).$$

Vähim m, mille korral $a^m \equiv 1 \mod p$, $\forall a \in \mathbb{Z}_p$, on $p - 1$; vähim m, mille korral $b^m \equiv 1 \mod q$, $\forall b \in \mathbb{Z}_q$, on $q - 1$. Vähim m, mille korral mõlemad kehtivad, on $\operatorname{lcm}(p-1, q-1)$. ∎

4.4 RSA definitsioon

Krüptosüsteem 4.4.1 (RSA) *Olgu* $n = pq$, *kus* p *ja* q *on algarvud. Olgu* $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$,

$$\mathcal{K} = \{(n, p, q, d, e) \colon n = pq,\ p, q \text{ on algarvud}, \\ \gcd(e, \lambda(n)) = 1,\ d := e^{-1} \mod \lambda(n)\}$$

$K = (n, p, q, d, e)$ korral defineerime

$$E_K(x) := x^e \mod n,$$
$$D_K(y) := y^d \mod n$$

($x, y \in \mathbb{Z}_n$). Siin n ja e on avalikud ning p, q ja d on privaatsed.

Kuna $d := e^{-1} \mod \lambda(n)$, siis ka $d := e^{-1} \mod \phi(n)$

Teoreem 4.4.2 *Suvalise $x \in \mathbb{Z}_n$ korral $x^{ed} \equiv x \mod n$.*

Proof:
Kuna $e \cdot d \equiv 1 \pmod{\lambda(pq)}$, siis leidub selline t, et $e \cdot d = 1 + t\lambda(pq)$.

1. $p \not| x$, $q \not| x$. Siis Euleri teoreemist $x^{\phi(n)} \equiv 1 \mod n$, seega $x^{ed-1} \equiv 1 \mod n$ ja $x^{ed} \equiv x \mod n$.
2. $p \mid x$, $q \not| x$. Siis $x^{q-1} \equiv 1 \mod q \Rightarrow x^{\phi(n)} \equiv \mod q \Rightarrow x^{j\phi(n)} \equiv 1 \mod q \Rightarrow x^{ed} \equiv x \mod q$. Kuna ka $x^{ed} \equiv x \mod p$, siis hiina jääkide teoreemist $x^{ed} \equiv x \mod n$.
3. $p \mid x$, $q \mid x$. Triviaalne.

■

Näide 4.4.3 *Üks lihtne näide lubamatult väikese mooduliga. Olgu $p = 113$ ja $q = 127$, siis $n = 14351$ ning $\lambda(n) = 1008$. Valime juhuslikult arvu $e = 5731$, nii et et täiesti juhuslikult ka $\gcd(e, 1008) = 1$ (viimast saab kontrollida Eukleidese algoritmiga). Selle sama Eukleidese algoritmi kõrvalnähuna saame kätte ka arvu $d := 3307 \equiv 5731^{-1} \pmod{1008}$.*

Seega Bob avalikustab arvud $n = 14351$ ja $e = 5731$. Oletame, et Alice tahab Bobile saata sõnumi CAT=$2 \cdot 32^2 + 0 \cdot 32 + 22 = 2070$, selleks arvutab ta, et $2070^{5731} \mod 14351 = 5815 = 5 \cdot 32^2 + 21 \cdot 32 + 23$ ja saadab Bobile teksti FŽU. Bob saab dešifreerides, et $5815^{3307} \mod 14351 = 2070$.

4.5 RSA realiseerimine

Olgu random(i, j) funktsioon, mis tagastab juhusliku arvu vahemikust $i \dots j$.
Võtme genereerimiseks tuleb

1. valida positiivne täisarv e (**avalik astendaja**).
2. salaja ja juhuslikult valida 2 erinevat paaritut algarvu p ja q, nii et $\gcd(p-1, e) = \gcd(q-1, e) = 1$.

3. arvutada avalik moodul $n := pq$.
4. $d := e^{-1} \pmod{\lambda(n)}$.

PKCS (*Public-Key Cryptography Standard*) näeb ette, et RSA salajane võti koosneb järgnevatest arvudest: n, e, d, p, q, $d \mod p-1$, $d \mod q-1$ ning $q^{-1} \mod p$. Tegelikult piisaks ainult n-ist ja d-ist, ülejäänud väärtused salvestatakse vaid efektiivsuse huvides (e olemasolu on vajalik ka selleks, et avaliku võtme genereerimine salajasest võtmest oleks lihtne).

Üks väga loomulik rünnak RSA pihta oleks faktoriseerida $n = pq$. See tehtud, oleks ülimalt lihtne arvutada arvu $\lambda(n) = \mathrm{lcm}(p-1, q-1)$ ning seejärel leida a, ehk teisisõnutsi, RSA lahtimurdmine on *polünomiaalselt taandatav* arvude faktoriseerimise ülesandele: polünomiaalse algoritmi olemasolu faktoriseerimisülesande jaoks tagab polünomiaalse algoritmi olemasolu ka RSA lahtimurdmisülesande jaoks. On oletatud, aga pole suudetud tõestada vastupidise väite kehtivust, seega võib leiduda meetodeid RSA lahtimurdmiseks, mis ei kasutada arvude faktoriseerimist.

Seega turvalisuse tagamiseks peab RSA moodul n olema piisavalt suur. Kaasaegsete faktoriseerimisalgoritmide abil on võimalik enam-vähem mõistlikus ajas faktoriseerida kuni 430-bitiseid arve (vt. paragrahv 6.2), mitmed RSA realiseeringud kasutavad 512-bitist moodulit, mis ei paku head pikaajalist turvalisust. Soovitatav oleks 1024-bitine moodul, eriti olulistel juhtudel ka pikem.

Märkus 4.5.1 *RSA turvalisuse tagamiseks tuleb päris palju vaeva näha, näiteks ei tohi d pikkus olla väiksem kui $\frac{1}{4}$ n-i pikkusest (sest sel juhul leiduks efektiivne rünnak). Et mitte selliseid pisiasju iga kord ise leiutada, on välja töötatudki PKCS, vaata [RSA91].*

Vaatleme nüüd lühidalt, kuidas teostada krüptimist ja dekrüptimist ehk arvu astendamist modulo n.

Algoritm 4.5.2 **($z := x^e \mod n$)** *Olgu $k := \lceil \log n \rceil$, $e = \sum_{i=0}^{k-1} e_i 2^i$ (arvu e kahendesitus). Meil on vaja leida $x^e \mod n$.*

```
funct modexp(x, e, n) ≡
  z := 1
  for i := k − 1 step −1 to 0 do
      z := z² mod n
      if e_i = 1 then z := z · x mod n fi
  od
  modexp := z
.
```

"Algkooli" algoritme kasutades saame k-bitist liitmist teostada ajas $O(k)$ ning korrutamist ajas $O(k^2)$[1]. $2k$-bitise arvu jagamisel k-bitise arvuga tekkivat

[1] Kavalamad algoritmid võimaldavad korrutamist teostada ajas $O(k^{1+\epsilon})$, suvalise $\epsilon > 0$ korral. Keegi pole suutnud tõestada või ümber lükata hüpoteesi, et korrutamist saab teostada ajas $O(k)$.

jääki saab samuti leida ajas $O(k^2)$, seega töötab antud algoritm ajas $O(k^3)$ ja samas ajas töötab ka RSA krüptimine ja dekrüptimine.

Märkus 4.5.3 *Paljudes rakendustes on e väärtus fikseeritud, tüüpiliselt kas $e = 3$ või $e = 65537$. Mõlemal juhtumil on tegu paaritu algarvuga, mille üheste bittide arv on minimaalne (2).*

Seega oleme leidnud RSA (de)krüptimise keerukuse. *Salajase astendaja d* leidmine on lihtne, selleks kasutatava Eukleidese algoritmi keerukus on Lamé teoreemi (1.2.8) põhjal $O((\log n)^3)$.

4.6 Randomiseeritud kordarvutestid

RSA krüptosüsteemi ülesseadmiseks on vajalik genereerida suuri (näiteks 100-kohalisi) algarvusid. Praktikas realiseeritakse see lihtsalt nii, et võetakse juhuslikult laest mingi suur arv ja siis kontrollitakse Monte Carlo algoritmi abil, kas antud arv on algarv. Sellised algoritmid võimaldavad meil seda testimist küll kiiresti läbi viia, ent võivad ka arvata, et x on algarv kui ta seda tegelikult pole. Samas, kui jooksutada antud algoritmi piisavalt suur arv kordi järjest, muutub vea tõenäosus ka piisavalt väikseks.

Oluline küsimus on ka see, kui paljusid (fikseeritud pikkusega) arve peab enne algarvu leidmist testima. Kuulus algarvude teoreem ütleb meile, et N-st väiksemate algarvude arv on umbes $\frac{N}{\ln N}$ ning seega juhuslikult valitud p on algarv tõenäosusega $\frac{1}{\ln p}$. Kui me kasutame 512-bitiseid mooduleid siis $\frac{1}{\ln p} \approx \frac{1}{177}$, seega oleks keskmiselt iga 177-s nii pikkadest arvudest algarv (kuid kui me näiteks genereerime ainult paarisarve, siis tõenäosus kahekordistub). Seega selliste "tõenäoliselt algarve" saab ökonoomselt genereerida.

Anname siinkohal ka Monte Carlo algoritmide formaalse definitsiooni.

Definitsioon 4.6.1 *Randomiseeritud (st. juhuarve kasutavat) algoritmi nimetatakse "jah"* ***Monte Carlo algoritmiks*** *veatõenäosusega $0 < \epsilon < 1$ kui selle algoritmi vastus "jah" on korrektne, ent vastus "ei" võib olla ebakorrektne tõenäosusega (üle algoritmi tehtud kõikvõimalike juhuslike valikute, kui talle on ette antud mingi kindel sisend), mis pole suurem kui ϵ. Duaalselt defineeritakse "ei" Monte Carlo algoritm.*

Toome esmalt ära ühe deterministliku testi.

Teoreem 4.6.2 (Wilsoni teoreem) *Täisarv p on algarv parajasti siis, kui $(p-1)! \equiv -1 \pmod{p}$.*

Proof: Oletame esiteks, et p on algarv. Juhtum $p = 2$ on triviaalne, olgu $p > 2$. Kuna $\mathbb{Z}_p$ on korpus, siis iga tema nullist erinev element on pööratav (modulo p) ja võrrandil $x^2 \equiv 1 \pmod{p}$ on ainult kaks lahendit: ± 1. Seega suvalisel arvul a, $1 < a < p-1$, on pöördarv modulo p, mis on erinev arvust endast. Sättides iga arvu paari tema pöördarvuga, saame

$$k := 2 \cdot 3 \cdots (p-2) \equiv 1 \pmod{p},$$

```
funct fermattest(n)
  a := random(0, n − 1)
  if gcd(a, n) > 1 ∨ a^{n−1} ≢ 1  mod n
    then
          fermattest := composite
     else
          fermattest := probably prime
  fi.
```

Joonis 4.1: **Fermat' test**

ja seega $(p-1)! = 1 \cdot k \cdot (p-1) \equiv 1 \cdot 1 \cdot -1 \equiv -1 \pmod{p}$.

Oletame, et p ei ole algarv, seega $\mathbb{Z}_p$ ei ole korpus ning järelikult leiduvad $a, b \in \mathbb{Z}_p \setminus \{0\}$, $a \cdot b \equiv 0 \pmod{p}$. Vaatleme korrutist $(p-1)! \cdot (p-1)! \mod p$. Kuna a esineb selle korrutise vasakus teguris ja b — paremas, siis $(p-1)! \cdot (p-1)! \equiv 0 \pmod{p}$. Kuna $0 \not\equiv 1 = (-1)^2 \pmod{p}$, siis oleme teoreemi tõestanud. ■

Siit saame arvu algarvulisuse tõestamiseks järgmise testi. Olgu $n \in \mathbb{Z}$; kui $(n-1)! \equiv -1 \pmod{n}$, siis on n algarv, vastasel juhul on n kordarv. Paneme tähele, et see test ütleb meile ainult, kas n on kordarv või mitte, väljastamata tema tegureid.

See test on praktikas aeglane: me peame korrutama $n-2$ korda, mis on tegelikult aeglasem kui läbijagamine (vt. peatükk 6.2).

4.6.1 Fermat' test

Fermat' test on toodud joonisel 4.1.

Fermat' väikesest teoreemist (teoreem A5.4) teame, et vastus "n on kordarv" on igal juhul tõene. Kuid näiteks $a^{560} \equiv 1 \mod 561$, suvalise $a < 561$, $\gcd(a, 561) = 1$ korral, kuid 561 on ometigi kordarv: $561 = 3 \cdot 11 \cdot 17$ ning seega pole tegu Monte Carlo algoritmiga.

Definitsioon 4.6.3 *Kordarvu n nimetatakse* ***Carmichael'i arvuks,*** *kui $\forall a < n, \gcd(a, n) = 1 \Rightarrow a^{n-1} \equiv 1 \pmod{n}$.*

561 on väikseim Carmichaeli arv, on tõestatud, et selliseid arve on lõpmatult palju. Ent proovime meie täbarat olukorda natuke leevendada.

Definitsioon 4.6.4 *Sümboliga $\mathrm{QR}(p)$ tähistame kõigi ruutjääkide hulka modulo p.*

Definitsioon 4.6.5 *Olgu p — algarv, $a \in \mathbb{Z}$ ning $a' := a \pmod{p}$.* ***Legendre'i sümbol***

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & p \mid a \\ 1, & p \not| a \wedge a' \in \mathrm{QR}(p) \\ -1, & p \not| a \wedge a' \notin \mathrm{QR}(p). \end{cases}$$

Definitsioon 4.6.6 *Olgu $n := \prod p_i^{k_i}, a \in \mathbb{Z}$. **Jacob'i sümbol***

$$\left(\frac{a}{n}\right) := \prod \left(\frac{a}{p_i}\right)^{k_i},$$

kus paremal pool on Legendre'i sümbol.

Lemmast 4.3.3 saame, et võrrandil $x^2 = a \mod p$ on ülimalt kaks lahendit ning peatüki A9 algusest saame, et tegelikult on teid kas 0 või 2 ning viimast parajasti siis kui $\left(\frac{a}{p}\right) = 1$ ehk (lemmast A9.1) parajasti siis kui $a^{\frac{p-1}{2}} \equiv 1 \pmod p$.

Algoritm 4.6.7 (Solovay-Strasseni test) *Kas paaritu n on kordarv?*

<u>**funct**</u> *sstest*(n)
  $a := \mathrm{random}(2, n-1)$
  <u>**if**</u> $\gcd(a,n) > 1 \lor \left(\frac{a}{n}\right) \not\equiv a^{(n-1)/2} \pmod n$
    <u>**then**</u>
        sstest := *composite*
    <u>**else**</u>
        sstest := *probably prime*
  <u>**fi**</u>.

Vaatleme esiteks küsimust, kas see algoritm toimib polünomiaalses ajas? Me oskame $a^{\frac{n-1}{2}} \mod n$ välja arvutada ajas $O((\log n)^3)$. Näitame ka, et Jacobi sümboli arvutamine on efektiivne.

Toome siia neli taandust, millest kaks on ekvivalentsed arvuteooria konspektis tõestatud lemmadega (ent on arvutuslikult lihtsamalt sõnastatud) ning veel kaks, mis tulenevad triviaalselt lauses A9.2 toodud Legendre'i sümboli omadustest ja Jacobi sümboli definitsioonist.

Olgu lsz() funktsioon, mis tagastab suurima kahe astme, millega antud arv jagub.

1. Kui n on paaritu ning $m_1 \equiv m_2 \pmod n$, siis
$$\left(\frac{m_1}{n}\right) = \left(\frac{m_2}{n}\right).$$

2. Kui n on paaritu, siis
$$\left(\frac{2}{n}\right) = \begin{cases} 1, & \text{kui } n \equiv \pm 1 \pmod 8, \\ -1, & \text{kui } n \equiv \pm 3 \pmod 8, \end{cases}$$
(samaväärne lemmaga A9.6).

3. Kui n on paaritu, siis

$$\left(\frac{m_1 m_2}{n}\right) = \left(\frac{m_1}{n}\right)\left(\frac{m_2}{n}\right).$$

Seejuures kui $k := \operatorname{lsz}(m)$, $k := \frac{m}{2^k}$, siis

$$\left(\frac{m}{n}\right) = \left(\frac{2}{n}\right)^k \left(\frac{t}{n}\right).$$

4. Kui m ja n on paaritud, siis

$$\left(\frac{m}{n}\right) = \begin{cases} -\left(\frac{n}{m}\right), & \text{kui } m \equiv n \equiv 3 \pmod 4, \\ \left(\frac{n}{m}\right), & \text{vastasel juhul.} \end{cases}$$

(samaväärne lemmaga A9.7).

Näide 4.6.8 *Olgu $a = 2070$, $n = 4323$. Siis*

$$\begin{aligned}\left(\frac{2070}{4323}\right) &= \left(\frac{2}{4323}\right)\left(\frac{1035}{4323}\right) = -\left(\frac{1035}{4323}\right) = \left(\frac{4323}{1035}\right) = \\ &= \left(\frac{183}{1035}\right) = -\left(\frac{1035}{183}\right) = -\left(\frac{120}{183}\right) = \\ &= -\left(\frac{2}{183}\right)^3\left(\frac{15}{183}\right) = -\left(\frac{15}{183}\right) = \left(\frac{183}{15}\right) = \left(\frac{3}{15}\right) = -\left(\frac{15}{3}\right) = 0.\end{aligned}$$

Ainsad kasutatavad operatsioonid siinjuures on aritmeetilised tehted modulo n ning jagamine arvu 2 astmetega. $\operatorname{lsz}(n)$ leidmine on teostatav ajas $O(|n|)$. Ei ole raske näidata, et tehteid on $O(\log n)$ ($\left(\frac{m}{n}\right)$ arvutamisalgoritm on enam-vähem analoogiline Eukleidese algoritmiga) ning kuna iga tehe on teostatav ajas $O((\log n)^2)$, siis saab Jacobi sümbolit välja arvutada ajas $O((\log n)^3)$. Ja jällegi, natuke täpsem analüüs annaks meile $O((\log n)^3)$ asemel $O((\log n)^2)$.

Lausest A9.1 saame, et kui n on algarv, siis

$$\left(\frac{a}{n}\right) \equiv a^{\frac{n-1}{2}} \pmod n. \tag{4.1}$$

Samas kui n on kordarv, võib (4.1) ikkagi veel kehtida.

Definitsioon 4.6.9 *Selliseid kordarve a, mille korral kehtib kongruents (4.1), nimetatakse **Euleri libaalgarvudeks** (*Euler pseudo-prime to the base n*) modulo n.*

Nii näiteks $\left(\frac{10}{91}\right) = -1 = 10^{45} \bmod 91$ ja seega on 10 Euleri libaalgarv modulo 91.

Lemma 4.6.10 *Olgu n paaritu. Kui $\left(\frac{m}{n}\right) \equiv m^{\frac{n-1}{2}} \pmod n$ iga $m \in \mathbb{Z}_n^*$ korral, siis on n algarv.*

Proof: Oletame vastuväiteliselt, et $\left(\frac{m}{n}\right) \equiv m^{\frac{n-1}{2}} \pmod n$ iga $m \in \mathbb{Z}_n^*$ korral ning n on kordarv.

Vaatleme esmalt juhtu, kus n on paarikaupa erinevate algarvude korrutis: $n = p_1 \dots p_k$. Olgu $r \in \mathbb{Z}_{p_1}^*$ selline, et $\left(\frac{r}{p_1}\right) = -1$ (miks selline leidub?). Siis saame Hiina jääkide teoreemist (teoreem 4.3.4), et leidub selline $m \in \mathbb{Z}_n^*$, et $m \equiv r \pmod{p_1}$ ning $m \equiv 1 \pmod{p_i}$, $2 \leq i \leq k$. Eelduse kohaselt $m^{\frac{n-1}{2}} \equiv \left(\frac{m}{n}\right) = \left(\frac{r}{p_1}\right)\left(\frac{1}{p_2}\right)\dots\left(\frac{1}{p_k}\right) \equiv -1 \pmod n$. Vaadeldes viimast võrrandit modulo p_2, saame $1 = 1^{\frac{n-1}{2}} \equiv m^{\frac{n-1}{2}} \equiv -1 \pmod{p_2}$, vastuolu (kuna n on paaritu).

Seega $n = p^l n_1$, kus $p > 2$ on algarv, $p \not| n_1$ ja $l \geq 2$. Olgu r algjuur modulo p^2 (selline leidub tänu teoreemile A7.5). Kuna $p \not| n_1$, siis $\gcd(p^2, n_1) = 1$ ning Hiina jääkide teoreemi põhjal leidub selline $r_1 \in \mathbb{N}$, et $r_1 \equiv r \pmod{p^2}$ ja $r_1 \equiv 1 \pmod{n_1}$. Siis ka r_1 on algjuur modulo p^2. Kuna r_1 on algjuur modulo p^2, siis $\gcd(r_1, p^2) = 1$ ja seega $p \not| r_1$. Et $r_1 \equiv 1 \pmod{n_1}$, siis leidub selline $u \in \mathbb{Z}$, et $r_1 + un_1 = 1$. Olgu $d := \gcd(r_1, n_1)$, saame et $d \mid 1$ ja järelikult $d = 1$. Kuna $\gcd(r_1, p^l) = 1$ ja $\gcd(r_1, n_1) = 1$, siis lemma A4.1 põhjal $\gcd(r_1, p^l n_1) = \gcd(r_1, n) = 1$. Seega $r_1 \in \mathbb{Z}_n^*$. Meie hüpoteesi põhjal $r_1^{\frac{n-1}{2}} \equiv \left(\frac{r}{n}\right) \pmod n$, seega $(r_1)^{n-1} \equiv 1 \pmod n$ ja järelikult ka $(r_1)^{n-1} \equiv 1 \pmod{p^2}$. Kuna r_1 on algjuur modulo p^2, siis tema järk rühmas $\mathbb{Z}_{p^2}^*$ on $\varphi(p^2) = p(p-1)$. Lemma A7.1 tõttu $p(p-1) \mid n-1$. Seega $p \mid n$ ja $p \mid n-1$, mis on võimatu.

■

Teoreem 4.6.11 *Kui n on paaritu kordarv, siis vähemalt poolte $m \in \mathbb{Z}_n^*$ puhul $\left(\frac{m}{n}\right) \not\equiv m^{\frac{n-1}{2}} \pmod n$.*

Proof: Eelmisest lemmast saame, et leidub vähemalt üks $a \in \mathbb{Z}_n^*$, nii et $\left(\frac{a}{n}\right) \not\equiv a^{\frac{n-1}{2}} \pmod n$. Olgu $B = \{b_1, \dots, b_k\} \subseteq \mathbb{Z}_n^*$ kõikvõimalike erinevate jäägiklasside hulk, mille korral $\left(\frac{b_i}{n}\right) \equiv b_i^{\frac{n-1}{2}} \pmod n$. Vaatleme hulka $a \cdot B = \{ab_1, \dots, ab_k\}$. Kui kehtiks $ab_i = ab_j$ mingite $i \neq j$ jaoks, siis $n \mid a(b_i - b_j)$, mis on võimatu (kuna $a \in \mathbb{Z}_n^*$ ning $0 < |b_i - b_j| < n$). Seega on arvud ab_i paarikaupa erinevad. Samuti $ab_i \notin B$, kuna $(ab_i)^{\frac{n-1}{2}} = a^{\frac{n-1}{2}} b_i^{\frac{n-1}{2}} \not\equiv \left(\frac{a}{n}\right)\left(\frac{b_i}{n}\right) = \left(\frac{ab_i}{n}\right) \pmod n$ ning seega oleme teoreemi tõestanud ja Solovay-Strasseni algoritm on Monte Carlo tüüpi algoritm veatõenäosusega $\frac{1}{2}$. ■

Lemma 4.6.12 ([Sti95], lk. 135) *Solovay-Strasseni algoritmi m korda käivitades saadakse "jah" Monte Carlo algoritm veatõenäosusega $\frac{\ln n - 2}{\ln n - 2 + 2^{m+1}}$.*

4.6.2 Miller-Rabini algoritm

Miller-Rabini test on toodud joonisel 4.2. Pole raske veenduda, et ka selle testi keerukus on $O((\log n)^3)$ (!), ent praktikas töötab Miller-Rabini test paremini. Meenutagem, et arvu x nimetati **ühejuureks** modulo n kui $x^2 \equiv 1 \pmod n$.

Teoreem 4.6.14 *Miller-Rabini algoritm on "jah" Monte Carlo algoritm.*

Algoritm 4.6.13 (Miller-Rabin) *Kas paaritu n on kordarv?*

```
funct mrtest(n) ≡
  k := lsz(n − 1)
  m := (n − 1)/2^k
  a := random(1, n − 1)
  b := a^m mod n
  if b ≡ 1 (mod n)
    then
          mrtest := probably prime
          exit
  fi
  for i := 0 to k − 1 do
      if b ≡ −1 (mod n)
        then
              mrtest := probably prime
              exit
        else
              b := b^2 mod n
      fi
  od
  mrtest := composite
```

Joonis 4.2: Miller-Rabini algoritm

Proof:

Oletame, et algoritm vastab "n on kordarv" algarvulise n korral ning saame sealt vasturääkivuse. Kuna algoritm vastab "n on kordarv," siis peab kehtima $a^m \not\equiv 1 \pmod{n}$. Vaatleme muutuja b erinevaid väärtusi tsükli sees: a^m, a^{2m}, $\ldots$, $a^{2^{k-1}m}$. Seega kuna algoritm vastab "n on kordarv," siis $a^{2^i m} \not\equiv -1 \pmod{n}$, $0 \leq i \leq k-1$. Kuna n on algarv, siis teoreemist A5.4 $a^{2^k m} = a^{n-1} \equiv 1 \pmod{n}$ ja seega on $a^{2^{k-1}m}$ ühejuur modulo n. Kuna n on algarv, siis on olemas ainult 2 ühejuurt modulo n, nimelt ± 1. Kuna $a^{2^{k-1}m} \not\equiv -1 \pmod{n}$, siis $a^{2^{k-1}m} \equiv 1 \pmod{n}$ ja seega peab $a^{2^{k-2}m}$ olema ühejuur ning analoogiliselt $a^{2^{k-2}m} \equiv 1 \pmod{n}$. Jne jne, kuni lõpuks saame, et $a^m \equiv 1 \pmod{n}$, vastuolu. ■

Saab näidata, et Miller-Rabini algoritmi veatõenäosus ei ületa $\frac{1}{4}$ (!).

Peatükk 5

RSA krüptoanalüüs

Kui krüptoanalüütik teaks lisaks arvule $n = pq$ ka väärtust $\phi(n) = (p-1)(q-1)$, siis oleks tal lihtne faktoriseerida arvu n : asendades teise võrrandisse $q = \frac{n}{p}$, saame p suhtes ruutvõrrandi:

$$p^2 - (n - \phi(n) + 1)p + n = 0,$$

selle võrrandi lahenditeks on p ja q. Seega $\phi(n)$-i teadmisest piisab n-i faktoriseerimiseks.

5.1 Salajase astendaja abil faktoriseerimine

Tõestame järgnevas ära ühe väga huvitava tulemuse: algoritmi, mis leiab salajase astendaja d, saab kasutada alamprogrammina (*oraaklina*) arvu n randomiseeritud faktoriseerimisalgoritmis ning seega ei ole d leidmine lihtsam kui arvu n faktoriseerimine (siiski jääb alles võimalus, et RSA-d saab lahti murda ka astendajat d arvutamata).

See tulemus näitab, et teades salajast võtit d saame faktoriseerida arvu n ning seega d avalikuks tulemisel peab Bob genereerima uue n-i.

Definitsioon 5.1.1 *Olgu $0 \leq \epsilon < 1$, reaalarv. Randomiseeritud algoritmi nimetatakse **Las Vegase algoritmiks** kui suvalise sisendi korral algoritm võib tõenäosusega ϵ anda vastuse "ei tea", samas iga teine vastus peab olema tõene.*

Mingile probleemile Las Vegase algoritmi abil vastuse leidmiseks tuleb seda algoritmi seni jooksutada, kuni saame vastuse. Tõenäosus, et me ei tea vastust pärast algoritmi m-kordset jooksutamist, on ϵ^m ning "keskmine arv kordi", mis me seda algoritmi jooksutama peame, et vastust saada, on $\frac{1}{1-\epsilon}$.

Olgu A algoritm, mis leiab salajase astendaja. Joonisel 5.1 on toodud A-d oraaklina kasutav Las Vegase algoritm. Algoritmi *lvfactor* puhul $\epsilon = \frac{1}{2}$ ning seega seda algoritmi m korda jooksutades saame arvu n faktoriseeritud tõenäosusega vähemalt $1 - 2^{-m}$.

$\underline{\textbf{funct}}$ $\mathit{lvfactor}(n)$
  $w := \mathrm{random}(1, n-1);\ x := \gcd(w, n)$
  $\underline{\textbf{if}}$ $1 < x < n$ $\underline{\textbf{then}}$ $\mathit{lvfactor} := x;$ $\underline{\textbf{exit}}$ $\underline{\textbf{fi}}$
  $d := A(e);$
  $s := \mathrm{lsz}(de-1);\ r := (de-1)/2^s$
  $v := w^r \mod n$
  $\underline{\textbf{if}}$ $v \equiv 1 \pmod n$ $\underline{\textbf{then}}$ $\mathit{lvfactor} := \mathit{don't\ know};$ $\underline{\textbf{exit}}$ $\underline{\textbf{fi}}$
  $\underline{\textbf{while}}$ $v \not\equiv 1 \pmod n$ $\underline{\textbf{do}}$ $v_0 := v;\ v := v^2 \mod n;$ $\underline{\textbf{od}}$
  $\underline{\textbf{if}}$ $v_0 \equiv -1 \pmod n$
    $\underline{\textbf{then}}$ $\mathit{lvfactor} := \mathit{don't\ know}$
     $\underline{\textbf{else}}$ $\mathit{lvfactor} := \gcd(v_0 + 1, n)$
  $\underline{\textbf{fi}}$.

Joonis 5.1: Las Vegase algoritm arvu n faktoriseerimiseks

See algoritm kasutab ühejuurte $\pmod n$ teatud kindlaid omadusi. Nimelt, on võrrandil $x^2 \equiv 1 \pmod p$ kaks lahendit, $x = \pm 1 \mod p$. Analoogilselt on kaks lahendit ka võrrandil $x^2 \equiv 1 \pmod q$, $x = \pm 1 \mod q$. Kuna aga $x^2 \equiv 1 \pmod n$ parajasti siis kui $x^2 \equiv 1 \pmod q$ ja $x^2 \equiv 1 \pmod p$, siis $x^2 \equiv 1 \pmod n$ parajasti siis kui $x = \pm 1 \mod p$ ja $x = \pm 1 \mod q$. Seega modulo n on olemas neli ühejuurt ja need kõik saab kätte, kasutades hiina jääkide teoreemi. Kaks lahendit on $x = \pm 1 \mod n$ (**triviaalsed ühejuured**, kahte ülejäänut nimetatakse **mittetriviaalseteks**, needki on teineteise vastandarvud.

Näide 5.1.2 *Olgu jällegi $p = 113$ ja $q = 127$, siis $n = 14351$. Tahame leida ühejuuri modulo n. Hiina jääkide teoreemi (teoreem 4.3.4) tõestuses toodud konstruktsiooni põhjal $m_1 = 127$ ning $k_1 = 105$, $m_2 = 113$ ning $k_2 = 9$, seega $x = 13335x_1 + 1017x_2$, kus $(x_1, x_2) = (\pm 1, \pm 1)$. Siit saame, et $x \in \{\pm 1, \pm 2033\}$.*

Kui x on mittetriviaalne ühejuur modulo n, siis $n \mid (x-1)(x+1)$, kuid $n \not| x \pm 1$. Siit järeldub, et $\gcd(x \pm 1, n) \in \{p, q\}$. Suurima ühisteguri arvutamiseks kasutame Eukleidese algoritmi ning järelikult teades mittetriviaalset ühejuurt modulo n, oskame me arvu n ka faktoriseerida (see tähtis fakt on paljude krüptograafiliste tulemuste aluseks). Ülaltoodud näites

$$\gcd(2034, 14351) = 113$$

ning

$$\gcd(2032, 14351) = 127.$$

Meie algoritm püüabki leida mittetriviaalseid ühejuuri modulo n. Kõigepealt üks pisike näide.

Näide 5.1.3 *Olgu $n = 14351$, $e = 5731$ ning $d = 3307$. Valime $w = 5689$. Siis $x = 1$, $s = 5$, $r = 592263$, $v = 7619$. Tsüklis oleks v järjestikkused väärtused 7619, 13717, 128, 2033, 1. Seega $p = \gcd(2034, 14351) = 113$.*

Analüüsigem nüüd meie algoritmi. Esiteks, kui $p \mid w$ või $q \mid w$, siis algoritm annab vastuse kohe alguses. Edasises oletame, et $\gcd(w, n) = 1$. Sel juhul arvutame $w^r, w^{2r}, w^{4r}, \ldots$, kuni lõpuks $w^{2^t r} \equiv 1 \pmod{n}$ mingi t korral. Kuna $2^s r = ed - 1 \equiv 0 \pmod{\phi(n)}$, siis $w^{2^s r} \equiv 1 \pmod{n}$ ning järelikult itereeritakse meie tsüklis maksimaalselt s korda. **while** tsükli lõpuks oleme leidnud sellise väärtuse v_0, et $v_0^2 \equiv 1 \pmod{n}$,aga $v_0 \not\equiv 1 \pmod{n}$. Kui $v_0 \equiv -1 \pmod{n}$, siis meie algoritm annab vastuseks "ei tea", vastasel juhul on v_0 mittetriviaalne ühejuur ja saame arvu n faktoriseerida.

Seega jääb üle veel näidata, et algoritm "teab" vastust tõenäosusega $\geq \frac{1}{2}$. Algoritm võib osutuda "ignorantseks" kahel juhul:

1. $w^r \equiv 1 \pmod{n}$
2. $w^{2^t r} \equiv -1 \pmod{n}$ mingi $0 \leq t \leq s-1$ jaoks,

(nn. "halvad juhud"). Kui juhuslik muutuja w on lahendiks vähemalt ühele neist $s+1$ kongruentsist, siis algoritm on ignorant. Loetlemegi järgnevalt, mitu lahendit on igal ülaltoodud kongruentsil.

Esiteks, kongruents $w^r \equiv 1 \pmod{n}$. Selleks vaatleme lahendeid modulo p ja modulo q eraldi ning kombineerime need siis, kasutades hiina jääkide teoreemi ($x \equiv 1 \pmod{n}$ parajasti siis kui $x \equiv 1 \pmod{p}$ ja $x \equiv 1 \pmod{q}$).

Vaatleme kongruentsi $x \equiv 1 \pmod{p}$. Kuna p on algarv, siis $U(\mathbb{Z}_p)$ on tsükliline rühm, olgu g selle rühma moodustaja ning $w = g^u$, $0 \leq u \leq p-1$. Siis $g^{ur} \equiv 1 \pmod{p}$ ning seega $p-1 \mid ur$. Olgu $p-1 = 2^i p_1$, kus p_1 on paaritu, ning analoogiliselt $q-1 = 2^j q_1$, q_1 on paaritu. Kuna $\phi(n) = (p-1)(q-1) \mid (ed-1) = 2^s r$, siis $2^{i+j} p_1 q_1 \mid 2^s r$. Seega $i + j \leq s$ ning $p_1 q_1 \mid r$. Kuna $p-1 = 2^i p_1$, siis $2^i p_1 \mid ur$. Kuna $p_1 \mid r$ ja r on paaritu, siis $2^i p_1 \mid ur$ parajasti siis kui $2^i \mid u$. Seega $u = k 2^i$, $k \in \mathbb{Z}_{p_1}$ ning kongruentsil $w^r \equiv 1 \pmod{p}$ on p_1 lahendit. Analoogiliselt on kongruentsil $w^r \equiv 1 \pmod{q}$ q_1 lahendit. Nende kahe kongruentsi lahendeid saab kombineerida, kasutades hiina jääkide teoreemi, iga lahendipaari korral saame ühese lahendi kongruentsile $w^r \equiv 1 \pmod{n}$ ning seega neid lahendeid on $p_1 q_1$.

Teiseks, vaatleme kongruentsi $w^{2^t r} \equiv -1 \pmod{n}$ fikseeritud $t \in \mathbb{Z}_s$ korral. Jällegi, vaatleme esmalt kongruentse modulo p ning modulo q (paneme tähele, et $w^{2^t r} \equiv -1 \pmod{n}$ parajasti siis kui $w^{2^t r} \equiv -1 \pmod{p}$ ja $w^{2^t r} \equiv -1 \pmod{q}$). Olgu siis $w^{2^t r} \equiv -1 \pmod{p}$ ning $w = g^u$ (nagu ennegi), järelikult $g^{u 2^t r} \equiv -1 \pmod{p}$. Kuna $g^{\frac{p-1}{2}} \equiv -1 \pmod{p}$, siis $u 2^t r \equiv \frac{p-1}{2} \pmod{p-1} \Rightarrow p-1 \mid \left(u 2^t r - \frac{p-1}{2}\right) \Rightarrow 2(p-1) \mid \left(u 2^{t+1} r - (p-1)\right)$, ehk $2^{i+1} p_1 \mid \left(u 2^{t+1} r - 2^i p_1\right)$, ehk $2^{i+1} \mid \left(\frac{u 2^{t+1} r}{p_1} - 2^i\right)$. Kui $t > i$, siis lahendid puuduvad, kuna $2^{i+1} \mid 2^{t+1}$, kuid $2^{i+1} \not| 2^i$.

Teiselt poolt, kui $t \leq i-1$, siis u on lahend parajasti siis kui $u = a \cdot 2^{i-t-1}$, mingi paaritu a korral (kuna $\frac{r}{p_1}$ on paaritu arv). Seega sel juhul on lahendeid $\frac{p-1}{2^{i-t-1}} \times \frac{1}{2} = 2^t p_1$. Analoogiliselt saame, et kongruentsil $w^{2^t r} \equiv -1 \pmod{q}$ pole lahendeid, kui $t \geq j$ ning on $2^t q_1$ lahendit kui $t \leq j-1$. Hiina jääkide teoreemist saame, et kongruentsil $w^{2^t r} \equiv -1 \pmod{n}$ pole lahendeid, kui $t \geq \min\{i, j\}$,

```
funct d-half(y) ≡
  for u := 0 to k do
      y_i := half(y)
      y := (y · e(2))  mod n
  od
  lo := 0
  hi := n
  for i := 0 to k do
      mid := (hi+lo)/2
      if y_i = 1
        then lo := mid
         else hi := mid
      fi
  od
  d-half := ⌊hi⌋
```

Joonis 5.2: $d(y)$ leidmine $half(y)$ olemasolul

ning on $2^{2t}p_1q_1$ lahendit, kui $t \leq \min\{i,j\} - 1$. Oletame ü.k., et $i \leq j$, seega lahendeid pole, kui $t \geq i$.

Juhusliku arvu w "halbu valikuid" kokku on $p_1q_1 + p_1q_1(1 + 2^2 + 2^4 + \cdots + 2^{2i-2}) = p_1q_1\left(1 + \frac{2^{2i}-1}{3}\right) = p_1q_1\left(\frac{2}{3} + \frac{2^{2i}}{3}\right)$. Kuna $j \geq i \geq 1$, siis $p_1q_1 < \frac{n}{4}$. Samuti, $2^{2i}p_1q_1 \leq 2^{i+j}p_1q_1 = (p-1)(q-1) < n$, ning seega $p_1q_1\left(\frac{2}{3} + \frac{2^{2i}}{3}\right) < \frac{n}{6} + \frac{n}{3} = \frac{n}{2}$.

Kuna ülimalt $\frac{n-1}{2}$ w valikut on halvad, siis algoritm pole "ignorantne" tõenäosusega $\geq \frac{1}{2}$.

5.2 Osaline informatsioon

Oletame, et RSA on turvaline. Ent kas me saame teda "osaliselt" lahti muukida? Paar näidet lekkida võiva osalise informatsiooni kohta:

1. Leida parity$(e(x))$, kus parity$(x) := x \mod 2$ (ehk LSB).

2. Leida half$(e(x))$, kus half$(x) = 0$, kui $0 \leq x < n/2$ ning half$(y) = 1$, kui $n/2 < x \leq n-1$.

Teoreem 5.2.1 *Arvutagu algoritm half suurust* half(x). *Siis leidub polünomiaalne algoritm, mis leiab $d(y)$ polünomiaalses ajas, kasutades algoritmi half oraaklina.*

Proof: Algoritm ise on toodud joonisel 5.2.

Esimese tsükli sees arvutame $y_i :=$ half$(y \cdot (e(2))^i) =$ half$(e(x \cdot 2^i))$, $0 \leq i \leq \log_2 n$.

Paneme tähele, et

$$\begin{aligned}
\mathrm{half}(e(x)) = 0 &\Leftrightarrow x \in [0, n/2) \\
\mathrm{half}(e(2x)) = 0 &\Leftrightarrow x \in [0, n/4) \cup [n/2, 3n/4) \\
\mathrm{half}(e(4x)) = 0 &\Leftrightarrow x \in [0, n/8) \cup [n/4, 3n/8) \cup [n/2, 5n/8) \cup [3n/4, 7n/8)
\end{aligned}$$

jne, seega saab $d(y)$ leida binaarse otsingu abil. Seda viimast teises tsüklis tehaksegi.

■

Lemma 5.2.2 $\mathrm{half}(y)$ *ja* $\mathrm{parity}(y)$ *on võrdrasked ülesanded.*

Proof: Tõestus tuleneb järgnevast kahest võrdusest: $\mathrm{half}(y) = \mathrm{parity}(y \cdot e(2) \mod n)$ ja $\mathrm{parity}(y) = \mathrm{half}(y \cdot e(2^{-1}) \mod n)$. ■

5.3 Rabini krüptosüsteem

Vaatle siinkohas ka **Rabini krüptosüsteemi** (joonis 5.3), mis on arvutuslikult turvaline valitud avateksti ründe vastu eeldusel , et $n = pq$ faktoriseerimine on raske.

Krüptosüsteem 5.3.1 (Rabini krüptosüsteem) *Olgu* n *kahe erineva algarvu* p *ja* q *korrutis, kus* $p, q \equiv 3 \pmod 4$, $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$ *ning* $\mathcal{K} := \{(n, p, q, B) \colon 0 \le B \le n-1\}$. $K = (n, p, q, B)$ *korral defineerime*

$$e_K(x) := x(x+B) \mod (n)$$

ning

$$d_K(y) := \sqrt{\frac{B^2}{4} + y} - \frac{B}{2}.$$

Väärtused n *ja* B *on avalikud,* p *ja* q *on salajased.*

Joonis 5.3: Rabini krüptosüsteem

Olgu ω üks neljast võimalikust arvu 1 ruutjuurest modulo n ning $x \in \mathbb{Z}_n$. Siis (tehted rühmas $\mathbb{Z}_n$, sealjuures $\frac{1}{x} := x^{-1} \mod n$)

$$\begin{aligned}
e_K\left(\omega\left(x + \frac{B}{2}\right) - \frac{B}{2}\right) &= \left(\omega\left(x + \frac{B}{2}\right) - \frac{B}{2}\right)\left(\omega\left(x + \frac{B}{2}\right) + \frac{B}{2}\right) \\
&= \omega^2\left(x + \frac{B}{2}\right)^2 - \left(\frac{B}{2}\right)^2 \\
&= x^2 + Bx + \frac{B^2}{4} - \frac{B^2}{4} \\
&= x^2 + Bx \\
&= e_K(x).
\end{aligned}$$

ning seega ei ole e_K injektsioon ning me ei saa dekrüptimist teostada üheselt, neli avateksti (x, $-x-B$, $\omega\left(x+\frac{B}{2}\right)-\frac{B}{2}$ ning $-\omega\left(x+\frac{B}{2}\right)-\frac{B}{2}$, kus ω on arvu 1 mittetriviaalne ruutjuur modulo n) krüptuvad üheks ja samaks krüptogrammiks $e_K(x)$ ning ei leidu meetodit, mis võimaldaks Bobil üldjuhul kindlaks teha, missugune antud neljast avatekstist on just see "õige".

Dekrüptimisel teab Bob krüptogrammi y ja ta ülesandeks on lahendada ruutvõrrand $x^2+Bx \equiv y \pmod n$ muutuja x suhtes. Siin lineaarne liidetav elimineerub, kui me asendame $x_1 := x+B/2$, saades võrrandi $x_1^2 \equiv \frac{B^2}{4}+y \pmod n$. Defineerides järgnevalt $C := \frac{B^2}{4}+y$, saame ruutvõrrandi $x_1^2 \equiv C \pmod n$ ning dekrüptimine taandub modulo n ruutjuurte leidmisele, mis on omakordne samaväärne kahe kongruentsi $x_1^2 \equiv C \pmod p$ ja $x_1^2 \equiv C \pmod q$ lahendamisega. Kuna arvul C on kaks ruutjuurt modulo p ja kaks ruutjuurt modulo q, siis hiina jääkide teoreemist saame 4 ruutjuurt modulo n. Euleri kriteeriumi[1] kasutades saame kindlaks teha, kas $C \in \mathrm{QR}(p) \cap \mathrm{QR}(q)$ (muidugi, $C \in \mathrm{QR}(p) \cap \mathrm{QR}(q)$ juhul, kui krüptiti korrektselt). Ent Euleri kriteerium ei aita meid ruutjuurte leidmisel, sest me saame vastuseks vaid "ja" või "ei".

Meie eelduse põhjal $p, q \equiv 3 \pmod 4$. Sellisel juhul on modulo p (q) ruutjuurte arvutamiseks olemas ilmutatud valem. Olgu $C \in \mathrm{QR}(p)$ ning $p \equiv 3 \pmod 4$. Siis $\left(\pm C^{(p+1)/4}\right)^2 = C^{(p+1)/2} = C^{(p-1)/2}C \equiv C \pmod p$ (kasutades jällegi Euleri kriteeriumi, mis ütleb, et $C \in \mathrm{QR}(p)$ korral $C^{(p-1)/2} \equiv 1 \pmod p$). Seega on arvu C kaheks ruutjuureks modulo p $\pm C^{(p+1)/4} \mod p$. Kõik eelnev kehtib ka, võttes p asemel q, ning seega on arvu C ruutjuurteks modulo q arvud $\pm C^{(p+1)/4} \mod q$. Kasutades hiina jääkide teoreemi leiame arvu C neli ruutjuurt x_1 modulo n.

Märkus 5.3.2 *Juhu $p \equiv 1 \pmod 4$ jaoks ei ole teada polünomiaalajas töötavat deterministlikku algoritmi, mis leiaks suvalise arvu $x \in \mathrm{QR}(p)$ ruutjuure. Samas on selleks teada polünomiaalajas toimiv Las Vegase algoritm.*

Teades arvu x_1 nelja võimalikku väärtust, leiame x-i väärtuse võrrandist $x = x_1 - B/2$, saades neli võimalikku avateksti ja oleme põhjendanud joonise 5.3 valemi d_K.

Näide 5.3.3 *Olgu $n = 77 = 7 \times 11$ ning $B = 12$. Siis*

$$e_K(x) := x^2 + 12x \mod 77$$

ning $d_K(y) := \sqrt{36+y} - 6 \mod 77$. Bobil on vaja dekrüptida krüptogramm $y = 12$, seega peame leidma arvu 48 ruutjuured modulo 7 ja modulo 11. Kuna $7 \equiv 11 \equiv 3 \pmod 4$, siis saame oma armsast valemist $48^{(7+1)/4} \equiv 6^2 \equiv 1 \mod 7$ ning $48^{(11+1)/4} \equiv 4^3 \equiv 9 \mod 11$. Hiina jääkide teoreemist saame, et arvu 23 neljaks ruutjuureks modulo 77 on $\pm 20 \mod 77$, $\pm 64 \mod 77$. Ning,

[1] kas on defineeritud?

at last but not at least, *saame kätte neli võimalikku avateksti:*

$$
\begin{aligned}
20 - 6 &\mod 77 = 14 \\
57 - 6 &\mod 77 = 51 \\
64 - 6 &\mod 77 = 58 \\
13 - 6 &\mod 77 = 7.
\end{aligned}
$$

On lihtne kontrollida, et suvaline neist neljast avatekstist krüptub krüptogrammiks 12.

Vaatleme järgnevalt Rabini krüptosüsteemi turvalisust, tõestades, et meie hüpoteetiline dekrüptimisalgoritm **A** sobib oraakliks Las Vegase algoritmilealgoritm!Las Vegase, mis faktoriseerib mooduli n tõenäosusega $\geq 1/2$. Algoritm on kujutatud joonisel 5.4.

Algoritm 5.3.4

funct *derabin* $\equiv$
  $r :=$ random$((,1), n-1)$
  $y := r^2 - B^2/4 \mod n$
  $x := \mathbf{A}(y)$
  $x_1 := x + B/2$
  if $x_1 \equiv \pm r \pmod n$
    then
      comment: *ebaedu*
      derabin $:= 0$
    else
      derabin $:= \gcd(x_1 + r, n)$
  fi
.

Joonis 5.4: Rabini mooduli faktoriseerimine dekrüptiva oraakli abil

Selgitame algoritmi tööd. Esiteks, $y = e_K\left(r - \frac{B}{2}\right)$. Sammu 4 järel $x_1^2 \equiv r^2 \pmod n$ ning seega $x_1 \equiv \pm r \pmod n$ või $x_1 \equiv \pm\omega r \pmod n$, kus ω on arvu 1 mittetriviaalne ruutjuur modulo n. Teisel juhul $n \mid (x_1 - r)(x_1 + r)$, kuid n ei jaga kumbagi paremal pool olevat tegurit. Seega $\gcd(x_1 + r, n)$ (või $\gcd(x_1 - r, n)$) arvutamine annab meile arvu n mittetriviaalse teguri p (või q, kuidas võtta).

Leiame meie algoritmi edu tõenäosuse üle juhusliku suuruse r kõikvõimalike $n-1$ valikute. Selleks defineerime hulgal $\mathbb{Z}_n \setminus \{0\}$ ekvivalentsusseose $\sim$: $r_1 \sim r_2 \Leftrightarrow r_1^2 \equiv r_2^2 \pmod n$. Hulga $\mathbb{Z}_n \setminus \{0\}$ kõikide ekvivalentsiklasside võimsuseks on 4: suvalise r korral $[r] = \{\pm r, \pm\omega r \mod n\}$, kus ω (üllatus üllatus!) on järjekordselt arvu 1 mittetriviaalne ruutjuur mooduli n järgi.

Joonisel 5.4 toodud algoritmi korral annavad suvalised kaks samasse ekvivalentsiklassi kuuluvat r-i tulemuseks sama y-i väärtuse. Vaatleme oraakli **A** poolt tagastatud väärtust $x := \mathbf{A}(y)$.

Meil on $[y] := \{\pm y, \pm\omega y \mod n\}$; kui $r = \pm y$, siis on meie algoritm ebaedukas; kui aga $r = \pm\omega y$, oleme omadega mäel. Kuna r sai valitud juhuslikult, on meie algoritm edukas tõenäosusega $\geq 1/2$.

Seega on Rabini krüptosüsteem tõestatavalt turvaline valitud avateksti ründe vastu. Samal ajal on antud süsteem absoluutselt turvavaba valitud šifferteksti ründe vastu. Meie endi süütu algoritm joonisel 5.4, mis serveeris meid seni esimese ründe vastaselt, serveerib meid samaaegselt teise ründe poolselt. Valitud krüptogrammi ründe korral asendatakse oraakel A Bobi dekrüptimisalgoritmiga.