Peatükk 8

ElGamali krüptosüsteem

8.1 Definitsioon

Definitsioon 8.1.1 *Olgu G — suvaline rühm, $\alpha \in G$ — algjuur ning $\beta \in G$. Arvu β* ***diskreetseks logaritmiks*** *alusel α nimetatakse üheselt määratud arvu a, $0 \leq a \leq |G| - 1$, et $\alpha^a = \beta$. Tähistame edaspidises $\log_\alpha \beta := a$.* ***Diskreetse logaritmi probleemiks*** *nimetatakse ülesannet leida $\log_\alpha \beta$ etteantud α ja β korral.*

Paragrahvis 8.2 toome ära mõningad tuntud algoritmid diskreetse logaritmi leidmiseks, seniks toome järgmise kriteeriumi.

Kriteerium 8.1.2 *Tuntud rünnete vältimiseks peab p olema vähemalt 150-kohaline ning arvul $p-1$ peab leiduma vähemalt üks "suur" algarvuline tegur.*

Kuna astendamine modulo p on kerge (kasutades algoritmi 5.5.1), diskreetse logaritmi leidmine aga (usutavalt) raske, siis on astendamine modulo p (usutavalt) ühesuunaline kujutus sobivalt valitud algarvude p korral). Sellel probleemil põhinev **ElGamali krüptosüsteem** on toodud joonisel 8.1.

ElGamali krüptosüsteemi mitteformaalne kirjeldus. Alice maskeerib avateksti, korrutades seda arvuga β^k ning saab nii arvu y_2, mille ta saadab edasi koos suurusega α^k. Bob (teades salajast astendajat a), leiab arvu β^k avaldisest α^k ning "demaskeerib" y_2, saades tagasi väärtuse a.

Paneme tähele, et ElGamali krüptosüsteem on randomiseeritud: krüptogramm sõltub avatekstist *ning* Alice'i poolt valitud juhuslikust väärtusest k, seega igale avatekstile vastab mitu erinevat krüptogrammi. ElGamali krüptosüsteemi (ja ka teiste diskreetse logaritmi probleemil baseeruvate tuntud krüptosüsteemide) puuduseks on teksti kahekordne pikenemine krüptimise käigus.

Näide 8.1.4 *Olgu $p = 2579$, $\alpha = 2$, $a = 765$, seega $\beta = 2^{765} \mod 2579 = 949$. Olgu $x = 231$, ning Alice valib $k = 187$. Seega $y_1 = 2^{187} \mod 2579 = 1045$, ning $y_2 = (231 \cdot 949^{187}) \mod 2579 = 295$, Alice saadab Bobile krüptogrammi $(1045, 295)$. Bob arvutab $x := 295 \cdot (1045^{765})^{-1} \mod 2579 = 295 \cdot 2290^{-1} \mod 2579 = 295 \cdot 1758 \mod 2579 = 231$.*

Krüptosüsteem 8.1.3 (ElGamal) *Olgu p — algarv, nii et diskreetse logaritmi probleem ringis $\mathbb{Z}_p$ oleks raske, olgu $\alpha \in \mathbb{Z}_p^*$ — algjuur. Olgu $\mathcal{P} = \mathbb{Z}_p^*$, $\mathcal{C} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ ning*

$$\mathcal{K} = \{(p, \alpha, a, \beta) \colon \beta \equiv \alpha^a \pmod p\},$$

kus p, α ja β on avalikud ning a on salajane.
Olgu $K = (p, \alpha, a, \beta)$ konkreetne võti ning $k \in \mathbb{Z}_{p-1}$ — juhuslik (salajane) arv. Defineerime

$$E_K(x, k) := (y_1, y_2),$$

kus

$$y_1 = \alpha^k \pmod p \text{ ja}$$
$$y_2 = x\beta^k \pmod p.$$

Suvalise paari $(y_1, y_2) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ korral

$$D_K(y_1, y_2) := y_2(y_1^a)^{-1} \mod p.$$

Joonis 8.1: ElGamal'i krüptosüsteem

8.2 Diskreetse logaritmi leidmise algoritmid

Fikseerime algarvu p ning algjuure $\alpha \in \mathbb{Z}_p^*$. Diskreetse logaritmi probleemi saab sõnastada järgmiselt: antud $\beta \in \mathbb{Z}_p^*$ korral leida üheselt määratud a, $0 \leq a \leq p-2$, mille korral $\alpha^a \equiv \beta \pmod p$.

On selge, et diskreetse logaritmi probleemi saab lahendada, vaadates läbi kõikvõimalikud a-d: selleks kulub $O(p)$ aega ja $O(1)$ mälu. Arvutades eelnevalt välja kõikvõimalikud α^a väärtused ning sorteerides paarid $(a, \alpha^a \mod p)$ teise koordinaadi järgi, saame diskreetse logaritmi probleemi lahendatud ajas $O(1)$, ent mälu kuluks $O(p)$ (ka eelarvutused võtaks aega $O(p)$) (siin kõikjal jätame tähelepanu alt välja logaritmilised tegurid keerukuses).

8.2.1 Shanks'i algoritm

Vaatleme Shanksi algoritmi (joonis 8.2). Sammu 1 saab eelnevalt ära teostada (ehkki see ei muuda "asümptootilist" lahendusaega). Kui $(j, y) \in L_1$ ja $(i, y) \in L_2$, siis $\alpha^{mj} \equiv y \equiv \beta\alpha^{-i} \pmod p$ ehk $\alpha^{mj+i} \equiv \beta \mod p$. Seega, iga β korral $\log_\alpha \beta = mj + i$, kus $0 \leq j, i \leq m-1$ ning samm 4 on edukas. Seejuures nii aega kui mälu kulub $O(m)$ (jällegi, ilma logaritmiliste teguriteta).

Olgu $m = \lceil\sqrt{p-1}\rceil$. Soovime leida väärtust $\log_\alpha \beta \pmod p$.

1. Arvuta $\alpha^{mj} \mod p$, $0 \le j \le m-1$ ning sorteeri m paari $(j, \alpha^{mj} \mod p)$ teise koordinaadi suhtes, saades massiivi L_1;
2. Arvuta $\beta\alpha^{-i} \mod p$, $0 \le i \le m-1$ ning sorteeri m paari $(j, \beta\alpha^{-i} \mod p)$ teise koordinaadi suhtes, saades massiivi L_2;
3. Leia paarid $(j, y) \in L_1$ ja $(i, y) \in L_2$ (st. paarid mille teised koordinaadid on võrdsed);
4. Vasta $\log_\alpha \beta := mj + i$;

Joonis 8.2: Shanksi algoritm diskreetse logaritmi probleemi lahendamiseks

8.2.2 Pohlig-Hellmani algoritm

Olgu $p - 1 = \prod_{i=1}^{k} p_i^{c_i}$, kus algarvud p_i on paarikaupa erinevad (seega $c_i = \mathrm{ord}_{p_i}(p-1)$). Teame, et $a := \log_\alpha \beta$ on määratud üheselt modulo $p-1$. Paneme tähele, et teades väärtusi $a \mod p_i^{c_i}$ suvalise $1 \le i \le k$ korral, saame hiina jääkide teoreemi kasutades leida suuruse $a \mod p-1$.

Järgnevas oletame, et q on algarv, $p - 1 \equiv 0 \pmod{q^c}$ ning $p - 1 \not\equiv 0 \pmod{q^{c+1}}$. Leiame avaldise $x = a \mod p^c$, $0 \le x \le q^c - 1$, väärtuse. Olgu $x = \sum_{i=0}^{c-1} a_i q^i$ (esitus q-ndsüsteemis), kus $a_i \in \mathbb{Z}_{q-1}$, $i \in \mathbb{Z}_{c-1}$. Olgu $a = x + q^c s$, $s \in \mathbb{Z}$.

Esiteks arvutame väärtuse a_0. Kuna

$$
\begin{aligned}
\frac{(p-1)(x+q^c s)}{q} - \frac{(p-1)a_0}{q} &= \frac{p-1}{q}(x + q^c s - a_0) = \\
&= \frac{p-1}{q}\left(\sum_{i=0}^{c-1} a_i q^i + q^c s - a_0\right) = \\
&= \frac{p-1}{q}\left(\sum_{i=1}^{c-1} a_i q^i + q^c s\right) = \\
&= (p-1)\left(\sum_{i=1}^{c-1} a_i q^{i-1} + q^{c-1} s\right) \equiv \\
&\equiv 0 \pmod{p-1},
\end{aligned}
$$

siis

$$\frac{(p-1)(x+q^c s)}{q} \equiv \frac{(p-1)a_0}{q} \pmod{p-1},$$

seega $\beta^{\frac{p-1}{q}} = \alpha^{\frac{(p-1)(x+q^c s)}{q}} \equiv \alpha^{\frac{(p-1)a_0}{q}} \pmod p$.

Arvutame $\beta^{(p-1)/q} \mod p$. Kui $\beta^{(p-1)/q} \equiv 1 \pmod p$, siis $a_0 = 0$. Vastasel korral leiame väärtused $\gamma = \alpha^{(p-1)/q} \mod p$, $\gamma^2 \mod p$ jne, kuni $\gamma^i \equiv \beta^{(p-1)/q} \pmod p$ mingi i korral. Kui see juhtub, siis $a_0 := i$.

Kui $c = 1$, siis oleme valmis. Vastasel juhul leiame a_1 väärtuse, defineerides $\beta_1 := \beta\alpha^{-a_0}$ ning $x_1 := \log_\alpha \beta_1 \mod q^c$. Pole raske näha, et $x_1 = \sum_{i=1}^{c-1} a_i q^i$, järelikult $\beta_1^{(p-1)/q^2} \equiv \alpha^{(p-1)a_1/q} \pmod p$. Seega arvutame $\beta_1^{(p-1)/q^2}$ ning leiame sellise i, et $\gamma^i \equiv \beta_1^{(p-1)/q^2} \pmod p$, siis $a_1 = i$.

Kui $c = 2$, siis oleme valmis, vastasel juhul kordame seda protsessi veel $c-2$ korda, saades $a_2, \ldots, a_{c-1}$. Algoritm ise on toodud joonisel 8.3.

Sisendid: α algjuur modulo p, q on algarv ning $p-1 \equiv 0 \pmod{q^c}$, kuid $p-1 \not\equiv 0 \pmod{q^{c+1}}$.
Väljundid: leiab arvud a_i ning $\log_\alpha \beta \mod q^c := \sum_{i=0}^{c-1} a_i q^i$.

funct *phlog* $\equiv$
  for $i := 0$ **to** $q-1$ **do**
      $\gamma_i := \alpha^{(p-1)i/q}$ **od**
  $j := 0$
  $\beta_0 := \beta$
  while $j \le c-1$ **do**
        $\delta := \beta_j^{(p-1)/q^{j+1}} \mod p$
        for $i := 0$ **to** $q-1$ **do**
            if $\delta = \gamma_i$ **then** $a_j := i$ **fi** **od**
        $\beta_{j+1} := \beta_j \alpha^{-a_j q^j} \mod p$
        $j := j + 1$.

Joonis 8.3: Pohlig-Hellman'i algoritm $\log_\alpha \beta \mod q^c$ arvutamiseks

Harjutus 8.2.1 *Diskreetset logaritmi saab leida suvalises tsüklilises rühmas G, eelnevalt vaatlesime ElGamali krüptosüsteemi vaid juhul $G = \mathbb{Z}_p^*$. Kirjeldada ElGamali krüptosüsteem üle suvalise tsüklilise rühma G. Kas diskreetne logaritm on asümptootiliselt sama keeruline iga G korral? Kui ei, siis tuua näide rühmadest G, mille korral diskreetne logaritm on lihtsalt lahenduv.*

k

Peatükk 9

Digitaalsignatuurid

9.1 Vajadus

Eelnevalt (paragrahv 1.1) on juba mainitud, et põhilisteks turvaeesmärkideks on konfidentsiaalsus, andmeterviklus, autentimine ning salgamise vääramine, kusjuures andmeallika autentimine tagab varjatud kujul ka andmetervikluse. Senini käsitletud krüptograafiliste primitiivide põhieesmärk on olnud tagada konfidentsiaalsust, käesolevas peatükis vaatleme autentsuse tagamist *digitaalsete signatuuride* (*digital signature*) kasutamise abil.

Paragrahvis 1.1 oli toodud näide sellest, et dokumendile käsitsi kirjutatud allkiri on mõeldud dokumendi eest vastutava isiku spetsifitseerimiseks ehk andmeallika autentimiseks, kuigi ainult allkirja kirjutamine on iseenesest tühipaljas identimine (autentimine saavutatakse antud juhul vaid vastavate organisatoorsete meetmete kasutamisel, mis sätestavad teatud tüüpi dokumentide blankettidele allkirja kirjutamist autentimisena), autentimiseks on vaja allkirja turvalist sidumist ülejäänud teatega, näiteks kogu teate käsitsi kirjutamist. Autentimise eesmärgiks on siduda olem tema poolt öelduga, ka elektroonilisel infovahetusel kasutatavad digitaalsed protseduurid peavad hõlmama kogu teadet. **Digitaalsed signatuurid** on üks meetod digitaalse informatsiooni autentimiseks.

Märkus 9.1.1 *Eeltoodut arvestades nimetatakse digitaalsignatuuriks* kogu *signeeritud sõnumit.*

Põhimõtteline erinevus digitaalsignatuuri ning tavalise allkirja vahel seisneb digitaalse informatsiooni kergesti kopeeritavuses: elektroonilise dokumendi koopia on identne originaaliga; samas on tavalise allkirja originaali ja koopiad enamik juhtudel võimalik eristada. Seetõttu peab hoolt kandma, et signatuure ei saaks taaskasutada. Hea näide on ost-müük läbi arvutivõrgu; osapooled peavad olema kindlad, et nende "tšekki" sooritatud ostu eest mitu korda ei kasutata. Seega peab dokument sisaldama informatsiooni (näiteks kuupäeva), mis välistab taaskasutamist.

Digitaalsignatuuri skeemid koosnevad kahest, teineteisest sõltuvast, protseduurist:

1. Signeerimine ehk protseduur, mis seostab signeeritava informatsiooni signeerijaga. Signeerimisalgoritm peab olema avalik; konkreetset olemit seob sõnumiga olemi signeerimisvõti.

2. Verifitseerimine ehk protseduur, mis kõrvutab signeeritud informatsiooni ja olemi signeerimisvõtit ning ütleb, kas antud olem oli signeerija või mitte.

Toome järgnevalt signatuuriskeemi formaalse definitsiooni.

Definitsioon 9.1.2 ***Signatuuriskeemiks*** *nimetatakse järgmisi tingimusi rahuldavat viisikut* $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$.

1. $\mathcal{P}$ *on võimalike teadete lõplik hulk.*

2. $\mathcal{A}$ *on võimalike signatuuride lõplik hulk.*

3. $\mathcal{K}$ *on võimalike võtmete lõplik hulk.*

4. *Suvalise* $K \in \mathcal{K}$ *jaoks leidub signeerimisalgoritm* $\mathrm{sig}_K \in \mathcal{S}$ *ning vastav verifitseerimisalgoritm* $\mathrm{ver}_K \in \mathcal{V}$. *Signeerimisalgoritm* $\mathrm{sig}_K : \mathcal{P} \to \mathcal{A}$ *ning verifitseerimisalgoritm* $\mathrm{ver}_K : \mathcal{P} \times \mathcal{A} \to \{\perp, \top\}$ *rahuldavad võrrandit*

$$\mathrm{ver}_K(x, y) := \begin{cases} \top & y = \mathrm{sig}_K(x) \\ \perp & y \neq \mathrm{sig}_K(x). \end{cases}$$

   suvalise teate $x \in \mathcal{P}$ *ja suvalise signatuuri* $y \in \mathcal{A}$ *jaoks.*

Signatuuriskeemi praktiliseks realiseeritavuseks peavad ver_K ning sig_K olema polünomiaalse keerukusega. Siinjuures ver_K on avalik ning sig_K salajane algoritm. Kurikaela jaoks peab olema arvutuslikult ebaökonoomne teise olemi signatuuri võltsida, st et suvalise teate x korral peab ainult volitatud osapool suutma leida sellist signatuuri y, et $\mathrm{ver}_K(x, y) = \top$. Signatuuriskeem ei saa olla tingimusteta turvaline, kuna Edgar võib täieliku läbivaatamise meetodil testida läbi kõiki signatuuride y jaoks, kas $\mathrm{ver}_K(x, y) = \top$. Ehk, signatuuriskeemi lahtimurdmise ülesanne on keerukusklassist $\mathcal{FNP}$. Meie eesmärk on leida arvutuslikult turvalisi signatuuriskeeme.

RSA avaliku võtme krüptosüsteemi võib kasutada signatuuriskeemina, vastav skeem on toodud joonisel 9.1.

Alice signeerib teate, krüptides selle salajase võtmega d_A. Kuna d_A on salajane, ei ole kurikael Edgar võimeline Alice'i signatuuri võltsima. Verifitseerimisalgoritm kasutab vastavat RSA krüptimisfunktsiooni e_A. Kõik saavad signatuuri verifitseerida, kuna e_A on avalik.

Märkus 9.1.4 *Alice'i signatuuri juhusliku* x*-i korral saab võltsida, arvutades* $x := e_A(y)$ *mingi* y *jaoks; siis* $y = \mathrm{sig}_K(x)$. *Üks võimalus sellist olukorda vältida on nõuda, et teated sisaldaksid piisavalt liiasust, mis garanteeriks, et seda*

Krüptosüsteem 9.1.3 (RSA signatuuriskeem) *Olgu $n = pq$, kus p ja q on algarvud. Olgu $\mathcal{P} = \mathcal{A} = \mathbb{Z}_n$ ning*

$$\mathcal{K} = \{(n, p, q, a, b) : n = pq, p \text{ ja } q \text{ on algarvud, } ab \equiv 1 \pmod{\phi(n)}\}.$$

Väärtused n ning b on avalikud, p, q ja a on salajased.
Võtme $K = (n, p, q, a, b)$ jaoks defineerime

$$\begin{aligned} \mathrm{sig}_K(x) &:= x^a \pmod{n} \\ \mathrm{ver}_K(x, y) &:= \top \Leftrightarrow x \equiv x^b \pmod{n}, \end{aligned}$$

kus $(x, y) \in \mathbb{Z}_n^2$.

Joonis 9.1: RSA signatuuriskeem

tüüpi võltsitud signatuur vastaks "mõttekale" signatuurile tühise tõenäosusega. Teine võimalus toodud tüüpi võltsimist vältida on kasutada räsifunktsioone koos signatuuriskeemiga.

Joonisel 9.2 on kujutatud juhtu, kus saadetavat teksti nii signeeritakse kui krüptitakse. Juhul kui Alice soovib Bobile saata signeeritud ja krüptitud teate, signeerib Alice selle esmalt, kasutades oma signeerimisvõtit S_A ning seejärel krüptib paari $(p, \mathrm{sig}_{S_A}(p))$, kasutades salajast krüptimisvõtit K_A^{-1}. Saadud tulemuse $\{p, s\}_{K_A}$ dekrüptib Bob, kasutades talle teadaolevat Alice'i avalikku krüptimisvõtit K_A ning seejärel kontrollib signatuuri protseduuri ver_{S_A} abil, kas $\mathrm{ver}_{S_A}(p, s) = \top$.

Harjutus 9.1.5 *Analüüsida olukorda, kus Alice esmalt krüptib ja seejärel signeerib teate. Kirjeldada võimalikult palju põhimõtteliselt erinevaid ründeid sellise lähenemise vastu.*

9.2 ElGamali signatuuriskeem

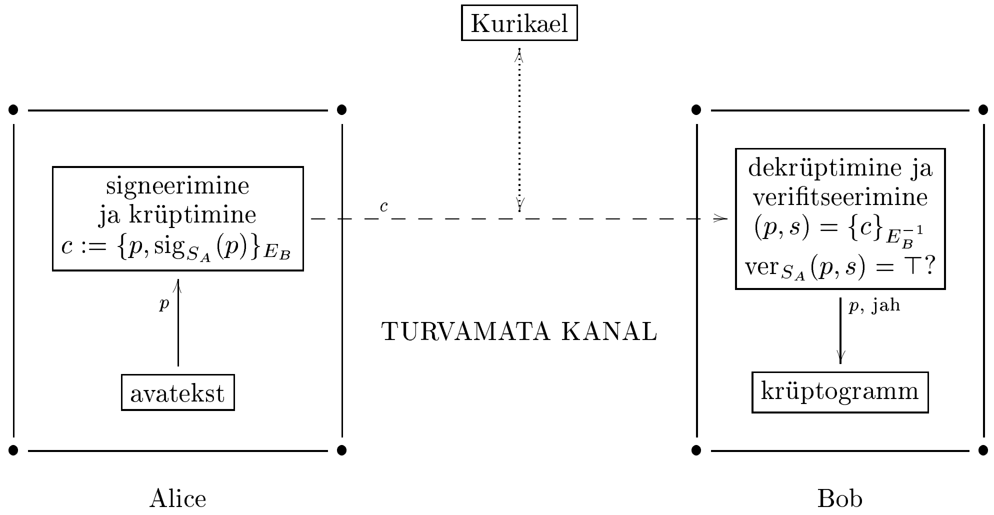
Kirjeldame järgnevas 1985. aastast pärinevat ElGamali signatuuriskeemi. ElGamali skeemi modifikatsioon on NIST-i (*National Institute of Standards and Technology*) poolt kinnitatud digitaalse signatuuri standardiks.

ElGamali signatuuriskeem, nagu ka ElGamali krüptosüsteem, on randomiseeritud. See tähendab, et suvalisele teatele vastavad paljud erinevad signatuurid; verifitseerimisalgoritm peab aktsepteerima suvalist "õiget" signatuuri. ElGamali signatuuriskeemi kirjeldus on antud joonisel 9.3.

Korrektse signatuuri verifitseerimine õnnestub, kuna

$$\beta^\gamma \gamma^\delta \equiv \alpha^{a\gamma} \alpha^{k\delta} \equiv \alpha^x \mod p,$$

kus me kasutasime fakti, et $a\gamma + k\delta \equiv x \mod p - 1$.



Joonis 9.2: Signeerimise ja krüptimise kooskasutus

Krüptosüsteem 9.2.1 *Olgu p selline algarv, et diskreetse logaritmi probleem $\mathbb{Z}_p$-s on raske ja olgu $\alpha \in \mathbb{Z}_p^*$ algjuur. Olgu $\mathcal{P} = \mathbb{Z}_p^*$, $\mathcal{A} = \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$ ning*

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \mod p\}.$$

Väärtused p, α ning β on avalikud; a on salajane. Fikseeritud võtme $K = (p, \alpha, a, \beta)$ ning salajase juhusliku suuruse $k \in \mathbb{Z}_{p-1}^*$ jaoks defineerime

$$\mathrm{sig}_K(x)k := (\gamma, \delta),$$

kus

$$\gamma := \alpha^k \pmod p$$

ja

$$\delta := (x - a\gamma)k^{-1} \pmod{p-1}.$$

Suvalise paari $(x, \gamma) \in (\mathbb{Z}_p^*)^2$ ning suvalise $\delta \in \mathbb{Z}_{p-1}$ jaoks defineerime

$$\mathrm{ver}_K(x, \gamma, \delta) := \top \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^x \mod p.$$

Joonis 9.3: ElGamali signatuuriskeem

Alice signeerib teate x, kasutades salajast väärtust a (mis on osa?võtmest) ning salajast juhuslikku arvu k (mida kasutatakse ühe teate signeerimiseks). Verifitseerimist saab teostada kasutades vaid avalikku informatsiooni.

9.2.1 ElGamali signatuuriskeemi turvalisus

Kurikaela eesmärgiks on võltsida antud teate x signatuuri, teadmata salajast väärtust a. Kui Edgar valib väärtuse γ, siis sellest lähtudes vastava δ leidmiseks peab ta arvutama diskreetse logaritmi $\log_\gamma \alpha^x\beta^{-\gamma}$. Teisest küljest, kui Edgar valib esmalt väärtuse δ, siis sellest lähtudes vastava γ leidmiseks peab ta lahendama võrrandi

$$\beta^\gamma\gamma^\delta \equiv \alpha^x \mod p$$

muutuja γ suhtes. Seda probleemi ei osata ökonoomselt lahendada, ehkki see ei tundu toetuvat mõnele hästi tuntud probleemile (nagu näiteks diskreetse logaritmi probleem). Ka ei ole välistatud võimalus, et Edgar saab paari (γ, δ) samaaegselt välja arvutada.

Kui Edgar valib arvud γ ja δ ning püüab saadud võrrandit lahendada x-i suhtes, on jällegi tegu diskreetse logaritmi probleemiga: ta peab leidma suuruse $\log_\alpha \beta^\gamma\gamma^\delta$. Seega ei saa Edgar antud lähenemisviisi kasutades "juhuslikku" teadet signeerida. On olemas meetod, kuidas Edgar saab juhuslikku teadet signeerida, valides kolmiku (x, γ, δ) samaaegselt. Nimelt, olgu $i, j \in \mathbb{Z}_{p-1}$, nii et $\gcd(j, p-1) = 1$. Kolmiku leiame järgmiselt:

$$\begin{aligned} \gamma &:= \alpha^i\beta^j \mod p \\ \delta &:= -\gamma j^{-1} \mod p-1 \\ x &:= -\gamma i j^{-1} \mod p-1, \end{aligned}$$

kus j^{-1} leidakse modulo $p-1$ (ja seega peab kehtima $\gcd(j, p-1) = 1$). Verifitseerimine annab tõesti, et $\beta^\gamma\gamma^\delta \equiv \alpha^x \mod p$.

Veel üheks võimalikuks ründeks on kasutada Alice'i poolt eelnevalt signeeritud teadet. Olgu $\mathrm{sig}_K(x) = (\gamma, \delta)$ kehtiv signatuur ning $(h, i, j) \in (\mathbb{Z}_{p-1})^3$, kus $\gcd(h\gamma - j\delta, p-1) = 1$. Arvutame järgmised suurused:

$$\begin{aligned} \lambda &:= \gamma^h\alpha^i\beta^j \mod p \\ \mu &:= \delta\lambda(h\gamma - j\delta)^{-1} \mod p-1 \\ x' &:= \lambda(hx + i\delta)(h\gamma - j\delta)^{-1} \mod p-1, \end{aligned}$$

kus $(h\gamma - j\delta)^{-1}$ arvutatakse modulo $p-1$. Kontrollimine annab, et $\beta^\lambda\lambda^\mu \equiv \alpha^{x'} \mod p$. Seega on (λ, μ) sobib signatuur x' jaoks.

Kumbki meetod võimaldab moodustada verifitseerimist läbivaid signatuure, kuid need meetodid ei võimalda kurikaelal signeerida enda poolt valitud teateid. Seega tundub, et need meetodid ei kujuta ohtu ElGamali signatuuriskeemi turvalisusele.

Mainime lõpuks mõningaid ettevaatusabinõusid, mida peab kasutama, rakendades ElGamali signatuuriskeemi.

1. signatuuri arvutamisel kasutatud juhuslik väärtus k peab jääma salajaseks. Kui k on teada, saab a leida valemist

$$a = (x - k\delta)\gamma^{-1} \mod p - 1$$

ning siis on signatuuriskeem lahti murtud.

2. erinevate teadete signeerimisel peab kasutama erinevaid juhuslikke väärtusi. Oletame, et teated x_1 ja x_2 on signeeritud, kasutades samat k väärtust, saades tulemuseks signatuurid (γ, δ_1) ning (γ, δ_2). Siis elementaarselt

$$\alpha^{x_1 - x_2} \equiv \gamma^{\delta_1 - \delta_2} \mod p.$$

Kuna $\gamma = \alpha^k$, saame järgmise võrrandi tundmatu k suhtes:

$$x_1 - x_2 \equiv k(\delta_1 - \delta_2) \mod p - 1. \tag{9.1}$$

Harjutus 9.2.2 *Leida tee ElGamali signatuuriskeemi murdmiseks, kui on teada võrdus (9.1).*

9.3 Digitaalsignatuuri standard

Digitaalsignatuuri standard (DSS, joonis 9.4) on ElGamali signatuuriskeemi modifikatsioon, mis võeti standardiks aastal 1994. Motiveerime esmalt ElGamali signatuuriskeemi tehtud muudatusi ning motiveerime tehtud muudatusi.

Paljudes olukordades on teadet vaja krüptida ja dekrüptida vaid üks kord; seega piisab, kui krüptosüsteem on "tugev" tema kasutamise hetkel. Samas võib signeeritud dokument kehtida kui õiguslik dokument või leping veel aastaid pärast signeerimist ennast. Seega on signatuuriskeemi turvalisuseks vaja veelgi enam vaeva näha kui krüptosüsteemi turvalisuseks. Fakt, et ElGamali signatuuriskeem ei ole turvalisem kui diskreetse logaritmi probleem sunnib meil kasutama signatuuri arvutamisel suuri algarve p (suurusjärgus 1024 bitti).

Kuid isegi 512-bitine moodul viiks signatuuri pikkuse juba 1024 bitini, mis on paljude rakenduste (näiteks kiipkaartide) jaoks liiga pikk. DSS-is sisse toodud muudatused adresseerivadki just seda küsimust. Nimelt signeeritakse 160-bitine teade 320-bitise signatuuriga, kuid kõik arvutused tehakse 512-bitise mooduli p järgi, "töötades" rühma $\mathbb{Z}_p^*$ alamrühmas, mille suurus on 2^{160}. DSS-i turvalisus sõltub oluliselt tehtavast eeldusest, et antud alamrühmas on digitaalse logaritmi probleem raske.

Esimene muudatus on δ definitsioonis miinuse vahetamine plussi vastu, nii et $\delta := (x + a\gamma)k^{-1} \pmod{p-1}$, mis tingib ka vastava muudatuse verifitseerimises:

$$\mathrm{ver}_K(x, \gamma, \delta) = \top \Leftrightarrow \alpha^{x\delta^{-1}}\beta^{\gamma\delta^{-1}} = \gamma \pmod{p}. \tag{9.2}$$

DSS-i innovatsioon seisneb aga järgnevas. Oletame, et q on 160-bitine algarv, $q \mid p - 1$, ning α on q-ndat järku ühejuur modulo p (sellist elementi on lihtne konstrueerida, näiteks kui α_0 on algjuur modulo p, siis võtta $\alpha := \alpha_0^{(p-1)/q}$

Krüptosüsteem 9.3.1 *Olgu p selline 512-bitine algarv, et diskreetse logaritmi probleem rühmas $\mathbb{Z}_p$ on raske. Olgu q 160-bitine arv, $q \mid p-1$. Olgu $\alpha \in \mathbb{Z}_p^*$ q-s ühejuur modulo p. Olgu $\mathcal{P} = \mathbb{Z}_q^*$, $\mathcal{A} = \mathbb{Z}_q \times \mathbb{Z}_q$, defineerime*

$$\mathcal{K} = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a \mod p\}.$$

Väärtused p, q, α ja β on avalikud, a on salajane. Konkreetse võtme $K = (p, q, \alpha, a, \beta)$ ja salajase juhusliku arvu $k \in \mathbb{Z}_q^*$ jaoks defineerime

$$\mathrm{sig}_K(x)k = (\gamma, \delta),$$

kus

$$\gamma = (\alpha^k \pmod p) \pmod q$$

ning

$$\delta = (x + a\gamma)k^{-1} \pmod q.$$

Olgu $x \in \mathbb{Z}_q^*$ ning $\gamma, \delta \in \mathbb{Z}_q$. Verifitseerimine toimub järgmiselt:

$$\begin{aligned} e_1 &:= x\delta^{-1} \pmod q \\ e_2 &:= \gamma\delta^{-1} \pmod q \end{aligned}$$

$$\mathrm{ver}_K(x, \gamma, \delta) = \top \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \pmod p) \pmod q = \gamma.$$

Joonis 9.4: Digitaalsignatuuri standard

(mod p)), siis on ka β ning γ q-ndat järku ühejuured. Seega saab astendajaid α, β ning γ redutseerida mooduli q järgi, ilma tingimust (9.2) rikkumata. Probleem on selles, et γ esineb võrrandi (9.2) vasakus pooles kui astendaja ning paremas pooles kui tavaline liige. Seega kui γ-t redutseeritakse modulo q, tuleb tervet võrrandi 9.2 vasakut poolt redutseerida.

Märkus 9.3.2 *Verifitseerimise õnnestumiseks ei tohi δ olla* $0 \pmod q$. *Kui signeerimise ajal genereeritakse δ, $\delta \equiv 0 \pmod q$ (selle tõenäosus on 2^{160}), siis tuleb genereerida uus k ning leida uus δ.*

DSS-i kriitika põhineb järgmistel asjaoludel:

1. Väljatöötamiskriteeriumid NSA poolt ei ole avalikustatud. Kahtlustatakse püünise olemasolu.

2. Signatuuri mooduli pikkus on 512, mis jääb ilmselgelt varsti väikseks. Uuemates DSS-i versioonides on mooduli pikkus varieeritav, p pikkus võib olla suvaline 64-ga jaguv arv 512 ja 1024 vahel.

3. DSS-iga signeerimine on tunduvalt kiirem kui verifitseerimine. Kuna enamus dokumentide puhul signeeritakse vaid üks kord, verifitseeritakse aga mitu korda, oleks loogiline omada kiiremat verifitseerimisalgoritmi. NIST-i vastus: pole tähtis kumb on kiirem kui mõlemad on piisavalt kiired.