Osnove kriptografije in El-Gamalov digitalni podpis

Nives Gošnjak, Samo Krejan, Hugo Trebše
Mentor: Nino Cajnkar

Povzetek

Pri projektu smo se spoznali z osnovami kriptografije in abstraktne algebre in ugotovili, da nam varnost pri kodiranju zagotavljajo dekodirni ključi in ne algoritmi. Pogledali smo si različne šifre in napade nanje, kako deluje Diffie-Hellmanova shema ter ElGamalov digitalni podpis in napad nanj.

1 Uvod

Kriptografija je znanstvena veda, ki se ukvarja z odkrivanjem in preučevanjem računalniških algoritmov za zagotavljanje varnega načina komuniciranja. Šifriranje se izvaja z enkripcijsko funkcijo s pomočjo šifrirnega ključa, dešifrira pa se z dešifrirno funkcijo s pomočjo dešifrirnega ključa, ki dešifrira prejet kriptogram.

Kriptografija se je začela uporabljati že v času pred našim štetjem in se je postopoma vedno bolj razvijala. V preteklosti se je uporabljala predvsem za vojne in politične namene, danes pa se, poleg za namene oboroženih sil, uporablja tudi za namene obveščevalnih služb, pri elektronskem poslovanju, medsebojnem komuniciranju ter za zaščito avtorskih pravic.

Eden od zanimivih primerov je način šifriranja, ki so ga uporabljali Špartanci. Ti so namreč okoli valja navili tanek list in nato besedilo napisali pravokotno na smer traku. Odvit list so nato poslali prejemniku, ki je za dekripcijo potreboval valj enakega premera. Zelo znana je tudi Cezarjeva šifra, ki je podrobneje predstavljena v nadaljevanju, uporabljal pa jo je Julij Cezar, predvsem za vojaške namene. Kriptografija je igrala zelo pomembno vlogo v 2. svetovni vojni, ko so nemške oborožene sile začele uporabljati šifrirni stroj ENIGMA. Beseda izhaja iz grščine in pomeni uganka, saj naj bi predstavljala uganko za Angleže, ki so se trudili dešifrirati nemška besedila.

2 Osnovne definicije

Kriptografija je veda o varni komunikaciji po nezaščitenem kanalu. Pošiljatelj pred oddajo sporočilo kodira, prejemnik pa ga dekodira. Pogoj za varnost komunikacije je, da nasprotnik ne pozna dekodirnega ključa.

Definicija 1. ***Kriptosistem*** *je kombinacija petih komponent* $(\mathcal{B}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$*, za katere velja:*

- $\mathcal{B}$ *je končna množica besedil,*
- $\mathcal{C}$ *je končna množica kriptogramov,*
- $\mathcal{K}$ *je končna množica ključev,*
- $\mathcal{E} = \{E_k : \mathcal{B} \to \mathcal{C}; k \in \mathcal{K}\}$,

- $\mathcal{D} = \{D_k : \mathcal{C} \to \mathcal{B}; k \in \mathcal{K}\}$,
- *za vsak* $e \in \mathcal{K}$ *obstaja* $d \in \mathcal{K}$, *da za vsak* $x \in \mathcal{B}$ *velja* $D_d(E_e(x)) = x$, *pri čemer je* x *čisto besedilo.*

Dve osebi si torej lahko izmenjujeta šifrirana besedila znotraj enega kriptosistema. Oseba A ima neko čisto besedilo iz množice $\mathcal{B}$, ki jo s pomočjo enkripcijske funkcije (tj. funkcije, s katero šifriramo besedilo) iz množice $\mathcal{E}$ spremeni v šifrirano besedilo oz. kriptogram. To besedilo nato pošlje osebi B, ki ga s pomočjo dekripcijske funkcije (tj. funkcije, s pomočjo katere dešifriramo) iz množice $\mathcal{D}$ dešifrira po nekem ključu iz množice $\mathcal{K}$ in posledično dobi prvotno, čisto besedilo, ki mu ga je želela oseba A poslati.

Poznamo **simetrične** in **asimetrične** sisteme. Pri simetričnih sistemih velja, da lahko iz šifrirnega ključa zelo hitro ugotovimo dešifrirni ključ. Primer takega sistema je v nadaljevanju razložena Cezarjeva šifra. Asimetričnim sistemom pravimo tudi kriptosistemi z javnim ključem, kjer javnost pozna šifrirne ključe, dešifrirni ključ pa je znan samo lastniku ključa.

Definicija 2. ***Algoritem*** *je v matematiki in računalništvu končno zaporedje natančno določenih, računalniško izvedljivih navodil, običajno namenjenih reševanju težav ali za izvajanje izračunov.*

Algoritmi so v kriptografiji postopki, s katerimi lahko neko sporočilo enkriptiramo in kasneje tudi dekriptiramo, če poznamo dešifrirni ključ.

Definicija 3. ***Grupa*** $(G, \circ)$ *je par neprazne množice* G *in binarne operacije na* G $\circ$, *ki jo imenujemo operacija grupe in mora zadoščevati pogojem (aksiomom) grupe:*

- *za* $\forall a, b, c \in G : \ (a \circ b) \circ c = a \circ (b \circ c)$,
- $\exists \ e \in G : \ \forall \ a \in G, e \circ a = a \circ e = a$,
- *za* $\forall \ a \in G, \exists \ a^{-1} : \ a \circ a^{-1} = a^{-1} \circ a = e$.

Definicija 4. *Grupa* $(G, \circ)$ *je* ***ciklična****, če obstaja tak* α, *da* $G = \{e, \alpha, \alpha^1, \cdots, \alpha^{n-1}\}$ *in* $e = \alpha^n$. *V takem primeru je* α *generator grupe.*

V tem članku bomo grupe uporabili v poglavjih o Diffie-Hellmanovi shemi in digitalnem podpisu.

3 Osnovne modularne aritmetike

Definicija 5. *Za celi števili a in b pravimo, da sta* ***kongruentni*** *po modulu n, če imata enak ostanek pri deljenju z n. Ekvivalentno*

$$a \equiv b \pmod{n} \iff n \mid a - b.$$

Za lažjo predstavo si poglejmo kak primer. Velja $10 \equiv 2 \pmod{8}$, saj imata obe števili pri deljenju z 8 ostanek 2, oziroma $8 \mid 10 - 2$. Podobno velja tudi $24 \equiv 3 \pmod{7}$. Kongruence imajo kup uporabnih lastnosti:

1. $n \mid a$, če in samo če je $a \equiv 0 \pmod{n}$,
2. če $d \mid n$ in je $a \equiv b \pmod{n}$, je $a \equiv b \pmod{d}$,
3. za $a \equiv b \pmod{n}$ in $u \equiv v \pmod{n}$ velja:
   - $a + u \equiv b + v \pmod{n}$,
   - $a - u \equiv b - v \pmod{n}$,
   - $au \equiv bv \pmod{n}$.

Vse te lastnosti lahko dokažemo direktno po definiciji kongruenc.

4 Napadi

V javnem forumu obstajajo ljudje, ki si želijo prebrati sporočila, ki jim niso namenjena. Ker so ta sporočila zakodirana, jih ne morejo napasti neposredno, lahko pa se lotijo napada na kriptosistem, s čimer bi seveda dobili možnost branja zakodiranih sporočil. V tem poglavju si bomo pogledali različne tipe napadov in napade na nekatere specifične šifre.

4.1 Vrste napadov

V osnovi delimo napade med *aktivne* in *pasivne*. Razlika se skriva v tem, katere informacije so napadalcu na voljo. Pod *pasivne* napade tako štejemo napade, kjer:

- nasprotnik pozna kriptogram c, ki ga lahko dešifrira, če pozna ključ,

- nasprotnik pozna nekaj parov besedil in kriptogramov (b, c), iz katerih poskusi izračunati ključ,
- nasprotnik napade z izbranim besedilom, ki ga šifrira, če pozna E, tako si ustvari pare (b, c). Pri simetričnih kriptosistemih to vrsto štejemo med aktivne napade.

Med *aktivne* napade pa sodijo:

- napad z izbranim kriptogramom c, ki ga napadalec ima, nato pa dobi informacijo o b ter s primerjanjem poskusi razbiti kriptosistem,
- prilagodljiv napad z izbranim kriptogramom c, kjer pa ima napadalec dostop do dešifrirnega algoritma D, s čimer si lahko generira pare (b, c), ki so mu v pomoč pri napadu.

4.2 Izčrpno iskanje ključa

Izčrpno iskanje ključev je algoritem, ki nam poišče ključ tako, da preveri vse možnosti.

Podatki: $x \in \mathcal{B}, y \in \mathcal{C}$ za kriptosistem $(\mathcal{B}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$.

Iščemo: $k \in \mathcal{K}$, za katerega je $E_k(x) = y$.

Postopek: $\forall\ k \in \mathcal{K}$, če velja $E_k(x) = y$, izpiši k.

Napad z izčrpnim iskanjem ne deluje primerih ko je velikost $\mathcal{K}$ dovolj velika. Da dosežemo minimalno raven varnosti, mora biti velikost $\mathcal{K}$ (tj. število možnih ključev) vsaj 2^{80}, vendar je v praksi željena velikost vsaj 2^{128}.
Rečemo, da je kriptosistem *razbit*, če lahko ključ najdemo mnogo hitreje kot z izčrpnim iskanjem ključev.

4.3 Frekvenčna analiza

Pri iskanju ključa je zelo koristna tehnika tudi tako imenovana *frekvenčna analiza*. Pri tej tehniki uporabimo frekvenčne diagrame, ki nosijo informacijo o neki karakteristični distribuciji različnih znakov, skupin znakov in besed.

Karakteristična distribucija je neko splošno znano dejstvo (lahko dosegljivo na internetu), katere lepa lastnost je ta, da se tudi po tem, ko besedilo šifriramo z nekaterimi kriptosistemi, ohrani. Torej, posamezne frekvenčne ponovitve ostanejo bolj ali manj nespremenjene, le da sedaj stojijo pri drugih znakih (besedah).
V slovenskem jeziku je tako najbolj pogosta črka e in če ugotovimo, da se v zakodiranem besedilu največkrat pojavi črka m, lahko sklepamo, da se precej verjetno e kodira v m. To seveda naredimo tudi za vse ostale znake in besede, kar nam da nek precej verjeten ključ.

4.4 Cezarjeva šifra

Cezarjeva šifra zamakne običajno abecedo za n znakov v desno. Pri $n = 1$ bi namesto A zapisali B, namesto B bi zapisali C in tako dalje. Za dešifracijo moramo poznati le n, da abecedo zamaknemo nazaj in s tem dobimo originalni tekst. Opisani kriptosistem bi matematično definirali sledeče:

- $\mathcal{B} = \mathcal{C} = \mathcal{K} = \{0, 1, \cdots, 24\} = \mathbb{Z}_{25}$,
- $\mathcal{E}_k(x) = x + k \pmod{25}$,
- $\mathcal{D}_k(x) = x - k \pmod{25}$.

To je ena izmed najstarejših znanih šifer, temu primerna je tudi težavnost napada nanjo, če namreč prestrežemo zakodirano sporočilo in ga želimo dešifrirati, je dovolj že samo, da preverimo vseh 25 možnih ključev, kar (predvsem za dandanašnje računalnike) ni preveč zahtevno.
Kljub temu, da izčrpno iskanje ključev pri Cezarjevi šifri ni zahtevno, si lahko vseeno močno pomagamo s frekvenčno analizo.

4.5 Vigenerjeva šifra

Vigenerjeva šifra je na nek način le razširjena Cezarjeva šifra, a je kljub temu precej močnejša. Tokrat namreč ključ sestavlja več števil. Število števil, ki sestavljajo ključ, imenujemo dolžina ključa, označimo jo z n. Kako

Vigenerjeva šifra deluje je najlažje kar na specifičnem primeru. Recimo, da želimo zakodirati besedilo $ABCDE$ s ključem $(0, 1, 2)$. Prvo črko v besedilu bomo prestavili za nič mest, drugo bomo prestavili za eno v desno, tretjo za dve, četrto pa nato spet za nič, ter tako dalje. Naše besedilo bi tako postalo $ACEDF$.
Kriptosistem formalno definiramo na naslednji način:

- $\mathcal{B} = \mathcal{C} = \mathcal{K} = \{0, 1, \cdots, 24\} = \mathbb{Z}_{25}^n$,
- $E_k(x) = \underline{x} + \underline{k} \pmod{25} = (x_1 + k_1, x_2 + k_2, \cdots, x_n + k_n) \pmod{25}$,
- $D_k(x) = \underline{x} - \underline{k} \pmod{25} = (x_1 - k_1, x_2 - k_2, \cdots, x_n - k_n) \pmod{25}$.

Vigenerjeva šifra je precej odpornejša na preproste napade kot Cezarjeva, saj, če bi se lotili izčrpnega iskanja ključa, bi se časovna zahtevnost izjemno povečala. Sedaj namreč ne rabimo preveriti le 25 možnosti ampak iz 25^n možnosti, s tem da sploh nimamo informacije o n. Velja, da je šifra približno varna pred napadom z izčrpnim iskanjem ključa, če mora računalnik pred razdrtjem preveriti vsaj 2^{80} možnih ključev, vendar je v praksi željenih vsaj 2^{128}. Če vzamemo $n = 25$, se temu že zelo približamo, saj je $25^{25} \approx 2^{126}$.
Poleg tega si pri Vigenerjevi šifri sprva ne moramo kaj dosti pomagati s frekvenčno analizo, saj se karakteristična distribucija ne ohranja. Frekvenčno analizo lahko uporabimo šele, ko nam uspe Vigenerjevo šifro razbiti v več Cezarjevih.

Vseeno poznamo trik, ki nam delo precej olajša. Da razbijemo Vigenerjevo šifro, je predvsem pomembno, da najprej ugotovimo dolžino ključa. To lahko dobimo s testom Kasiskega.
Za ta test najprej predpostavimo, da je dolžina besedila vsaj nekajkrat daljša od dolžine ključa. Takrat se namreč lahko zgodi, da bodo bloki besedila, ki se večkrat ponovijo, tudi enako zakodirani, če bodo ležali na nekem večkratniku n narazen. Test Kasiskega gre nato takole.

1. Poiščemo pare $p_1, p_2, \cdots, p_m$ identičnih blokov besedila dolžine vsaj tri,
2. z d_i označimo razdaljo med parom p_i,
3. poiščemo $\gcd(d_1, d_2, \cdots, d_m)$, ki je nato zelo dober kandidat za n.

Ko poznamo n, kriptogram lahko *razrežemo* na n različnih Cezarjevih šifer. To naredimo po sledečem postopku.

1. Vsak znak označimo z d_i, kjer i označuje mesto znaka v besedilu.
2. Definiramo n različnih množic.
   - $\mathcal{C}_1 = \{d_i \mid i = 1 \pmod n\}$
   - $\mathcal{C}_2 = \{d_i \mid i = 2 \pmod n\}$
   - $\cdots$
   - $\mathcal{C}_n = \{d_i \mid i = n = 0 \pmod n\}$
3. $\mathcal{C}_j$ tako postane svoja Cezarjeva šifra γ_j, ki jo definiramo kot zaporedje znakov v $\mathcal{C}_j$, tako da si indeksi sledijo v naraščajočem zaporedju.

Preostane nam še, da poiščemo posamezne ključe za te Cezarjeve šifre. Ko nam to uspe, so koraki do popolnoma dešifriranega besedila trivialni. To najlažje storimo s formulo najmanjšega odstopanja.
Najprej definiramo sledeče:

- s je število simbolov v $\mathcal{B}$,
- N je število vseh simbolov v B,
- $m = \frac{N}{n}$,
- f_i je frekvenca črke i v γ_j,
- p_i je frekvenca črke i v podobnih tekstih (dobimo iz frekvenčnih diagramov),
- k_j je željen ključ za γ_j.

Verjetnega kandidata za k_j lahko dobimo po formuli

$$k_j = \min \left\{ \sum_{i=1}^{s} \left(p_i - \frac{f_{i+k \pmod s}}{m} \right)^2 \right\}.$$

S tem algoritmom se časovna zahtevnost precej zmanjša, zato za sodobno tehnologijo Viginerjeva šifra ni več med varnejšimi.

5 Problem diskretnega logaritma

Denimo, da za neka $\alpha, y \in \mathbb{Z}_p^\times$, kjer je α generator grupe $\mathbb{Z}_p^\times$ in $p \in \mathbb{P}$, velja

$$y = \alpha^x \pmod{p}$$

ter želimo izračunati $x \in \{1, 2, ..., p-1\}$.

Za praštevilo p, ne glede na velikost le tega, je splošno mnenje med računalničarji, da ne obstaja algoritem, ki bi lahko določil x v polinomskem času. Izčrpno iskanje ključev je zato posebej neučinkovit način iskanja diskretnega logaritma v $\mathbb{Z}_p^\times$. Zaradi te računske kompleksnosti je problem diskretnega logaritma pogosto uporabljen v namene šifriranja.

Za razliko od mulitplikativne grupe $\mathbb{Z}_p^\times$ je iskanje diskretnega logaritma v grupi $\mathbb{Z}_p^+$ zelo preprost problem, ki ga lahko rešimo z uporabo razširjenega Evklidovega algoritma

$$y = \alpha^x = \alpha x \pmod{p} \Rightarrow \alpha x + pl = y$$

za nek $l \in \mathbb{Z}$, kjer so vrednosti α, y, p poznane ter jih obravnavamo kot elemente $\mathbb{Z}$. Rešitev te linearne Diofantske enbačbe je enolična za vrednosti n po modulu p, saj so vsi pari (n, l), ki rešijo enačbo, oblike

$$\left(n_1 - m\frac{p}{\gcd(\alpha, p)}, l_1 + m\frac{\alpha}{\gcd(\alpha, p)}\right) = (x_1 - mp, y_1 + m\alpha)$$

za neko rešitev (n_1, l_1) za poljuben $m \in \mathbb{Z}$. Implikacija velja, ker je α generator grupe $\mathbb{Z}_p^\times$ in posledično tuj p.

6 Diffie-Hellmanova shema ter napad srednjega moža

6.1 Diffie-Hellmanova shema

Denimo, da se Alica in Brane nahajata na javnem mestu (vaškem forumu), kjer nimata možnosti zasebne komunikacije. Da bi slednjo lahko dosegla, bi se želela dogovoriti za zasebni ključ - vrednost, ki je znana le njima. Diffie-Hellmanova shema jima omogoča varno izmenjavo zasebnega kriptografskega ključa z uporabo problema diskretnega logaritma. Predstavimo shemo komunikacij.

1. Alica in Brane se javno dogovorita za praštevilo p ter α, ki je generator grupe $\mathbb{Z}_p^\times$.

2. Skrivno izbereta poljubni naravni števili a in b, ki sta v množici ostankov modulo p.

3. Zaporedoma vsak zase izračunata $A = \alpha^a \pmod p$ ter $B = \alpha^b \pmod p$.

4. Alica in Brane si na javnem forumu izmenjata vrednosti A in B.

5. Oba izračunata skupen zasebni ključ

$$K = \alpha^{ab} = A^b = B^a \pmod p,$$

ki ga uporabljata kot njun zasebni ključ.

V praksi uporabimo še nekaj dodatnih pogojev, da se izognemu napadu izčrpnega iskanja ključev. Želeli bi, da je p čim večji, da preprečimo ugibanje ključa iz elementov $\mathbb{Z}_p^\times$. Prav tako je zaželjeno, da sta reda α^a in α^b čim večja, saj se učinkovitost izčrpnega iskanja ključev v nasprotnem primeru znatno poveča.

Nasprotnik, ki je spremljal komunikacijo med Alico in Branetom, pozna vrednosti $p, \alpha, A = \alpha^a \pmod p$ in $B = \alpha^b \pmod p$, želel pa bi izračunati $K = \alpha^{ab} \pmod p$. Problem diskretnega logaritma zagotavlja, da nasprotnik nima načina, da bi na učinkovit način izračunal a iz A ter b iz B, zaradi česar vrednost K nasprotniku ostaja neznana.

6.2 Napad srednjega moža na Diffie-Hellmanovo shemo

Predstavimo korake napada srednjega moža:

1. Denimo, da Cene prestreže Aličino sporočilo, ki vsebuje $\alpha^a \pmod p$, ter nato Alici pošlje vrednost $\alpha^c \pmod p$ za nek $c \in \mathbb{Z}_p$ pod pretvezo, da je sporočilo poslal Brane.

2. Cene stori enako v svoji komunikaciji z Branetom, namreč prestreže vrednost $\alpha^b \pmod p$ ter mu pošlje vrednost $\alpha^c \pmod p$ pod pretvezo, da je sporočilo poslala Alica.

3. Sedaj sta Alica in Brane pod vtisom, da lahko varno komunicirata z uporabo zasebnega ključa, za katerega sta se dogovorila.

4. Ko Alica pošlje sporočilo, namenjeno Branetu, Alica nevedoč uporabi zasebni ključ $\alpha^{ac} \pmod{p}$, kar omogoča Cenetu, da sporočilo sprejme, ga dešifrira, prebere ter poljubno spremeni. Nato Cene sporočilo zakodira z zasebnim ključem $\alpha^{bc} \pmod{p}$ ter ga pošlje Branetu, ki bo sporočilo prejel misleč, da je prišlo direktno od Alice.

Ta postopek omogoča Cenetu, da poljubno dešifrira ter spreminja komunikacijo med Alico in Branetom, ne da bi rešil problem diskretnega logaritma.

V praksi so uporabljene variacije Diffie-Hellmanove sheme, ki se izognijo napadu srednjega moža z uporabo digitalnih podpisov ter certifikatov. To naredi problem napada na Diffie-Hellmanovo shemo ekvivalentnega rešitvi problema diskretnega logaritma.

7 Digitalni podpis

Digitalni podpis je nadomestek za lastnoročni podpis pri elektronski izmenjavi in digitalnem hranjenju podatkov. Podpis prejemniku besedila omogoča preverjanje izvora in celovitost besedila ter neki tretji osebi preprečuje ponarejanje. Besedila, podpisanega z digitalnim podpisom, ne moremo spreminjati, saj s tem uničimo veljavnost digitalnega podpisa in prejemnik ve, da gre za ponaredek. Prav tako podpisana oseba ne more trditi, da besedila ni podpisala. Podpis je prav tako pomemben za osebi v Diffie-Hellmanovi shemi, da lahko preverita pristnost prejetih podatkov druga od druge pri izmenjavi zasebnega ključa.

Algoritem digitalnega podpisa sestavljajo:

- algoritem generiranja ključa,
- algoritem generiranja digitalnega podpisa sig_A, ki sporočilu priredi podpis osebe A,
- algoritem preverjanja digitalnega podpisa ver_A, ki sporočilu in podpisu priredi vrednost "veljaven"ali "neveljaven"glede na pristnost podpisa.

Sistem za digitalno podpisovanje je peterka $(\mathcal{B}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, pri čemer je:

- $\mathcal{B}$ končna množica besedil,
- $\mathcal{A}$ končna množica podpisov,

- $\mathcal{K}$ končna množica ključev,
- $\mathcal{S}$ končna množica šifrirnih funkcij,
- $\mathcal{V}$ končna množica verifikacijskih funkcij.

Velja še:

- $\forall K \in \mathcal{K},\ \exists sig_K \in \mathcal{S} :\ sig_K : \mathcal{B} \to \mathcal{A}$,
- $\forall K \in \mathcal{K},\ \exists ver_K \in \mathcal{V} :\ ver_K : \mathcal{B} \times \mathcal{A} \to \{true, false\}$,
- $ver_K(sig_K(X)) = true$.

7.1 ElGamalov podpis

ElGamalov digitalni podpis temelji na težavnosti izračuna diskretnega logaritma in tako prejemniku besedila zagotavlja, da lahko preveri njegovo pristnost. Da ustvarimo digitalni podpis, sledimo naslednjim korakom.

1. Izberemo veliko praštevilo p, tako de je težko računati diskretni logaritem po modulu p.
2. Izberemo poljuben generator $\alpha \in \mathbb{Z}_p^\times$.
3. Izberemo ključ k, tako da $1 < k < p-1$.
4. Izračunamo $y = \alpha^x$.
5. Potem je $\mathcal{K} = (\alpha, k, y, x)$, pri čemer so α, k, y javni ključi, x pa je zasebni.

Pošiljatelj besedilo podpiše tako, da izbere poljuben k, za katerega velja $1 < k < p-1$ in $\gcd(k, p-1) = 1$. Nato izračuna $sig_K(m, k) = (r, s)$, kjer je $r = \alpha^k \pmod p$ in $s = (m - xr) \cdot k^{-1}$, pri čemer je m besedilo, ki ga želimo podpisati.

Prejemnik lahko pristnost podpisa preveri z verifikacijsko funkcijo, in sicer velja:

$$ver_K(m, r, s) = T \Leftrightarrow y^r r^s = \alpha^m \pmod p.$$

Dokaz za to pa je naslednji:

$$y^r r^s = y^r \left(\alpha^k\right)^{(m-xr)k^{-1}} = \alpha^{x^r} \cdot \alpha^{m-xr} = \alpha^{xr} \frac{\alpha^m}{\alpha^{xr}} = \alpha^m \pmod p.$$

7.1.1 Napad na El-Gamalov podpis s ponaredkom

Če imamo nek veljaven podpis (r, s) za neko besedilo m, lahko naredimo ponaredek (r', s') za novo besedilo m'. Ponarejen podpis bo pri verifikaciji dal odgovor T $(True)$, a pri tem postopku nimamo vpliva na to, kakšno bo novo besedilo m'. Za izračun ponaredka si najprej izberemo tri velika poljubna števila A, B, C tako, da velja $\gcd(Ar - Cs, p - 1) = 1$. Nato izračunamo še:

$$r' = r^A \alpha^B y^C \pmod{p}$$

in

$$s' = \frac{r's}{Ar - Cs} \pmod{p-1}.$$

Novo besedilo m' dobimo po enačbi

$$m' = \frac{r'(Am + Bs)}{Ar - Cs} \pmod{p-1}.$$

Besedilo m' lahko izračunamo, saj želimo, da je $ver_K(m', r', s') = T$, kar pa velja takrat, ko velja

$$y^{r'} r'^{s'} = \alpha^{m'} \pmod{p}.$$

Iz tega lahko izračunamo besedilo m':

$$\alpha^{m'} = y^{r'} r'^{s'} = y^{r^A \alpha^B y^s} = y^{r' + Cs'} r^{As'} \alpha^{Bs'} \pmod{p}.$$

Potrebno je najti tak t, da bo veljalo

$$y^{r' + Cs'} r^{As'} = (y^r r^s)^t \pmod{p}.$$

Nato primerjamo eksponente

$$r' + Cs' = rt \pmod{p-1}$$

$$As' = st \pmod{p-1}$$

Prvo enačbo pomnožimo s s, drugo z r, ju enačimo ter izpostavimo s'

$$r's + Cs's = srt = As'r \pmod{p-1}$$

$$s'(Ar - Cs) = r's \pmod{p-1}$$

$$s' = \frac{r's}{Ar - Cs} \pmod{p-1}$$

Sedaj lahko iz enačbe $As' = st (\mathrm{mod}\, p-1)$ izračunamo še t:

$$t = \frac{Ar'}{Ar - Cs} \ (\mathrm{mod}\, p-1).$$

Ko vse skupaj združimo, dobimo:

$$\alpha^{m'} = y^{r'+Cs'} r^{As'} \alpha^{Bs'} = \alpha^{m^t} \alpha^{Bs'} \ (\mathrm{mod}\, p).$$

Nato znova primerjamo eksponente in posledično dobimo

$$\begin{aligned} m' &= mt + Bs' (\mathrm{mod}\, p-1) \\ &= m\frac{Ar'}{Ar - Cs} + B\frac{r's}{Ar - Cs} \ (\mathrm{mod}\, p-1) \\ &= \frac{Ar'm + Br's}{Ar - Cs} \ (\mathrm{mod}\, p-1) \\ &= \frac{r'(Am + Bs)}{Ar - Cs} \ (\mathrm{mod}\, p-1) \end{aligned}$$

Par (r', s') je veljaven podpis za besedilo m', pri čemer nas niti ne zanima kaj točno je novo besedilo m', saj želimo le prikriti originalno besedilo m tako, da ga zamenjamo z drugim besedilom m' in je podpis vseeno veljaven. V posebnem primeru, če izberemo $A = 0$ sta nov podpis in besedilo neodvisna od r, in sicer velja

- $r' = \alpha^B y^C \ (\mathrm{mod}\, p)$,
- $s' = -r'C$,
- $m' = -\frac{Br'}{C}$.

Literatura

[1] L. Horjak, 2020. *Modularna aritmetika*. Dostopno na: `http://www.dmfa.si/Tekmovanja/MaSSA/Priprave.aspx`

[2] P. Corn, A. Ellinor, M. Jain, etc. *Diffie-Hellman protocol*. Dostopno na: `https://brilliant.org/wiki/diffie-hellman-protocol/`