UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Računalništvo in matematika – 2. stopnja

Tim Kalan

VEČSTRANSKI PODPISI

Magistrsko delo

Mentor: doc. dr. Tilen Marc

Ljubljana, 2024

Zahvala

Neobvezno. Zahvaljujem se ...

Kazalo

Program dela

Mentor naj napiše program dela skupaj z osnovno literaturo.

Osnovna literatura

1. S. Micali, K. Ohta in L. Reyzin, *Accountable-subgroup multisignatures*, v: Proceedings of the 8th ACM conference on Computer and Communications Security (ur. P. Samarati), ACM, Philadelphia, PA, USA, 2001, str. 245–254, DOI: 10.1145/501983.502017, dostopno na https://doi.org/10.1145/501983.502017.

Podpis mentorja:

Večstranski podpisi

Povzetek

Tukaj napišemo povzetek vsebine. Sem sodi razlaga vsebine in ne opis tega, kako je delo organizirano.

Multisignatures

Abstract

An abstract of the work is written here. This includes a short description of the content and not the structure of your work.

Math. Subj. Class. (2020): 94A60, 11T71
Ključne besede: digitalni podpis, kriptografija
Keywords: digital signature, cryptography

1 Uvod

Odkar se je na svetu pojavil koncept (ročnega) podpisa, je večina primerov uporabe temeljila na pridobivanju podpisov več deležnikov. Odličen primer je npr. Deklaracija neodvisnosti Združenih držav Amerike. SLIKA?.

V prejšnjem stoletju je vzpon računalnika in napredek v kriptografiji privedel do *digitalnih podpisov*. Ti odlično nadomeščajo ročni podpis, prav tako omogočajo, da se skupina podpiše tako, da vsak član poda svoj podpis. Vendar tu lahko z malo matematike poskrbimo, da se skupina lahko podpiše tako, da vsi člani skupaj oddajo en sam podpis, ki priča o podpisu celotne skupine. Tako razbremenimo preverjalca podpisov, kar je ključno v sistemih, kjer je računska moč omejena ali pa draga (npr. pri tehnologiji veriženja blokov).

2 Kriptografske osnove

Preden si lahko pogledamo točno kako lahko skupina generira en sam podpis besedila, si moramo pogledati nekaj kriptografskih osnov. Bolj kompliciranе stvari bodo opisane sproti, ideja tega poglavja je predstaviti stvari, ki so predpogoj za branje kakršnegakoli kriptografskega besedila.

2.1 Aritmetika v $\mathbb{Z}_p^*$

V kriptografiji imamo pogosto opravka z multiplikativnimi grupami, najenostavnejša med njimi (in tudi tradicionalno največ uporabljena) je *multiplikativna grupa naravnih števil modulo p* $\mathbb{Z}_p^*$. Njeni elementi so števila v $\{0, 1, \ldots, p-1\}$, ki so tuja številu p. V posebnem primeru, ko je p praštevilo, so to torej števila {1, 2, ..., p - 1} in je red grupe $\text{ord}(\mathbb{Z}_p^*) = |\mathbb{Z}_p^*| = p - 1$. Operacija v tej grupi je, kot ime že nakazuje, množenje modulo p.

Spomnimo se, da je red elementa g najmanjše naravno število q, da velja $g^q \equiv 1 \pmod{p}$, kjer je 1 enota za množenje. V primeru, da je p praševilo, je grupa grupa $\mathbb{Z}_p^*$ *ciklična*, kar pomeni, da v njej obstaja element g, katerega red je enak redu grupe, torej $\text{ord}(g) = p - 1$. V tem primeru se g imenuje *generator*.

Primer 2.1 (Grupa $\mathbb{Z}_{11}^*$)**.** Ker je 11 praštevilo, v grupi $\mathbb{Z}_{11}^*$ obstaja generator, oz. je grupa ciklična z redom $10 = 11 - 1$. Z zaporednim računanjem potenc lahko vidimo, da je $\text{ord}(2) = 10$, torej je 2 generator.

$$
\begin{aligned}
2^1 &\equiv 2 \pmod{11} & \qquad 2^6 &\equiv 9 \pmod{11} \\
2^2 &\equiv 4 \pmod{11} & \qquad 2^7 &\equiv 7 \pmod{11} \\
2^3 &\equiv 8 \pmod{11} & \qquad 2^8 &\equiv 3 \pmod{11} \\
2^4 &\equiv 5 \pmod{11} & \qquad 2^9 &\equiv 6 \pmod{11} \\
2^5 &\equiv 10 \pmod{11} & \qquad 2^{10} &\equiv 1 \pmod{11}
\end{aligned}
$$

♢

Opomba 2.2. Spomnimo se *kongruence*: $a \equiv b \pmod{m} \iff m \mid a - b$.

2.2 Zgoščevalne funkcije

V grobem so (kriptografske) *zgoščevalne funkcije* funkcije, ki prejmejo poljubno dolg binarni niz (ki lahko predstalja besede, številke, celotne dokumente, ...), vrnejo pa binarni niz, ki ima vnaprej določeno dolžino. Tem rezultatom pravimo *zgostitve*. Namen zgoščevalnih funkcij je za dokument ustvariti unikaten niz, ki zelo verjetno identiificira dokument. V grobem si od zgoščevalnih funkcij želimo naslednje lastnosti:

- **Določenost** pomeni, da bo zgoščevanje enakih nizov vedno privedlo do enake zgostitve.
- **Učinkovitost** pomeni, da lahko računalnik izračuna poljubno zgostitev v doglednem času. Izračun zgostitve mora biti računsko učinkovit.
- **Enosmernost** pomeni, da iz predložene zgostitve zelo težko ugotovimo, kater niz je funkcija prejela kot vhod. Tej lastnosti pravimo tudi *odpornost na prasliko*.
- **Odpornost na drugo prasliko** pomeni, da če poznamo niz in njegovo zgostitev, je zelo težko najdemo drug niz z enako zgostitvijo.
- **Skoraj brez trčenj** pomeni, da je verjetnost, da imata dva izraza enako zgostitev, majhna. Želimo tudi, da je zelo težko najti dva niza z enako zgostitvijo.
- **Učinek plazu** pomeni, da majhna sprememba v vhodnem nizu povzroči veliko spremembo v zgostitvi.

Definicija 2.3. Kriptografska zgoščevalna funkcija $H : \{0,1\}^* \to \{0,1\}^n$ je funkcija, ki slika binarne nize m poljubne dolžine v njihove **zgostitve** $H(m)$, tj. binarne nize vnaprej določene dolžine n. Zadoščati mora naslednjim lastnostim:

- (določenost) $\forall m : ((h_1 = H(m) \land h_2 = H(m)) \implies h_1 = h_2)$.
- (učinkovitost) Izračun funkcije H mora biti računsko učinkovit.
- (odpornost na prasliko) Če poznamo zgostitev h je računsko neizvedljivo najti niz m, da velja $h = H(m)$.
- (odpornost na drugo prasliko) Če poznamo niz m_1 je računsko neizvedljivo najti zgostitev m_2, da velja $H(m_1) = H(m_2)$.
- (odpornost na trčenja) Računsko neizvedljivo je najti dva niza m_1 in m_2, da velja $H(m_1) = H(m_2)$.
- (učinek plazu) Vsaka sprememba vhoda povzroči, veliko spremembo v zgostitvi. Vsak bit zgostitve se spremeni z verjetnostjo vsaj $1/2$.

Primer 2.4. Ena izmed najbolj znanih zgostitvenih funkcij je `SHA-256`. Njeno ime pomeni *Secure Hash Algorithm* (slov. varen zgostitveni algoritem), 256 pa predstavlja dolžino zgostitve. Pogostokrat to ime zasledimo pri nameščanju programske opreme, služi kot avtentikator, da smo res naložili pravo stvar.

Za primer si lahko ogledamo zgostitvi dveh podobnih nizov, *Ljubljana* in *Ljubljena*. Kljub podobnosti bomo videli, da sta rezultata popolnoma drugačna, kar si tudi želimo pri zgostitvenih funkcijah.

```
SHA-256(Ljubljana) =
b7f147d8b4a6703a951336654355071f9752385f85d0860379e99b484aee7a82

SHA-256(Ljubljena) =
995d2d8ffb40e1838219e65dd2c665701ba34a90e11f7195a4b791838b6787fe
```

Za preglednost nismo prevajali besed v binarne nize, to bi storili npr. z `ASCII` ali `UTF-8` tabelo. Prav tako smo rezultat napisali v šestnajstiškem sistemu, saj je tako krajši. ♢

2.3 Kriptografija javnega ključa

Prve šifre, ki smo jih uporabljaji ljudje, so bile *simetrične*, kar pomeni, da sta osebi za komunikacijo obe morali poznati skriven *ključ*, ki je definiral, kako je bila šifra ustvarjena.

Primer 2.5 (Cezarjeva šifra)**.** Ena najbolj znanih šifer, ki izvira iz Antičnega Rima, je *Cezrjeva šifra*. Njen ključ je število, ki je krajše od dolžine naše abecede, v Cezarjevem primeru je bilo to število 3. Šifra potem deluje tako, da vsako črko zamaknemo za toliko mest v abecedi, kolikor definira ključ. Npr. za slovensko abecedo, bi šifra zamaknila črke:

```
A B C Č D E F G H I J K L M N O P R S Š T U V Z Ž
Č D E F G H I J K L M N O P R S Š T U V Z Ž A B C
```

To bi izraz `JAVNI KLJUČ` preslikalo v `MČARL NOMŽF`. Cezarjeva šifra se imenuje tudi *zamična šifra*. ♢

V prejšnjem stoletju pa se je pojavila alternativa, imenovana *asimetrična kriptografija*, oz. *kriptografija javnega ključa*. Glavna prednost te je, da osebi za komunikacijo ne rabita poznati enakega skrivnega ključa, vendar ima vsak od njiju par ključev, ki ju imenujemo *javni ključ* (angl. *public key*) in *zasebni ključ* (angl. *secret/private key*)in označimo kot par $(\mathrm{pk}, \mathrm{sk})$. Vsaka oseba objavi svoj javni ključ in poskrbi, da nihče ne izve, kaj je njen zasebni ključ.

Šifriranje potem poteka tako, da pridobimo javni ključ od osebe, s katero želi komunicirati, ga uporabi za šifriranje in objavi šifrirano sporočilo. Lastnik ustreznega zasebnega ključa (vsakemu javnemu pripada natanko en zasebni) potem pridobi šifrirano sporočilo in ga z zasebnim ključem odšifrira. Kriptosistemi delujejo na način, da lahko sporočilo, šifrirano z javnim ključem odšifrira samo ustrezen zasebni ključ. Tako zagotovimo varno komunikacijo.

Primer 2.6 (RSA)**.** En prvih algoritmov javnega ključa, ki se uporablja še danes, je *RSA*. Njegova varnost izhaja iz (domnevne) težavnosti problema iskanja prafaktorjev. Svoj ključ definiramo tako, da si izberemo dve (zelo veliki) praštevili p in

q, ter ju zmnožimo v $n = pq$. Za primer vzemimo $p = 23$ in $q = 17$. n je potem enak 391. Izbrati si moramo še eksponent e, vzemimo npr. $e = 3$. Naš javni ključ je potem par

$$(n, e) = (391, 3).$$

Postopek šifriranja poteka tako, da oseba, s katero komuniciramo, izbere sporočilo m, npr. $m = 10$, pridobi naš javni ključ, in izračuna šifro c kot

$$c = m^e \bmod n = 10^3 \bmod n = 218.$$

Dogovoriti se moramo še o zasebnem ključu. Za to bomo potrebovali eksponent za odšifriranje d, tako da bo veljalo

$$(m^e)^d \equiv 1 \pmod{\varphi(n)},$$

kjer φ označuje Eulerjevo funkcijo fi. Iščemo torej multiplikativni inverz eksponenta e, modulo $\varphi(n)$. V našem primeru je to $d = 235$. Zasebni ključ je potem

$$(p, q, d) = (23, 17, 235).$$

Iz zasebnega ključa torej lahko kadarkoli izračunamo javnega, saj enostavno zmnožimo p in q ter izračunamo inverz, v splošnem pa iz n učinkovito ne moremo pridobiti faktorjev p in q, kar nam daje varnost.

Ko prejmemo šifrirano sporočilo c, ga odšifriramo tako, da izračunamo

$$m = c^d \bmod n = 218^{235} \bmod 391 = 10.$$

♢

Poleg šifriranja, brez da bi si delili ključ, pa je kriptografija javnega ključa omogočila tudi *digitalne podpise*. Ti so uporabljeni vsakič, ko pošljemo e-pošto ali dostopamo do katerekoli spletne strani. Delujejo na podoben način, kot šifriranje z javnim ključem, le da najprej uporabimo zasebni ključ na sporočilu, prek javnega ključa pa preverjamo veljavnost podpisa. Ponavadi sta šifrianje in podisovanje uporabljena hkrati, saj tako pošljemo šifrirano sporočilo, za katerega lahko oseba, s katero komuniciramo preveri, da je res prišlo od nas.

2.4 Digitalni podpisi

Ideja *kriptografskih* ali *digitalnih* podpisov je, da služijo kot izboljšava človeškega ročnega podpisa. Za razliko od ročnega podpisa, lahko z digitalnim dosežemo pravo identifikacijo posameznika, ki temelji na njegovem zasebnem ključu. Tako smo lahko za digitalno podpisan dokument prepričani, da ga je res podpisal lastnik točno določenega zasebnega ključa.

Podpis dokumenta poteka nekoliko drugače, kot pri ročnih podpisih. Pri ročnem podpisu ta postane del dokumenta, digitalni podpis pa je od njega ločen, vseeno pa nastane s pomočjo zgostitve podpisanega dokumenta, zato bo podpis za dva različna dokumenta vedno drugačen.

Ostane še vprašanje preverjanja avtentičnosti podpisa. Pri ročnem podpisu to lahko storimo prek primerjave z znanim, preverjeno avtentičnim podpisom. Ta

postopek je zamuden in nenatančen, veliko večino ročnih podpisov je moč ponarediti z nekaj prakse. Preverjanje digitalnega podpisa pa temelji na kriptografiji javnega ključa. Ker je podpis nastal s pomočjo podpisnikovega zasebnega ključa, lahko s pomočjo ujemajočega javnega ključa preverimo avtentičnost.

Definicija 2.7. Digitalni ali **kriptografski podpis** $\mathcal{S} = (G, S, V)$ je trojica učinkovitih algoritmov G za ustvarjanje ključa, S za podpisovanje in V za preverjanje podpisa. Definirana je nad končno množico možnih sporočil $\mathcal{M}$, vrnjeni podpis pa leži v v končni množici podpisov Σ.

- G je naključnostni algoritem za ustvarjanje para ključev (pk, sk), ki ne prejme nobenega argumenta. pk je javni kljč za preverjanje avtentičnosti podpisa, sk pa je zasebni ključ za podpisovanje.
- S je naključnostni algoritem, ki za svoja argumenta prejme zasebni ključ sk in sporočilo m, vrne pa podpis σ spročila m z zasebnim ključem sk oz.

$$\sigma = S(\text{sk}, m).$$

- V je deterministične algoritem, ki preverja veljavnost podpisov. Za svoje arugmente prejme javni ključ pk, sporočilo m in podpis σ, vrne $veljaven$, če je podpis veljaven in $neveljaven$, sicer. Velja torej

$$V(\text{pk}, m, \sigma) = \begin{cases} veljaven, & \sigma = S(\text{sk}, m), \\ neveljaven, & \sigma \neq S(\text{sk}, m). \end{cases}$$

2.5 Varnost

Glavna stvar, ki nas zanima pri obravnavi kateregakoli kriptosistema, je njegova *varnost*. Ker je cilj digitalnih podpisov sogovorniku zagotoviti, da sporočilo res pošiljamo mi, nas glede varnosti najbolj skrbi, da bi *napadalec* lahko ponaredil naš podpis in nam s tem ukradel identiteto. Pri tem je lahko uspešen na več nivojih, ki so od najmanj do najbolj škodljivega:

- **Eksistencialno ponarejanje** (angl. *existential forgery*) pomeni, da napadalec lahko ponaredi vsaj en podpis. To pomeni, da lahko najde vsaj en par (m, σ), da velja $V(\text{pk}, m, \sigma) = veljaven$.
- **Selektivno ponarejanje** (angl. *selective forgery*) pomeni, da lahko napadalec z nezanemarljivo verjetnostjo podpiše sporočilo, ki mu ga da nekdo drug in ga mi še nismo podpisali. Torej, če napadalcu nekdo predloži spročilo m, lahko z nezanemarljivo verjetnostjo najde podpis σ, da velja $V(\text{pk}, m, \sigma) = veljaven$.
- **Popoln zlom** (angl. *total break*) pomeni, da je napadalec ugotovil naš zasebni ključ in s tem podpisovalni algoritem. V našem imenu lahko podpiše karkoli.

Poleg zgoraj definiranih *ciljev napadalca*, lahko za vsak kriptosistem definiramo tudi *model napada*, in pa *tip varnosti*, ki ga zagotavlja shema. Varnost večine shem za digitalne podpise temelji na (domnevni) težavnosti določenih matematičnih problemov.

Stinson [3] definira naslednje modele napada:

- **Napad samo s ključem** je napad, kjer napadalec pozna samo naš javni ključ pk. Pozna torej algoritem za preverjanje podpisov V.
- **Napad z znanimi sporočili** je napad, kjer napadalec poseduje seznam parov sporočil in njihovih podpisov $(m_1, \sigma_1), (m_2, \sigma_2), \ldots$, kjer za vsak i velja $\sigma_i = S(\text{sk}, m_i)$.
- **Napad z izbranimi sporočili** je napad, kjer nam napadalec da seznam sporočil $m_1, m_2, \ldots$, mi pa mu vrnemo seznam podpisov, da za vsak i velja $\sigma_i = S(\text{sk}, m_i)$.

Ostane nam še predled varnosti, ki jo lahko pričakujemo oz. zahtevamo od sheme za podpisovanje. Takšna shema ne more biti *brezpogojno varna*, kar bi pomenilo, da je tudi z neomejenimi računskimi zmožnostmi nemogoče ponarediti podpis. To je zato, ker lahko napadalec sistematično preveri vse podpise za neko sporočilo s pomočjo algoritma V, dokler ne najde pravega. Pričakujemo pa lahko *računsko varnost*, kar pomeni, da napadalec ne more najti ponaredka v doglednem času, če ima omejene računske sposobnosti, ali pa *dokazljivo varnost*, kar pomeni, da lahko varnost prevedemo na težavnost nekega matematičnega problema.

3 Schnorrov podpis

Eden izmed najenostavnejših, dokazano varnih podpisov je ravno *Schnorrov podpis*. Kot vsi podpisi, tudi ta potrebuje tri algoritme: za generiranje ključa, podpisovanje in preverjanje podpisa.

Za generiranje para ključev, je potrebno najprej generirati dve praštevili p in q, tako da q deli $p - 1$. Potem je potrebno izbrati element g iz multiplikativne grupe modulo p, torej $g \in \mathbb{Z}_p^*$, ki je reda q. Za konec je potreben še izbor naključnega števila $s \in [0, q-1]$ in izračun

$$I = g^s \bmod p.$$

Ko vse to opravimo, smo uspešno ustvarili par ključev

$$\begin{aligned} \text{pk} &= (p, q, g, I), \\ \text{sk} &= s. \end{aligned}$$

Ideja v ozadju teh števil je, da p določa multiplikativno grupo števil $\mathbb{Z}_p^*$. Za podpis je potrebno najti podgrupo, katere red je praštevilo, je pa vseeno dovolj velika, da omogoča varnost. Red te podgrupe je q, določa pa jo generator g. Iz varnostnih razlogov mora p imeti 2048 bitov, q pa 224. Čeprav je grupa $\mathbb{Z}_p^*$ zelo velika, je Schnorrov podpis vseeno učinkovit, saj večinoma deluje znotraj podgurpe, ki jo generira g.

KAKO DOBIMO PARAMETRE??

Poleg parametrov p, q in g, morata podpisnik in preverjalec določiti oz. imeti dostop do varne kriptografske zgoščevalne funkcije $H : \{0,1\}^* \to \mathbb{Z}_q$. Za varno funkcijo smatramo vsako, ki zadošča lastnostim iz definicije 2.3. Velikost kodomene te funkcije definira velikost končnega podpisa. Iz zgostitve, dolge $\log_2 q$ bitov, dobimo podpis, dolg $2 \log_2 q$ bitov [3].

Za podpis enega sporočila mora podpisnik generirati naključno število $r \in [0, q-1]$ in izračunati *zavezo*

$$X = g^r \bmod p.$$

Ta korak je podoben zadnjemu delu generiranja ključev, le da je skrivni del ključa s uporabljen večkrat, r pa mora biti generiran za vsako sporočilo znova. Potem z uporabo funkcije H izračunamo *izziv*

$$e = H(X||m),$$

kjer $||$ označuje stikanje nizov. Za konec je potrebno izračunati še

$$y = es + r \bmod q,$$

podpis sporočila m pa je potem par (X, y) oz.

$$S(s, m) = (X, y).$$

Za preverjanje veljavnosti podpisa (X', y') sporočila m, je potrebno izračunati

$$e' = H(X'||m)$$

in preveriti, če velja

$$g^{y'} \stackrel{?}{\equiv} X' \cdot I^{e'} \pmod p. \tag{3.1}$$

Za to moramo uporabiti nekaj lastnosti cikličnih grup in modularne aritmetike.

Trditev 3.1. *Naj bosta p in q praštevili, $q \mid p-1$. Naj bo g element grupe $\mathbb{Z}_p^*$ reda q, kar pomeni, da je $g^q \equiv 1 \pmod p$. Naj bo k naravno število. Potem velja*

$$g^k \bmod p = g^{k \bmod q} \bmod p.$$

Dokaz. Po osnovnem izreku o deljenju naravnih števil, lahko k na en sam način zapišemo kot $k = nq + r$, kjer velja $n \in \mathbb{N}, r < q$.

Leva stran enačbe se potem prepiše

$$\begin{aligned} g^k \bmod p &= g^{nq+r} \bmod p = \\ &= (g^q)^n g^r \bmod p = \\ &= 1^n g^r \bmod p = \\ &= g^r \bmod p. \end{aligned}$$

Desna stran pa se prepiše kot

$$\begin{aligned} g^{k \bmod q} \bmod p &= g^{(nq+r) \bmod q} \bmod p = \\ &= g^r \bmod p. \end{aligned}$$

Ker sta obe strani enaki, je trditev dokazana. □

Po trditvi 3.1 lahko levo stran enačbe za preverjanje Schnorrovega podpisa (3.1) prepišemo

$$\begin{aligned} g^{y'} \bmod p &= g^{es+r \bmod q} \bmod p = \\ &= g^{es+r} \bmod p. \end{aligned}$$

Za pretvorbo desne strani moramo uporabiti nekaj lastnosti modularne aritmetike.

Trditev 3.2. *Naj bodo a, b in p naravna števila. Potem za modularno množenje in potenciranje velja*

$$a \cdot b \bmod p = (a \bmod p) \cdot (b \bmod p) \bmod p, \tag{3.2}$$

$$a^b \bmod p = (a \bmod p)^b \bmod p. \tag{3.3}$$

Dokaz. a in b lahko po osnovnem izreku o deljenju naravinih števil na en sam način zapišemo kot

$$a = n_a p + r_a,$$
$$b = n_b p + r_a,$$

kjer velja $r_a < p$.

(3.2): Levo stran preoblikujemo

$$\begin{aligned} a \cdot b \bmod p &= (n_a p + r_a) \cdot (n_b p + r_b) \bmod p = \\ &= (n_a n_b p^2 + n_a p r_b + n_b p r_a + r_a r_b) \bmod p = \\ &= r_a r_b \bmod p, \end{aligned}$$

desno pa

$$\begin{aligned} (a \bmod p) \cdot (b \bmod p) \bmod p &= (n_a p + r_a \bmod p) \cdot (n_b p + r_b \bmod p) \bmod p = \\ &= r_a r_b \bmod p. \end{aligned}$$

Ker se strani ujemata, je trditev dokazana.

(3.3): Ker je potenciranje samo zaporedna uporaba množenj, lahko trditev pokažemo z indukcijo na b in enačbo (3.2):

- $b = 2$: Primer, ko je $b = 1$ (ali $b = 0$) je trivialen, če pa je $b = 2$, pa se problem reducira v
$$a \cdot a \bmod p \stackrel{?}{=} (a \bmod p) \cdot (a \bmod p) \bmod p,$$
kar drži neposredno po enačbi (3.2).

- $n \to n + 1$: Predpostavimo, da enačba (3.3) drži za $b = n$ (I.P.). Ko je $b = n + 1$, dobimo
$$\begin{aligned} a^{n+1} \bmod p = a^n a \bmod p &= \\ &\stackrel{(3.2)}{=} (a^n \bmod p)(a \bmod p) \bmod p = \\ &\stackrel{\text{I.P.}}{=} (a \bmod p)^n (a \bmod p) \bmod p = \\ &= (a \bmod p)^{n+1} \bmod p. \end{aligned}$$

S tem je indukcija končana in trditev dokazana. □

Desno stran enačbe (3.1) torej lahko prepišemo

$$\begin{aligned} X' \cdot I^{e'} \bmod p &= g^r \bmod p \cdot (g^s \bmod p)^e \bmod p = \\ &= (g^r \bmod p) \cdot (g^{es} \bmod p) \bmod p = \\ &= g^{es+r} \bmod p, \end{aligned}$$

kjer smo pri prehodu v drugo vrstico uporabili lastnost (3.3), pri prehodu v tretjo pa lastnost (3.2). Ker se obe strani ujemata za veljavne podpisne vrednosti, ta enačba res preveja Schnorrov podpis.

KAKO PA POKAŽEMO, DA SE ZA NAPAČNE VREDNOSTI NE UJEMA??

3.1 Varnost Schnorrovega podpisa

Najbolj očitna nevarnost kateregakoli kriptosistema z javnimi ključi bi bila možnost izračuna zasebnega ključa iz javnega. Slednji je dostopen vsem, zato bi lahko kdorkoli pridobil zasebni ključ, kar popolnoma izniči pomen šifriranja ali podpisovanja.

Pri Schnorrovem podpisu je javni ključ poleg parametrov uporabljene grupe p, q in g, še število I, izračunano kot

$$I = g^s \bmod p.$$

Za izračun je torej neposredno uporabljen zasebni ključ s. Zaradi notacije bi morda kdo hitro pomislil, da lahko zgornjo enačbo obrnemo in s izračunamo kot

$$s = \log_g(I) \bmod p.$$

Taki izračuni v grupah $\mathbb{Z}_p^*$ žal niso tako enostavni, prišli smo do koncepta *diskretnega logaritma*.

Definicija 3.3 (Problem diskretnega logaritma [1])**.** Naj bo G ciklična grupa reda q, ki jo generira element g. Naj bo h naključni element iz grupe G. Naj velja $g^x = h$. x Potem imenujemo **Diskretni logaritem (DL)**.

Zamislimo si igro, kjer izzivalec in nasprotnik kot vhod prejmeta opis grupe G (torej q in $g \in G$). Izzivalec potem izbere naključen element $\alpha \in G$ in izračuna $h = g^\alpha$. h pošlje nasprotniku, le-ta pa mora odgovoriti nazaj z elementom α. To igro imenujemo **problem diskretnega logaritma (PDL)** (angl. *discrete logarithm problem*).

Pri tej igri nas zanima verjetnost pravilnega odgovora nasprotnika, ki je računsko omejen. S tem mislimo, da ima na voljo polinomsko mnogo časa (glede na velikost grupe). Če je grupa G takšna, da je verjetnost zanemarljiva, pravimo, da za grupo G drži *predpostavka diskretnega logaritma*.

Izkaže se, da za grupe, kot je $\mathbb{Z}_p^*$, ne poznamo učinkovitega algoritma za izračun diskretnega logaritma, torej v njih drži predpodstavka DL. To torej pomeni, da ob pridobljenem javnem ključu, napadalec ne more učinkovito izračunati zasebnega.

4 Pregled skupinskih podpisov

Ko pridemo do podpisovanja skupin, si lahko zamislimo več različnih rešitev. Micali v [2] definira dve lastnosti oz. spektra, ki jim lahko zadošča podpis skupine:

- **Prilagodljivost** (angl. *flexibility*): Popolnoma prilagodljiv podpis skupine je takšen, ki ga lahko proizvede katerakoli podskupina originalne skupine podpisnikov. Ko je podpis preverjen, se mora tisti, ki ga je preveril odločiti, če je ustrezen del skupine podal svoj podpis. Popolnoma neprilagodljiv podpis bi bil takšen, ki ga lahko v imenu skupine ustvari katerkoli član.

- **Odgovoronost** (angl. *accountability*): Če lahko iz podpisa ugotovimo, kateri člani so sodelovali pri ustvarjanju, nam podpis omogoča odgovornost. Ta lastnost je lahko zaželena, če se želimo prepričati, ali je ustrezen del skupine

sodeloval pri podpisu (npr. ali je pri podpisovanju sodeloval generalni direktor podjetja). V drugih primerih pa si želimo anonimnost posameznih članov (npr. če bi generiranje podpisa predstavljalo nekakšno glasovanje, bi želeli vedeti samo, koliko članov je sodelovalo).

V nadaljevanju bomo skupino potencialnih podpisnikov (torej podpisnikov, ki imajo možnost sodelovati pri podpisovanju) označili z $G = P_1, \ldots, P_L$, kjer ima skupina L članov. Dejanski podpis pa bo generiral samo del skupine $S \subseteq G$.

4.1 Skupinski podpisi

Skupinski podpis (angl. *group signature*) v imenu celotne skupine G ustvari en anonimen član. To torej pomeni, da je podpis popolnoma neprilagodljiv, saj ni možno prisiliti skupine, da bi podpis ustvaril več kot en član. Prav tako v splošnem noben član, niti tisti, ki preverja podpis, ne more ugotoviti, kdo je podpis ustvaril. Da skupinski podpisi omogočijo vsaj delno odgovornost, skupina določi *vodjo skupine*, ki ima možnost razkriti identiteto podpisnika, če pride do težav. V tem primeru seveda vodja predstavlja atraktivno tarčo za napad.

4.2 Pragovni podpisi

Če želimo zagotoviti, da se s podpisom strinja zadosten delež skupine, lahko uporabimo *pragovni podpis* (angl. *threshold signature*). Ta nam omogoča določeno mero prilagodljivosti, saj lahko katerkoli zadosten delež skupine ustvari podpis. Še vedno je nemogoče upoštevati morebitno hierarhično strukturo skupine. Pragovni podpisi omogočajo tudi popolno anonimnost podpisnikov, in s tem torej nično odgovornost. Intuicija tu je, da večina pragovnih podpisov temelji na interpolaciji polinoma $(l-1)$-stopnje z l točkami. Podpis je potem ustvarjen s pomočjo vrednosti polinoma v neki točki. Po interpolaciji se informacija o tem, točno katere točke smo uporabili, izgubi. Take podpise imenujemo tudi *l-od-L sheme*. Primer uporabe je odklepanje sefa v banki. Recimo, da ne zaupamo samo osebi z odklepanjem in želimo, da je prisotnih l od L pooblaščenih oseb, ni nam pa važno, katerih. Tu je pragovni podpis odlična rešitev.

4.3 Večstranski podpisi

Za nekatere uporabe podpisov, bi si od njih želeli podobne lastnosti, kot jih ima večstranski ročen podpis. Pri njem lahko hitro preberemo podpisnike, torej imamo popolno prilagodljivost. Vidimo lahko seznam podpisnikov, torej lahko presodimo, če so med njimi tisti, ki smo jih želeli. Prav tako podpisniki nosijo popolno odgovornost, saj na papirju piše njihovo ime.

Podoben učinek bi z digitalnimi podpisi lahko dosegli, če bi namesto enega podpisa skupine, od članov zbrali individualne podpise in jih nanizali v seznam. Dobili bi torej digitalni podpis skupine, ki ponuja popolno prilagodljivost in odgovornost. Težava je samo, da je dolžina podpisa (in s tem čas preverjanja) proporcionalna številu podpisnikov. *Večstranski podpisi* (angl. textitmultisignatures) ohranijo lastnosti seznama podpisov, rezultat sheme je pa en sam podpis, ki je enako dolg ne glede

na število podpisnikov, prav tako je od števila neodvisen čas preverjanja. So torej odlična posplošitev ročnih podpisov skupin, ki vseeno ohrani učinkovito preverjanje.

DEFINICIJA??

4.4 Agregirani podpisi

5 Večstranski Schnorrov podpis

Povzeto po [2].

6 Večstranski podpisi v splošnem

Literatura

[1] D. Boneh in V. Shoup, *A graduate course in applied cryptography*, Stanford University, Stanford, 2023.

[2] S. Micali, K. Ohta in L. Reyzin, *Accountable-subgroup multisignatures*, v: Proceedings of the 8th ACM conference on Computer and Communications Security (ur. P. Samarati), ACM, Philadelphia, PA, USA, 2001, str. 245–254, DOI: 10.1145/501983.502017, dostopno na https://doi.org/10.1145/501983.502017.

[3] D. R. Stinson in M. B. Paterson, *Cryptography: theory and practice*, Textbooks in Mathematics, CRC Press, 2018.