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

Računalništvo in matematika – 2. stopnja

Tim Kalan

SKUPINSKO GENERIRANI 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:

Skupinsko generirani 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 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*. V grobem si od zgoščevalnih funkcij želimo naslednje lastnosti:

- **Določenost** pomeni, da bo zgoščevanje enakih nizov vedno privedlo do enakega odgovora.
- **Učinkovitost** pomeni, da lahko računalnih izračuna poljubno zgostitev v doglednem času.
- **Enosmernost** pomeni, da iz predložene zgostitve zelo težko ugotovimo, kater niz je funkcija prejela kot vhod.
- **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.

Primer 2.1. Ena izmed najbolj znanih zgostitvenih funkcij je `SHA-256`. Njeno ime pomeni *Secure Hashing 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. ♢

2.1.1 Model slučajnega oraklja

2.2 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 2.2 (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 2.3 (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.

2.3 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 2.4. 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 deterministično 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

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

2.3.1 Ustvarjanje ključa

2.3.2 Podpisovanje

2.3.3 Preverjanje podpisa

2.3.4 Primer: Schnorrov podpis

2.4 Drugo

Merklova drevesa, dokazi brez razkritja znanja, slučajni oraklji, fiat-shamirjeva hevristika,

3 Matematično ozadje

3.1 Modularna aritmetika

kongruence, modulo,

3.2 Multiplikativne grupe modulo n

multiplikativni inverz, red elementa, diskretni logaritem

4 Pregled skupinskih podpisov

5 Skupinsko generirani podpisi na podlagi Schnorrovega podpisa

Povzeto po [1].

6 Skupinsko generirani podpisi v splošnem

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.