Digitalna poštna znamka

Projektna naloga pri predmetu Kriptografija in teorija kodiranja 2

Darko Pevec

avgust 2009

Povzetek

Poštne znamke nas spremljajo že več kot 150 let, a v današnjem visokotehnološkem svetu je običajna papirnata poštna znamka postala zelo izpostavljena ponarejanju. Visoko zmogljivi tiskalniki so postali dostopni vsakomur, cena tiska je nizka in zato se ponarejanje izplača. Tako je nastala potreba po novih, varnih načinih evidentiranja plačila poštnine.

Kazalo

1 Uvod

V tem delu se bomo spopadli s problematiko evidentiranja plačila poštnine, s kriptografskega vidika. Večina poštnih organizacij zahteva predplačilo za svoje storitve, kar pomeni, da morajo biti vse poštne pošiljke opremljene s preverljivo označbo tega predplačila.

Zgodovinsko prvi primer take označbe predstavlja poštna znamka, ki se je pojavila šele v sredini 19. stoletja. Običajna znamka ima kljub mnogim dobrim lastnostim, tudi svoje pomankljivosti. Njeni dve največji hibi sta enostavno ponarejanje ter draga izdelava in distribucija. Ravno ta dva problema bomo rešili s pomočjo kriptografije in sodobne računalniške infrastrukture, tj. interneta.

V sledečih podrazdelkih najprej naredimo vpogled v zgodovino same znamke, nato si ogledamo kakšni so današnji problemi ter uvod zaključimo s postavitvijo zahtev, katere mora digitalna poštna znamka upoštevati. V nadaljnjih poglavjih bomo najprej predstavili zelo preprost in eleganten koncept digitalne poštne znamke, si zatem ogledali potrebna in primerna kriptografska orodja ter nato še pokazali njeno varnost.

1.1 Vpogled v zgodovino

Ljudje so si izmenjevali sporočila še preden so Egipčani izumili papirus. Ob izumu papirja, so sporočila postala zapisana in vsaka kultura oz. vladavina je za svoje potrebe ustanovila svojo kurirsko službo. Prvo dobro dokumentirano poštno službo so imeli Rimljani. Ustanovljena je bila za časa cezarja Avgusta (62 p.n.š – 14 n.š) in velja za prvo pravo poštno službo, saj so poštne storitve prvič postale dostopne navadnim državljanom.

Sprva so stroške plačevali prejemniki, a v 19. stoletju, ko so države ustanavljale in imele v lasti nacionalne poštne službe, se je pojavila tudi prva poštna znamka. Tako je stroške poštnih storitev prevzel pošiljatelj in znamka je služila kot dokazilo predplačila storitve.

Znano je, da je že leta 1835 Lovrenc Košir predlagal dunajski pošti uvedbo znamke, a je bil njegov predlog sprva zavrnjen. Zaradi tega je splošno sprejeti izumitelj poštne znamke postal Rowland Hill, ki je leta 1840 v Veliki Britaniji vpeljal znamko poimenovano *Penny Black*. Leta 1843 sta z vpeljavo poštne znamke sledili najprej Švica in Brazilija, v naslednjih dveh desetletjih pa se je poštna znamka razširila po celem svetu.

1.2 Problemi navadne znamke

Čeprav poštni sistem v današnji dobi velja za manj učinkovitega od ostalih naprednejših načinov komunikacije, ostaja še vedno edini univerzalni sistem ra-

znašanja sporočil. Razlogov za to je kar nekaj: ponuja razvejano infrastrukturo za raznos po sprejemljivi ceni, ima prednost varnega prenosa in omogoča, da pošta še vedno ostaja zakonska podlaga vsakršnega poslovanja.

Poštne znamke so se izkazale za dobre na veliko načinov, imajo pa tudi precej pomanjkljivosti. Izdelava in distribucija sta dragi, poleg tega so tudi predmet kraje. Znamke same ne nosijo nobenih informacij o času pošiljanja ali o prejemniku. Vemo tudi, da ponujajo le omejeno varnost plačevanja poštnine. Frankirni stroji so težave nekoliko omilili, njihova varnost pa temelji na predpostavki, da ni mogoče izdelati učinkovitega ponaredka poštne znamke. To pa v današnjih časih, ko so se pojavili tiskalniki visoke kvalitete in ki so dostopni vsakomur, enostavno ne drži več.

Razvoj digitalne tehnologije je doprinesel k dramatičnim spremembam v procesu pisanja poštnih pošiljk, njihovi obdelavi in celo raznašanju. Ocene iz leta 2000 pravijo, da je približno 80% pošte, ki nastane v industriji, napisane s pomočjo računalnika in tiskalnika ter da je vsaj polovica teh računalnikov povezanih v medmrežje [1]. Napačno bi bilo misliti, da ta odstotek ni in ne bo več rasel. Ravno zato je naša digitalna poštna znamka zasnovana tako, da jo je moč kupiti preko interneta in jo natisniti na navadnem domačem tiskalniku.

1.3 Cilji in zahteve

Preden se lotimo načta, si moramo najprej zastaviti cilje in potrebe, katerim želimo zadostiti. Kaj pričakujemo od nove znamke?

1. Znamka naj bo *varna*.
   Kar tukaj zares mislimo je, da od znamke pričakujemo pametno uporabo kriptografije. Zadali smo si naslednje cilje:

   - enostavno, tj. hitro preverljive avtentičnost, integriteta in veljavnost – želimo, da lahko kdorkoli preveri avtentičnost in veljavnost znamke, z uporabo čim bolj računsko nezahtevnih algoritmov.
   - odpornost proti goljufijam – morebitni poskusi zlorabe naj bodo neizvedljivi ali neizplačljivi.
   - anonimnost pošiljatelja – to lastnost navadne znamke želimo obdržati, saj ima pošiljatelj pravico do anonimnosti.
   - preprečitev večkratne uporabe – nimamo nadzora nad tem, koliko kopij veljavne znamke uporabnik ustvari. Zato potrebujemo mehanizem, s katerim preprečimo večkratno uporabo iste znamke.
   - odpornost na manjše poškodbe – znamke se lahko namerno ali nenamerno poškodujejo, zato potrebujemo dodatno robustnost zapisanih podatkov.

2. Znamka naj omogoča hrambo dodatnih informacij.
   Dodatne informacije potrebujemo za namene kriptografske varnosti, hkrati pa odpirajo možnosti za dodatne storitve, lažjo strojno obdelavo, itd. Sprva zahtevajmo vključitev naslednjih podatkov:

   - serijska oznaka
   - datum izdaje
   - datum zapadlosti
   - naslov prejemnika
   - poštna št. prejemnika
   - tip in podtip znamke
   - vrednost plačane poštnine
   - možnost razširitve

3. Znamka naj bo enostavno dostopna.
   Konvencionalnih stroškov tiska in distribucije se lahko znebimo tako, da tisk prepustimo uporabnikom:

   - nakup preko interneta
   - tisk na povprečnem domačem tiskalniku (vsaj 300dpi)

4. Znamka naj ustreza fizičnim omejitvam.

   - površina znamke naj bo pod $25mm \cdot 25mm$, tj. $6,25cm^2$
   - branje naj bo možno tudi z nizkocenovnimi čitalci (ločljivost vsaj $0,25mm$)

2 Predlog digitalne poštne znamke

V tem razdelku bomo sestavili novo, svojo digitalno poštno znamko, katero bi lahko uporabljali v Sloveniji. Usmerjali nas bodo vtisi že obstoječih rešitev (USPS IBIP [6], Royal Mail SmartStamp in Deutsche Post StampIt). Želimo, da je zasnova čim enostavnejša in da ne uporablja konceptov podvrženih plačljivim patentom.

2.1 Ideja - proti protokolu

Najprej semantično predstavimo, kako deluje naša rešitev. Začnimo z enostavno skico trenutnega postopka nakupa in uporabe poštne znamke:

1. Stranka gre na pošto,
2. Izrazi, da želi kupiti poštne znamke,
3. Izvrši plačilo,
4. Prejme znamke,
5. Znamke nalepi na pisemske ovojnice,
6. Pisma odda v najbližji nabiralnik,
7. Pošta pred sortiranjem preveri ustrezno plačano poštnino.

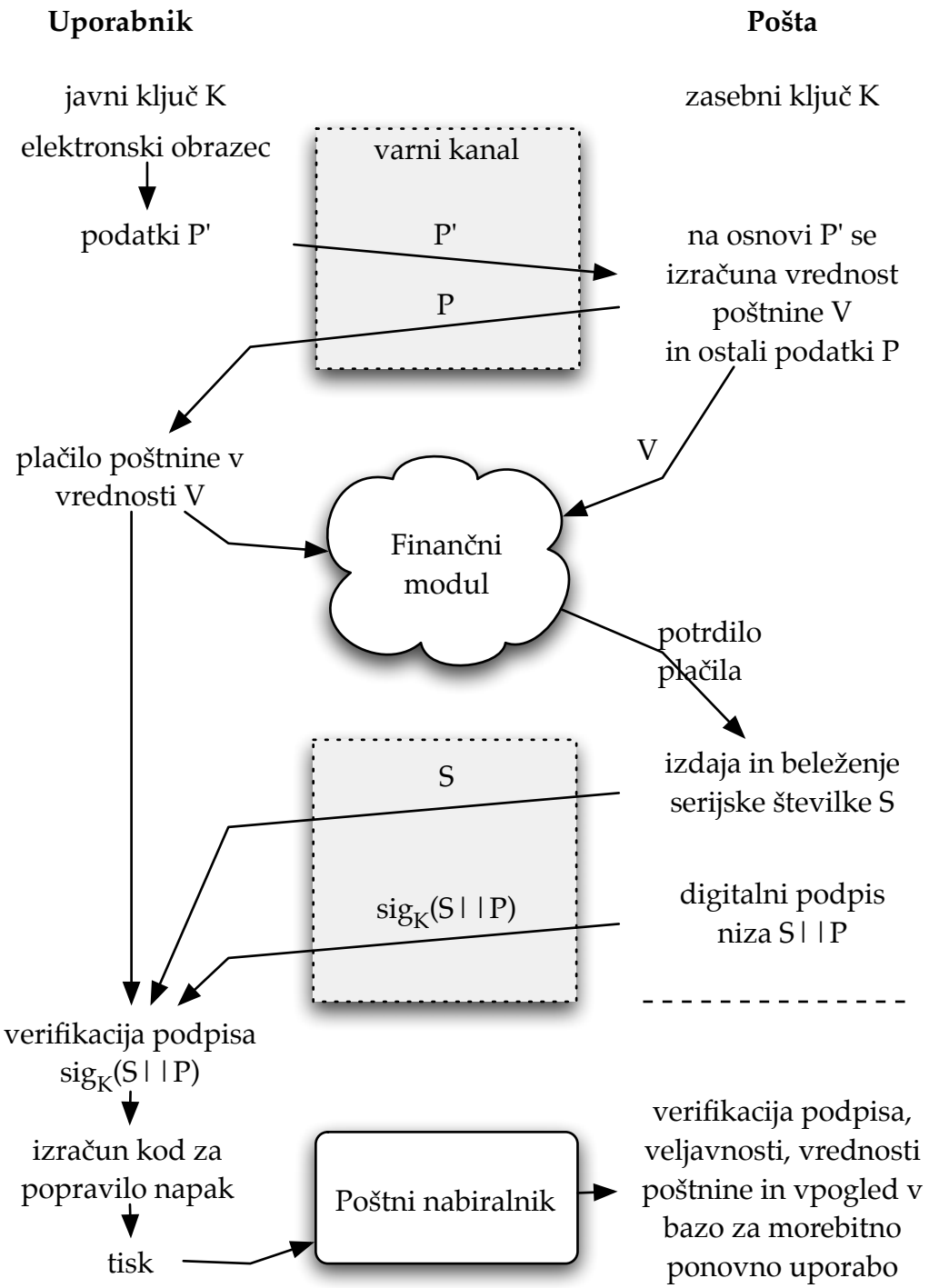
Če zgornje točke "digitaliziramo", dobimo semantično predstavo, kako bo naš protokol deloval:

1. Uporabnik obišče spletno stran Pošte Slovenije in poišče stran *Nakup znamk*,
2. Med Pošto in stranko se vzpostavi varen komunikacijski kanal,
3. Stranka izpolni elektronski obrazec s potrebnimi podatki za znamko,
4. Izvrši se elektronsko plačilo storitve,
5. Pošta ustvari digitalni podpis, v bazi zabeleži izdano zaporedno številko in znamko izroči stranki,
6. Stranka znamko natisne na pisemsko ovojnico ter pismo odda v najbližji nabiralnik,
7. Pošta pred sortiranjem preveri digitalni podpis in podatke znamke ter iz podatkovne baze izbriše uporabljeno zaporedno številko znamke.

2.2 Opis protokola

Shema 1 prikazuje osnovni potek protokola, kakor smo ga napovedali v prejšnjem razdelku. Zaradi ekonomičnosti težimo k minimalni komunikaciji. Natančnega bralca bi motile nekoliko nenatančne oznake – ko bomo natančno opredelili uporabljena orodja, bomo diagram "popravili".

Varni komunikacijski kanal smo vpeljali zato, da zaščitimo uporabnika pred morebitnimi prisluškovalci za npr. podatke naslovnikov pošte, in pred krajo izdane znamke. Varnih kanalov, kot je SSL, tu ne bomo natančneje obravnavali, saj gre za že dobro utemeljeno in ustaljeno tematiko [3] [11].



Shema 1: Osnovni potek protokola

Protokol deluje tako, da uporabnik najprej izpolni elektronski obrazec; sestavi delno sporočilo P', ki je sestavljeno iz naslova prejemnika, poštne številke prejemnika, tipa in podtipa znamke. Pošta izračuna vrednost poštnih stroškov, med podatke P doda še datum izdaje in datum zapadlosti, nato pa te podatke posreduje nazaj uporabniku. Uporabnik mora prejeti vse na znamki zapisane podatke, da lahko preveri pravilnost podatkov. Ko se uporabnik s podatki strinja in izvrši elektronsko plačilo [5] poštnine, pošta zgenerira serijsko številko, jo pošlje uporabniku ter jo zabeleži v svoji podatkovni bazi kot izdano. To je, kot bomo videli, nujno za potrebe zaznave večkratne uporabe. Tako uporabnik kot pošta imata v tej točki vse podatke znamke, manjka le digitalni podpis. Kot bo bolj razvidno v kasnejših poglavjih, ta elegantno razreši problem avtentičnosti in integritete. Digitalni podpis ustvari pošta in ga pošlje uporabniku. Uporabnik lahko sedaj preveri avtentičnost znamke, če želi. Poudariti moramo, da je tu potrebno uporabnikovo zaupanje v poštno organizacijo, saj uporabnik nima garancije, da mu pošta kasneje ne bi mogla zavrniti znamke (češ, da je le-ta že bila uporabljena).

Vse podatke je potrebno za odpornost pred morebitnimi poškodbami, praskami in madeži še zakodirati s kodami za popravilo napak. Z njimi je v omejenem obsegu možno odpraviti nastale napake in podatke v celoti prebrati.

Ko fizično pismo skupaj z znamko prispe nazaj na pošto, jo je potrebno preveriti. Če podatki niso bili spremenjeni, bo integriteto in pristnost zapisanih podatkov potrdil digitalni podpis. Za namene preverjanja veljavnosti, je na znamki zapisan datum izdaje in datum zapadlosti. Pošta ob preverjanju znamke naredi tudi poizvedbo nad podatkovno bazo; če znamka še ni bila uporabljena, bo v bazi zapisana njena serijska številka. Če je znamka še veljavna, pismo nadaljuje svojo pot, iz podatkovne baze pa se izbriše vnos te znamke, da ne bi bila možna ponovna uporaba.

Zaradi fizičnih omejitev površine, je potreben kompromis med stopnjo varnosti in količino zapisanih podatkov – želimo, da je podpis prostorsko čim bolj varčen, hkrati pa še vedno računsko varen. Anonimnost pošiljatelja se ohrani, saj noben podatek ni vezan na pošiljatelja in nič ne preprečuje, da znamko uporabi druga oseba, kot tista, ki jo je kupila.

3 Uporabljena orodja

Prišli smo do točke, kjer se moramo natančno opredeliti kaj in kako tvori naš protokol. Vemo, da potrebujemo digitalni podpis, saj ta zagotavlja, da je znamko generirala poštna organizacija in da so podatki, ki sestavljajo digitalno poštno znamko, celoviti. Dvodimenzionalno črtno kodo bomo uporabili kot fizični nosilec podatkov, kodiranje pa potrebujemo zaradi kodov za detekcijo in odpravo

napak, s katerimi se dosega robustnost in odpornost na morebitne poškodbe natisnjene znamke.

3.1 Digitalni podpis – DSA

Poznamo mnogo različnih algoritmov za digitalne podpise. Uporabili bomo algoritem, ki temelji na javni kriptografiji, saj želimo, da lahko kdorkoli preveri izdano znamko oz. njen podpis. Podpisovalec, ki želi podpisati sporočilo M, tvori podpis $sig(M)$ tako, da s svojim privatnim ključem zakriptira željeno sporočilo. Kdor želi podpis preveriti, preslika $sig(M)$ s podpisnikovim javnim ključem nazaj v M. Ponovimo definicijo sistema za digitalno podpisovanje:

Definicija : Sistem za digitalno podpisovanje je peterka (P, A, K, S, V), za katero velja:

1. $\mathcal{P}$ je končna množica sporočil,
2. $\mathcal{A}$ je končna množica podpisov,
3. $\mathcal{K}$ je končna množica ključev,
4. $\mathcal{S}$ je končna množica funkcij za podpisovanje:

$$\forall K \in \mathcal{K} : sig_K \in \mathcal{S}, \quad sig_K : \mathcal{P} \longrightarrow S$$

5. $\mathcal{V}$ je končna množica funkcij za preverjanje podpisa:

$$\forall K \in \mathcal{K} : ver_K \in \mathcal{V}, \quad ver_K : \mathcal{P} \times A \longrightarrow \{\top, \bot\}$$

Funkciji sig_K in ver_K imata to lastnost, da za vsako sporočilo $x \in \mathcal{P}$ in vsak podpis $y \in \mathcal{A}$ velja

$$ver_K(x, y) = \begin{cases} \top ; & y = sig_K(x) \\ \bot ; & y \neq sig_K(x) \end{cases}$$

Uporabili bomo *DSA* (Digital Signature Algorithm), katerega klasificiramo med nedeterministične digitalne podpise, ki so dodatek sporočilu. Poglavitna prednost DSA pred drugimi algoritmi je velikost samega podpisa. Kakor ostali podobni algoritmi, tudi ta uporablja zgoščevalno funkcijo h za "kompresijo"podpisovanega sporočila m. Kot je iz algoritma razvidno, naša inačica DSA ustvari 512 bitni podpis. Za primerjavo, RSA bi za enako stopnjo varnosti potreboval 3072 bitov velik dodatek [14]. ECDSA sicer ustvari enako velik podpis, z

manjšimi ključi in ob ekvivalentni varnosti, a je žal podvržen mnogim plačljivim patentom (kar je v nasprotju z našim ciljem neplačljivih patentov).

3.1.1 Algoritem

Naj bo p praštevilo velikosti 3072 bitov in q praštevilo velikosti 256 bitov, tako da $q|p-1$. Naj bo še $g \in \mathbb{Z}_p^*$ q-ti koren enote po modulu p, g lahko izračunamo po sledeči formuli: $t \in \mathbb{Z}_p^*$, $g = t^{(p-1)/q} \mod q > 1$. Definirajmo $\mathcal{P} = \mathbb{Z}_p^*$, $\mathcal{A} = \mathbb{Z}_q \times \mathbb{Z}_q$ in

$$\mathcal{K} = \{(p, q, g, x, y) \ : \ y \equiv g^x (\text{mod } q)\}.$$

Terica (p, q, g, y) tvori javni ključ, število x pa mora ostati zasebno.

3.1.2 Podpisovanje

Podpisnik izbere naključno in skrito število k, $2 \leq k \leq q-2$ ter izračuna

$$r \equiv (g^k \mod p) \mod q \quad \text{in} \quad s \equiv k^{-1}(h(m) + xr) \mod q.$$

V primeru, da $s \equiv 0 \ (\text{mod } q)$, je potrebno postopek podpisovanja ponoviti z drugače izbranim k. Digitalni podpis tvori par (r, s), tj.

$$sig_{(p,q,g,y),x}(m, k) = (r, s).$$

3.1.3 Preverjanje podpisa

Najprej izračunamo

$$u_1 \equiv h(m)\, s^{-1} \mod q \qquad in \qquad u_2 \equiv r\, s^{-1} \mod q.$$

Potem velja, da je podpis veljaven, $ver_{(p,q,g,y)}(m, r, s) = \top$, če in samo če

$$(g^{u_1}\, y^{u_2} \mod p) \mod q = r.$$

Parametre algoritma (ki je podrobneje predstavljen v [14]) smo določili po priporočilih [12], varnost bomo pokazali v posebnem poglavju, izbrani zgoščevalni funkciji h =SHA-256 pa posvetimo naslednji razdelek.

3.2 Zgoščevalne funkcije – SHA-2

Zgoščevalne funkcije enolično preslikajo poljubno dolg niz znakov v blok konstantne dolžine, ki je nekakšen prstni odtis oziroma povzetek vhodnega niza (*message digest, message fingerprint*). Od zgoščevalne funkcije se pričakuje, da:

- je postopek zgoščevanja, je računsko enostaven, rezultat pa determinističen,
- je neizvedljivo najti sporočilo, ki bi se preslikalo v dano zgostitev,
- je neizvedljivo spremeniti sporočilo, brez da se zgostitev spremeni,
- je neizvedljivo najti dve različni sporočili, ki bi se preslikali v isto zgostitev.

Postopek izbrane zgoščevalne funkcije SHA-2 je natanko opisan v [13]. Zgoščevalne funkcije so komplicirane za matematično obravnavo, zato njihovo varnost obravnavamo v modelu *naključnega preroka*, dokler se ne pojavijo učinkovitejši napadi.

Definicija: **Naključni prerok** je prerok (teoretična črna škatla), ki se na vsako poizvedbo odzove z enakomerno naključno izbrano vrednostjo iz zaloge vrednosti, vendar pa hkrati za določeno poizvedbo, vsakič ko je povprašan, vedno vrne isti odgovor. Lahko rečemo, da gre za matematično preslikavo iz vsakega možnega vhoda v naključen element zaloge vrednosti.

Obravnavati moramo naslednje preproste napade:

- Naj bo (x, y) podpisano sporočilo in $y = sig_K(h(x))$. Izračunamo $z = h(x)$ in nato poiščemo tak od x različen x' , da je $h(x') = h(x)$. Za zgoščevalno funkcijo h pravimo, da je **šibko brez trčenj**, če v doglednem času ni možno izračunati tak x'.
- poiščemo taka x in x' , da je $x = x'$ in $h(x') = h(x)$. Če nekako pridobimo podpis sporočila x, je potem (x', y) ponarejen podpis. Za zgoščevalno funkcijo h pravimo, da je **krepko brez trčenj**, če v doglednem času ni možno najti takšnega para x in x'.
- recimo, da nam uspe ponarediti podpis naključnega števila z. Nato poiščemo tak x, da velja $z = h(x)$. Ta napad preprečimo z enosmernimi funkcijami.

V poglavju o varnosti bomo pokazali, da zgoščevalni algoritem SHA-2, ki vrne 256 bitni povzetek, v modelu naključnega preroka ni dovzeten za zgornje napade.

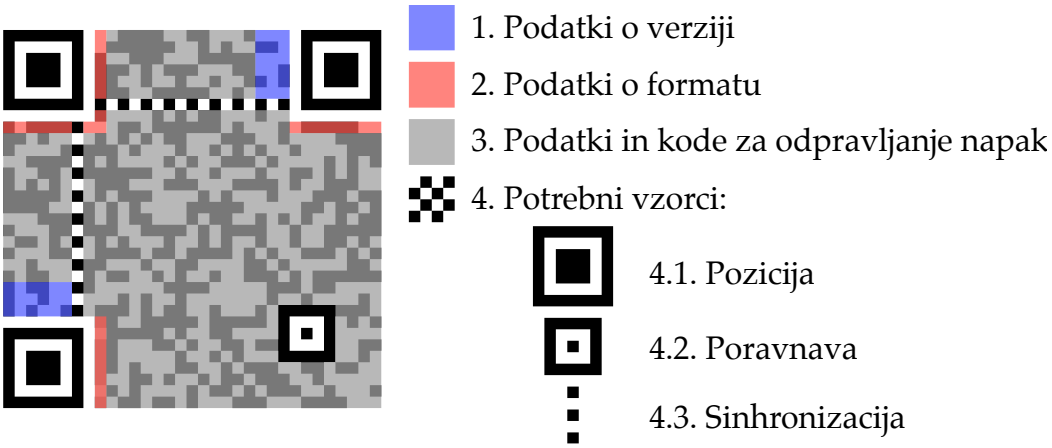
3.3 Dvodimenzionalne črtne kode - QR koda

Dvodimenzionalne črtne kode so naravni naslednik navadnih črtnih kod. Od klasičnih se razlikujejo po dodatni dimenziji, katera bistveno poveča podatkovno gostoto nosilca. Po nekaj raziskovanja preko internera, je bila sestavljena neslednja preglednica.

	QR koda	PDF417	DataMatrix	MaxiCode
izgled				
razvijalec	DENSO	Symbol Technologies	RVSI Acuity CiMatrix	UPS
nosilnost				
numerična	7 089	2 710	3 116	138
znakovna	4 296	1 850	2 355	93
binarna	2 953	1 018	1 556	
lastnosti				
visoka nosilnost	•	•	•	
hitrost razpoznave	•			•
gostota zapisa	•		•	
Reed-Solomon kodi	•	•	•	•

Ameriška digitalna poštna znamka zavzema obliko PDF417, Angleška in Nemška pa obliko DataMatrix. Naša digitalna poštna znamka bo zavzela fizično obliko dvodimenzionalne črtne kode QR, saj izpolnjuje vse iskane lastnosti. Te so: visoka podatkovna nosilnost, enostavnost in hitrost branja, majhna fizična velikost in uporaba kodov za zaznavo in odpravo napak. Njena glavna aduta, zakaj je najustreznejša za ditalno poštno znamko, sta enostavnost razpoznavanja in majhna fizična lastnost. Kot ostale, se poslužuje korekcijskih kodov, kar doprinese k odpornosti na fizične poškodbe znamke. Na shemi 2 je prikazana struktura zapisa QR kode, kjer se vidi kolikšen del površine je namenjen pozicioniranju oz. enostavnosti razpoznavanja.

Kot zanimivost – dandanes je uporaba QR-kod največja na daljnem vzhodu, kjer vsi sodobni mobilniki prebirajo hiper-povezave s plakatov in reklam preko QR-kod; v zadnjih letih je njihova popularnost nevzdržno začela rasti širom sveta, zlasti v oglaševalskih sferah.



Shema 2: Zgradba QR kode

3.4 Kodiranje – Reed-Solomon kod

Poglavitni del teorije kodiranja so kodi za odkrivanje in popravlilo napak. V osnovi te kode delimo na bločne in konvolicijske. Reed-Solomon kod, ki je zelo učinkovit in očitno široko uporabljan, spada med linearne bločne kode, zato hitro preletimo njihove osnove.

3.4.1 Teoretične osnove

Definicija: Kod $\mathcal{C}$ nad abecedo Σ je končna podmnožica $C \subset \Sigma^*$, kjer je Σ neprazna končna množica kodnih simbolov, imenovana *kodna abeceda* in $\Sigma^* = \cup_{n=0}^{\inf} \Sigma^n$. Elementi $\mathcal{C}$ so *kodne besede*. Kod $\mathcal{C} \subseteq \Sigma^*$ je *bločni kod dolžine* n, če je $\mathcal{C} \subseteq \Sigma^n$.

Pripomba: Pogosto je $\Sigma = GF_q$. Kode nad abecedo $GF_2 = \mathbb{Z}_2 = \{0,1\}$ imenujemo *dvojiški kodi*.

Definicija: Naj bo $\Sigma = GF_q$. Kod $\mathcal{C} \subseteq \Sigma^n$ je *linearen*, če je $\mathcal{C}$ vektorski podprostor prostora GF_q^n nad obsegom GF_q.

Definicija: Naj sta $x, y \in \Sigma^n$. Metriko $d(x,y) = |\{i; x_i \neq y_i\}|$ (število mest, na katerih se besedi razlikujeta) imenujemo *Hammingova razdalja*.

Definicija: Količina $d = \min_{x,y \in \mathcal{C}, x \neq y} d(x,y)$ je *minimalna razdalja* ali *razmaknjenost* koda $\mathcal{C}$.

Definicija: $\Sigma = GF_q$, $x \in \Sigma^*$. *Teža besede* x je $t(x) = |\{i;\ x_i \neq 0\}|$

Definicija: Kod $\mathcal{C}$ je (n, M, d)-*kod*, če:

- bločna dolžina $\mathcal{C}$ enaka n,
- $|\mathcal{C}| = M$,
- razmaknjenost $\mathcal{C}$ je enaka d.

Definicija: Za linearen (n, M, d) kod dimenzije k velja $M = q^k$:
Naj bo $\{x^1, \ldots, x^k\}$ baza $\mathcal{C}$. Potem je $\mathcal{C} = \{\sum_{i=1}^{k} \lambda_i x^i; \lambda_i \in GF_q\}$ in velja $|\mathcal{C}| = |\{(\lambda_1, \ldots, \lambda_k\}); \lambda_i \in GF_q\}| = q^k$.
Linearen (n, q^k, d)-kod označimo kot $[n, k, d]$-kod.

Izrek: Linearen $[n, k, d]$ kod, $\Sigma = GF_q$:

- odkrije $d-1$ napak, če za vse $x \in \mathcal{C}$ in vse *sindrome napak* $e \in \Sigma^n$ s težo $t(e)$, $1 \leq t(e) \leq d-1$. Velja $x + e \notin \mathcal{C}$
  Dokaz: $d(x, x+e) = t((x+e)-x) = t(e) \leq d-1;\ t(e) \neq 0 \Rightarrow x+e \notin \mathcal{C}$.
- popravi $\lfloor \frac{d-1}{2} \rfloor$ napak, če za vse pare različnih kodnih besed $x, y \in \mathcal{C}$, $x \neq y$ in vse napake $t \in \Sigma^n$ s težo $t(e) \leq \lfloor \frac{d-1}{2} \rfloor$ velja $d(x+e, x) < d(x+e, y)$.
  Dokaz: $d(x, x+e) = t(e) \leq \lfloor \frac{d-1}{2} \rfloor \leq \frac{d-1}{2} \quad 2d(x, x+e) \leq d-1$

$$\begin{aligned} &\leq d(x,y) - 1 && \Rightarrow d(x, x+e) \leq (x+e, y) - 1 \\ &\leq d(x,e) + d(x+e,y) - 1 && \Rightarrow d(x, x+e) \leq \dot{(}x+e, y) \end{aligned}$$

Definicija: Linearen $[n, k, d]$-kod podajamo z *generatorsko matriko* $G \in GF_q^{k \times n}$, kjer vrstice ustrezajo kodnim besedam iz neke baze koda.
Za vsak $[n, k, d]$ obstaja ekvivalenten kod z generatorsko matriko oblike $[I_k|A]$, kjer je I_k enoksta matrika $k \times k$. To je *standardna oblika* generatorske matrike.

Definicija: Naj bo $\mathcal{C}$ $[n, k, d]$-kod. $\mathcal{C}^\perp = \{x \in \Sigma^n;\ Gx^T = 0\}$ je dualni kod koda $\mathcal{C}$. Generatorsko matriko koda $\mathcal{C}^\perp$ imenujemo *nadzorna matrika* koda $\mathcal{C}$.

Naj je $G \in GF_q^{k \times n}$, $G = [I_k|A]$ generatorska matrika v standardni obliki. Pripadajoča nadzorna matrika $H \in GF_q^{(n-k) \times n}$ je $H = [-A^T|I_{n-k}]$.
Dokaz: $G \cdot H^T = 0$.

3.4.2 Kodiranje

Naj je $\mathcal{C}$ linearen kod nad GF_q.

- Baza koda: $\{x^1, \ldots, x^k\}$
- Sporočilo: $S = s_1 s_2 \ldots s_k \in GF_q^k$
- Kodirano sporočilo:

$$y = \sum_{i=1}^{k} s_i x^i \in \mathcal{C}$$

Če je generatorska matrika v standardni obliki, potem ima produkt lepo lastnost: $y = s_1 s_2 \ldots s_k y_{k+1} \ldots y_n$. Torej je v kodiranem sporočilu kar celotno sporočilo, sledi pa redundantni del za zaznavo in odpravo napak.

3.4.3 Dekodiranje

Naj je $\mathcal{C}$ linearen kod nad GF_q.

- Poslana beseda: $x \in \mathcal{C}$
- Prejeta beseda: $y \in \Sigma^n$
- Odkrivanje napak:

$$Hy^T \neq 0 \Rightarrow y \notin \mathcal{C} \Rightarrow y \neq x$$

- Popravljanje napak:

$$y = x + e \Rightarrow e = y - x \text{ je napaka}$$

Definicija: $Hy^T = Hx^T + He^T = He^T$ imenujemo *sindrom napake*.
Naj je $t(e_1), t(e_2) \leq s = \lfloor \frac{d-1}{2} \rfloor$ in $e_1 \neq e_2$. Potem velja $He_1^T \neq He_2^T$.
Dokaz: Recimo, da je $He_1^T = He_2^T$. Potem velja $H(e_1 - e_2)^T = 0 \Rightarrow e_1 - e_2 \in \mathcal{C}$.
Sledi, da $t(e_1 - e_2) = d(e_1, e_2) \leq d(e_1, 0) + d(e_2, 0) = t(e_1) + t(e_2) \leq 2s \leq d - 1$.
To pomeni, da je $e_1 - e_2 = 0$, sicer besede ne bi bile razmaknjene za vsaj d. Torej je $e_1 = e_2$.

Za postopek dekodiranja se navadno pripravi T tabela parov (He^T, e) za $\forall e \in \Sigma^n,\ t(e) \leq \lfloor \frac{d-1}{2} \rfloor$. Ko sprejmemo besedo $y \in \Sigma^n$, preverimo, če je prišlo do napake, tj. preverimo $Hy^T \stackrel{?}{=} 0$. Če napake ni, $y \in \mathcal{C} in x = y$, sicer pa za Hy^T poiščemo e v tabeli T in $x = y - e$.

3.4.4 Reed-Solomon kod

Reed-Solomon kod je kod, ki temelji na prevzorčenju polinoma, katerega definirajo podatki. Kot vemo, osnovni teorem algebre pravi, da k točk enolično določa polinom stopnje največ $k-1$. Podatki določijo ta polinom, redundantne podatke pa predstavljajo dodatno vzorčene točke tega polinoma. Dokler se pravilno prenese dovolj koeficientov, se originalne podatke da rekonstruirati. Formulirajmo to matematično:

Naj je $F[x]$ polinomski obseg nad $GF(2^m)$ in n, k takšni števili, da velja $1 \leq k \leq n = 2^m - 1$, ter α primitivni koren enote. Točke vzorčenja tvorijo potence α, tj. $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, kjer je $\alpha_i = \alpha^{i-1}$. Kod $\mathcal{C}$ sestavljajo vsi polinomi stopnje manjše od k, nad vrednostmi točk vzorčenja α^i:

$$\mathcal{C} = \{(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)) \mid f \in F[x], deg(f) < k\}$$

Kod $\mathcal{C}$ je $[n, k, n-k+1]$-kod.

Veliko lažje je, če kod obravnavamo s polinomi. Naj bo

$$g(x) = (x - \alpha_1)(x - \alpha_2) \cdots (x - \alpha_{n-k})$$

generatorski polinom. Tedaj je kodiranje množenje polinomov, $y(x) = g(x) \cdot s(x)$, kjer je $s(x) = s_1 + s_2 x + \cdots + s_{n-k} x^{n-k-1}$ podatkovni polinom. Računsko je lažje izračunati

$$y(x) = x^{n-k+1} \cdot s(x) + p(x)$$

kjer je $p(x) = s(x) \cdot x^{n-k} \mod g(x)$. Sindromski polinom ima obliko

$$h(x) = h_1 + h_2 x + \cdots + h_{n-k} x^{n-k-1};\ h_i = y(\alpha_1)$$

Za tem se izračuna iskalni polinom (*error locator polynomial*) s Berlekamp-Macey algoritmom in poišče njegove ničle z Chan-ovim iskalnim algoritmom. Inverzni elementi ničel dajo mesta, kjer je prišlo do napake. Napake se za tem izračuna z Forney-evim algoritmom, ter prišteje k pokvarjenemu koeficientu.

Pokažimo še, da Reed-Solomon kod doseže Singletonovo mejo, torej da velja

$$d = n - k + 1$$

Ker je vsak polinom stopnje največ $k-1$, ima največ $k-1$ ničel, tj. doseže vrednost nič v največ $k-1$ točkah. Po drugi strani to pomeni, da obstaja vsaj $n - (k-1)$ elementov z neničelno vrednostjo, ker smo v končnih obsegih. Sledi, da ima vsaka neničelna vrednost težo vsaj $n - k + 1$. Ker imamo opravka z linearnim kodom, je minimalna razmaknjenost koda enaka teži kodne besede z najmanjšo težo, zato $d = n - k + 1$.

4 Načrt digitalne poštne znamke

4.1 Podatki na znamki

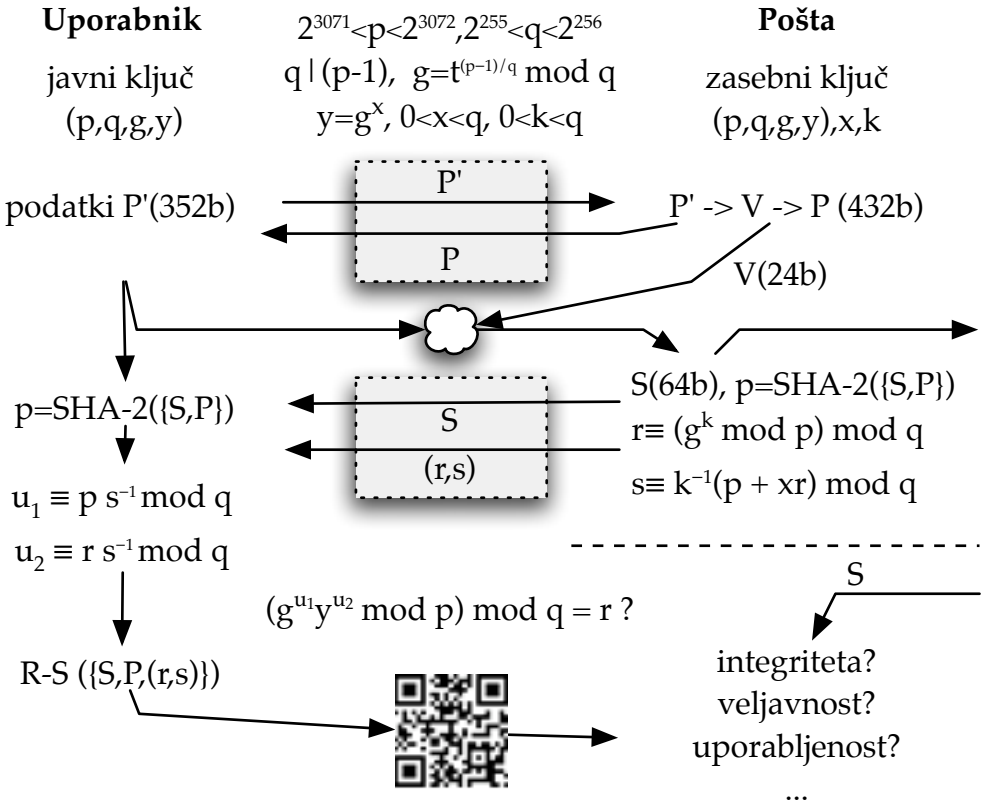
podatek	dolžina (bit)	opomba
verzija znamke	8	
naslov prejemnika	320	40 znakov · 8bit
poštna št. prejemnika	16	4 cifre · 4bit
datum pošiljanja	24	6 cifer · 4bit
datum zapadlosti	24	6 cifer · 4bit
zaporedna št. znamke	64	
tip znamke	8	
podtip znamke	8	
plačana poštnina	24	6 cifer · 4bit
digitalni podpis	512	
skupaj	**1008**	

Zgornja tabela prikazuje koliko bitov potrebujemo za posamezne podatke, katere nosi digitalna poštna znamka. Celokupni seštevek je 1008 bitov, kar je 126B. QR-koda, ki ima primerno nosilnost je verzija 11H in sprejme 137B podatkov, tako ostane še 88 bitov prostih za manjše razširitve vključenih podatkov. Ob želji, da znamko lahko natisnemo na vsakem domačem tiskalniku, privzamemo da je minimalna ločljivost 300dpi. Ob taki ločljivosti je velikost kodnega modula $0,33mm$, verzija kode 11H pa ima velikost 61×61 modulov. Tako je stranica digitalne poštne znamke dolga $20,13mm$, njena površina pa $4,05cm^2$. Ob morebitnih večjih razširitvah ima že verzija 12H nosilnost 155B, torej dodatnih 232 bitov, ob povečanju stranice znamke za zgolj $4 \cdot 0,33mm = 1,32mm$.

4.2 Protokol

Uporabnik ima javno dostopen javni ključ (p, q, g, y), pošta pa lasti v svojem zasebnem ključu še skrivno število x (glej razdelek 3.1). Da je centralizirano izdajanje znamk računsko učinkovitejše, se lahko v naprej določi nekaj skrivnih vrednosti $k_1 \ldots k_n$ in ustrezne $r_{i \in [1..n]} = (g^i \mod p) \mod q$. To smemo storiti, saj bodo zaradi vsebovanja enoličnih serijskih številk, vsa podpisovana sporočila različna.

Preverjanje znamk se izvaja pri ponudniku storitev in predstavlja ozko grlo sistema. Ker je preverjanje vsake pošiljke praktično težko izvedljivo, se pregledovanje ponavadi izvaja statistično, kjer se vpelje t.i. *planiran obseg poštnih zlorab* [1]. V te namene smo v znamko vključili naslov prejemnika in zaporedno številko. Z naslovom prejemnika smo pošiljko povezali z znamko, s tem pa



Shema 3: Diagram protokola

onemogočili zlorabo znamke na način, da se isto znamko uporabi za pošiljke na različne naslove. Težava nastopi, če uporabnik pošilja več pošiljk istega dne na isti naslov. Tako bi lahko uporabnik plačal eno znamko ter jo poljubnomnogokrat uporabil za istega naslovnika. Zaplet smo rešili z uporabo unikatne identifikacijske – serijske številke znamke.

Oceniti moramo še, če je zastavljeni sistem dovolj dimenzioniran za uporabo v realnem svetu. Zastavimo, da se uporabljani par ključev za digitano podpisovanje zamenja vsakih 5 let. Možnih serijskih številk je 2^{64}. Da bi jih vse porabili, bi moralo v povprečju biti vsak dan izdanih $2^{64}/365 \cdot 5 = 1.0107805 \cdot 10^{16}$ znamk (če bi imeli 2^{32} serijskih številk, bi bila kapaciteta "zgolj" 2 353 406 znamk na dan).

Prenos podatkov med nakupom znaša $352 + 432 + 64 + 512 = 1360$ bitov. Za zelo grobo oceno, recimo da mora izdajalni strežnik na znamko prenesti vsaj $10x$ toliko podatkov, približno $1,66kB \simeq 2kB$. Na 10Mb povezavi s svetom to pomeni, da bi strežnik v eni sekundi lahko izdal $60\% * 1280kB/2kB = 384$ znamk, tj. v enem dnevu $384 \cdot 60 \cdot 60 \cdot 24 = 33\,177\,600$. Kljub temu, da je to

veliko, nam s takšno "obremenitvijo"ne bi uspelo porabiti vseh serijskih številk (glej zgoraj). Velikost preproste podatkovne baze za hrambo vseh izdanih serijskih številk, bi se na tako "ploden"dan povečala za $33\,177\,600 \cdot 64b \doteq 253MB$. Seveda pa obstajajo bolj učinkovite izvedbe takšne podatkovne baze [4].

5 Varnost

Hipoteza 1: Digitalna poštna znamka je zaradi uporabe algoritma DSA, z zasebnim ključem velikosti 3072 bitov in 256 bitnim javnim ključem ter zgoščevalno funkcijo SHA-2 z 256 bitnim rezultatom, računsko varna.

Pokazali bomo, da so na tako opisanem algoritemu in trenutnem znanju, znani napadi računsko prezahteveni, da bi bili računsko izvedljivi (*feasible*).

1.1 Definicija: **MIPS stroj** je stroj, ki je zmožen opraviti en milijon poljubnih operacij na sekundo (*million instructions per second*).

1.2 Definicija: **100MIPS stroj** je skoraj enak stroju iz prejšnje definicije, le da opravi 100 milijonov operacij v sekundi. Njega bomo uporabljali kot referenčni računski stroj, s katerim bomo prikazovali oceno današnje časovne zahtevnosti napadov. Za primerjavo, najzmogljivejši procesor danes, Intel Core i7 Extreme 965EE, premore 76,4 MIPS.

1.3 Lema: Zgoščevalna funkcija h=SHA-2 z rezultatom zgostitve $n = 256$ bitov je, v modelu naključnega preroka, šibko brez trčenj.

Napad z grobo silo potrebuje 2^{256} operacij:
Algoritem 1.3 : Poišči drugo prasliko (x)

- $y = h(x)$
- $\forall x' \in X \setminus x:$ če $h(x') = y$ vrni x'.
- sicer vrni $\perp$.

MIPS stoj bi potreboval $2^{256} - 1/(10^6 \cdot 60 \cdot 60 \cdot 24 \cdot 365) = 3.67174306 \cdot 10^{63}$ let za uspešno poneverbo. Tudi če bi skupaj zbrali milijon 100MIPS strojev, bi vselej potrebovali več kot $3.6 \cdot 10^{55}$ let. □

1.4 Lema: Zgoščevalna funkcija h=SHA-2 z rezultatom zgostitve $n = 256$ bitov je, v modelu naključnega preroka, odporna pred kolizijami, oz. je krepko brez trčenj.

Napad s pomočjo rojstnodnevnega paradoksa potrebuje 2^{128} operacij:
Verjetnostni algoritem 1.4 : Poišči kolizijo ($p_{uspeha} = \frac{1}{2}$)

- $X \overset{enakomerno\ naključno}{\longrightarrow} Q; |Q| = 1.17 \cdot 2^{128}$ [9]
- $\forall x \in Q: \ y_x = h(x)$
- če $y_x = y_{x'}$ za nek $x \neq x'$, potem vrni (x, x')
- sicer vrni $\perp$

Kar smo pokazali je, da že za samo iskanje dveh sporočil, kateri bi imeli enak digitalni podpis, bi MIPS stoj potreboval več kot $2^{128}/(10^6 \cdot 60 \cdot 60 \cdot 24 \cdot 365) = 1.07902831 \cdot 10^{25}$ let. Tudi če bi skupaj zbrali milijon 100MIPS strojev, bi vselej potrebovali več kot 10^{17} let. □

1.5 Lema: Zgoščevalna funkcija h =SHA-2: $X \longrightarrow Z$ je enosmerna funkcija.

Naj obstaja algoritem $\mathcal{A}$ za računanje obrata zgoščevalne funkcije. Uporabimo ga kot podprogram probabilističnega algoritma $\mathcal{B}$, ki išče trčenja:

- izberi element $x \in X$,
- izračunaj $z = h(x)$,
- izračunaj $y = \mathcal{A}(z)$,
- če $y = x$, potem x in y trčita glede na h, sicer neuspeh.

Torej, če funkcija h ne bi bila enosmerna, bi obstajal učinkovit algoritem za iskanje trkov. Ker pa takega algoritem ne poznamo, je SHA-2 enosmerna funkcija. □

1.6 Trditev: Algoritem DSA je varen pred eksistenčnim ponarejanjem z znanim sporočilom.

To je najbolj očiten poskus ponarejanja digitalnega podpisa: Napadalec pridobi veljaveno podpisano sporočilo (x, y), kjer je $y = sig_K(h(x))$. Nato sam izračuna $z = h(x)$ in skuša najti tak $x' \neq x$, da velja $h(x') = h(x)$. Če mu to uspe, je par (x', y) veljavno podpisano sporočilo, y je ponarejen podpis sporočila x'. Lema 1.3 pravi, da je zgoščevalna funkcija SHA-256 varna pred takšnim napadom druge praslike. □

1.7 Trditev: Algoritem DSA je varen pred eksistenčnim ponarejanjem z izbranim sporočilom.

Napadalec poišče dve sporočili $x' \neq x$, za kateri velja $h(x') = h(x)$. Ko da napadalec pošti v podpis sporočilo x oz. njeno zgostitev $h(x)$, pridobi podpis $y = sig_K(h(x))$. Par (x', y) je veljavno podpisano sporočilo in y je ponarejen podpis sporočila x'. Lema 1.4 pravi, da je zgoščevalna funkcija SHA-256 varna pred kolizijami, zato tak napad ni izvedljiv. □

1.8 Trditev: Algoritem DSA je varen pred eksistenčnim ponarejanjem ob poznavanju zgolj javnega ključa.

Takšni napadi se prevedejo na reševanje problema diskretnega logaritma, in sicer v multiplikativni grupi $\mathbb{Z}_p^*$ (a) in ciklični podgrupi reda q (b):
a) Najučinkovitejši algoritem za reševanje diskretnega logaritma v takšni grupi je *index-calculus*. Hevristična ocena časovne zahtevnosti tega algoritma je $L_p[\frac{1}{3}, c] = \mathcal{O}(e^{(c+\mathcal{O}(1))(\ln p)^{\frac{1}{3}}(\ln \ln p)^{\frac{2}{3}}})$, kar je približno reda $4 \cdot 10^{50}$ operacij, oz. $1,4 \cdot 10^{37}$ let računanja na MIPS stroju in $1,4 \cdot 10^{29}$ let na milijonu 100MIPS strojev.
b) Trenutno najboljši algoritmi za takšne grupe dosegajo korensko časovno zahtevnost. Za uspešen napad bi tako npr. z algoritmom *Pollard rho* potrebovali 2^{128} operacij, za kar smo pri lemi 1.4 pokazali, da je računsko nedosegljivo. □

Izrek o varnosti DSA: Algoritem DSA je z zasebnebnim ključem velikosti 3072 bitov in 256 bitov javnega ključa ter uporabo zgoščevalne funkcije SHA-256 računsko varen.

Pokazali smo, da so znani napadi tako na problem diskretnega logaritma, kot napadi na zgoščevalne funkcije računsko prezahtevni, da bi bili izvedljivi. Zato verjamemo, da je naš algoritem DSA računsko varen. ■

6 Zaključek: Doseženi cilji in zahteve

S pomočjo prejšnjega poglavja lahko skoraj trivialno izpeljemo željene posledice:

Posledica 1 : Avtentičnost (a), integriteta (b) in veljavnost (c) digitalne poštne znamke so enostavno, tj. hitro preverljivi, izračunljivi.

a, b) Avtentičnost in integriteto podatkov garantira digitalni podpis. Predstavljena instanca algoritma DSA je računsko varna pred znanimi napadi (Izrek o varnosti DSA). Zato lahko zaključimo, da lahko le lastnik zasebnega ključa ustvari digitalni podpis podatkov, oziroma njihovo zgostitev. □

c) Veljavnost znamke se prebere s same znamke, saj je vključen tako datum izdaje, kot datum zapadlosti. Da sta datuma pravilna in veljavna, jamči digitalni podpis. ■

Posledica 2 : Digitalna poštna znamka je odporna proti goljufijam.

Dokaz: V poglavju Varnost smo pokazali, da so vsi trenutno znani načini ponaredb njenega ključnega digitalnega podpisa računsko nedostopni v razumnem času. ■

Posledica 3 : Večkratna uporaba znamke je preprečena.

Dokaz: Med podatke na znamki smo vključili naslov in poštno številko prejemnika. Tako bi nepošteni uporabniki še vedno lahko znamko večkrat uporabili za pošiljke namenjene na isti naslov. Še to možnost smo preprečili s pomočjo beleženja veljavnih izdanih serijskih številk. ■

Posledica 4 : Pošiljatelj je še vedno anonimen.

Dokaz: Nobeni podatki na znamki, ne pri poštni organizaciji ne povezujejo digitalne poštne znamke in posledično pošiljke s pošiljateljem. ■

Posledica 5 : Znamka je odporna na poškodbe.

Dokaz: Robustnost smo pridobili z uporabo Reed-Solomon koda (glej 3.4.4). Znamka bo ostala berljiva, tudi če je do 30% podatkov poškodovanih. ■

Posledica 6 : Znamka je prostorsko varčna.

Dokaz: Potreben prenos podatkov znaša 1360 bitov, znamka vsebuje 1008 bitov informacij, stranica natisnjene kvadratne znamke pa znaša malo več kot $2cm$. ■

Literatura

[1] Mojca Mikac, *Evidenca poštnih plačil v digitalni dobi*, Fakulteta za matematiko in fiziko, Ljubljana (2001).

[2] Boštjan Mešetič, *Elektronski podpis in infrastruktura za varno elektronsko poslovanje*, Fakulteta za elektrotehniko, Ljubljana (2001).

[3] Rok Urbas, *Protokol SSL*, Fakulteta za računalništvo in informatiko, Ljubljana (2003).

[4] Ludvik Kos, *Digitalna poštna znamka*, Fakulteta za računalništvo in informatiko, Ljubljana (2004).

[5] Aleš Rink, *Elektronsko bančništvo in varnost poslovanja*, Fakulteta za računalništvo in informatiko, Ljubljana (2006).

[6] The United States Postal Service (USPS), *Information-Based Indica Program: Performance criteria for information-based indica and security architecture for closed IBI postage metering systems (PCIBI-C)*, The United States Postal Service (1999).

[7] Dominic Welsh, *Codes and Cryptography*, Oxford University Press (1995).

[8] Boštjan Makarovič, Goran Klemenčic,..., *Internet in pravo : izbrane teme s komentarjem Zakona o elektronskem poslovanju in elektronskem podpisu*, Pasadena, Ljubljana (2001).

[9] Douglas R. Stinson, *Cryptography: Theory and Practice, 3th edition*, Chapman and Hall/CRC (2005).

[10] William Stallings, *Cryptography and network security : principles and practices, 4th edition*, Pearson/Prentice Hall (2006).

[11] Pavla Lah, *Uporaba kriptografije v internetu*, http://www.ca.gov.si/kripto/ (2007).

[12] National Institute of Standards and Technology, *Recommendation for Key Management (SP-800-57)*, http://csrc.nist.gov/publications/ (2007).

[13] National Institute of Standards and Technology, *Secure Hash Standard (SHS) (FIPS PUB 180-3)*, http://csrc.nist.gov/publications/ (2008).

[14] National Institute of Standards and Technology, *Digital Signature Standard (DSS) (FIPS PUB 186-3)*, http://csrc.nist.gov/publications/ (2009).