Univerza v Ljubljani
Fakulteta za računalništvo in informatiko
Magistrski študij, smer Informacijski sistemi in odločanje

Projektna naloga

BRANDSOV SISTEM DIGITALNEGA DENARJA Z NADZORNIKOM, TEMELJEČ NA PROBLEMU REPREZENTACIJE

Ljubljana, 5. november 2002 Marko Perme

KAZALO

POVZETEK

V projektni nalogi bo opisan konkreten sistem digitalnega denarja z nadzornikom, temelječ na problemu reprezentacije. Predstavljeni bosta dve kriptografski tehniki, ki igrata bistveno vlogo v protokolih tega sistema. To sta problem reprezentacije (v povezavi s tem tudi dokaz poznavanja reprezentacije) in omejeni slepi podpisi. Pred predstavitvijo konkretnega sistema bodo v jeziku, razumljivem bralcu brez kriptografskega znanja, predstavljeni protokoli in lastnosti večine danes poznanih sistemov digitalnega denarja.

UVOD

V zgodovini človeka so menjalna sredstva že od svojega pojava naprej spreminjala obliko glede na časovna obdobja in razvitost civilizacij, ki so jih uporabljala. Od dragih kovin, prek kovancev in bankovcev, do – kovancev in bankovcev dandanes. Smo na pragu spremembe?

Živimo v informacijski dobi v kateri je informacija postala glavna dobrina. Digitalizacija informacij ob koncu preteklega stoletja je verjetno glavni razlog za tako radikalno spremembo. Tudi denar je pogosto le še v dobro zavarovani bančni podatkovni bazi zabeležena številka, ki se občasno poveča ali zmanjša. Pa vendar, medij za prenašanje in hranjenje vrednosti pri večini navadnih smrtnikov še vedno v veliki meri ostaja papir ali kovina. Če bo digitalizacija prodrla tudi na to področje, bo denar le še kos digitalne informacije.

V zadnjih nekaj desetletjih so se razvili kriptografski algoritmi in strojne rešitve, ki omogočajo varno (s stališča banke) in anonimno (s stališča potrošnika) uporabo digitalnega denarja. V tem projektu so predstavljeni nekateri od kriptografskih algoritmov, splošne lastnosti digitalnega denarja in konkreten primer sistema digitalnega denarja.

Prvo poglavje predstavi problem reprezentacije. Gre za posplošitev problema diskretnih logaritmov. Predstavljen bo dokaz poznavanja reprezentacije brez razkritja le-te. Na koncu je še kratka predstavitev omejenih slepih podpisov. Vse naštete kriptografske tehnike so gradniki v tretjem poglavju predstavljenega sistema digitalnega denarja.

Drugo poglavje se loti digitalnega denarja in sistemov digitalnega denarja v splošnem. Opiše postopke in lastnosti sistemov ter našteje nekaj problemov, ki se pojavljajo pri uvajanju digitalnega denarja.

Tretje poglavje predstavi konkreten sistem digitalnega denarja – Brandsov sistem z nadzornikom, temelječ na problemu reprezentacije. Predstavljen je koncept nadzornika. Po korakih je razložen vsak od postopkov (odprtje računa, dvig denarja, plačilo, polog denarja). Na koncu je navedenih še nekaj možnih razširitev sistema, kot jih je predlagal S. Brands v [BRA93].

1 KRIPTOGRAFSKI TEMELJI

1.1 Problem reprezentacije

Problem reprezentacije je posplošitev problema diskretnih logaritmov. Rešitev problema reprezentacije zahteva računanje k diskretnih logaritmov, imenovanih k-terica eksponentov. Njegova zahtevnost je zato pogojena z zahtevnostjo problema diskretnih logaritmov.

V nadaljevanju je G_q grupa (znanega) praštevilskega reda q, za katero so znani algoritmi polinomske časovne zahtevnosti za ugotavljanje enakosti med elementi in pripadnosti grupi, za računanje inverzov in produktov ter za naključno izbiro elementov.

Definicija 1: Naj bo $k \geq 2$ konstanta. Imamo k-terico generatorjev $(g_1,\ldots,g_k)$, kjer so $g_i \in G_q$ za vse $i \in \{1, \ldots, k\}$, paroma različni in različni od 1. Imamo k-terico eksponentov $(a_1, \ldots, a_k)$, kjer so $a_i \in \mathbb{Z}_q$ za vse $i \in \{1, \ldots, k\}$. Za poljuben $h \in G_q$ je **reprezentacija** k-terica eksponentov $(a_1, \ldots, a_k)$, tako da $\prod_{i=1}^{k} g_i^{a_i} = h$.

Za $h = 1$ je enostavno najti reprezentacijo $(0,\ldots,0)$. To k-terico imenujemo trivialna reprezentacija. Če bi vzeli $k = 1$, bi dobili dobro znani diskretni logaritem z $a \in \mathbb{Z}_q$, ki je reprezentacija za $h \in G_q$ z upoštevanjem $g \neq 1$. Ker se želimo tej situaciji izogniti, je v definiciji naveden pogoj $k \geq 2$.

Trditev 2: Za vse $h \in G_q$ in vse k-terice generatorjev je natančno q^{k-1} reprezentacij (k-teric eksponentov) za h.

Dokaz: Ker je G_q praštevilskega reda, je vsak $g \in G_q \setminus \{1\}$ generator v G_q. Torej lahko izberemo prvih k-1 elementov k-terice eksponentov $\in_R \mathbb{Z}_q$ (naključni elementi iz $\mathbb{Z}_q$). Zadnji, k-ti element je potem enolično določen kot diskretni logaritem od $h / \prod_{i=1}^{k-1} g_i^{a_i}$ glede na g_k.

□

Ta ugotovitev implicira, da je gostota reprezentacij za h zanemarljivo majhna v primerjavi z množico vseh k-teric eksponentov. Torej ima poljuben algoritem polinomske zahtevnosti, ki uporablja metodo izčrpnega preiskovanja za iskanje ustrezne reprezentacije, zanemarljivo verjetnost uspeha. V naslednji trditvi še pokažemo, da ni boljše metode iskanja - ob predpostavki, ki velja za problem diskretnega logaritma.

Trditev 3: Naj bo V množica vseh funkcij oblike $q(\cdot)/r(\cdot)$, tako da sta $q(\cdot)$ in $r(\cdot)$ polinoma s celoštevilskimi koeficienti in celoštevilsko domeno. Naj bo $q(k) > r(k) \geq 1$ za vse dovolj velike k. Za poljubne funkcije $f_1(\cdot), f_2(\cdot), f_3(\cdot), f_4(\cdot) \in V$ so naslednje štiri trditve ekvivalentne:

(1) Obstaja algoritem $A_{(1)}$ polinomske časovne zahtevnosti, ki vzame na vhodu k-terico generatorjev in $h \in G_q$ ter vrne reprezentacijo za h z verjetnostjo uspeha najmanj $1/f_1(|p|)$ za vse dovolj velike p.

(2) Obstaja algoritem $A_{(2)}$ polinomske časovne zahtevnosti, ki vzame na vhodu k-terico generatorjev in vrne netrivialno reprezentacijo za 1 z verjetnostjo uspeha vsaj $1/f_2(|p|)$ za vse dovolj velike p.

(3) Obstaja $h \in G_q \setminus \{1\}$ in algoritem $A_{(3)}$ polinomske časovne zahtevnosti, ki vzame na vhodu k-terico generatorjev in vrne reprezentacijo za h z verjetnostjo uspeha vsaj $1/f_3(|p|)$ za vse dovolj velike p.

(4) Obstaja algoritem $A_{(4)}$ polinomske časovne zahtevnosti, ki reši problem diskretnega logaritma z verjetnostjo uspeha vsaj $1/f_4(|p|)$ za vse dovolj velike p.

Dokaz: Za dokaz zgornjih trditev je potrebno pokazati transformacijo verjetnosti v okviru polinomske časovne zahtevnosti za trditev (4) v vsako od (1), (2) in (3). Če imamo $A_{(4)}$, lahko namreč zlahka pridemo tudi do ustreznih $A_{(1)}$, $A_{(2)}$ in $A_{(3)}$. Predpostavimo, da imamo dva elementa $g, h \in G_q$, $g \neq 1$ in da hočemo izračunati $\log_g h$.
Algoritem $A_{(4)}$ izražen z $A_{(1)}$:

Korak 1: Tvori naključno k-terico $(u_1, \ldots, u_k)$ in izračunaj k-terico generatorjev $(g_1, \ldots, g_k)$ glede na $g_i = g^{u_i}$ za $i \in \{1,\ldots,k\}$. Podaj k-terico generatorjev skupaj s h algoritmu $A_{(1)}$.

Korak 2: Algoritem $A_{(1)}$ vrne k-terico eksponentov $(a_1,\ldots,a_k)$. Če to ni reprezentacija za h, pojdi na Korak 1.

Korak 3: Izračunaj in vrni $\log_g h = \sum_{i=1}^{k} a_i u_i \bmod q$.

Po pričakovanem številu ponovitev $f_1(|p|)$ Koraka 2, $A_{(1)}$ vrne reprezentacijo za h. Ostale transformacije so prikazane v [BRA93], strani 16 in 17.

□

Zapišimo problem reprezentacije v formalni obliki:

Ime: Problem reprezentacije.
Instanca: Grupa G, elementi $g_1, \ldots, g_k \in G$, $h \in G$.
Vprašanje: Ali obstaja reprezentacija za h glede na $(g_1, \ldots, g_k)$? Če obstaja, najdi eno.

V poglavju 3 opisan sistem digitalnega denarja, ki temelji na problemu reprezentacije, uporablja samo omejeno oblika tega problema – z grupami praštevilskega reda. Trditev 3 pravi, da je problem reprezentacije za grupe praštevilskega reda po računski zahtevnosti ekvivalenten problemu diskretnega logaritma.

Zapišimo pomembno posledico, ki izhaja iz Trditve 3:

Posledica 4: Ob predpostavki zahtevnosti problema diskretnega logaritma ne more obstajati algoritem polinomske časovne zahtevnosti, ki vzame na vhodu naključno izbrano k-terico generatorjev $(g_1, \ldots, g_k)$ in vrne število $h \in G_q$ ter dve reprezentaciji za h z verjetnostjo, ki ni zanemarljiva.

Dokaz: Če bi obstajal tak algoritem A, bi ga lahko uporabili za tvorbo naslednjega algoritma:

Korak 1: Podaj k-terico generatorjev $(g_1,\ldots,g_k)$ algoritmu A. Ta vrne h in dve k-terici eksponentov $(a_1,\ldots,a_k)$ in $(a'_1,\ldots,a'_k)$, ki sta reprezentaciji za h.
Korak 2: Vrni $(a_1 - a'_1 \bmod q,\ldots,a_k - a'_k \bmod q)$.
Izdelali smo algoritem $A_{(2)}$ iz Trditve 3, kar pa je v protislovju s predpostavko, ki velja za problem diskretnega logaritma.

1.2 Dokaz poznavanja reprezentacije

Ker do poznavanja reprezentacije ne moremo priti brez računanja diskretnih logaritmov, lahko to izkoristimo za preverjanje identitete tistega, ki mu je reprezentacija poznana. Seveda zahtevamo, da med postopkom dokazovanja reprezentacija ni razkrita tistemu, ki preverja.

Recimo, da dokazovalec (prover) pozna reprezentacijo za $h = \prod_{i=1}^{k} g_i^{a_i}$. To pomeni, da mu je ob dani k-terici generatorjev $(g_1,\ldots,g_k)$ poznana k-terica eksponentov $(a_1,\ldots,a_k)$. Dokazovalec želi dokazati poznavanje k-terice eksponentov brez razkrivanja le-te.

Preverjevalec (verifier) pozna h in $(g_1,\ldots,g_k)$, želi pa se prepričati, da pozna tudi $(a_1,\ldots,a_k)$.

Korak 1: Dokazovalec tvori k naključnih števil $(w_1,\ldots,w_k)$ izbranih iz G_q in pošlje -ju število $z = \prod_{i=1}^{k} g_i^{w_i}$.

Korak 2: prejme z in tvori naključno število – izziv c, ki ga pošlje nazaj -ju.

Korak 3: izračuna odzive $r_i = w_i + ca_i \bmod q$, za $i = 1, \ldots, k$ in jih pošlje nazaj -ju.

Korak 4: izračuna zh^c in to primerja z $\prod_{i=1}^{k} g_i^{r_i}$. Če sta obe števili enaki, zaključi, da pozna reprezentacijo.

Pokažimo še, zakaj enakost iz Koraka 4 dokazuje, da pozna reprezentacijo:

$$z \cdot h^c = \prod_{i=1}^{k} g_i^{w_i} \cdot \left(\prod_{j=1}^{k} g_j^{a_j}\right)^c = \prod_{i=1}^{k} g_i^{w_i + c \cdot a_i} = \prod_{i=1}^{k} g_i^{r_i}$$

Enakost torej obstaja samo v primeru, ko za izračun eksponentov r_i uporabi a_i.

1.3 Omejeni slepi podpisi

Shema digitalnega denarja, ki temelji na problemu reprezentacije in je opisana v poglavju 3, uporablja omejene slepe podpise ob dvigu denarja z banke. Glavna značilnost slepih podpisov je, da lahko prejemnik pridobi podpis $sign(m)$ na sporočilo $m \neq 1$ na tak način, da ostane $(m, sign(m))$ neznan podpisniku in da je tudi z neomejeno računsko močjo podpis $(m, sign(m))$ nemogoče povezati s kako izvedbo digitalnega podpisovanja.

Naj bo m sporočilo. Podpisnik to sporočilo podpiše tako, da m dvigne na potenco x:

$z = signed(m) = m^x$

Postopek slepega podpisa je sledeč:

Korak 1: Prejemnik pošlje sporočilo *m* podpisniku (ali pa je *m* znan in ga prejemnik ne določa). Ta naj bi sporočilo podpisal s svojim zasebnim ključem *x* in posredoval prejemniku tudi podpisan dokaz, da je uporabljeni zasebni ključ *x* veljaven. To je dokazljivo z javnim ključem *h*, kajti velja $\log_g h = x = \log_m z$.

Korak 2: Podpisnik tvori naključno število $w \in_R \mathbb{Z}_q$ in pošlje prejemniku podpisano sporočilo $z = m^x$, $a = g^w$ (kjer *a* poimenujemo psevdo-javni ključ) in $b = m^w$ (kjer *b* poimenujemo psevdo-podpisano sporočilo).

Prejemnik ima sedaj dve sporočili, podpisani z zasebnim ključem (*x*) podpisnika – podpisano sporočilo $z = m^x$ in podpisnikov javni ključ, ki je pravzaprav podpisan generator $h = g^x$. Prav tako ima prejemnik tudi dve sporočili, podpisani z naključno izbranim *w* (psevdo-zasebnim ključem): psevdo-javni ključ *a* in psevdo-podpisano sporočilo *b*.

Korak 3: Prejemnik tvori izziv *c* tako, da najprej tvori štiri naključna števila *s*, *t*, *u* in *v*. Z uporabo *s* in *t* uporabnik izračuna spremembe sporočila *m* in *z*, torej slepo sporočilo *m*' in podpisano slepo sporočilo *z*':

$$M' = M^s \cdot g^t$$
$$z' = z^s \cdot h^t = (M^x)^s \cdot (g^x)^t = (M^s \cdot g^t)^x = M'^x$$

Prejemnik torej dobi veljaven podpis na spremenjenem sporočilu *m*', ki ga podpisnik ne pozna.

Z uporabo *u* in *v* prejemnik izračuna spremembe *a* in *b*, torej *a*' in *b*':

$$a' = a^u \cdot g^v = (g^w)^u \cdot g^v = g^{w'}$$

$$b' = a^{u \cdot t} \cdot b^{u \cdot s} \cdot M'^v = (g^w)^{u \cdot t} \cdot (M^w)^{u \cdot s} \cdot M'^v = (g^t)^{u \cdot w} \cdot (M^s)^{u \cdot w} \cdot M'^v = M'^{u \cdot v} \cdot M'^v = M'^{w'}$$

Kjer je $w' = uw + v \bmod q$.

Nad dobljenimi vrednostmi *m*', *z*', *a*' in *b*' prejemnik izvede zgoščevalno funkcijo $\Omega(m', z', a', b') = c'$ in vrne podpisniku izziv $c = c' / u \bmod q$.

Izziv *c* je izračunan tako, da podpisnik ne more odkriti spremenjenega sporočila *m*', ki ima veljaven podpis z zasebnim ključem podpisnika (*x*). Prav tako podpisnik ne more povezati plačil s pomočjo naključnega števila *w*, saj je bilo to spremenjeno v novo vrednost *w*'.

Korak 4: Podpisnik odgovori z $r = w + cx \bmod q$.

Korak 5: Prejemnik z izzivom *c* in odgovorom *r* preveri enakosti $ah^c = g^r$ in $bz^c = m^r$. Če je enakost potrjena, je podpis sprejet. Četvorka (*z*, *a*, *b*, *r*) sama še ni pravilni podpis za *m*. Lahko pa prejemnik izračuna $r' = ur + v \bmod q$ in tako dobi veljavni podpis za *m*', to je $sign(m') = (z', a', b', r')$.

Pokažimo, zakaj veljajo enakosti v Koraku 5:

$$a \cdot h^c = g^w \cdot (g^x)^c = g^{w+c \cdot x} = g^r$$

$$b \cdot z^c = M^w \cdot (M^x)^c = M^{w+c \cdot x} = M^r$$

2 DIGITALNI DENAR IN SISTEMI DIGITALNEGA DENARJA

Pojem digitalni denar (ali tudi elektronski denar) je pogosto napačno uporabljen za nekatere oblike denarja, ki so se »znebile« tradicionalne fizične oblike. Ko govorimo o digitalnem denarju, torej ne mislimo plačilnih kartic, debetnih kartic, spletnih bank ipd., katerih lastništvo ali dostopnost nam pravzaprav zgolj omogoča upravljanje z dematerializiranim (knjižnim) denarjem v bančnih in tem podobnih ustanovah.

Digitalni denar je v nasprotju s prej omenjenimi sredstvi po namembnosti bliže klasičnemu denarju. Uporabniki ga lahko dvignejo z bančnega računa, hranijo v svojih (elektronskih) denarnicah, prenašajo med (elektronskimi) denarnicami, plačujejo z njim in polagajo na bančne račune.

Kako torej opisati digitalni denar? Lahko rečemo, da gre za digitalni zapis ali sporočilo, ki

- služi kot denar,
- se ga ne da zlahka ponarediti,
- mu je možno potrditi avtentičnost s strani institucije, ki ga je izdala in
- je lahko varno prenesen drugam (v drugo denarnico, na bančni račun …)

Na kratko, digitalni denar je plačilno sporočilo, ki vsebuje digitalni podpis in služi kot medij za izmenjavo ali hranjenje vrednosti (povzeto po [ORL]).

Pogosto podpisan kos digitalne informacije z zgornjimi lastnostmi imenujemo kar **kovanec**.

Sistemi digitalnega denarja opisujejo tudi interakcije med udeleženci in operacije nad denarjem. Udeleženci so banke, trgovci in uporabniki. Nujno potrebne operacije so postavitev sistema, odprtje računa, dvig denarja z računa, plačilo in polog na račun. Opisi operacij, ki sledijo, so dovolj splošni, da veljajo za večino danes aktualnih sistemov digitalnega denarja.

2.1 Postavitev sistema

Pred odprtjem prvega računa banka tvori ustrezne parametre, npr. števila – generatorje izbrane grupe, pare zasebnih/javnih ključev in podobno. Del teh parametrov se javno objavi in so potrebni pri vseh ostalih operacijah, preostali (npr. zasebni ključi) pa ostanejo znani samo banki.

2.2 Odprtje računa

Odprtje računa poteka ob interakciji med uporabnikom in banko. Banka uporabniku določi enolično identifikacijo, obe strani si izmenjata vse potrebne parametre (javne ključe), če gre za sistem, ki vključuje denarnico z nadzornikom, se uporabniku dodeli še nadzornika (npr. v obliki pametne kartice).

2.3 Dvig denarja z računa

Dvig denarja je interakcija med uporabnikom in banko. Uporabnik najprej dokaže lastništvo nad svojim računom. Če gre za sistem, ki uporablja slepe podpise, se izvede slepi podpis banke nad kosom digitalne informacije, ki predstavlja denar. Če imamo denarnico z nadzornikom, se med zapisom digitalne informacije v uporabnikovo denarnico izvede še

interakcija z nadzornikom. Rezultat je s strani banke podpisan kos digitalne informacije, ki predstavlja denar.

2.4 Plačilo

Plačilo je interakcija med uporabnikom in trgovcem. V postopku plačila se kovanec, to je digitalno podpisan kos digitalne informacije, prenese od uporabnika k trgovcu. Še pred tem trgovec pošlje izziv uporabniku, na katerega ta odgovori. Če ima denarnico z nadzornikom, se pred odzivom izvede še interakcija z nadzornikom, saj brez te pravilen odziv ni mogoč. Namen vključevanja nadzornika v postopek plačevanja je preprečevanje dvojne porabe.

2.5 Polog

Polog je interakcija med trgovcem in banko. Trgovec v določenih časovnih intervalih (npr. dnevno ali tedensko) položi prejet denar na bančni račun. Banka pred tem preveri veljavnost denarja (preprečevanje dvojne porabe s strani uporabnika oz. dvojnega pologa s strani trgovca). Če je veljavnost potrjena, se kovanec v banki zapiše v bazo porabljenih.

V primeru, da pride do napake oz. goljufije s strani trgovca, je banki identiteta krivca znana. V primeru, da je prišlo do napake oz. goljufije (dvojne porabe) s strani uporabnika, lahko banka ugotovi identiteto tega.

2.6 Lastnosti sistemov digitalnega denarja

Večino lastnosti, ki so zaželene v sistemih digitalnega denarja, lahko uvrstimo v dve skupini:

- lastnosti, povezane z uporabnostjo in
- lastnosti, povezane z zasebnostjo in varnostjo.

Digitalna oblika seveda bistveno povečuje uporabnost, kljub temu pa sama po sebi ne zagotavlja nekaterih osnovnih uporabnosti klasičnih oblik. **Prenosljivost** denarja med elektronskimi denarnicami, **deljivost** (vračanje drobiža) so nekatere med njimi. Obstajajo pa tudi lastnosti, ki imajo bolj posreden vpliv na uporabnost. To so na primer **zanesljivost**, **razširjenost** in **interoperabilnost**. Različni sistemi digitalnega denarja se različno lotevajo zagotavljanja teh lastnosti.

Na drugi strani se zaradi narave interakcije med vpletenimi (banke, trgovci, uporabniki) pojavi problem anonimnosti uporabnika in v okviru tega zahteva po **nesledljivosti** in **nepovezljivosti**. Digitalni denar, ki zadošča tem zahtevam, pogosto imenujemo anonimni digitalni denar. S stališča banke in trgovcev je bistvena varnost in v okviru tega predvsem mehanizmi za preprečevanje dvojne porabe. Digitalni denar je namreč tako kot vsako digitalno informacijo lahko kopirati.

V nadaljevanju so podrobneje predstavljeni v zadnjih dveh odstavkih omenjeni problemi, ki jih sistemi digitalnega denarja različno rešujejo (ali pa jih sploh ne).

2.6.1 Zasebnost uporabnika

Iz plačila, ki ga izvede uporabnik, naj banka ne bi mogla ugotoviti identitete plačnika, niti če sodeluje z vsemi trgovinami. Ta zahteva je uresničljiva, če zagotovimo, da banka ne more povezati informacij iz plačila in predhodno opravljenega dviga denarja. Potrebno in zadostno je, da je informacija, ki jo uporabnik dobi ob dvigu denarja, statistično neodvisna od informacije, ki jo ta razkrije trgovini ob plačilu. To lastnost imenujemo **nesledljivost** (untraceability).

Nepovezljivost (unlinkability) še zaostri zgornjo zahtevo. Sistem, ki zagotavlja nepovezljivost, preprečuje banki, da bi ta povezala (vsaj) dve plačili, ki jih je opravil isti uporabnik. V primeru, da se razkrije identiteta uporabnika ob enem plačilu, namreč na tak način pridemo do informacije o vseh plačilih v verigi povezanih.

2.6.2 Varnost

S stališča banke, pa tudi trgovcev, je v sistemih digitalnega denarja tipa »off-line« največji varnostni problem dvojna (ali večkratna) poraba denarja. Digitalni denar je (tako kot vsako informacijo v digitalni obliki) enostavno kopirati.

Obstajata dva pristopa k preprečevanju poizkusov dvojne porabe:

- Preverjanje ob nalaganju denarja na račun. To preverjanje se opravi po že izvedenem plačilu s podvojenim digitalnim denarjem - pogosto nekaj dni ali več kasneje. Ker je preverjanje naknadno, blago ali storitev, ki jo dobimo v zameno, pa je tipično izročeno ali izvedena takoj, je tak način za banke, posebej pa trgovce, nesprejemljiv. To velja tudi v primerih, ko lahko odkrijemo identiteto uporabnika, ki je plačal s kopijo.

- Preverjanje ob plačilu je mnogo bolj sprejemljivo za banke in trgovce (ki nalagajo zbrani denar na račun), saj do plačila ne pride, če poizkuša uporabnik plačati s kopijo. Mehanizem, ki to omogoča, so denarnice z nadzorniki.

Običajno sta uporabljena oba pristopa. Preverjanje ob nalaganju denarja na račun je namreč nujno v primerih, ko uporabniku uspe razbiti zaščito nadzornika v elektronski denarnici.

Banka pa lahko v sistemih tipa »off-line« tudi lažno trdi, da je prišlo do dvojne porabe (framing). Zato mora obstajati možnost, da pošteni uporabniki dokažejo svojo nedolžnost.

2.6.3 Deljivost in prenosljivost digitalnega denarja

Deljivost digitalnega denarja se nanaša na možnost uporabe dela digitalno podpisanega denarja (torej sporočila o plačilu). S kriptografskega stališča ta zahteva ni trivialna. Če jo upoštevamo, so običajno potrebni kompromisi pri nekaterih drugih zahtevah, kot je na primer nesledljivost.

Če npr. banka s svojim privatnim ključem digitalno podpiše kovanec z neko fiksno vrednostjo, uporabnik pa želi porabiti samo del te vrednosti, smo pred problemom. Zaradi digitalnega podpisa je namreč sporočilo nedeljivo. Problem je lažje rešljiv v sistemih tipa »on-line«, medtem ko rešitve za sisteme tipa »off-line«, ki ne bi imela negativnih stranskih učinkov (v smislu učinkovitosti ali zagotavljanja zasebnosti), ni.

Prenosljivost digitalnega denarja se nanaša na možnost uporabnikov, da med sabo prenašajo denar brez sodelovanja banke. O prenosljivosti denarja je seveda smiselno govoriti le v sistemih tipa »off-line«. Prenosi denarja se izvajajo po dvigu denarja z banke in pred končnim plačilom. Velikost digitalnega kovanca z vsakim prenosom naraste, zato je največje število prenosov običajno omejeno. Tako kot za deljivost tudi za prenosljivost velja, da zahteva kompromise glede zasebnosti uporabnika.

2.6.4 Razširljivost, razširjenost in zanesljivost

Zasnova sistema digitalnega denarja lahko vpliva na razširljivost (scalability) sistema kot celote. Potreba po centralnem obdelovanju transakcij na primer ovira rast števila uporabnikov, zato se sistemi takim zahtevam poizkušajo izogniti.

Uporabniki želijo isti sistem digitalnega denarja uporabljati na več bankah in pri več trgovcih, skratka, želijo uporabljati sistem, ki je dovolj razširjen med bankami in trgovci. Če ne to, je zaželena vsaj interoperabilnost, to je zmožnost prenosa in pretvarjanja digitalnega denarja iz enega v drug sistem.

Dvig in plačevanje z digitalnim denarjem je pravzaprav prenašanje sporočil med dvema udeležencema. Zanesljivost pri hranjenju in prenašanju digitalnega denarja je nujen pogoj za uspešnost sistema.

3 BRANDSOV SISTEM DIGITALNEGA DENARJA Z NADZORNIKOM, TEMELJEČ NA PROBLEMU REPREZENTACIJE

V sistemu poteka interakcija med tremi tipi udeležencev: bankami, trgovci in uporabniki. Izberemo si banko , trgovca in uporabnika . si lahko predstavljamo tudi kot uporabnikovo pametno kartico, računalnik ipd.

3.1 Nekaj lastnosti

1. Sistem uporabnikom zagotavlja **zasebnost** (nesledljivost in nepovezljivost), bankam in trgovcem pa ustrezno **varnost** (preprečevanje dvojne porabe – tako ob plačilu, kot tudi ob pologu).
2. Sistem omogoča bankam odkrivanje identitete uporabnikov, ki poizkusijo večkrat porabiti isti kovanec.
3. Uporabnik je zaščiten pred bankami, ki bi ga lažno obtožile dvojne porabe (framing).
4. Uporabniki ne morejo zanikati izvedenega plačila (nonrepuditation).
5. Sistem dovoljuje obstoj večih bank.
6. Sistem uporablja Schnorrov protokol ob dvigu denarja (ko banka izvede omejeni slepi podpis) in ob interakciji s trgovcem (ko uporabnik izvede Schnorrov dokaz lastništva).
7. Možne so razširitve, ki omogočajo deljivost, uvedbo čekov ipd.

3.2 Koncept nadzornika

Digitalni denar je, še veliko bolj kot klasičen, podvržen nevarnosti ponarejanja oz. dvojne porabe. Ugotavljanje kopij šele ob polaganju denarja na račun običajno ni sprejemljivo. Zato uvedemo nadzornika , ki v obliki pametne kartice ali zaščitenega modula v kaki drugi obliki sodeluje pri dvigih in plačilih uporabnika. Brez nadzornika omenjenih dveh operacij uporabnik ne more izvesti pravilno. Nadzornik tudi prepreči, da bi uporabnik večkrat izvedel plačilo z istim kovancem. Banka pred odprtjem računa uporabniku zagotovi modul z nadzornikom, do vsebine katerega po uporabnik ne sme imeti dostopa. Na drugi strani mora biti tudi uporabnik prepričan, da nadzornik trgovcu ali banki ne posreduje informacij, ki bi utegnile razkriti njegovo identiteto.

Če uporabnik razbije zaščito nadzornika in dobi dostop do tam shranjenih podatkov, se varnost zmanjša na tisto, ki je značilna za Brandsov sistem, temelječ na problemu reprezentacije brez nadzornika.

Da dosežemo preprečevanje dvojne porabe pred plačilom, mora biti informacija, ki jo uporabnik dvigne (kovanec), razdeljena med uporabniški modul elektronske denarnice in nadzornika. Tako uporabnik ne more plačati brez sodelovanja nadzornika.

3.3 Postavitev sistema

Banka naključno tvori naslednje elemente:

- Javni ključ banke (g, h), s katerim izvaja podpise v postopku dviga denarja. Eksponent $x = \log_g h$ je zasebni ključ.
- Generatorja (g_1, g_2).

- Generator g_3 (»dummy« generator) ki zagotavlja, da identiteta poštenih uporabnikov ne more biti izračunana iz postopka plačila.

Grupo G_q, vpeljano v razdelku 1.1, konkretiziramo: G_q naj bo podgrupa Z^*_p, kjer je $p = kq + 1$ (p je praštevilo, $k \in$ N). Praktična implementacija sistema bi zahtevala vsaj 512 bitov dolg p in vsaj 140 bitov dolg q ([BRA93], stran 13).

3.4 Dokaz poznavanja reprezentacije

V tem razdelku je predstavljen postopek dokaza reprezentacije, ki je v splošnem enak tistemu iz poglavja 1.2, vendar je bolj specifičen v tem, da na strani prejemnika upošteva dva sodelujoča – uporabnika in nadzornika.

Imamo naslednjo situacijo: Uporabnik □ in nadzornik □ morata skupaj banki □ ali trgovcu □ (skratka, preverjevalcu □) dokazati poznavanje reprezentacije za m_1m_2, kjer je nadzorniku znana reprezentacija za m_1 in uporabniku reprezentacija za m_2. Zahtevamo, da se nadzorniku □ prepreči kazanje kakršnekoli informacije o svoji reprezentaciji za m_1 in da mu ni potrebno poznati m_2, da izvrši svoj del protokola.

Pred začetkom postopka predvidevamo, da nadzornik □ pozna reprezentacijo (x_1, …, x_k) za m_1 glede na (g_1, …, g_k) in da uporabnik □ pozna reprezentacijo (y_1, …, y_k) za m_2 glede na (g_1, …, g_k). □ pozna tudi $m = m_1m_2$.

Korak 1: □ tvori k naključnih števil $v_i \in_R$ □$_q$, izračuna $A' = \prod_{i=1}^{k} g_i^{v_i}$ in pošlje A' uporabniku □.

Korak 2: □ tvori k naključnih števil $w_i \in_R$ □$_q$ in število $d \in_R$ □$_q$. Potem izračuna $A = A' \cdot (\prod_{i=1}^{k} g_i^{w_i}) \cdot m_1^d$ in pošlje A preverjevalcu □.

Korak 3: □ tvori enoličen in naključen izziv $c \in_R$ □$_q$ in ga pošlje uporabniku □.

Korak 4: Uporabnik □ izračuna $c' = c + d \bmod q$ in pošlje izziv c' nadzorniku □.

Korak 5: Nadzornik □ se odzove s k števili $r'_i = v_i + c'x_i \bmod q$.

Korak 6: Uporabnik □ preveri, če je odziv nadzornika □ pravilen, to pomeni, če velja $\prod_{i=1}^{k} g_i^{r'_i} = A' m_1^{c'}$. Če je enakost potrjena, □ izračuna za $i \in \{1, …, k\}$ odzive $r_i = r'_i + w_i + cy_i \bmod q$ in jih pošlje preverjevalcu □.

□ verjame dokazu o poznavanju reprezentacije v primeru, ko je $\prod_{i=1}^{k} g_i^{r_i} = Am^c$.

3.5 Odpiranje računa

Uporabnik □ želi odpreti račun pri banki □:

Korak 1: □ tvori tri naključna števila u_1, u_2 in $t \in_R$□□□ $_q$. □ izračuna A□= $g_1^{u_1} g_2^{u_2}$ in pošlje $T' = A$□g^t nadzorniku □. Število t uporabimo zgolj za zakrivanje A□.

Korak 2: □ tvori dve naključni števili o_1 in $o_2 \in_R$□□ $_q$ in izračuna A□ = $g_1^{o1} g_2^{o2}$. Nadzornik potem podpiše $T = T'A$□ s svojo shemo za digitalni podpis in pošlje oboje, A□ in podpis uporabniku □.

Korak 3: □ verificira pravilnost podpisa nadzornika □ in če je verifikacija uspešna pošlje $T = T'A$□, nadzornikov podpis in t banki □.

Korak 4: □ verificira nadzornikov podpis in če je ta pravilen, izračuna T/g^t. Če je bil □ pošten, je rezultat deljenja enak $g_1^{o1+u1} g_2^{o2+u2}$.

Korak 5: □ in □ skupaj dokažeta poznavanje reprezentacije za T/g^t glede na (g_1, g_2). Glej opis postopka dokazovanja. V tem konkretnem primeru velja k = 2.

Če □ sprejme dokaz o poznavanju reprezentacije s strani □ in □, shrani $I = T/g^t$ in ostale podatke uporabnika skupaj z novo odprtim računom.

Digitalni podpis nadzornika v Koraku 2 zagotavlja banki, da □ ni simuliral nadzornika □.

3.6 Dvig denarja

Pred dvigom denarja (kovanca) z računa morata □ in □ skupaj dokazati banki □ poznavanje reprezentacije za I z uporabo protokola, opisanega na začetku tega poglavja (gre torej za dokazovanje lastništva nad bančnim računom). Uporabimo k = 2. Če je poznavanje reprezentacije potrjeno, se za vsak kovanec izvede naslednji protokol:

Korak 1: □ tvori dve naključni števili o_3 in $o_4 \in_R$□□□$_q$, izračuna B□ = $g_1^{o3} g_2^{o4}$ in pošlje B□ uporabniku □. B□ je število, ki ga □ uporabi, ko želi □ plačati z informacijo (kovancem), ki jo je dvignil.

Korak 2: □ odšteje ustrezno količino denarja (v našem primeru fiksno) z računa in izračuna število $m = Ig_3$. Ker lahko tudi □ sam izračuna to informacijo, mu je □ ne prenaša. □ pošlje $z = m^x$, $a = g^w$ in $b = m^w$ uporabniku □. Število w je naključno izbrano iz □$_q$.

Korak 3: □ tvori naključna števila α, β, $\gamma \in_R$□□□$_q^*$ in u ter $v \in_R$□□□$_q$. Izračuna $m = (A$□A□$)g_3$ in ga zakrije z uporabo α v $m' = m^\alpha = g_1^{u_1\alpha} g_2^{u_2\alpha} g_3^{\alpha}$. Sledi razdelitev m' na dva dela $m' = AB$. □ izračuna oba dela: $A = (A$□A□$)^\alpha B$□$^\beta g_3^\gamma$, $B = B$□$^{-\beta} g_3^{\alpha-\gamma}$. Nato zakrije še $z \rightarrow z' = z^\alpha$, $a \rightarrow a' = a^u g^v$, $b \rightarrow b' = b^{\alpha u}(m')^v$ in izračuna c' = □(m', z', a', b', A). Končno pošlje uporabnik □ banki □ izziv $c = c'/u \bmod q$.

Korak 4: Banka □ pošlje nazaj uporabniku odziv $r = xc + w \bmod q$.

Korak 5: Uporabnik □ sprejme denar, če potrdi enakosti $g^r = h^c a$ in $m^r = z^c b$. □ izračuna še $r' = v + ru \bmod q$.

Kovanec je torej predstavljen z {*A*, *B*, (*z*', *a*', *b*', *r*')}, kjer je tretji element (četverka) podpis banke nad *A* in *B*. Kljub temu, da je podpis opravljen z zasebnim ključem *x*, banka ne pozna *A* in *B*. Predstavitev kovanca ne vključuje njegove vrednosti, ker smo predvideli, da so vsi kovanci enake vrednosti.

3.7 Plačilo

Iz postopka za dvig denarja so uporabniku ostala naslednja števila: *A*, *B*, *sign*(*A*, *B*) = (*z*', *a*', *b*', *r*') in naključna števila α, β in $\gamma \in_R \square\square\square_q$. Postopek plačevanja sledi:

Korak 1: pošlje trojico *A*, *B* in *sign*(*A*, *B*) trgovcu .

Korak 2: Trgovec se prepriča, da je $AB \neq 1$ in preveri podpis po shemi za slepe digitalne podpise. Če je podpis veljaven, pošlje izziv $c \in \square_q^*$ uporabniku .

Korak 3: preveri, da je $c \neq 1$ in če je, izračuna $c' = \beta(1-c)/\alpha \bmod q$ ter to kot izziv pošlje nadzorniku .

Korak 4: Nadzornik preveri, če kovanec, ki ga želi porabiti, ni bil že porabljen. Če ni bil in če $c' \neq 0$, potem izračuna dva odziva $r'_1 = o_1 + c'o_3 \bmod q$ in $r'_2 = o_2 + c'o_4 \bmod q$ in ju pošlje uporabniku .

Korak 5: preveri, če velja enakost $g_1^{r'_1} g_2^{r'_2} = A\square B\square^{c'}$. Če velja, potem z odzivoma nadzornika izračuna $r_1 = \alpha(r'_1 + u_1) \bmod q$ in $r_2 = \alpha(r'_2 + u_2) \bmod q$. Tretji odziv r_3 lahko izračuna brez pomoči nadzornika: $r_3 = \gamma + c(\alpha - \gamma) \bmod q$. Vse tri odzive pošlje trgovcu .

Trgovec sprejme kovanec samo če potrdi enakost $g_1^{r_1} g_2^{r_2} g_3^{r_3} = AB^c$.

Uporabnik ne pozna reprezentacije svojega *I* (ker ne pozna o_1 in o_2), torej tudi ne pozna *A* brez pomoči nadzornika. Torej brez nadzornika kovanca ni mogoče porabiti, razen v primeru, ko uporabnik razbije zaščito nadzornika in pride do skritih števil.

3.8 Polog

Ko želi trgovec naložiti denar v banko , pošlje tej za vsak kovanec naslednje podatke: *A*, *B*, *sign*(*A*, *B*), *c*, r_1, r_2 in r_3.

Banka v svoji bazi porabljenih kovancev preveri, če se tam nahaja kovanec (*A*, *B*, *sign*(*A*, *B*)). Če obstaja in je trojica odzivov ter *c* enaka, gre za poizkus trgovca , da dvakrat unovči isto kovanec. tak poizkus prepreči.

Če kovanec v bazi obstaja, trojica odzivov (r_1, r_2 in r_3) in *c* pa so različni, potem gre za poizkus dvojne porabe s strani uporabnika . Recimo, da banka v drugo dobi (r'_1, r'_2, r'_3) in *c*'. Z dvema naboroma navedenih števil lahko razkrije identiteto uporabnika . Veljata naslednji relaciji:

$$g_1^{r1} g_2^{r2} g_3^{r3} = AB^c$$
$$g_1^{r'1} g_2^{r'2} g_3^{r'3} = AB^{c'}$$

Iz teh dveh relacij lahko izračunamo (celoten postopek je opisan v [BRA93], strani 65-67) vrednosti $o_1 + u_1$ in $o_2 + u_2$, kar pa nam da identiteto uporabnika $I = g_1^{o1+u1} g_2^{o2+u2}$

Dejstvo, da banka pozna podatka u_1 in u_1 (poleg že prej poznanih o_1 in o_2) je tudi dokaz, da je uporabnik poizkusi z dvojno porabo. Za računanje u_1 in u_2 brez podvojenega kovanca bi bilo namreč potrebno izračunati diskretni logaritem.

3.9 Razširitve opisanega sistema

V nadaljevanju so naštete nekatere od možnih razširitev predstavljenega sistema. Zgolj nakazane so tudi potrebne spremembe, ki razširitve omogočijo.

3.9.1 Različne denominacije

V sistemu digitalnega denarja, ki smo ga opisali v prejšnjem poglavju, je privzeta poenostavitev, da so vsi kovanci enake vrednosti. Na tem mestu bosta nakazani dve alternativi za razširitev sistema s kovanci različnih denominacij.

Prva možnost, ki je sicer relativno enostavna, vendar ne najbolj elegantna, je ta, da banka uporabi različne javne ključe (h, g) za podpisovanje kovancev različnih vrednosti. Banka torej objavi množico parov (g, h_1), (g, h_2), …, (g, h_k) in zase zadrži ustrezne zasebne ključe x_1, …, x_k. Na tak način lahko banka izdaja bankovce s k različnimi denominacijami.

Druga možnost je, da namesto enega generatorja g_3 uporabimo množico »denominacijskih« generatorjev g_{31}, g_{32}, …, g_{3k}. Vsak od generatorjev določa neko količino denarja. Če generator g_{3i} določa vrednost 2^{i-1} denarnih enot, potem lahko uporabljamo vse denominacije od 1 enote do 2^k enot, če le zmnožimo generatorje, ki ustrezajo binarni predstavitvi denominacije.

3.9.2 Možnost večkratne porabe kovanca

V sistemu, opisanem v prejšnjem poglavju, je identiteta uporabnika ob pologu razkrita, če je ta poizkušal z dvojno porabo. Precej enostavno lahko uvedemo spremembo, ki razkritje omogoča šele ob k+1 pologu istega kovanca. Tako lahko banka uporabniku namesto k kovancev izda samo enega, ki pa ga ta lahko k-krat porabi.

Taka razširitev je smiselna v sistemih, kjer se skuša porabo prostora v elektronski denarnici minimizirati. Plačila, ki so izvedena z istim kovancem, se da med sabo povezati - torej ni popolnoma zagotovljena nepovezljivost, eden od elementov zasebnosti uporabnika.

3.9.3 Uvedba čekov

Ena od slabosti predstavljenega sistema je tudi ta, da uporabniki za plačilo pogosto porabijo velike količine kovancev, posebej če je vrednost teh majhna. Uvedba čekov omenjeni problem reši, vendar pa zahteva nov protokol za povračilo denarja in spremembe v obstoječih postopkih.

Uporabnik ob dvigu čeka določi njegovo vrednost. Kasneje pri plačilu navede, kolikšen del te vrednosti porabi. Trgovec ček unovči na banki, uporabnik pa lahko v primeru, ko ni porabil celotne vrednosti, zahteva povračilo preostanka s strani banke.

Uvedba čekov zahteva razširitev obstoječe predstavitve kovanca. Vrednosti *AB* dodamo (produkt) denominacijski del. Uvedemo *k* parov $\{(e_1, f_1), \ldots, (e_k, f_k)\}$, kjer vsak par (e_i, f_i) predstavlja določeno količino denarja, recimo 2^{i-1} denarnih enot. Na tak način lahko vsak znesek do 2^k-1 denarnih enot predstavimo z ustreznim $e_1 f_1$. Ob dvigu tvorimo naključni *k*-terici exponentov $(a_i, \ldots, a_k)$ in $(b_i, \ldots, b_k)$ ter a_{k+1}, kjer so vsa števila med sabo različna in različna od 0. Število g_4 je naključni »dummy« generator, ki zagotavlja nepovezljivost med postopkom povračila uporabniku. Zapišimo denominacijski del čeka:

$$(\prod_{i=1}^{k} e_i^{a_i} f_i^{b_i}) \cdot g_4^{a_{k+1}}$$

V nadaljevanju so na kratko in zelo v grobem opisane spremembe v že znanih postopkih dvigovanja, plačevanja in pologa ter nov postopek povračila neporabljenega denarja uporabniku.

Med postopkom dvigovanja uporabnik dodatno tvori ustrezna števila (obe *k*-terici in a_{k+1}), izračuna denominacijski del m_1 in ga pošlje banki, ki ga shrani. V nadaljevanju postopka se m_1 uporabi kot vhodni parameter pri izračunu *m* oziroma *m*' na strani banke oziroma uporabnika.

Ko želi uporabnik porabiti določeno količino denarja, razkrije trgovcu (*A*, *B*, *sign*(*A*, *B*)) in znesek, predstavljen z ustreznimi eksponenti, to je vrednostmi *k*-teric iz postopka dvigovanja. Recimo, da so to $(a_1, \ldots, a_j)$ in $(b_1, \ldots, b_j)$, kjer je $k \geq j \geq 1$.

Ko trgovec unovči ček v banki, ta na listo povračil shrani eksponente, ki določajo porabljeno vrednost, recimo, da so to $(a_1, \ldots, a_j)$ in $(b_1, \ldots, b_j)$, kjer je $k \geq j \geq 1$.

Ko želi uporabnik dobiti povračilo za neporabljen del čeka, pošlje $(a_{j+1}, \ldots, a_k)$ in $(b_{j+1}, \ldots, b_k)$ banki. Če banka potrdi, da gre za del uporabnikovega čeka in če poslane vrednosti še niso na seznamu povračil, se izvede povračilo. Eksponenti se shranijo na listo povračil. V primeru, da uporabnik drugič unovči isti ček (pa čeprav le z delom vrednosti), se njegova identiteta odkrije.

3.9.4 Deljivost

Deljivost lahko dosežemo z modifikacijo v prejšnjem poglavju predstavljenega sistema s čeki. Dodatno zahtevamo, da uporabnik ob ponovnem unovčenju istega čeka ni razkrit, če gre za še neporabljen del čeka. To lahko dosežemo tako, da vsakemu $e_i^{a_i} f_i^{b_i}$ iz denominacijskega dela čeka dodamo še podatek, ki govori o identiteti uporabnika: $e_i^{a_i} f_i^{b_i} h_i^{a_i \alpha}$.

Za plačila, ki so izvedena z istim čekom, ni mogoče zagotoviti nepovezljivosti.

4 VIRI

[BRA93] S. Brands, An Efficient Off-line Electronic Cash System Based On The Representation Problem, CWI, 1993

[ORL] J. Orlin Grabbe, Digital Finance, http://orlingrabbe.org/digital_index.htm

[VUK01] M. Vuk, Digitalni denar in omejeni slepi podpisi, diplomsko delo, Fakulteta za matematiko in fiziko, 2001