Univerza v Ljubljani,
Fakulteta za računalništvo in informatiko

Seminar za predmet
Kriptografija in računalniška varnost :

Slepi podpis

Ljubljana, dne 5.10.2004 Slavko Drnovšček

1. Uvod

V seminarski nalogi sem poskušal podati neko zaključeno predstavitev slepega elektronskega podpisa. Začetek vsebuje pojasnitve nekaterih osnovnih zahtev in značilnosti elektronskega podpisa. V naslednjem poglavju pa se posvetim slepemu podpisu. V sami nalogi predstavim dva tipa slepih podpisov in sicer Chaum-ov slepi podpis, ki je bil prvi predstavljen slepi podpis, in Schnorr-ov slepi podpis. Kot posebni implementaciji slepega podpisa sem omenil tudi pošten slepi podpis, predlagan je bil za uporabo pri digitalnem plačevanju, in skupinski slepi podpis, ki je namenjen predvsem sistemom z več enakovrednimi podpisovalci. Zadnje poglavje je namenjeno varnosti in analizi slepih podpisov ter pregledu napadov.

1.1. Elektronski podpis

Glavni cilj kriptografije je, da dva osebka vzpostavita povezavo in varno komunicirata preko povezave, ki ne zagotavlja varnega prenosa podatkov. Nepovabljeni poslušalci se lahko pojavijo kjerkoli in kadarkoli med samim komuniciranjem. Zato, da se zavarujemo pred nepovabljenimi poslušalci, ki se pojavijo na prenosnem kanalu, poslane informacije šifriramo.

V zgodnji dobi šifriranja so bili sistemi zaščiteni s skritim ključem. Dva osebka, ki sta želela komunicirati med seboj, sta se najprej zmenila za skriti ključ. Ključ je moral ostati skrit pred ostalimi, če smo želeli ohraniti varno komunikacijo. Sam proces določitve ključa je lahko kompliciran in pojavljali so se pomisleki, ali se ne bi dalo šifrirati podatke brez predhodnega dogovora ali izmenjave ključa.

Diffie in Hellman [14] sta predstavila princip javne kriptografije, ki rešuje problem izmenjave ključev, ter nekatere druge kriptografske probleme. Ideja javne kriptografije je v uporabi dveh različnih ključev: javnega ključa za šifriranje in skritega ključa za dešifriranje. Ključa sta pridobljena na tak način, da poznavanje javnega ključa ne pove nič o skritem ključu in obratno.

Pojav javne kriptografije je pripeljal do pojma elektronskega podpisa, ki je elektronska različica prostoročnega podpisa. Namen elektronskega podpisa je, da uporabnik »podpiše« elektronski dokument in s tem zagotavlja, da je on ta dokument res sam napisal, poslal, itd. Uporaba elektronskega podpisa se je razširila na veliko področij in skladno s tem so se razvile različice elektronskega podpisa. Ena od teh različic je tudi slepi elektronski podpis.

1.2. Slepi elektronski podpis

Glavna ideja slepega podpisa se lahko prikaže iz znanega primera paprinatih dokumentov. Papirna verzija slepega podpisa se lahko izvede s pomočjo kuverte obložene z indigo papirjem. Podpis na zunanji strani kuverte se prepiše na list vsebovan v kuverti. Pri tem sam podpisovalec ne ve kaj je podpisal.

Slepi elektronski podpis je razvil Chaum [4] za potrebe anonimnosti uporabnika v sistemu elektronskega denarja. Uporablja se na področjih, kjer je anonimnost zelo pomembna; najbolj znani področji sta elektronski denar in elektronske volitve.

2. Digitalni podpis

Digitalni podpis je osnoven gradnik overjanja, pooblastitev in nezatajljivosti. Namen digitalnega podpisa je poskrbeti za način, kako oseba prilepi informaciji delček svoje identitete. Proces podpisovanja vsebuje preoblikovanje podatka in skrite informacije v enoto z imenom podpis.

2.1. Delovanje in uporaba digitalnih podpisov

V veliki večini zahodnih držav je digitalni podpis pravno izenačen s fizični podpisom na papirju. Same zahteve digitalnega podpisa po Kwangjo [10] so naslednje:

- učinkovitost (hitre in računsko nezahtevne operacije),
- neponarejenost podpisa (poznavanje določenega števila podpisov ne omogoča napadalcu generiranja podpisa v imenu nekoga drugega),
- overjanje uporabnika (iz podpisa se enolično določi podpisovalca),
- neuporabljiv na drugih sporočilih (podpisa enega sporočila se na da pripisati drugemu sporočilu),
- nespremenljiv (po podpisu ni mogoče spremeniti sporočila),
- preprečevanje zanikanja (podpisovalec sporočila po podpisu svojega podpisa ne more zatajiti).

Uporaben digitalni podpis mora v praksi po Menezes, van Oorschot in Vansone [2, str. 475-488] imeti naslednje značilnosti:

- podpisovanje mora biti učinkovito,
- preverjanje identitete podpisa mora biti učinkovito,
- zadosti velika življenska doba (računsko varen pred ponarejanjem vsaj toliko časa, kot je uporaben podpis).

Digitalne podpise delimo v dve skupini (po Menezes et al. [2, str. 475-488]) in sicer na:

- digitalne podpise z dodatkom (ang. appendix), ki potrebuje za preverjanje identitete tudi originalno sporočilo,
- digitalne podpise z izračunom sporočila (ang. message recovery), ki originalno sporočilo prebere iz samega podpisa.

3. Slepi podpis

3.1. Princip delovanja

V sistemu slepega podpisa nastopata dva. Uporabnik (A), ki želi podpisan dokument in podpisovalec (B), ki podpiše osebi A sporočilo. Seveda želimo, da oseba B podpiše sporočilo tako, da ne ve kaj je podpisal in, da tudi, če kasneje dobi originalno in šifrirano sporočilo in ju med seboj primerja, ne zna izračunati, kateri uporabnik je zahteval podpis sporočila.

Slepi podpis vsebuje po Chaum [4] naslednje dve funkcije:

- podpisovalno funkcijo S', ki je poznana le podpisovalcu B in ustrezna inverzna funkcija S, ki je javno objavljena. Veljati seveda mora naslednje :
  - $S(S'(x)) = x$ in
  - iz javne funkcije S se ne da izračunati zasebne funkcije S'.
- slepilno funkcijo C in njeno inverzno funkcijo C', ki sta poznani le uporabniku A. Velja pa naslednje :
  - $C'(S'(C(x))) = S'(x)$ in
  - iz poznanih $C(x)$ in S' ne moremo izračunati x.

Veljati tudi mora predikat r, ki preveri ali je iskanje veljavnih podpisov računsko prezahtevno. Ta predikat je pomemben predvsem iz varnostnih razlogov, njegova sestava pa je odvisna od implementacije slepega podpisa.

Zgoraj definirane funkcije so uporabljene v naslednjem vrstnem redu in sestavljajo protokol slepega podpisa:

- uporabnik A sestavi $C(x)$ iz sporočila x tako, da velja predikat $r(x)$ in ga pošlje v podpis osebi B,
- podpisovalec B podpiše sporočilo $C(x)$ s funkcijo S' in vrne podpisano sporočilo $S'(C(x))$ osebi A,
- uporabnik A nad podpisanim sporočilom uporabi inverzno slepilno funkcijo C' in sicer $C'(S'(C(x))) = S'(x)$,
- vsakdo lahko z uporabo javne dekriptirne funkcije S preveri, da je sporočilo podpisal podpisovalec B. Lahko pa tudi preveri veljavnost predikata $r(S(S'(x)))$; torej je iskanje veljavnih podpisov računsko prezahtevno.

Slepi podpis, sestavljen iz zgoraj navedenih funkcij in protokola, mora imeti naslednje varnostne lastnosti:

- digitalni podpis – vsakdo lahko z uporabo funkcije S preveri, kdo je podpisal sporočilo,
- slepi podpis – podpisovalec B ne ve nič o vsebini sporočila, ki ga podpiše; ne zna povezati niti komu je bilo to sporočilo poslano, pa čeprav prejemnik sporočila preveri verodostojnost sporočila pri njemu (podpisovalcu B)

(podpisovalec pri podpisovanju vidi skriti podatek in osebo A, pri preverjanju pa vidi podpisano sporočilo in prejemnika; ne zna pa iz skritega podatka izračunati podpisano sporočilo, ter tako povezati prejemnika sporočila z osebo A, ki je to sporočilo poslala v podpis; edina možnost je ugibanje med podpisi, ki jih je naredil, kjer pa je verjetnost zadetka obratnosorazmerna s številom podpisov),

- hranjenje podpisov je brez pomena, saj tudi če imamo shranjena predhodna podpisana sporočila ne znamo iz njih ponarediti podpisa za neko novo sporočilo (osebe, razen podpisovalca B, ne poznajo podpisovalne funkcije S' s katero bi lahko podpisali novo sporočilo).

3.2. Chaum-ov slepi podpis

David Chaum [4] je prvi predlagal in predstavil slepi podpis. Njegova shema sloni na RSA shemi za podpis.

3.2.1 Podpis z RSA shemo

Po Stinson [1] naj bo $n = pq$, kjer sta p in q veliki praštevili in naj bo e tak, da je $D(e,\phi(n)) = 1$. Naprej naj bo d tak, da je $de = 1 \pmod{\phi(n)}$. Podpisovalec ima javen ključ sestavljen iz (e,n) in zaseben ključ iz (d,p,q). Po Boneh [17] se je od uvedbe RSA sheme nanjo zvrstilo veliko število različnih napadov, a uničujoče učinkovitega napada še niso odkrili. Napadi so sicer odkrili nekaj pasti, ki se jih je dobro izogniti pri implementiranju RSA, vendar se je mogoče večini napadov izogniti tako, da pravilno pripravimo sporočilo pred šifriranjem ali podpisovanjem.

Šifriranje

$$E(e,n)(x) = x^e \pmod{n}$$

Odšifriranje

$$D(d,p,q) = y^d \pmod{n}$$

Šifriranje in odšifriranje sta inverzni operaciji. Za $x \in Z_n^*$ to sledi iz Eulerjeve kongruence :

$$(x^e)^d \equiv x^{r\varphi(n)+1} \equiv \left(x^{\varphi(n)}\right)^r x \equiv x \pmod{n}$$

Za podpis sporočila m mora oseba A narediti naslednje :

1. izračuna $M = H(m)$, kjer je $m \in \{0,1\}^*$ in H zgoščevalna funkcija,
2. izračuna $s = M^d \pmod{n}$,
3. s je podpis za sporočilo m.

Oseba B, ki preverja podpis osebe A pa naredi :

1. vzame overjeno kopijo javnega ključa (n, e) osebe A,
2. izračuna $M = H(m)$,
3. izračuna $M' = s^e \pmod n$
4. sprejme (m, s) če in samo če je $M = M'$

3.2.2 Slepi RSA podpis (Chaum-ov slepi podpis)

Predpostavimo, da imamo dve osebi :

- osebo A, ki rabi podpis osebe B za sporočilo M in,
- osebo B, ki to sporočilo podpiše.

Sam protokol Chaum-ovega slepega podpisa poteka v več korakih in je po Ramzan [12] naslednji :

- korak 1(oseba A)
  1. oseba A želi, da mu oseba B slepo podpiše sporočilo M . O tem ga tudi obvesti.
  2. Oseba A izbere $r \in_R Z_n^*$ in izračuna »zaslepljeno« sporočilo $\hat{M}$
     $\hat{M} = H(M) \cdot r^e \pmod n$
     kjer sta n in e vzeta iz javnega ključa osebe B.
  3. Oseba A pošlje $\hat{M}$ osebi B.

- Korak 2 (oseba B)
  1. oseba B pobere sporočilo $\hat{M}$ in ga podpiše
     $\sigma(\hat{M}) = \hat{M}^d \pmod n$
     pri čemer je
     $\hat{M}^d \pmod n = (H(M) \cdot r^e)^d \pmod n = H(M)^d \cdot r \pmod n$
  2. oseba B pošlje $\sigma(\hat{M})$ osebi A.

- Korak 3 (oseba A)
  1. oseba A vzame podpis $\sigma(\hat{M})$, ki je podpisano »zaslepljeno« sporočilo $\hat{M}$ in ga pretvori v podpis sporočila M
     $\sigma(M) = \sigma(\hat{M}) / r \pmod n$; deljenje je realizirano z množenjem inverznega elementa r^{-1}, ki ga običajno izračunamo z uporabo razširjenega evklidovega algoritma.
  2. par $(M, \sigma(M))$ je sedaj veljaven par sporočila in podpisa osebe B.

Najpomembnejši dejstvi, ki ju tu lahko opazimo sta :

- oseba B originalnega sporočila ne vidi (vidi le »zaslepljeno« sporočilo, ki je

zanj le naključno število in lahko le ugiba o njegovi vsebini)

- če oseba B kasneje naleti na par $(M, \sigma(M))$, z lahkoto ugotovi, da ga je sama podpisala; je pa omejena pri determiniranju komu in kdaj je to podpisala. V najboljšem lahko le ugiba med zabeleženimi podpisi, ki jih je naredila.

3.3. Schnorr-ov slepi podpis

Schnorr-ov digitalni podpis sloni na problemu diskretnega logaritma in je po Ramzan [12] varen v »random oracle« modelu.

3.3.1 Schnorr-ov digitalni podpis

Naj bo G podgrupa grupe Z_n^* z redom q, za neko vrednost n in praštevilo q. Izberemo generator $g \in G$ pri katerem je računanje diskretnega logaritma v G računsko nedosegljivo. Naj bo $x \neq 0$ skriti ključ in $y = g^x$ pa njegov javni ključ. Na koncu naj bo H zgoščevalna funkcija iz $\{0,1\}^*$ in njena ciljna množica je Z_q.

Za sporočilo $m \in \{0,1\}^*$ je par (c, s) veljaven Schnorr-ov podpis nad m, če zadovoljuje naslednjo verifikacijsko enačbo :

$$c = H(m, g^s y^c),$$

kjer se $(m, g^s y^c)$ razume kot seštevek m in $g^s y^c$.

Pri zgornji enačbi lahko vidimo, da se vrednost c pojavlja na obeh straneh enačbe. Z domnevo, da je H »collision resistant«, je mogoče sestaviti podpis le, če poznamo diskretni logaritem od y pri osnovi g.

Generiranje podpisa nad sporočilom m poteka po naslednjem vrstnem redu :

- izbere se naključno stevilo $r \in_R Z_q$,
- izračuna se $c = H(m, g^r)$,
- izračuna se $s = r - cx$, par (s, c) predstavlja veljaven Schnorr-ov podpis.

Za preverjanje podpisa je potrebno najprej izračunati :

$$g^s y^c = g^{r-cx} (g^x)^c = g^r \pmod{n}$$

in podpis se sprejme v primeru, da je :

$$H(m, g^s y^c) = H(m, g^r) = c.$$

Schnorr-ov podpis je učinkovit in praktičen. Čas potreben za podpis je predvsem odvisen od računanja enega potenčnega računa, ki ga lahko naredimo neodvisno od sporočila in tudi pred prejemom samega sporočila. Pri preverjanju podpisa je potrebno izračunati dva potenčna računa ($g^s y^c$), ki se ju lahko po Menezes et al. [2, str. 617] izračuna v približno istem času kot en potenčni račun.

Varnost sheme se lahko po Koblitz in Menezes [19] prevede na reševanje diskretnega logaritma v približno istem času.

3.3.2 Protokol slepega Schnorr podpisa

Sam protokol je povzet po Ramzan [12] in je bolj kompliciran, kot RSA slepi podpis. Vsebuje namreč več korakov za pridobitev slepega podpisa, pa tudi več je samega računanja.

Protokol pridobitve slepega podpisa (oseba A želi sporočilo slepo podpisano od osebe B) je naslednji :

- korak 1 (oseba B)
  1. izbere $\hat{r} \in_R Z_q$,
  2. nastavi se $\hat{t} = g^{\hat{r}} \pmod n$ in se ga pošlje osebi A.

- Korak 2 (oseba A)
  1. izbere se $\gamma, \delta \in_R Z_q$,
  2. nastavi se $t = \hat{t} g^{\gamma} y^{\delta} \pmod n$,
  3. nastavi se $c = H(m,t)$,
  4. nastavi se $\hat{c} = c - \delta \pmod q$ in se pošlje osebi B.

- Korak 3 (oseba B)
  1. nastavi se $\hat{s} = \hat{r} - \hat{c}x \pmod q$ in se pošlje osebi A.

- Korak 4 (oseba A)
  1. nastavi se $s = \hat{s} + \gamma \pmod q$

Slepi podpis je sedaj (c, s) in ni težko videti, da je ta podpis res slep, podpisovalec (oseba B) nikoli ne vidi vrednosti c -ja in s -ja. Obe vrednosti sta skriti s pomočjo naključnih vrednosti γ in δ. Lahko se tudi prepričamo, da je podpis veljaven in sicer:

$$g^s y^c = g^{\hat{s}+\gamma} y^{\hat{c}+\delta} = g^{\hat{r}-\hat{c}x+\gamma+\hat{c}x} y^{\delta} = \hat{t} g^{\gamma} y^{\delta} = t \pmod n$$

kar pomeni $c = H(m,t) = H\left(m, g^s y^c\right)$.

3.4. Pošten slepi podpis

Bistvo slepega podpisa je, da zagotavlja uporabnikom sistema anonimnost. Podpisovalec ne ve kaj je vsebovano v sporočilu, niti ne ve, komu je sporočilo podpisal, če slučajno naleti na neko podpisano sporočilo. Sistem je bil zasnovan za potrebe anonimnega digitalnega plačevanja. Vendar se pri samem poslovanju z banko pojavi problem, ki se mu reče pranje denarja. S tem, da je zagotovljena popolna anonimnost, se odpre velika možnost za opravljanje transakcij, v sklopu katerih bi nelegalno zaslužen denar lahko prikazali kot legalno pridobljen.

Pri reševanju problema zlorab anonimnosti slepega podpisa, pa tudi zlorab pri uporabi kriptografije, se je pojavila ideja o poštenem (ang. fair) slepemu podpisu. Ta deluje ideja razširjuje sam slepi podpis z mehanizmom sodnika, ki ima možnost videti vsako sporočilo in tudi povezovati sporočilo s pošiljateljem.

Po Micali [15] je slepi podpis S je lahko tudi pošteni slepi podpis, ko :

»S omogoča določeni skupini (in samo tej skupini) pod določenimi pogoji, ki jih določa zakon (in samo pod temi pogoji), razumevanje in sledenje vseh podpisov S, tudi če se uporabnik s tem ne strinja in/ali o tem sploh ne ve.«

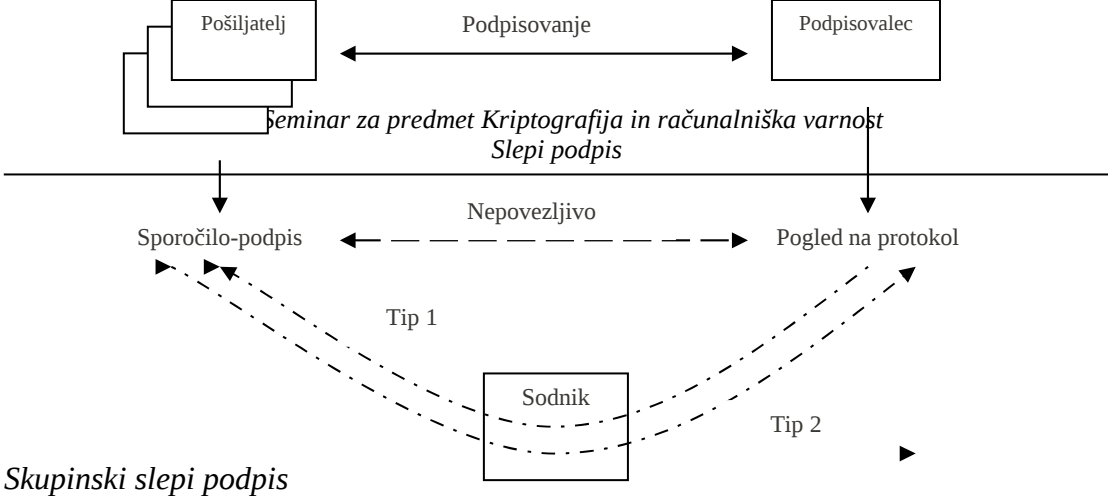
Model fair slepega podpisa je sestavljen iz :

- pošiljateljev,
- podpisovalca,
- zaupanja vredno osebo npr.: sodnika (ang. judge),
- protokol (slepega) podpisovanja med pošiljateljem in podpisovalcem in
- protokol pridobivanja povezav med podpisovalcem in sodnikom.

Protokol slepega podpisovanja generira slepi podpis, pri katerem podpisovalec omogoča anonimnost in nesledljivost. Podpisovalec s klicem protokola za pridobivanje povezav, pridobi od sodnika podatke, ki mu omogočajo prepoznati par slepega podpisa (sporočilo-podpis) in ostale podatke. Glede na to katere podatke pridobi podpisovalec ločimo dva tipa poštenega slepega podpisa :

- tip 1 (ang. type 1) :
  sodnik pošlje podatke, ki omogočijo podpisovalcu (ali komurkoli) prepoznavanje para sporočilo-podpis (prebiranje sporočila)
- tip 2 (ang. type 2) :
  sodnik pošlje podatke, ki podpisovalcu omogočijo identifikacijo pošiljatelja.

Teoretično lahko pri obeh tipih dobimo vse informacije. Pri tipu 1, kjer samo preberemo sporočilo, lahko z iskanjem po vseh podpisanih sporočilih ugotovimo kdo je bil pošiljatelj. Pri tipu 2 pa velja isto. Problem je samo v zamudnosti teh operacij, ker je potrebno pregledati vse možne podpisane podatke.



3.5. Skupinski slepi podpis

3.5.1 Skupinski digitalni podpis

Po Ramzan [12] se v skupinski digitalni shemi nahaja več podpisovalcev, od teh pa lahko vsak podpiše uporabniku sporočilo v imenu celotne skupine. Preverjanje podpisa pa se vrši s skupnim javnim ključem za celotno skupino. Zagotavljena je tudi anonimnost podpisovalca, nihče razen ustreznega skupinskega upravitelja ne zna ugotoviti, kdo je podpisal to sporočilo.

Podjetja lahko uporabljajo skupinske podpise pri preverjanju cenikov ali pomoči uporabnikom. Zaposleni v podjetju bi bili vključeni v skupino digitalnega podpisa, s katerim bi jamčili, da je res predstavnik podjetja odposlal sporočilo. Z lastnostjo anonimnosti, pa bi podjetje skrilo samo notranjo strukturo podjetja. Glavni nadzornik pa bi še zmeraj lahko, v primeru zlorab, ugotovil, kdo je sporno sporočilo podpisal.

Skupinski slepi podpis sta prva predlagala in implementirala Chaum in van Heyst.

3.5.2 Skupinski slepi digitalni podpis

Skupinski slepi podpis je kombinacija skupinskega digitalnega podpisa in slepega podpisa. Njihova uporaba je mogoča v veliki večini primerov, kjer se lahko pojavijo navadni slepi podpisi. Primer uporabe bi lahko bil plačilni sistem več povezanih bank skupaj ali pa volilni sistem porazdeljen na več volilnih mest (vsak bi imel svoj strežnik, ki bi sprejemal in podpisoval volilne listke).

3.5.3 Procedure dovoljene v skupinskem slepem digitalnem podpisu

Dovoljene procedure v skupinskem digitalnem slepem podpisu so po Ramzan [12] naslednje (iste procedure se uporabljajo tudi v skupinskem digitalnem podpisu) :

- $Setup$ - probabilističen algoritem, ki generira skupinski javni ključ γ in skriti nadzornikov ključ S .
- $Join$ - interaktivni protokol med nadzornikom in novim pripadnikom skupine; zgenerira se skriti ključ novega pripadnika in njegov članski certifikat.
- $Sign$ - interaktivni protokol med pripadnikom skupine B in zunanjo osebo A, ki vzame sporočilo m osebe A in skriti ključ osebe B x in naredi podpisano sporočilo s .
- $Verify$ - algoritem, ki ima na vhodu m , s in γ in preveri ali je s podpisano sporočilo m glede na javni ključ γ.
- $Open$ - algoritem, ki ima na vhodu s in S ter dobi identiteto pripadnika skupine, ki je podpisal sporočilo s .

3.5.4 Varnostne zahteve pri skupinskem slepem digitalnem podpisu

Varnostne zahteve pri skupinskem digitalnem slepem podpisu so naslednje (zahtevam skupinskega digitalnega podpisa se doda zahtevo po slepoti podpisa) :

- Zaslepljen podpis (ang. blindness of signatures)
  podpisovalec ne vidi vsebino sporočila, ki ga podpisuje. Podpisovalec tudi ne sme imeti nobenega pomnenja o samem podpisovanju sporočila, čeprav lahko preveri, da ga je res on podpisal.
- Neponarejenost podpisa (ang. unforgeability)
  samo člani skupine lahko izdajajo veljavne podpise v imenu celotne skupine.
- Pogojna anonimnost podpisovalca (ang. conditional signer anonymity)
  če vzamemo par sporočilo/podpis, se lahko takoj preveri, da je podpis opravil član skupine. Točno kdo je podpisal, pa lahko ugotovi le nadzornik skupine.
- Nezatajljiva identiteta podpisovalca (ang. undeniable signer identity)
  nadzornik skupine lahko v vsakem času preveri, kdo je izdal podpis. Še več, nadzornik lahko neki tretji osebi pokaže identiteto podpisovalca točno določenega sporočila, ne da bi pri tem ogrozil identitete podpisovalca in vseh njegovih prejšnjih in bodočih podpisov.
- Nepovezanost podpisov (ang. unlinkability)
  preverjanje ali sta bili dve sporočili podpisani s strani enega in istega podpisovalca je računsko neizvedljivo.
- Varnost pred napadi podtikanja (ang. security against framing attacks)
  nobena podmnožica članov skupine (vključno z nadzornikom) ni sposobna podpisati sporočila v imenu nekega drugega pripadnika skupine, ki pa ni del omenjene podmnožice. $Open$ procedura mora zmeraj vrniti identiteto osebe, ki je podpisala sporočilo.
- Varnost pred koalicijami (ang. coalition resistance)
  katerakoli podmnožica članov skupine (ki ne vključuje nadzornika) ne sme biti sposobna generirati veljavnega podpisa, ki pa ne vsebuje informacije o samem podpisovalcu. Torej, ne smejo biti sposobni generirati podpisa, ki ga procedure $Verify$ potrdi, procedure $Open$ pa ne vrne identitete podpisovalca.

4. Varnost slepega podpisa

4.1. Uvod

Vse od začetka javne kriptografije je bilo predlaganih veliko novih shem in veliko izmed njih je bilo razbitih. Najboljše zagotovilo, da je shema varna, je, da shema vzdrži večletne in številne kriptoanalitične napade ljudi s področja varnosti.

En koncept sestave shem v kriptografiji je sestavljanje take sheme, da se samo razbijanje prevede na že poznane probleme, npr.: faktorizacija pri RSA, ali pa katerikoli NP-polni problem. Takrat lahko rečemo, da je shema dokazano (ang. provably) varna. Vendar samo dejstvo, da lahko problem skrčimo na tak težak problem, nam ne ponuja dokaza, da je shema varna. Pove nam samo to, da če bi razbili to shemo, bi lahko v približno istem času lahko rešili problem, na katerega smo bazirali varnost naše sheme.

Idealno želimo, da ima shema naslednjo lastnost, da četudi nasprotnik (napadalec) ve, da je čistopis eden od dveh sporočil m_0 in m_1 in ima možnost izbrati obe sporočili in nato videti y^*, pridobljen tako, da šifriramo eno od sporočil, ima nasprotnik še zmeraj le približno 50% možnosti ugibanja katero sporočilo je bilo šifrirano. Možnosti se ne bi smele spremeniti niti v primeru, da lahko nasprotnik zahteva šifriranje kateregakoli sporočila pred in po izbiri m_0 in m_1. Ta lastnost nam zagotavlja, da je shema nerazpoznavno varna pred napadom z izbranim čistopisom (ang. indistinguishability-secure from chosen-ciphertext attack).

Pri slepemu podpisu se srečujemo z dvema različnima subjektoma: uporabnikom in podpisovalcem. V primeru, da gre za uporabo slepega podpisa za elektronsko plačevanje, sta to komitent in banka. V tem primeru je v igri tudi velik interes – denar. Uporabnik (komitent) ima skrit dokument in želi, da mu ga banka podpiše, ne da bi banka ta dokument prebrala. Po drugi strani, pa mora biti varnost podpisa taka, da uporabnik ne more ponarediti podpisa, pa čeprav je od banke lahko dobil že veliko število podpisanih dokumentov. Torej se pri varnosti slepega podpisa pojavljata dve lastnosti, ki ju moramo doseči :

- slepost (ang. blindness),
- neponaredljivost (ang. non-forgeability).

Raziskovalci se poslužujejo dveh načinov, s katerimi je mogoče dokazati varnost slepega podpisa. Ta dva načina sta po Juels, Luby in Ostrovsky [8] naslednja :

- dokazovanje o zapletenosti (ang. complexity-based proofs),
- dokazi na osnovi napovedovalno naključnega modela (ang. proofs based on random oracle model).

4.1.1 Dokazovanje o zapletenosti (ang. complexity-based proofs)

Ta način sta najprej predstavila Diffie in Hellman [14]. Njihovo dokazovanje v bistvu sloni na določenih poznanih problemih, ki so težavni. Če samo varnost kriptografske sheme, natančneje enosmerne zgoščevalne funkcije, reduciramo na razrešitev takih problemov, potem je shema dokazano varna. Pri čem je potrebno paziti tudi na celoten protokol sheme. Skozi zgodovino javne kriptografije je bila

večina uspešnih napadov na sheme usmerjena na sam protokol delovanja sheme in ne na zgoščevalno funkcijo, katero je mogoče reducirati na težak problem.

Ta pristop se je pokazal kot zelo učinkovit pri celi vrsti shem, podpisov in varnostnih protokolih. Dokazovanje varnosti s tem načinom ima prednost pred dokazovanjem na osnovi napovedovalno naključnega modela. Vendar se je doslej po Juels et al. [8] vse dokaze o varnosti slepega podpisa dokazalo le na osnovi napovedovalno naključnega modela.

4.1.2 Dokazi na osnovi napovedovalno naključnega modela (ang. random oracle model)

Ta dokaz se uporabi v primeru, ko ni mogoče dokazati varnosti s pomočjo dokazovanja o kompleksnosti. V tem primeru se predpostavi, da se sami osnovni gradniki v kriptografiji (ang. cryptographic primitive) npr.: DES, MD5, obnašajo kot idealne naključne funkcije. Varnost sheme se nato dokaže z domnevo, da se osnovni gradniki obnašajo skoraj idealno. Taki dokazi seveda niso tako »močni«, kot dokazi s pomočjo dokazovanja o kompleksnosti.

Pojavile so se tudi sheme (sicer precej drugačne kot sheme v uporabi) zgrajene tako, da je shema dokazano varna v napovedovalno naključnem modelu, vendar se je v praksi pokazala kot nezanesljiva.

4.2. Napadi na slepi podpis

4.2.1 Uvod – potrebne definicije

Avtorji Goldwasser, Micali in Rivest so v [16] podali naslednje definicije, ki nam pridejo prav pri dokazovanju varnosti.

Shema za podpis (ang. signature scheme) je definirana s tremi algoritmi $(Gen, Sign, Verify)$.

- $Gen(l^k)$ - probabilističen algoritem, ki v polinomskem času iz vhodnega niza varnostnega parametra vrne par javnega in privatnega ključa (pk, sk).
- $Sign(pk, sk, m)$ - probabilističen algoritem, ki v polinomskem času iz vhodnih podatkov (privatni ključ, javni ključ in sporočilo) izračuna podpis sporočila $\sigma(m)$ in nov skriti ključ sk' (v večini shem je ta ključ vedno isti).
- $Verify(pk, m, \sigma(m))$ - je deterministični algoritem, ki v polinomskem času iz vhodnih podatkov (privatni ključ, sporočilo, podpisano sporočilo) vrne $accept / reject$.

Vsako sporočilo, ki ga podpišemo najprej z uporabo prvega algoritma za pridobitev ključa in nato z drugim algoritmom za podpis sporočila, mora tretji algoritem sprejeti kot veljavno.

Varnost podpisovalne sheme

Če imamo napadalca A, ki lahko zahteva veliko število podpisov po svoji izbiri, in lahko iz teh pridobljenih podpisov izračuna podpis za neko novo vrednost, ki ga algoritem za verifikacijo sprejme, potem je napad na podpisovalno shemo uspel. Podpisovalna shema je definirana kot varna za vse konstante c in za vse probabilistične polinomske čase A-ja, če obstaja varnostni faktor $k_{c,A}$, da ja za vsak

$k > k_{c,A}$ verjetnost, da je A uspešen manjša od $1/k^c$.

Slepa shema za podpis je definirana z dvema interaktivnima turingovima strojema $(Signer, User)$ in dvema algoritmoma $(Gen, Verify)$.

$Gen(l^k)$ - probabilističen algoritem, ki v polinomskem času iz vhodnega niza varnostnega parametra vrne par javnega in privatnega ključa (pk, sk).

$Signer(pk, sk)$ in $User(pk, m)$ - sta par polinomsko vezanih probabilističnih interaktivnih turingovih strojev. Oba stroja imata (ločene) naslednje sestavine :

- bralni vhodni trak,
- pisalni izhodni trak,
- bralno/pisalni delovni trak,
- bralni trak z naključnimi vrednosti,
- dva komunikacijska trakova (eden bralni, drugi pisalni).

Oba stroja imata na vhodnih trakovih (kot privzeto vrednost) napisan pk, ki je ustvarjen s prvim algoritmom. Dodatno ima $User$ na vhodnem traku še sporočilo m. Dolžine vseh vhodnih trakov morajo biti polinomsko velike (glede na varnostni parameter). Oba stroja komunicirata po protokolu, ki traja polinomsko veliko korakov (glede na varnosti parameter). Na koncu ostane $Signer$ v enem od dveh stanj $completed$ ali $non-completed$; $User$ pa ima vrednost ali $fail$ ali $\sigma(m)$.

$Verify(pk, m, \sigma(m))$ - je deterministično algoritem, ki v polinomskem času iz vhodnih podatkov (privatni ključ, sporočilo, podpisano sporočilo) vrne $accept / reject$.

Varnost slepe podpisovalne sheme

Podpisovalna shema je definirana kot varna za vse konstante c in za vse probabilistične polinomske čase A-ja, če obstaja varnostni faktor $k_{c,A}$, da ja za vsak $k > k_{c,A}$ verjetnost, da je A uspešen manjša od $1/k^c$, če upoštevamo naslednje :

- slepost (ang. blindness)
  Naj bo $b \in_R \{0,1\}$ (b je naključen bit, ki je skrit A-ju). A, ki je podpisovalec, požene naslednje korake (pri tem želi ugotoviti b):
  - korak 1 : $(pk, sk) \leftarrow Gen(l^k)$,
  - korak 2 : $\{m_0, m_1\} \leftarrow A(l^k, pk, sk)$,
    (A naredi dva dokumenta, polinomska (glede na l^k), ta dva dokumenta sta leksikografsko urejena in sta lahko odvisna od pk in sk.
  - Korak 3 : naj bosta $\{m_b, m_{1-b}\}$ dokumenta $\{m_0, m_1\}$ urejena glede na vrednost b (ki ga A ne pozna). $A(l^k, pk, sk, m_0, m_1)$ paralelno zažene dva, med seboj neodvisna, protokola $User(pk, m_b)$ in $User(pk, m_{1-b})$.
  - Korak 4 : če prvi $User$ vrne vrednost $\sigma(m_b)$ (in ne $fail$) in tudi

drugi vrne $\sigma(m_{1-b})$ (in ne $fail$), potem si A doda vhod $\{\sigma(m_b), \sigma(m_{1-b})\}$ urejen glede na (m_0, m_1) (eden od obeh ali tudi oba lahko odgovorita tudi s $fail$).

- Korak 5 : A odgovori z bitom $\tilde{b}$ (glede na korake od 1 do 3 in korak 4, če so zadovoljivi pogoji).

Verjetnost, da je A ugotovil vrednosti b -ja je največ $\frac{1}{2} + \frac{1}{k^c}$.

- Neponaredljivost
  A je sedaj uporabnik in poskuša narediti »eden več« podpis.
  - Korak 1 : $(pk, sk) \leftarrow Gen(1^k)$
  - Korak 2 : A sedaj pridobi l veljavnih podpisov ($Signer$ je odgovoril z $completed$).
  - Korak 3 : A uporabi zbirko $\{(m_1, \sigma(m_1)), ..., (m_j, \sigma(m_j))\}$, kjer je vsak par $(m_i, \sigma(m_i)) \quad \forall i : 1 \le i \le j$, sprejet z algoritmom $Verify(pk, m_i, \sigma(m_i))$.

  Verjetnost, da je $j > l$ je največ $\frac{1}{k^c}$.

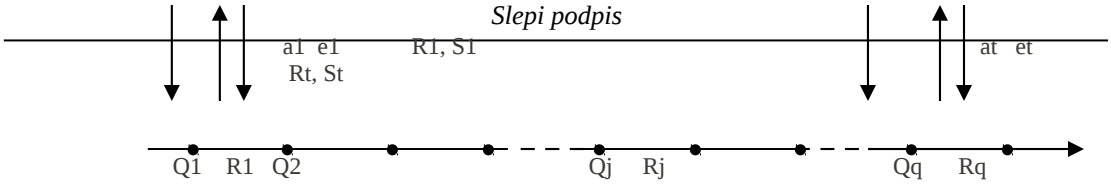
4.2.2 Uporaba ponaredkov

Pri slepemu podpisu so bolj nevarni napadi s ponaredki podpisov, kot pa sami napadi na elektronski podpis. To je posledica polja uporabe slepih podpisov – elektronsko poslovanje in elektronski denar. Poznamo naslednje napade s ponaredki :

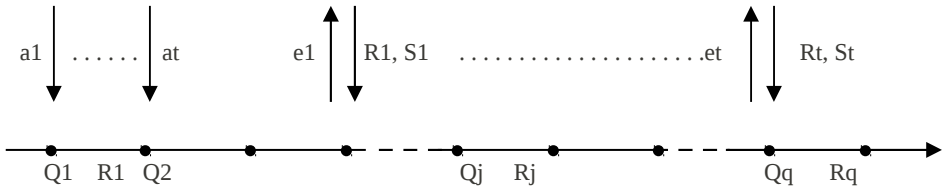
- $(l, l+1)$ ponaredek (ang. $(l, l+1)$ forgery)
  Iz nekega poljubnega števila l veljavnih podpisov napadalec sproducira $l+1$-i veljaven podpis.
- »še eden« ponaredek (ang. »one more« forgery)
  to je verzija $(l, l+1)$ ponaredka, z razliko, da je tukaj število l polinomsko veliko (glede na varnostni parameter k).
- Močan »še eden« ponaredek (ang. strong »one more« forgery)
  to je verzija $(l, l+1)$ ponaredka, kjer je l polilogaritmičen glede na varnostni parameter k ($\exists \alpha : l \le (\log k)^\alpha$).

Pri napadih (uporaba slepega podpisa na področju elektronskega denarja) pride ponavadi do dveh scenarijev :

- zaporedni napad (ang. sequential attack)
  napadalec zaporedno komunicira s podpisovalcem. To je npr.: ko napadalec zaporedoma dviguje denar.



- Paralelni napad (ang. parallel attack)
  napadalec l krat kontaktira paralelno s podpisovalcem. Napadalec začne novo sejo še preden se je stara končala. Ta napad je močnejši in je izveden s pomočje večje skupine napadalcev ob istem času.



4.2.3 Analiza RSA slepega podpisa

Pri lastnosti slepega podpisa, da podpisovalec ne vidi vsebine dokumenta, ki ga podpisuje, se je potrebno vprašati kako bi lahko podpisovalec odkril vrednost argumenta r. Oseba A, ki želi podpis od osebe B, mora po protokolu izbrati neko naključno število iz množice celih števil Z_n^* (n je vzet iz javnega ključa osebe B in je zelo veliko število). S tem številom oseba A »zaslepi« sporočilo. Edina možnost, da iz zaslepljenega sporočila dobimo originalno sporočilo, je s pomočjo števila r oziroma njegovega inverza r^{-1}.

Če predpostavimo, da je funkcija naključnega izbiranja števila idealna, potem je verjetnost, da oseba B ugane število r in s tem prebere originalno sporočilo enaka $\frac{1}{n}$ (seveda moramo upoštevati, da za r ne izberemo števila, ki sporočila sploh ne »zaslepi«, kot je na primer število 1).

Potrebno je še pogledati ali lahko napadalec iz j parov sporočila in podpisa lahko sam generira $j+1$ par. Primer ponaredljivosti podpisa se prestavi v razbivanje RSA sheme, pri čemer je tu napad olajšan, ker se lahko podpisovalcu pošlje v podpis sporočila po izbiri napadalca. Po Bohak [17] je to t.i. tehnika »skrivanja« (ang. blinding) in ni pretirano nevarna.

4.2.4 Analiza (slepih) skupinskih podpisov

Da imamo varen skupinski podpis je potrebno dokazati, da veljajo vse varnostne lastnosti podpisa (slepota podpisa, neponarejenost podpisa, pogojna anonimnost podpisovalca, nezatajljiva identiteta podpisovalca, nepovezanost podpisov, varnost pred napadi podtikanja in varnost pred koalicijami).

Po Wang [18] se skupinske sheme podpisov deli na dva tipa :

- sheme, ki izvirajo iz podpisov znanja (ang. signatures of knowledge),
- sheme, ki so izpeljane ad-hoc.

Med prvimi dobimo tudi nekatere računsko varne sheme, vendar so vse te sheme neučinkovite. Najbolj učinkovita shema tega tipa še zmeraj potrebuje okoli 13.000 RSA modularnih množenj pri podpisovanju in preverjanju skupinskega podpisa.

Drugi tip shem so učinkovite, za podpisovanje in preverjanje skupinskega podpisa potrebujemo le nekaj deset standardnih podpisov, vendar po Wang [18] do sedaj ni bila še nobena dokazana kot računsko varna. Veliko shem ne izpolnjuje pogoja neponarejenosti podpisa, nekaj pa tudi nepovezanosti podpisov. Ta dva pogoja sta kot kaže najtežja za zagotoviti pri skupinskem podpisu.

5. Uporaba

Vsakič, ko uporabimo telefon, kupimo blago z uporabo kreditne kartice ali si naročimo karto za kino preko interneta, se nekje informacija, da smo mi nekaj kupili oz. naredili zapiše v določeno zbirko podatkov. Vsi ti podatki se lahko celo povežejo med seboj in tvorijo dosje našega življenja. Ti podatki v rokah nepooblaščenih oseb lahko pomenijo izgubo svobodne izbire, denarja ali celo življenja.

Princip slepega podpisa omogoča dve stvari, in sicer :

- zmanjšuje možnost zlorabe,
- ohranja anonimnost uporabnikov.

Prva predstavitev pojma slepega podpisa je bila vezana na elektronsko poslovanje in na elektronski denar. Cilj je bil narediti elektronski denar neizsledljiv, tako kot je »navaden« papirnat denar. Ideja je, da bi uporabniku banka slepo podpisala kovanec, on bi ga pa predal prodajalcu v zameno za blago. Prodajalec bi lahko takoj preveril podpis banke in se s tem prepričal o veljavnosti kovanca. Nato pa bi si ta kovanec naložil na svoj račun. Pri tem ne bi banka vedela komu in za kaj je uporabnik porabil določen kovanec. Prav tako banka ne bi vedela od koga je prodajalec dobil denar, torej kdo je kupil blago pri njemu.

Sama uporaba slepega podpisa pa ni omejena le na elektronsko poslovanje. Z njim bi se lahko zaščitilo pravico do zasebnosti pri vrsti dejavnosti, ki pripadajo oz. jih izvaja veliko število ljudi, kot so na primer elektronske volitve, pa tudi pri plačevanju cestnin in pa podzemnih železnic.

Ena izmed idej je tudi implementacija slepega podpisa v borzno poslovanje. S tem bi se lahko zmanjšalo nihanje delnic v primeru, da določene delnice prodaja vplivna investicijska družba. Prodajalec bi bil anonimen in manjša bi bila verjetnost sprožitve plaza prodaje delnic raznih borznih špekulantov.

6. Reference

1. *D. R. Stinson* : Cryptography Theory and Practice, second edition, CRC Press, 2002.
2. *A. Menezes, P. van Oorschot, S. Vansone* : Handbook of Applied Cryptography, CRC Press, 1996.
3. *P. Horster, M. Michels, H. Petersen* : Meta-Message Recovery and Meta-Blind signature schemes based on the discrete logarithm problem and their applications, Proc. Asiacrypt '94, University of Wollongong, NSW, Australia.
4. *David Chaum* : Blind Signatures for Untraceable payments, CRYPTO .82, pages 199-203, New York 1983, Plenum Press.
5. *M. Stadler, J-P. Piveteau, J.Camenisch* : Fair Blind Signatures, Advances in Cryptology - Eurocrypt '95, Springer-Verlag (1995), 209-219.
6. *David Chaum* : Achieving Electronic Privacy, Scientific American, August 1992, pp 96-101.
7. Electronic Cash Protocol, www.cse.unsw.edu.au/~cs4012/cash3.ps.
8. *A. Juels, M.Luby, R. Ostrovsky* : Security of Blind Digital Signature, CRYPTO 97, Lecture Notes in Computer Science, pages 150-164, Springer-Verlag 1997.
9. *D. Pointcheval, J. Stern* : Provable Secure Blind Signature Schemes. Lecture Notes in Computer Science No. 1294, Proceedings of Asiacrypt 96, LNCS 1163, pp 252-265.
10. *K. Kwangjo* : Digital Signature, ICU prosojnice za predavanje, http://caislab.icu.ac.kr/course/2001/spring/ice605/down/010508.PDF.
11. *D. Pointcheval, J. Stern* : Security Arguments for Digital Signatures and Blind Signatures, Journal of Cryptology, pp 361-396, Springer-Verlag, 2000.
12. *Z. A. Ramzan* : Group Blind Digital Signatures: Theory and Applications, partial fulfillment on the Master of Science degree, MIT, May 1999, http://theory.lcs.mit.edu/~cis/theses/ramzanms.ps.
13. *Gerrit Bleumer* : Chaum Blind Signature, April 2004, http://www.win.tue.nl/~henkvt/GBI.ChaumBlindSignature.pdf
14. *W. Diffie, M. E. Hellman* : New directions in cryptography, IEEE Trans. Inform. Theory, IT-22:644-654, November 1976.
15. *S. Micali* : Fair Cryptosystems, Technical Report MIT/LCS/TR-579.b, November 1993
16. *S. Goldwasser, S. Micali, R. Rivest* : A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks, SIAM Journal of Computing Vol. 17, No 2 April 1988, pp 281-308.
17. *D. Boneh* : Twenty Years of Attacks on the RSA Cryptosystem, Notices of AMS, Feb. 1999, pp 203-212.
18. *G. Wang* : Security Analysis of Several Group Signature Schemes, IndoCrypt December 2003, pp 252-265.
19. *N. Koblitz, A. J. Menezes* : Another Look at »Provable Security«, Combinatorics & Optimization 2004-20, August 2004.