Zaklenjena skrivnost

Sheme za zapriseganje

Marinka Žitnik

3. julij 2012

Povzetek

We study commitment schemes in this project. Informally, a commitment scheme abstracts the notion of a locked box: the content of the box is hidden and a key is needed to unlock the box and reveal its content. First, we give a formal definition of a non-interactive commitment scheme accompanied by explanation. We then give a few constructions of commitment schemes and consider compositions of commitment schemes including bit-by-bit and hash-then-commit methods. A relaxed notion of commitment is presented, which suffices for some commitment applications. Finally, several applications of commitment schemes are provided along with an introduction and brief explanation of zero knowledge proofs.

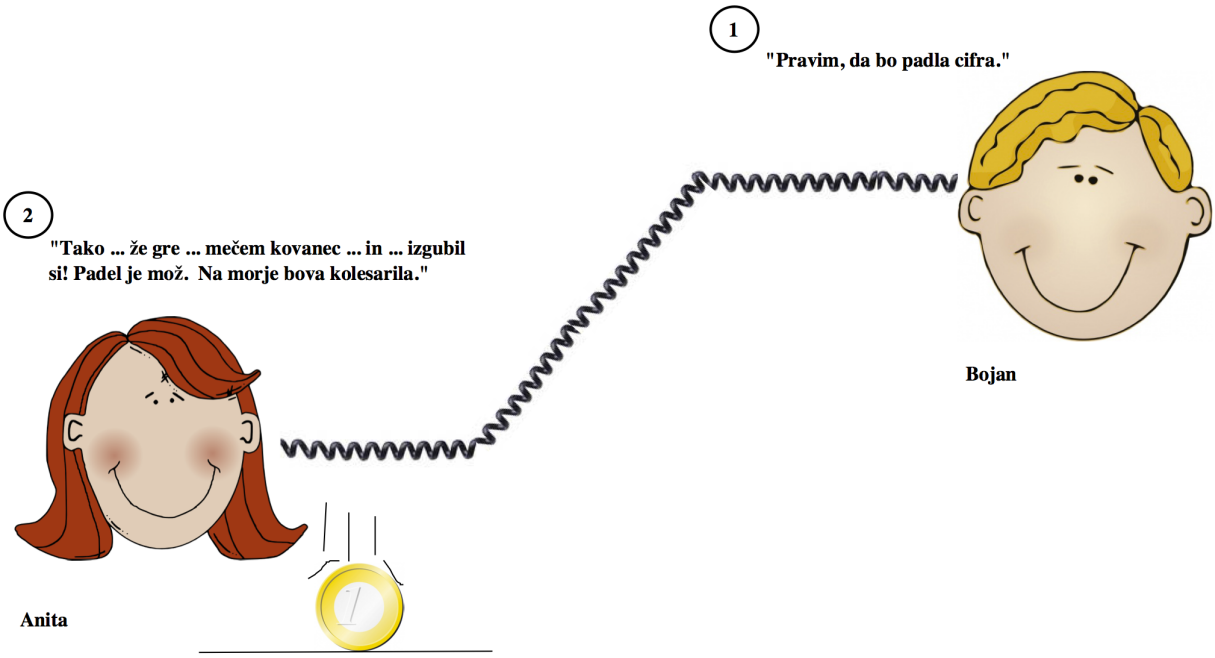
1 Uvod

Začnimo z zanimivim problemom metanja kovanca po telefonu, ki ni enostaven brez uporabe **sheme za zapriseganje.** Problem je prvi predstavil Manuel Blum, eden izmed začetnikov shem za zapriseganje, leta 1981 [2]. Recimo, da se naša znanca Anita in Bojan odpravljata na skupne poletne počitnice. Veselita se snidenja in poletnih radosti, saj živita v oddaljenih mestih in se nista videla več kot leto dni. Preden se odpravita na pot, se morata dogovoriti o načinu prevoza. Anita je zagreta kolesarka, Bojan nad kolesi ni navdušen; raje bi se peljal z avtomobilom. Uspelo se jima je sporazumeti, da bo najbolje če vržeta kovanec. Lažje rečeno, težje storjeno. Anita in Bojan drug drugemu ne zaupata povsem in imata premalo časa, da bi našla tretjo osebo, ki bi ji zaupala, da vrže kovanec namesto njiju. Bojan ni zadovoljen s protokolom na sliki 1, v katerem on napove *cifro* in Anita vrže kovanec ter oznani po telefonu: "Tako, že gre... Mečem kovanec... in... Izgubil si! Padel je mož in na morje bova kolesarila!" Kako lahko rešimo to zagato? Vsekakor ne sme Anita vreči kovanca in razkriti rezultat Bojanu, preden Bojan izbere *cifro* ali *moža*. V tem slučaju bi bila nezadovoljna Anita. Izgleda, da smo zašli v slepo ulico – nihče ne želi prvi razkriti izbrane vrednosti. Nastalo situacijo lahko enostavno rešimo s preprosto shemo za zapriseganje

1. Anita zapriseže naključno izbrani bit b_A in posreduje svojo izbiro c Bojanu. Bralec si lahko predstavlja, da je c zapečatena kuverta, ki hrani b_A ali šifrirana vrednost Anitine izbire. Bojan nima ključa, da bi odprl zapečateno kuverto.

2. Bojan izbere bit b_B in izbranega sporoči Aniti.

3. Sedaj Anita Bojanu razkrije b_A (posreduje mu ključ za odpiranje kuverte) in oba lahko izračunata rezultat, ki je $b = b_A \oplus b_B$.



Slika 1: Znanca Anita in Bojan se odpravljata na skupne počitnice in se morata odločiti o prevozu do morja. Anita bi se na pot odpravila s kolesom, Bojan pa z avtomobilom. Bojan prvi napove cifro in Anita v oddaljenem mestu vrže kovanec ter sporoči rezultat. Anita ravna v svojo korist in oznani, da je padel mož. To pomeni, da je Bojan izgubil, in bosta na morje kolesarila.

V razdelku 5.1 opišemo več shem, ki rešujejo problem meta kovanca po telefonu, in utemeljimo njihovo varnost.

Od Blumovih začetkov je bilo objavljenih veliko konstrukcij shem za zapriseganje, ki so primerne za uporabo v bolj zapletenih scenarijih. Juels in Wattenberg sta v [12] predlagala shemo za zapriseganje, ki je primerna za biometrično avtentikacijo z veliko šuma. Izkaže se, da so sheme za zapriseganje zelo uporabne v kriptografiji, zato je njihov razvoj aktivno področje raziskovanja. Pomembne so predvsem konstrukcije, ki zagotovljajo različne stopnje varnosti, ki so opisane v nadaljevanju projekta.

Organizacija projekta je sledeča. V razdelku 2 vpeljemo oznake in podamo osnovno definicijo shem za zapriseganje. Varnost shem obravnavamo v razdelku 3, kjer definiramo lastnosti brezpogojnega in računskega skrivanja ter brezpogojne in računske zaveze. V razdelku 4 je podanih več konkretnih konstrukcij shem za zapriseganje in možnosti za sestavljanje novih shem iz že obstoječih. Primeri uporabe shem so predstavljeni v razdelku 5, kjer se osredotočimo na uporabo shem pri dokazovanju brez razkritja znanja.

2 Sheme za zapriseganje

Sheme za zapriseganje lahko predstavimo kot matematične modele, s katerimi opišemo pojem **zaklenjenih škatel** v vsakdanjem življenju. Če v škatlo shranimo pomemben

predmet ali dokument in škatlo zaklenemo, običajno pričakujemo, da velja:

- vsebina v škatli je skrita in nedostopna brez ključa,
- vsebino lahko razkrijemo le tako, da škatlo odklenemo z ustreznim ključem.

Sheme za zapriseganje izhajajo iz želje po pošteni izmenjavi informacij med vpletenimi – nihče ne sme izvedeti za izbrane vrednosti drugih vpletenih dokler sam ne razkrije svoje vrednosti. Nadaljnje, želimo, da je razkritje vrednosti enolično oziroma, da lahko Bojan razkrije le vrednost, ki jo je Anita resnično izbrala. Rečemo, da se je Anita **zavezala k vrednosti** ali **vrednost zaprisegla**. S tega gledišča shema za zapriseganje posnema zaklenjeno škatlo s shranjeno vrednostjo.

V projektu raziščemo neinteraktivne sheme, pri katerih vsa komunikacija poteka v smeri od Anite proti Bojanu in ne obratno. V splošnih shemah si lahko Anita in Bojan izmenično pošiljata sporočila. Shema za zapriseganje preslika sporočilo m v par (c, d), kjer c predstavlja zaklenjeno škatlo in je d ključ, tako da velja:

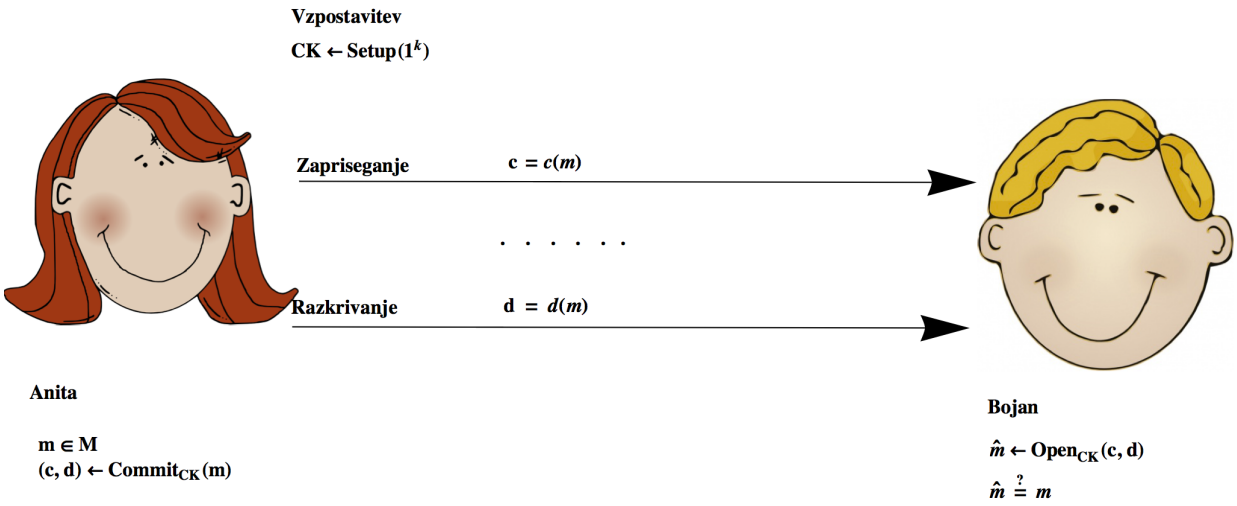
- zaklenjena škatla c ne razkriva nobene informacije o sporočilu m,
- in skupaj škatla s ključem (c, d) razkrije sporočilo m ter
- ni mogoče najti različnega ključa d', ki bi prav tako odprl zaklenjeno škatlo c ter razkril različno sporočilo od prvotnega, t.j. $m' \neq m$.

Definicija 2.1. Shema za zapriseganje je trojica algoritmov $(\texttt{Setup}, \texttt{Commit}, \texttt{Open})$, tako da velja:

1. Naključno izberemo **javni ključ** za zapriseganje dolžine k bitov $\texttt{CK}$, tako da uporabimo algoritem $\texttt{CK} \leftarrow \texttt{Setup}(1^k)$ (**Vzpostavitev**).
2. Za vsako sporočilo m iz množice možnih sporočil M z algoritmom $\texttt{Commit}_{\texttt{CK}}$ določimo pripadajoči *par za zapriseganje in razkritje* sporočila (c, d), t.j. $(c, d) \leftarrow \texttt{Commit}_{\texttt{CK}}(m)$. Par sestoji iz **zapriseženе vrednosti** ali **zaprisege** c in **vrednosti razkritja** ali **odpirača** d (**Zapriseganje**).
3. Z algoritmom $\texttt{Open}_{\texttt{CK}}$ razkrijemo sporočilo, t.j. $\texttt{Open}_{\texttt{CK}}(c,d) \rightarrow \tilde{m} \in M \cup \{\text{'fail'}\}$, kjer 'fail' označuje, da zapriseženа vrednost c ni veljavna za nobeno sporočilo m (**Razkrivanje**).

Definicija 2.2. Shema za zapriseganje je **pravilna**, če za vsako sporočilo $m \in M$ velja, $\texttt{Open}_{\texttt{CK}}(\texttt{Commit}_{\texttt{CK}}(m)) = m$.

Oglejmo si primer uporabe sheme za zapriseganje na sliki 2. Recimo, da Anita želi zapriseči sporočilo m Bojanu (s ključem $\texttt{CK}$). Anita sprva generira par za zapriseganje (c, d) in posreduje zaprisego c Bojanu. Kasneje ko Anita želi razkriti sporočilo m, Bojanu pošlje vrednost za razkritje d. Bojan sedaj pozna par za razkritje sporočila (c, d), zato z algoritmom $\texttt{Open}_{\texttt{CK}}$ razkrije sporočilo $\tilde{m}$. Če je shema pravilna, velja $m = \tilde{m}$.



Slika 2: Shema za zapriseganje. Ključ `CK` je javen, a ni povsem določeno, kdo ga generira, Anita, Bojan ali center zaupanja. V slučaju, da ključ generira ena stranka, lahko le-ta namreč izkoristi pomanjkljivosti sheme in ključ generira sebi v korist. Vzpostavitev sheme je zato odvisna od varnostnih lastnosti konkretne konstrukcije.

2.1 Varnostne zahteve

Kot smo omenili v uvodu, želimo zagotoviti vsaj dve lastnosti shem za zapriseganje:

- zaprisežena vrednost c Bojanu ne daje nobene informacije o sporočilu m in
- Anita ne more razkriti c na več različnih načinov.

Ti dve lastnosti imenujemo *skrivanje* in *zaveza*. Oglejmo si ju podrobneje.

1. **Skrivanje.** Računsko težko izvedljivo je, da nasprotnik generira dve sporočili $m_0, m_1 \in M$, tako da lahko razlikuje med zapriseženima vrednostima c_0, c_1. To z drugimi besedami pomeni, da $c(m)$ ne razkriva nobene informacije o sporočilu m. To lastnost definiramo z zahtevo, da je za vsak strank (A_1, A_2) verjetnost, da A_2 razpozna sporočilo m_0 iz zaprisege c približno enaka verjetnosti razpoznave sporočila m_1. Pišemo $c(m_0) \approx c(m_1)$ za vsak (m_0, m_1), ki ga zapriseže A_1. Omenimo le, da sta v spošnem stranki A_1 in A_2 probabilistična Turingova stroj s polinomsko časovno zahtevnostjo [9] (PPT – angl. *Probabilistic Polynomial Time*).

2. **Zaveza.** Računsko težko izvedljivo je, da nasprotnik izračuna *trčenje* (c, d, d') tako, da sta (c, d) in (c, d') veljavna para za zapriseganje za sporočili m in m', kjer $m \neq m'$. Za nas to pomeni, da je verjetnost, da lahko nasprotnik razkrije več različnih sporočil iz zapriseženе vrednosti, zanemarljiva.

Ob opisovanju varnostnih zahtev shem za zapriseganje moramo pomisliti na vzpostavitev sheme, t.j. generiranje javnega ključa `CK`. Nepremišljena izbira stranke, ki požene algoritem `Setup` za generiranje `CK`, lahko izniči lastnosti skrivanja in zaveze. Ključ `CK` je namreč javen, a ni povsem jasno, kdo ga generira: pošiljatelj ali prejemnik. V slučaju,

da ključ generira ena stranka, lahko le-ta izkoristi pomanjkljivosti sheme in ključ generira sebi v korist. Za ponazoritev; prejemnik Bojan bi lahko generiral ključ `CK`, s katerim bi lahko iz vrednosti c delno razkril sporočilo m; prekršena bi bila varnostna zahteva skrivanja. Podobno bi pošiljateljica Anita generirala ključ `CK`, s katerim bi zaprisego c lahko razkrila na več različnih načinov; prekršena bi bila zahteva po zavezi.

Izkaže se, da na to vprašanje ni lahkega odgovora. Možnih rešitev je več.

- Algoritem `Setup` požene tretja stranka, kateri zaupata obe stranki. Takim shemam pravimo sheme za zapriseganje *z javnimi parametri*.

- Uporabimo shemo. ki zagotavlja varnost, če bodisi pošiljatelj bodisi prejemnik generira ključ `CK` in ga javno objavi. Če je shema *informatično-teoretsko zavezujoča* (t.j., sporočilo je skrito le računsko in je v teoriji vsebovano v c), pogosto vsak ključ `CK` zagotavlja zavezo. To pomeni, da pošiljatelj ne more izbrati ključa, ki ne bi zagotavljal zaveze in je v njegovem interesu izbrati ključ, ki zagotavlja skrivanje. V tem slučaju lahko ključ generira pošiljatelj. Podobno, če je shema *informatično-teoretsko skrivna* (t.j., sporočilo je neodvisno od c, a je računsko zahtevno prekršiti lastnost zaveze), pogosto vsak ključ `CK` zagotavlja skrivanje. To pomeni, da prejemnik ne more izbrati ključa, ki ne bi zagotavljal skrivanja in je v njegovem interesu izbrati ključ, ki zagotavlja zavezo. V takih primerih lahko ključ generira prejemnik.

- Prejemnik generira nov ključ `CK` za vsako sporočilo in shema za zapriseganje postane *interaktivna*. Prejemnik in pošiljatelj lahko določita ključ skupaj po postopku interaktivnega protokola.

Shema za zapriseganje kot smo jo definirali, je nepravična do prejemnika, čemur pravimo **asimetričnost**. Čeprav pošiljatelj z zaprisego c zapriseže sporočilo m, prejemnik ne more spoznati m, vse dokler pošiljatelj ne pošlje d, ki razkrije sporočilo. To pomeni, da lahko pošiljatelj enostavno zavrne prejemnika tako, da mu ne posreduje d. Izkaže se, da se tovrstni asimetriji ne moremo izogniti v dvostranskih protokolih; ena stranka ima vedno prednost v smislu prekinitve protokola, preden druga stranka izve za vrednost izhoda.

3 Varnost shem

Motivacija definicije 3.1 je sledeča. Če ponavljamo eksperiment, v katerem se nek dogodek zgodi z zanemarljivo verjetnostjo, potem je pričakovano število ponovitev do prvega uspešnega poskusa super-polinomska funkcija (t.j. pričakovano število ponovitev ne moremo navzgor omejiti s polinomom). Poenostavljeno to pomeni, da se dogodki, ki se zgodijo z zanemarljivo verjetnostjo, zgodijo tako redko, da v polinomskih algoritmih niso realizirani [6].

Definicija 3.1. Funkcija $\varepsilon : \mathbb{N} \to \mathbb{R}$ je **zanemarljiva**, če za vsako naravno število c obstaja pripadajoče naravno število k_c, da velja $\varepsilon(k) \leq k^{-c}$ za vse $k > k_c$.

Definicija 3.2. Naj bosta dani slučajni spremenljivki U in V nad diskretno množico X. **Statistična razdalja** med U in V je $d(U, V) = \sum_{x \in X} |U(x) - V(x)|$.

Opišimo željene varnostne lastnosti shem za zapriseganje.

Brezpogojna zaveza [5]**.** Anita ne more spremeniti zapriseženе vrednosti, tudi če bi imela neomejeno računsko moč. V shemah z brezpogojno zavezo Anita požene vzpostavitveni algoritem $\texttt{CK} \leftarrow \texttt{Setup}(1^k)$ in generirani ključ $\texttt{CK}$ posreduje Bojanu. V shemah z brezpogojno zavezo je sporočilo enolično določeno iz zaprisege in Bojan sprejme nepravilno generiran ključ z zanemarljivo verjetnostjo.

Računska zaveza. Če Anita nima zelo velike računske moči, je verjetnost, da lahko spremeni že zapriseženo vrednost, zelo majhna. Bojan požene vzpostavitveni algoritem $\texttt{CK} \leftarrow \texttt{Setup}(1^k)$. Naj bo ε verjetnost, s katero Anita vrne zapriseženo vrednost z dvema veljavnima različnima razkritjema, t.j. $m \leftarrow \texttt{Open}_{\texttt{CK}}(c, d), m' \leftarrow \texttt{Open}_{\texttt{CK}}(c, d')$, $m' \neq m$. Funkcija ε mora biti zanemarljiva.

Brezpogojno skrivanje [5]**.** Zaprisega $c = c(m)$ ne razkriva (skoraj) nobene informacije o m, tudi če ima Bojan neomejeno računsko moč. Bojan požene vzpostavitveni algoritem $\texttt{CK} \leftarrow \texttt{Setup}(1^k)$ in posreduje javni ključ $\texttt{CK}$ Aniti. V slučaju enobitnih sporočil morata biti porazdelitvi, ki ju določata $\texttt{Commit}_{\texttt{CK}}(0)$ in $\texttt{Commit}_{\texttt{CK}}(1)$ statistično neločljivi – njuna statistična razdalja je zanemarljiva. Bojan sprejme nepravilno generiran ključ z zanemarljivo verjetnostjo. V najboljšem primeru je porazdelitev $\texttt{Commit}_{\texttt{CK}}(m)$ neodvisna od sporočila m – zaprisega c ne razkrije nobene informacije o m. Sheme s slednjo lastnostjo so sheme s *popolnim skrivanjem.*

Računsko skrivanje. Bojan s polinomsko računsko močjo težko prepozna sporočilo iz objavljene zaprisege. V shemah z računskim skrivanjem Anita požene vzpostavitveni algoritem. Naj bo ε_m verjetnost, da je zaprisega $c = 0$, če Anita zapriseže enobitno sporočilo $m \in \{0, 1\}$. Potem je statistična razdalja $d(\varepsilon_0, \varepsilon_1)$ zanemarljiva. Z drugimi besedami, nasprotnik ne more učinkovito določiti sporočila iz zaprisege z verjetnostjo, večjo od naključnega ugibanja sporočila.

Vsekakor sta lastnosti brezpogojne zaveze in skrivanja bolj zaželjeni od njunih računskih različic, zato bi si želeli le shem za zapriseganje ki bi bile brezpogojno varne, t.j. zanje bi veljala brezpogojna zaveza in brezpogojno skrivanje. Izkaže se, da take sheme ne obstajajo.

Lema 3.3. *Sheme za zapriseganje z lastnostmi brezpogojnega skrivanja in brezpogojne zaveze ne obstajajo.*

Dokaz. Recimo, da obstaja shema `(Setup, Commit, Open)` z lastnostima brezpogojnega skrivanja in brezpogojne zaveze in Anita z njo zapriseže sporočilo $m = 0$, t.j. $(c, d) = \texttt{Commit}_{\texttt{CK}}(0)$. Potem mora obstajati d', da velja $(c, d') = \texttt{Commit}_{\texttt{CK}}(1)$. Če tak d' ne obstaja, lahko Bojan zaključi, da c ne zaprisega sporočila $m = 0$, kar krši lastnost brezpogojnega skrivanja. Toda Anita lahko z neomejeno računsko močjo poišče ustrezni d' in spremeni zapriseženo vrednost m na $m' = 1$, ne da bi to ugotovil Bojan. To krši lastnost brezpogojne zaveze. □

4 Konstrukcije shem za zapriseganje

V tem razdelku predstavimo konkretne konstrukcije shem za zapriseganje, analizo varnosti posameznih shem in pristope k sestavljanju večbitnih shem za zapriseganje iz enobitnih shem.

4.1 Kriptografski sistem z zaprisego

Lastnost skrivanja shem za zapriseganje povsem ustreza zahtevi sistemov za kriptografijo z javnimi ključi (odpornih na napade z izbranim sporočilom), namreč $c(m_0) \approx c(m_1)$ za katerakoli m_0, m_1. Lastnost zaveze lahko podobno interpretiramo v okolju šifrirnih shem. Zaveza v shemah za zapriseganje pomeni, da je iz zaprisege c mogoče razkriti kvečjemu eno sporočilo. Za šifriranje bi to pomenilo, da dani kriptogram enolično šifrira sporočilo. Večina znanih kriptografskih sistemov za javno kriptografijo zadošča tej zahtevi.

To pomeni, da Anita s svojim tajnim ključem $\mathtt{SK}$ pravilno odšifrira vsako sporočilo, ki je bilo zakodirano z njenim javnim ključem $\mathtt{PK}$, t.j. $\forall m, \mathtt{SK}, \mathtt{PK} : \ D_{\mathtt{SK}}(E_{\mathtt{PK}}(m)) = m$. Torej, če obstaja kriptogram c in tajna ključa $\mathtt{SK}_0, \mathtt{SK}_1$, ki ustrezata javnemu $\mathtt{PK}$ ter $m_0 = D_{\mathtt{SK}_0}(c) \neq D_{\mathtt{SK}_1}(c) = m_1$, potem Bojan ob poslanem m_0 ne more biti prepričan, ali bo $E_{\mathtt{PK}}(m_0)$ pravilno odšifriran. Lahko se namreč zgodi, da Anita dešifrira s tajnim ključem $\mathtt{SK}_1$ in ne $\mathtt{SK}_0$, zato bo odšifrirala sporočilo m_1, ki ga Bojan v resnici ni poslal. Sistemi za kriptografijo, za katere kriptogram c z opisano lastnostjo ne obstaja, so zavezujoči in se imenujejo *kriptografski sistemi z zaprisego*[1] (angl. *commiting encryption*).

Lema 4.1. *Kriptografski sistem z zaprisego omogoča konstrukcijo varne sheme (tj. omogoča skrivanje in zavezo) za zapriseganje.*

Dokaz. Naj bo $S = (G, E, D)$ kriptografski sistem z zaprisego z javnimi ključi, ki je odporen na napade z izbranim sporočilom. Sestoji iz generatorja javnega in zasebnega ključa G, šifrirne funkcije E in odšifrirne funkcije D. Definirati želimo shemo za zapriseganje $(\mathtt{Setup}, \mathtt{Commit}, \mathtt{Open})$. To je enostavno, saj uporabimo $E_{\mathtt{PK}}(m)$ za zaprisego in tajni ključ $\mathtt{SK}$ za razkritje zapriseženе vrednosti. Težava je, kje shraniti $\mathtt{SK}$? $\mathtt{SK}$ ne moremo hraniti kot del ključa za zapriseganje $\mathtt{CK}$. Odgovor je preprost, skriti ključ ni potreben. c lahko razkrijemo z demonstracijo naključnosti r, uporabljenega pri šifriranju.

Definirajmo shemo za zapriseganje $(\mathtt{Setup}, \mathtt{Commit}, \mathtt{Open})$ za sporočilo $m \in M = \{0,1\}^k$, da velja:

- Algoritem $\mathtt{Setup}(1^k)$ naj vrne javni ključ $\mathtt{CK} = \mathtt{PK}$, kjer sta dana javni in zasebni ključ kriptografskega sistema $(\mathtt{PK}, \mathtt{SK}) \leftarrow G(1^k)$, ter je k varnostni parameter sheme.
- Algoritem za zapriseganje definiramo, da velja $(c, (m; r)) \leftarrow \mathtt{Commit}(m; r)$, kjer je r naključno izbran in je zaprisega enaka $c = E_{\mathtt{PK}}(m; r)$.
- Algoritem za razkrivanje definiramo, da velja $\tilde{m} \leftarrow \mathtt{Open}(c, (m; r))$, kjer $\tilde{m} = m$, če $c = E_{\mathtt{PK}}(m; r)$ in $\tilde{m} =$ 'fail' sicer.

Shema za zapriseganje je varna, ker je S kriptografski sistem z zaprisego (glej še uvod razdelka 4.1). □

V zgornjem dokazu skriti ključ ni bil uporabljen. To ponazarja ključno razliko med šifriranjem in zapriseganjem. Šifriranje zahteva zmožnost odšifriranja na osnovi c in univerzalnim skritim ključem (*neodvisno od sporočila* m), drugače, zapriseganje dovoljujejo razkrivanje z *sporočilno odvisnim* ključem d. Torej, d pogosto vsebuje sporočilo m v originalni obliki. Brez škode za splošnost lahko domnevamo $d = (m; r)$ in $\mathtt{Open}(m; (m, r))$ le preveri $c = \mathtt{Commit}(m; r)$ ter vrne m, če je preverjanje uspešno.

[1] Včasih je zaželeno (ohlapno) šifriranje z možnostjo napak pri odšifriranju. Kriptografski sistemi brez zaprisege so pomembni v elektronskem glasovanju in varnem skupnem računanju.

4.2 Enobitna shema s psevdonaključnim generatorjem

V tem razdelku si oglejmo preprosto enobitno shemo za zapriseganje, ki uporablja psevdonaključni generator.

Definicija 4.2. Naj bo dan psevdonaključni generator (PRG) $G : \{0,1\}^k \to \{0,1\}^{3k}$. Definirajmo shemo za zapriseganje $(\texttt{Setup}, \texttt{Commit}, \texttt{Open})$ za enobitno sporočilo $b \in M = \{0,1\}$, da velja:

- Za vzpostavitev sheme naključno izberemo binarni niz R dolžine $3k$ bitov, pri čemer je k varnostni parameter sheme.
- Par za zapriseganje je dan z $(c,(s,b)) \leftarrow \texttt{Commit}(b)$, pri čemer je s naključno izbran niz dolžine k bitov. Če je $b = 0$, je zaprisežena vrednost enaka $c = G(s)$, sicer je $c = G(s) \oplus R$.
- Ob razkritju zaprisege preverimo, ali je $c = G(s)$ (če $b = 0$) oziroma $c = G(s) \oplus R$. V nasprotnem primeru v proceduri `Open` vrnemo 'fail'.

Za uporabnost opisane sheme moramo preveriti, ali ima želene varnostne lastnosti (t.j. skrivanje in zaveza).

Skrivanje je zagotovljeno, ker se zanašamo na lastnosti psevdonaključnega generatorja G. Ob zaprisegi sporočila $b = 0$ dobimo zaprisego $c(0) = G(s)$, sicer je zaprisega $c(1) = G(s) \oplus R$ in $G(s) \approx R$.

Vprašajmo se še, ali je shema zavezujoča. Poglejmo si dva veljavna para za zapriseganje $(c,(s_0,G))$ in $(c,(s_1,G))$. Naj velja, da s prvim parom razkrijemo $\texttt{Open}(c,(s_0,G)) = 0$ in z drugim parom $\texttt{Open}(c,(s_1,G)) = 1$. Iz definicje sheme to pomeni, da mora veljati $G(s_0) = c$ in $G(s_1) \oplus R = c$. Nadaljnje, $G(s_0) = G(s_1) \oplus R$ ali zapisano drugače $G(s_0) \oplus G(s_1) = R$. Sedaj je največ 2^k možnih vrednosti za vsako vrednosti $G(s_0)$ in $G(s_1)$ in največ 2^{2k} možnih vrednosti, ki jih zavzame izraz $G(s_0) \oplus G(s_1)$. Spomnimo se, da niz R lahko zavzame katerokoli izmed 2^{3k} možnih vrednosti. Torej velja

$$P(\exists s_0, s_1 : G(s_0) \oplus G(s_1) = R) \leq \frac{2^{2k}}{2^{3k}} = \frac{1}{2^k} = \varepsilon(k).$$

Verjetnost, da taka s_0 in s_1, ki bi povzročila trčenje z naključnim R sploh obstajata, je zanemarljiva, zato je shema zavezujoča.

Predlagano shemo iz definicije 4.2 želimo posplošiti na večbitno shemo. To lahko storimo na dva načina:

1. sestavljanje po bitih, pri čemer pošiljatelj zapriseže vsak bit posebej in neodvisno;
2. v shemi iz definicije 4.2 povečamo množico možnih sporočil ter prilagodimo algoritme `Setup, Commit, Open`.

Odločimo se za drugi način. Naj bo prostor sporočil $M = \{0,1\}^k$ in $\mathbb{F}$ končno polje velikosti 2^{5k}. Elemente polja $\mathbb{F}$ naravno predstavimo kot $5k$-bitne nize. $\mathbb{F}$ ima karakteristiko 2, zato seštevanje in odštevanje v taki predstavitvi ustrezata operaciji XOR (oz. $\oplus$). Posplošitev povzema definicija 4.3.

Definicija 4.3. Naj bo dan PRG $G : \{0,1\}^k \to \{0,1\}^{5k}$. Definirajmo shemo za zapriseganje $(\texttt{Setup}, \texttt{Commit}, \texttt{Open})$ za sporočilo $m \in M = \{0,1\}^k$, da velja:

- Za vzpostavitev sheme naključno izberemo binarni niz R dolžine $5k$ bitov, pri čemer je k varnostni parameter sheme.

- Par za zapriseganje je $(c, (s, m)) \leftarrow \texttt{Commit}(m)$, kjer je s naključno izbran niz dolžine k bitov. Zaprisego izračunamo kot $c = G(s) \oplus (m \cdot R)$. Seštevanje in množenje se opravita v polju $\mathbb{F}$. Sporočilo m spredaj dopolnimo s fiksnim nizom dolžine $4k$, da je operacija množenja izvedljiva.

- Algoritem `Open` ob razkritju preveri, ali velja $c = G(s) \oplus (m \cdot R)$, sicer vrne 'fail'.

Lastnost skrivanja je zagotovljena kot v shemi iz definicije 4.2, saj vrednosti $G(s)$ le prištejemo fiksno sporočilo.

Pokažimo še, da velja tudi zaveza. Predpostavimo, da sta sporočili m_0 in m_1 različni in si oglejmo dva veljavna para za zapriseganje podana z $(c, (s_0, G))$ in $(c, (s_1, G))$. Iz definicije sheme za zapriseganje sledi $G(s_0) \oplus (m_0 \cdot R) = G(s_1) \oplus (m_1 \cdot R)$. Potem lahko izrazimo element R

$$R = (G(s_0) \oplus G(s_1)) \cdot (m_0 \oplus m_1)^{-1},$$

inverz je v polju $\mathbb{F}$. Za vrednost $(G(s_0) \oplus G(s_1)) \cdot (m_0 \oplus m_1)^{-1}$ imamo na voljo kvečjemu 2^{4k} možnih vrednosti (t.j. naključno izberemo vse bite nizov s_0, s_1, m_0, m_1). Vprašajmo se, kolikšna je verjetnost, da je naključno izbrani niz R ustrezne oblike. Verjetnost, da je R take oblike, znaša kvečjemu $2^{4k}/2^{5k} = 2^{-k} = \varepsilon(k)$.

4.3 Enobitna shema z enosmerno permutacijo

V tem razdelku predstavimo shemo za zapriseganje, ki temelji na obstoju enosmernih permutacij. Pred tem definirajmo **trdi bit** (angl. *hard-core*). Trdi bit enosmerne funkcije f je je predikat h (t.j. funkcija, ki na izhodu vrne en bit), katere vrednost $h(x)$ je enostavno izračunati, če je dan x. Če je dana le vrednost $f(x)$, ne obstaja učinkovit algoritem za izračun $h(x)$. Poenostavljeno, težko je izračunati inverz funkcije f.

Definicija 4.4. Naj bo dana enosmerna permutacija (OWP) f in h naj bo trdi bit za funkcijo f. Definirajmo shemo za zapriseganje $(\texttt{Setup}, \texttt{Commit}, \texttt{Open})$ za sporočilo $b \in M = \{0, 1\}$, da velja:

- Vzpostavitveni algoritem `Setup` določi opis funkcij f in h.

- Par za zapriseganje je določen z $(c, x) \leftarrow \texttt{Commit}(b)$, pri čemer je x naključno izbran binarni niz fiksne dolžine in velja $h(x) = b$ in $c = f(x)$.

- Ob razkritju zaprisege algoritem `Open` preveri, ali velja $b = h(x)$ in $c = f(x)$, sicer vrne 'fail'.

Z drugimi besedami, uporabimo vrednost $f(x)$, da zaprisežemo trdi bit $h(x)$.

Premislimo, kakšna je varnost opisane sheme. Skrivanje je doseženo, ker je določitev sporočila b iz pripadajoče zaprisege c enakovredna določitvi $h(x)$ le iz dane vrednosti $f(x)$. Ker je h trdi bit za f, nasprotnik ne more prepoznati b iz $c(b)$ in razlikovati med $c(0)$ ter $c(1)$ z verjetnostjo večjo od $1/2 + \varepsilon(k)$.

Zaveza je dosežena, ker je funkcija f permutacija in sta zaprisega $c = f(x)$ ter $b = h(x)$ enolično določeni.

V shemi iz definicije 4.4 lahko algoritem za vzpostavitev `Setup` požene bodisi pošiljatelj bodisi prejemnik. Slabost opisane sheme je, da omogoča zaprisego le enega bita. Če je več bitov permutacije f trdih, lahko zaprisežemo tudi več bitov, a to ni pogosta uporaba.

4.4 Shema z zgoščevalno funkcijo

Naj bo H družina zgoščevalnih funkcij, ki so odporne za trčenja (CRHF). Intuitivno je enostavno doseči (računsko) zavezo z uporabo naključne funkcije $h \in H$, tako da zaprisežemo sporočilo x z $h(x)$. Toda $h(x)$ običajno ne skrije sporočila x, zato moramo shemo prilagoditi za dosego skrivanja.

Definicija 4.5. Naj bo dana družina zgoščevalnih funkcij, ki so odporne za trčenja $H : \{0,1\}^L \to \{0,1\}^l$. Naj bo množica sporočil dana z $M = \{0,1\}^n$ in naj velja $L >> l + n$. Naj bo U družina popolnih univerzalnih zgoščevalnih funkcij, $U : \{0,1\}^L \to \{0,1\}^n$. Definirajmo shemo za zapriseganje (`Setup`, `Commit`, `Open`) za sporočilo $m \in M$ tako, da velja:

- Vzpostavitveni algoritem `Setup` naključno izbere funkcijo h iz družine zgoščevalnih funkcij H.
- Par za zapriseganje je določen z $(c, (u, x)) \leftarrow$ `Commit`(m), pri čemer je x naključno izbran binarni niz fiksne dolžine in u naključno izbrana univerzalna zgoščevalna funkcija iz U. Algoritem `Commit` zaprisego izračuna z $c = (u, h(x))$, da velja $u(x) = m$.
- Ob razkritju zaprisege algoritem `Open` preveri, ali velja $m = u(x)$ in $c = (u, h(x))$, sicer vrne 'fail'.

Sedaj nas zanimajo varnostne lastnosti sheme. Zaveza je dosežena, ker trčenje oblike $c = (u, y), d = (u, x), d' = (u, x')$ implicira $h(x) = h(x') = y$. Ker je h naključno izbrana iz družine H, je nasprotnik prisiljen uporabiti $x = x'$, a potem dobimo $m = u(x) = u(x') = m'$, torej sporočili nista različni.

Skrivanje je težje dokazati. Tukaj uporabimo pogoj iz definicije 4.5, $L >> l + n$. Intuitivno, $h(x)$ razkrije l izmed L bitov informacije o izbranem x. Potem univerzalnost U in pogoj $L - l >> n$ implicirata, da izbira u, čeprav z omejitvijo $u(x) = m$, ohrani porazdelitev u škoraj"enakomerno in neodvisno od m. To pomeni, da sta vrednosti u in $h(x)$ neodvisni od sporočila $m \in \{0,1\}^n$ in izgledata naključno za nasprotnika.

Iz definicije sheme 4.5 opazimo, da je velikost zaprisežene n-bitne vrednosti $\mathcal{O}(L) >> n$ (u predstavimo z $\mathcal{O}(L)$ biti). Metodo s CRHF lahko uporabimo, da zmanjšamo velikost zaprisege na $\mathcal{O}(l)$, kar je lahko veliko manjše od velikosti sporočila m.

4.5 Pedersenova enobitna shema

Pedersenova shema za zapriseganje je primer sheme, katere varnost temelji na zahtevnosti problema diskretnega logaritma.

Definicija 4.6. Definirajmo enobitno Pedersenovo shemo (`Setup`, `Commit`, `Open`) za zapriseganje sporočila $b \in M = \{0,1\}$ tako, da velja:

- Vzpostavitveni algoritem `Setup` določi trojico števil $(p, g, y) \leftarrow$ `Setup`. Pri tem je p praštevilo, y naključno izbran element iz grupe $\mathbb{Z}_p^*$ in g naključno izbran generator grupe $\mathbb{Z}_p^*$.
- Par za zapriseganje je določen z $(c, (r, b)) \leftarrow$ `Commit`(b), pri čemer je r naključni element $\mathbb{Z}_p^*$. Zaprisego izračunamo $c = g^r y^b \ mod \ p$.
- Ob razkritju zaprisege algoritem `Open` preveri, ali velja $c = g^r y^b$, sicer vrne 'fail'.

Predstavimo varnostne lastnosti enobitne Pedersenove sheme za zapriseganje. Skrivanje je doseženo, ker je r izbran naključno iz $\mathbb{Z}_p^*$ in zato sta oba $c(0) = g^r$ in $c(1) = g^r y$ naključna elementa iz $\mathbb{Z}_p^*$.

Po drugi strani bi iskanje r_0, r_1, tako da bi veljalo `Open`$(c, (r_0, 0)) = 0$ in `Open`$(c, (r_1, 1)) = 1$, zahtevalo $g^{r_0} = g^{r_1} y$. Iz slednje enačbe izrazimo y in dobimo $y = g^{r_0 - r_1}$. To pomeni, da bi nasprotnik moral izračunati diskretni logaritem naključno izbranega y. Shema za zapriseganje iz 4.6 je zavezujoča pod predpostavko težavnosti diskretnega logaritma.

4.6 Sestavljanje novih shem za zapriseganje iz obstoječih

4.6.1 Bit-po-bit sheme

Recimo, da je na voljo varna shema za zapriseganje za majhen prostor sporočil, $M = \{0, 1\}$. Zanima nas, ali lahko zgradimo varno shemo za zapriseganje večbitnih sporočil z večkratno zaporedno uporabo enobitne sheme.

Lema 4.7. *Če je $C =$ (`Setup`, `Commit`, `Open`) varna shema za zapriseganje (t.j., izpolnjeni sta zahtevi skrivanja in zaveze) za prostor $M = \{0, 1\}$, potem je spremenjena shema C', dobljena iz C s $p(k)$-kratnim bitnim sestavljanjem, varna shema za zapriseganje za prostor sporočil $M' = \{0, 1\}^{p(k)}$ za vsak polinom $p(k)$.*

Dokaz, da shema za zapriseganje C' v lemi 4.7 zadošča lastnosti skrivanja, je podoben kot v primeru šifriranja. Za potrditev zaveze velja, da iskanje trčenja za različni sporočili $m_0, m_1 \in \{0, 1\}^{p(k)}$ pomeni iskanje 0/1-trčenja za nek $i \in \{1, \ldots p(k)\}$.

4.6.2 Zgosti-nato-zaprisezi sheme

Podobno kot lahko zgoščevalne funkcije koristno uporabimo v digitalnih podpisih (zgosti-nato-podpiši), je možno tudi pri shemah za zapriseganje. Osnovna zahteva digitalnih podpisov in shem za zapriseganje je, da mora biti iskanje trčenj dveh (ali več) sporočil težavno.

Lema 4.8. *Naj bo dana družina zgoščevalnih funkcij odpornih za trčenja H, pri čemer so funkcije iz družine preslikave iz L-bitnih vrednosti v l-bitne. Naj bo $C' =$ (`Setup`$'$, `Commit`$'$, `Open`$'$) varna shema za zapriseganje l-bitnih sporočil. Potem je shema $C =$ (`Setup`, `Commit`, `Open`) varna za zapriseganje L-bitnih sporočil, če velja:*

- *Javni ključ sheme zgradimo kot* `CK` $=$ (`CK`$'$, h), *kjer je* `CK'` *javni ključ sheme C' in $h \leftarrow H$, zgoščevalna funkcija.*
- *Algoritem za zapriseganje je $(c', (d', m)) \leftarrow$ `Commit`$_{\texttt{CK}}(m)$, kjer $(c', d') \leftarrow$ `Commit`$'_{\texttt{CK}'}(h(m))$ par za zapriseganje, dobljen iz sheme C' za zgoščeno sporočilo m.*

- *Sporočilo razkrijemo z algoritmom* $\texttt{Open}_{\texttt{CK}}(c', (d', m)) = \tilde{m}$. *Velja* $\tilde{m} = m$, *če je* (c', d') *par za zapriseganje v shemi* C' *za sporočilo* $h(m)$, *sicer* $\tilde{m} =$ *'fail'*.

Osnutek dokaza. Lastnost skrivanja enostavno sledi iz C': $\texttt{Commit}'(h(m_0)) \approx \texttt{Commit}'(h(m_1))$. Za zavezo velja, kolizijski trojček $(c', (d'_0, m_0), (d'_1, m_1))$ implicira bodisi $h(m_0) = h(m_1)$ trčenje v h bodisi da (c', d'_0, d'_1) povzroči trčenje za par $h(m_0) \neq h(m_1)$. □

Pripomnimo, da s kombiniranjem shem za zapriseganje z uporabo CRHF in pristopa zgosti-nato-zaprisezi lahko dobimo izredno kompaktne in učinkovite sheme za zapriseganje, pri katerih je velikost zaprisege za poljubno dolga sporočila kvečjemu tolikšna, kot je velikost varnostnih parametrov [11].

5 Primeri uporabe shem

5.1 Metanje kovanca po telefonu

Osnovni primer uporabe pri obravnavi shem za zapriseganje je metanje kovanca po telefonu (oziroma preko medijev, kjer Anita in Bojan ne vidita meta kovanca). V tem primeru želimo zagotoviti varnost pred nepošteno stranko, Anito ali Bojanom. Podajamo tri konstrukcije shem za zapriseganje, ki so uporabne za metanje kovance. Njihova varnost temelji na:

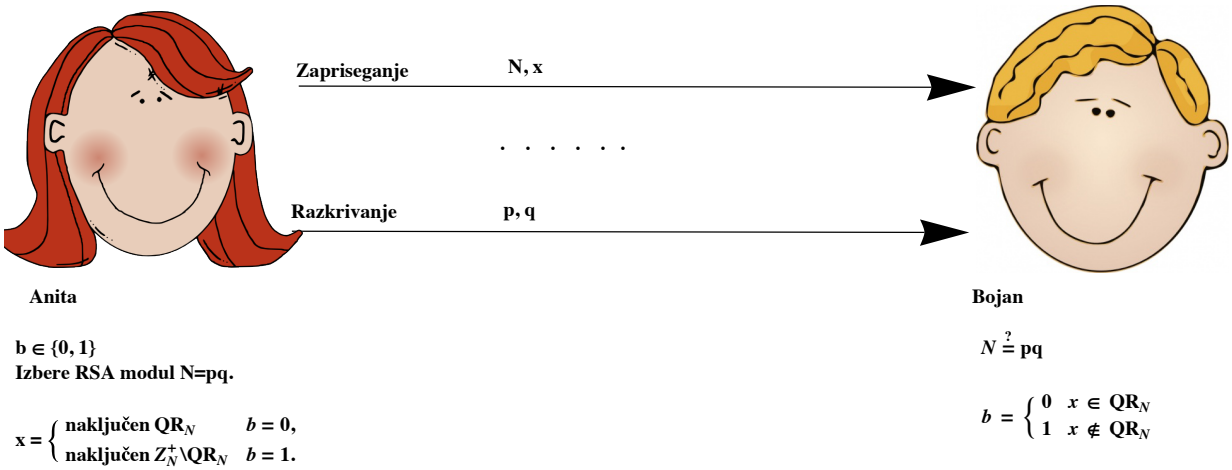
- težavnosti kvadratnih ostankov (angl. QRA – *quadratic residuosity assumption*); slika 3,
- težavnosti problema diskretnega logaritma; slika 4,
- generatorju psevdonaključnih števil [15]; slika 5.

Oglejmo si podrobneje varnost navedenih shem za metanje kovanca.

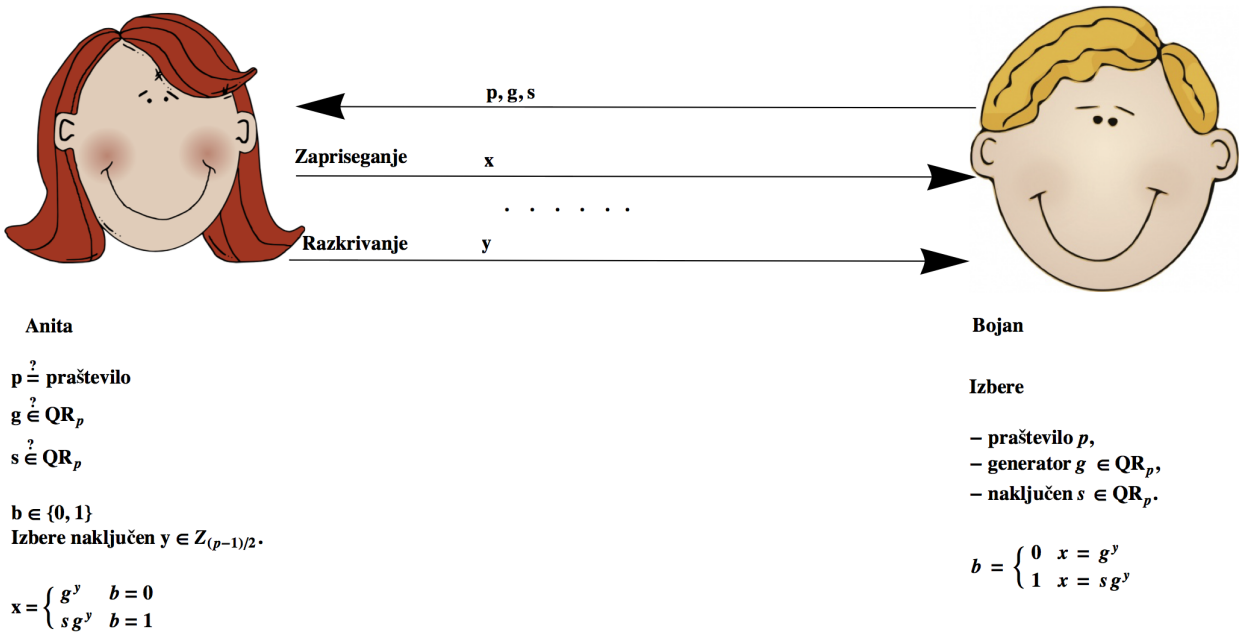
Shema na sliki 3 je brezpogojno zavezujoča in ima lastnost računskega skrivanja. Recimo, da Anita pošlje Bojanu (N, x) in poglejmo, katere vrednosti lahko Bojan razkrije. Možnosti so: (i) če N ni RSA modul, bo Bojan vedno zaznal napako; (ii) če $x \in \mathrm{QR}_N$, Bojan lahko odpre le bit $b = 0$; (iii) če $x \notin \mathrm{QR}_N$, Bojan lahko odpre le bit $b = 1$. Da bi Bojan vnaprej lahko razpoznal zapriseženo vrednost $b = 0$ ali $b = 1$, bi moral znati ločiti QR_N od drugih elementov $\mathbb{Z}_N$. Slednje je računsko težak problem za sestavljeni N.

Shema na sliki 4 je računsko zavezujoča in ima lastnost brezpogojnega skrivanja. Da bi Anita lahko odprla zapriseženo vrednost x na dva različna načina, bi morala poznati y in y', tako da obstaja x z lastnostjo $g^y = x = s \cdot g^{y'}$. To pomeni, da $g^{y-y'} = s$ in Anita bi morala poznati diskretni logaritem $\log_g s$. Predpostavljamo, da je računanje diskretnega logaritma v QR_p težko. Vrednost x je naključen element v QR_p, zato ima shema lastnost brezpogojnega skrivanja.

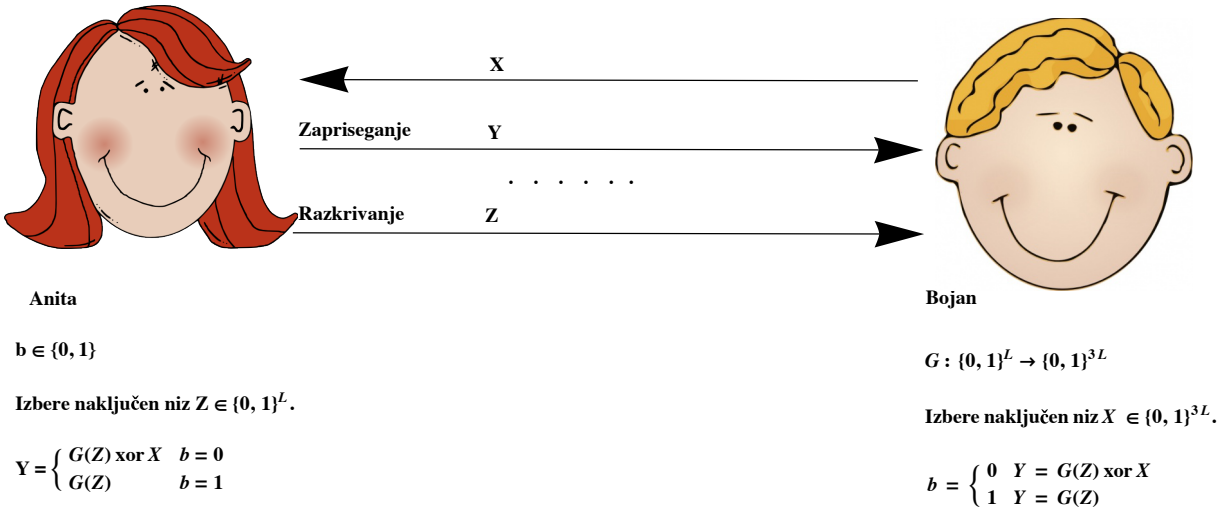
Shema na sliki 5 je brezpogojno zavezujoča in ima lastnost računskega skrivanja, če je G varen psevdonaključni generator. Da bi Anita lahko odprla zapriseženo vrednost Y na dva različna načina, bi morala najti Z in Z', tako da obstaja Y z lastnostjo $G(Z) \oplus X = Y = G(Z')$, to pa pomeni, da $G(Z) \oplus G(Z') = X$. Poglejmo, koliko elementov $X \in \{0,1\}^{3L}$ ima lastnost, da obstajata Z in Z', $Z \neq Z'$ ter velja $G(Z) \oplus G(Z') = X$. Takih X je največ $(2^L)^2 = 2^{2L}$. Torej je verjetnost, da ima naključno izbran $X \in \{0,1\}^{3L}$ to lastnost, največ 2^{-L}.



Slika 3: Konstrukcija sheme za varno metanje kovanca na osnovi QRA.



Slika 4: Konstrukcija sheme za varno metanje kovanca na osnovi DLP.



Slika 5: Konstrukcija sheme za varno metanje kovanca na osnovi PRG [15].

5.2 Ponudba in dražba

Oglejmo si sledeč primer med kupcem Bojanom in Anito, prodajalko. Bojan je zadovoljen, če kupi izdelek po ceni, nižji od prodajne cene izdelka b. Anita je zadovoljna, če proda izdelek po višji ceni od ponujene cene povpraševanja a. Naj bo $a \leq b$ in a in b sta tajni vrednosti, ki ju Anita (zadrži a) in Bojan (zadrži b) ne želita razkriti.

Recimo, da se Anita in Bojan dogovorita za pošten protokol, po katerem si izmenjata denar in izdelek po srednji ceni $p = \frac{a+b}{2}$. Po naivnem protokolu bi Anita Bojanu posredovala ceno a, nato bi Bojan poslal Aniti b in lahko bi izračunala povprečno ceno. Toda Bojan lahko zlorabi protokol in si zniža stroške nakupa, tako da posreduje ceno $b' = a$ in je $p' = a$. Podobno, če najprej Bojan posreduje ceno b Aniti, lahko Anita zlorabi sistem, da izbere $a' = b$ in potem $p' = b$. Rešitev je uporaba sheme za zapriseganje.

Sprva Anita zapriseže vrednost a in Bojanu pošlje zaprisego $c = c(a)$. Lastnost skrivanja zagotavlja, da Bojan iz c ne izve ničesar o a. Nato Bojan Aniti pove vrednost b in Anita odpre zaprisego c z $d = d(a)$. Sedaj oba izračunata $p = \frac{a+b}{2}$.

Scenarij se da posplošiti na bolj kompleksne situacije, kot so dražbe. Ideja je preprosta. Najprej neka stranka zapriseže vrednost (za katero ne želi, da jo izvedo ostali vpleteni pred odločevanjem), po zaprisegi stranka izve za vrednosti drugih vpletenih, čemur sledi razkritje zaprisege. Težava se lahko pojavi, če stranka zavrne razkritje zapriseženе vrednosti – sklepamo lahko, da je izzid za stranko neugoden in razveljavimo protokol.

5.3 Šifriranje z avtentikacijo in sproščena lastnost zaveze

S šifriranjem sporočila želimo doseči predvsem njegovo tajnost, z digitalnim podpisom pa avtentikacijo pošiljatelja oziroma integriteto sporočila. Pogosto želimo doseči oboje hkrati, tajnost sporočila in avtentikacijo pošiljatelja.

Oglejmo si primer uporabe sheme za zapriseganje $C = (\texttt{Setup}, \texttt{Commit}, \texttt{Open})$ za učinkovito in varno šifriranje z avtentikacijo. Predpostavimo, da imamo na voljo varno šifrirano shemo E in varen sistem za digitalno podpisovanje S. Z $E(m)$ označimo kriptogram sporočila m in z $S(m)$ podpis (m, σ).

Preprost postopek je zaporedna uporaba; $E(S(m))$ ali $S(E(m))$ v splošnem dosežeta oba naša cilja, t.j. šifriranje in avtentikacijo sporočila m. A s takim postopkom izvajamo dve morebitni zahtevni računski operaciji zaporedno, eno za drugo. Namesto tega lahko z uporabo shem za zapriseganje obe računski operaciji izvajamo vzporedno in dosežemo boljšo učinkovitost.

Lema 5.1. *Naj bo* $(c, d) \leftarrow \texttt{Commit}(m)$ *par za zapriseganje v shemi* C*. Definirajmo šifrirno shemo, za katero je kriptogram sporočila* m *podan z* $(c, E(d))$*. Ta šifrirna shema je varna natanko tedaj, ko ima shema za zapriseganje* C *lastnost skrivanja. Odšifriranje izvedemo, da sprva odšifriramo* d *in nato vrnemo* $\texttt{Open}(c, d)$*.*

Osnutek dokaza. ($\Leftarrow$) Shema C ima lastnost skrivanja, zato c ne razkriva nobene informacije o m. Enako velja za $E(d)$, saj je E varna šifrirna shema.

Podobno v smeri ($\Rightarrow$).

Lema 5.1 je sama po sebi neuporabna, saj lahko m šifriramo direktno z uporabo šifrirne sheme E, $E(m)$. Šele lema 5.1 skupaj z lemo 5.2 dosega želeni cilj.

Pred tem si oglejmo sheme za zapriseganje s **sproščeno lastnostjo zaveze**. To so sheme, v katerih je lastnost zaveze nadomeščena s sproščeno zavezo. Nasprotnik A

računsko težko določi sporočilo m za katerega velja: ob generiranju zaprisege $(c, d) \leftarrow \texttt{Commit}(m)$, nasprotnik A z nezanemarljivo verjetnostjo poišče d', da je (c, d') veljavna zaveza za neko drugo sporočilo $m', m' \neq m$. Torej, A ne more najti trčenja za naključno generiran $c(m)$, čeprav A sam izbere m.

Lema 5.2. *Naj bo $(c, d) \leftarrow \texttt{Commit}(m)$ par za zapriseganje v shemi C. Definirajmo sistem za digitalno podpisovanje za sporočilo m z $(S(c), d)$. Ta podpis je varen natanko tedaj, ko ima shema za zapriseganje C sproščeno lastnost zaveze. Novi podpis preverimo tako, da sprva preverimo podpis sporočila c in nato preverimo ali razkritje vrne veljavno sporočilo, $\texttt{Open}(c, d) \neq$ 'fail'.*

Osnutek dokaza. ($\Leftarrow$). Upoštevamo sproščeno lastnost zaveze C, v kateri (c, d) predstavlja par za zapriseganje in razkritje sporočila m. To pomeni, da je težko ponovno uporabiti veljaven podpis $(S(c), d)$ sporočila m za ponarejanje $(S(c), d')$ sporočila m', saj bi prišlo do trčenja (c, d, d'). Zato mora napadalec ponarediti nov podpis $S(c')$, a to ni mogoče, ker je S varen sistem za digitalno podpisovanje.

Podobno v smeri ($\Rightarrow$).

Podobno ko prejšnja lema, je tudi lema 5.2 samo po sebi neuporabna. Sporočilo m se lahko podpiše direktno ali se uporabi cenejši pristop z zgoščevalno funkcijo, če je ovira velikost podpisa. Lemi združuje naslednji izrek.

Izrek 5.3. *Naj bo $(c, d) \leftarrow \texttt{Commit}(m)$ par za zapriseganje v shemi C. Definirajmo shemo za šifriranje z avtentikacijo (angl.* signcryption*) za sporočilo m z $(S(c), E(d))$. Ta shema je varna natanko tedaj, ko je C je varna shema za zapriseganje s sproščeno zavezo. Šifriranje in preverjanje podpisa je opisano v lemah 5.1 in 5.2.*

Rezultat iz izreka 5.3 se lahko uporablja za vzporedno izvajanje operacij šifriranja in podpisovanja sporočil.

5.4 Dokazi brez razkritja znanja

Pri dokazovanju brez razkritja znanja običajno sodelujeta probabilistična algoritma dokazovalnik P in preverjevalnik V. Oba imata skupen podatek (objekt) x. Dokazovalnik P pozna določeno lastnost podatka x in želi preko javnega kanala prepričati preverjevalnik V, da ima podatek x omenjeno lastnost. Bolj natančno, pravimo, da je x DA-primer nekega odločitvenega problema. Pri tem preverjevalnik še vedno ne ve, kako bi sam dokazal, da ima x to lastnost.

Dokaz brez razkritja znanja sestavlja več krogov, ki so bodis preverjevalnikovi izzivi ali dokazovalnikovi odgovori. Informacije, ki jih dobi preverjevalnik in vsa sporočila, ki si jih zimenjata P in V, imenujemo **zapis** (sled). Če je dokazovalnik uspešen pri vseh izzvih, preverjevalnik na koncu protokola dokaz **sprejme**, sicer ga **zavrne**.

Preden definiramo dokaz brez razkritja znanja, si oglejmo sisteme za interaktivno dokazovanje.

Definicija 5.4. Sistem za interaktivni dokaz za odločitveni problem Π je protokol, ki izpolnjuje naslednja pogoja:

- **Uglašenost** (angl. *soundness*) – verjetnost, da dokazovalnik sprejme neveljavni dokaz (t.j. dokazovalnik sprejme dokaz ob negativnem odgovoru odločitvenega problema Π) je zanemarljiva.

- **Polnost** (angl. *completeness*) – dokazovalnik vedno sprejme veljavni dokaz (t.j. ob pozitivnem odgovoru odločitvenega problema).

Dokaz brez razkritja znanja je primer sistema za interaktivno dokazovanje.

Definicija 5.5. Naj bo dan sistem za interaktivni dokaz za odločitveni problem $\Pi \in \mathcal{NP}$ s pričo R_Π in algoritem S s polinomsko časovno zahtevnostjo, ki služi kot simulator v izmenjavi sporočil med dokazovalnikom in preverjevalnikom. Naj bo $T(x)$ množica možnih zapisov, ki jih dokazovalnik in preverjevalnik izmenjata v sistemu za interaktivni dokaza za DA-primer x. Naj bo $F(x)$ množica možnih ponarejenih zapisov simulatorja S Sistem za interaktivni dokaz je **dokaz brez razkritja znanja** (računski), če velja

- $T(x) = F(x)$ in
- za vsak zapis $t \in T(x)$ je verjetnost, da je t zapis interaktivnega dokaza, polinomsko neločljiva od verjetnosti, da je zapis t ponarejen zapis simulatorja.

Popolni dokaz brez razkritja znanja se od definicije 5.5 razlikuje le v zahtevi po enakosti obeh verjetnosti in ne le njuni polinomski neločljivosti (več v [3], [5] in [10]). Definicija 5.5 z drugimi besedami pove, da lahko preverjevalnik po izvedbi protokola stori kvečjemu toliko kot lahko simulator, potem ko je zgeneriral ponarejeni podpis.

Izrek 5.6. *Če obstajajo enosmerne funkcije, potem vsakemu jeziku v $\mathcal{NP}$ pripada dokaz brez razkritja znanja.*

Dokaz izreka 5.6 bomo zaradi krajšega protokola (trije krogi namesto štirih) pokazali na primeru enosmernih permutacij. Pomagali si bomo z $\mathcal{NP}$-polnim problemom pravilnega 3-barvanja grafa, ki ga v nadaljevanju označujemo z 3COLOR. To je odločitveni problem, ki vprašuje, ali obstaja pravilno 3-barvanje grafa.

Izrek 5.7. *Če obstajajo enosmerne permutacije, potem ima vsak jezik L v $\mathcal{NP}$ dokaz brez razkritja znanja.*

Dokaz. Dokaz poteka v dveh korakih. Najprej podamo (računski) dokaz brez razkritja znanja za problem 3COLOR. Nato reduciramo originalni jezik L na 3COLOR z uporabo Cook-ove redukcije. Slednja zagotavlja, da lahko ob prevedbi vprašanja $x \in L$ na $x' \in L_{3COLOR}$ prevedemo tudi pričo w na pričo $w' \in R_{3COLOR}(x)$ in poženemo dokaz brez razkritja znanja za 3COLOR z vhodom x', w'.

3COLOR je jezik, sestavljen iz 3-obarvljivih grafov s standardnim postopkom barvanja grafov. Naj bodo dani graf $G = (V, E)$ z $n = |V|$ vozlišči, barve $c_i \in \{0, 1, 2\}$ in neka priča (t.j. dodelitev barv vozliščem grafa) $w = c_0, c_1, \ldots, c_n$.

Definirajmo dokaz brez razkritja znanja za 3COLOR. Naj bo $C = (\texttt{Setup}, \texttt{Commit}, \texttt{Open})$ shema za zapriseganje definirana z enosmernimi permutacijami kot v definiciji 4.4. Pri tem predpostavljamo obstoj enosmerih permutacij, v [1] je podan alternativni dokaz brez te predpostavke. Naj bo ϕ dano barvanje grafa G, t.j. funkcija $\phi : V(G) \to \{0, 1, 2\}$.

1. Dokazovalnik P enakomerno naključno izbere permutacijo π nad $\{0, 1, 2\}$ nekega fiksnega barvanja ϕ. Za $i = 1, 2, \ldots, n$ dokazovalnik P pošlje zaprisego $\texttt{Commit}(\pi(\phi(i)))$ preverjevalniku V.
2. Preverjevalnik V enakomerno izbere naključno povezavo $e \in E$ in izbiro pošlje dokazovalniku P.

3. Ob sprejemu $e = (i, j) \in E$, dokazovalnik P razkrije i-ti in j-ti vrednosti, zapriseženi v koraku 1.

4. Preverjevalnik V preveri, da sta razkriti vrednosti $\phi(i), \phi(j)$ različna elementa iz $\{1, 2, 3\}$ in da se ujemata z zaprisegami iz koraka 1.

Polnost. Če $G \in$ 3COLOR in je ϕ veljavno barvanje, potem bo očitno dokazovalnik P vedno razkril ustrezni vrednosti za $\phi(i)$ in $\phi(j)$ in bo preverjevalnik V sprejel dokaz.

Uglašenost. Če $G \notin$ 3COLOR, potem ϕ ni veljavno 3-barvanje grafa G. Torej, obstajati mora vsaj ena povezava $e = (i, j) \in E$, da $\phi(i) = \phi(j)$. Preverjevalnik V izbere povezavo v koraku 2 enakomerno naključno, zato je verjetnost, da izbere nepravilno povezavo vsaj $\frac{1}{|E|}$. Če torej V izbere nepravilno povezavo, dokazovalnik P zagotovo ne bo razkril vrednosti, ki bi jih V preveril in sprejel kot ustrezni.

Nerazkrivanje znanja. Z danim (morebitno goljufivim) preverjevalnikom V lahko zgradimo simulator S_V na sledeč način. Naj bo dan vhod (x, y), kjer je x kodirana predstavitev grafa G. Simulator S_V naključno določi barve vozliščem grafa G in zapriseže izbire barv. Nato simulator S_V simulira algoritem preverjevalnika V in izbere povezavo $e \in E$. Če sta vozlišči izbrane povezave e različno pobarvani, potem S_V razkrije barvi in izpiše rezultat. Če sta vozlišči enako pobarvani, S_V zavrže izbiro in poskusi ponovno. Verjetnost, da sta vozlišči povezave e enako pobarvani, je $\frac{1}{3}$, zato je pričakovano število poskusov do prve uspešne izbire povezave e enako 3. Vsaka ponovitev poskusa zahteva polinomski čas in potrebujemo pričakovano konstantno število ponovitev, zato simualtor S_V teče v pričakovano polinomskem času. Dodelitev barv v simulatorju S_V in permutacija barvanja v dokazovalniku P so izbrane enakomerno naključno, zato je verjetnost, da gre za zapis iz dokaza brez razkritja znanja polinomsko neločljiva od verjetnosti da gre za ponarejen zapis simulatorja.

Dani protokol ima nezanemarljivo napako v uglašenosti. Težavo lahko omilimo z večkratno zaporedno ponovitvijo protokola za dokaz brez razkritja znanja. Če opravimo $n \cdot |E|$ ponovitev protokola, napako v uglašenosti zmanjšamo na $(1 - \frac{1}{|E|})^{n \cdot |E|} \approx e^{-n}$. Dokaz bi lahko ponovili vzporedno, a žal se lastnost nerazkrivanja znanja ne ohranja pri vzporednem sestavljanju. □

Sheme za zapriseganje so primitiv za učinkovito sestavljanje dokazov brez razkritja znanja, kar je pokazano v [7].

6 Odprti problemi

Omenimo še nekaj odprtih vprašanj s področja shem za zapriseganje in njihove navezanosti na dokaze brez razkritja znanja. Precej problemov se nanaša na obstoj enosmernih funkcij, saj je očitno da mora biti funkcija `Commit` enosmerna, če naj bi bila zagotovljena varnost sheme za zapriseganje.

Nedavno je bila predlagana shema z dvema krogoma, ki zagotavlja popolno skrivanje in računsko zavezo pod predpostavko obstoja enosmernih permutacij [16]. V [13] je pokazan obstoj shem za zapriseganje le na osnovi enosmernih funkcij in drugih primitivov, kot so generatorji naključnih števil. Če predpostavimo obstoj enosmernih funkcij, potem obstajajo sheme za zapriseganje z brezpogojno zavezo in računskim skrivanjem (dokaz v [13]). Če obstajajo ena-na-ena surjektivne enosmerne funkcije, potem obstajajo sheme

za zapriseganje z popolnim skrivanjem in računsko zavezo (dokaz v [14]). Toda še vedno ni znano, ali obstoj enosmernih funkcij implicira obstoj shem za zapriseganje z brezpogojnim skrivanjem.

Literatura

[1] M. Ben-Or et. al: Multi-prover interactive proofs: how to remove intractability assumptions. V *Proceedings of the twentieth annual ACM symposium on Theory of computing*, 113–131, 1988.

[2] M. Blum: Coin flipping by telephone a protocol for solving impossible problems. *SIGACT News* 15, 1 23–27.

[3] M. Blum: How to Prove a Theorem So No One Else Can Claim It. V *Proceedings of the International Congress of Mathematicians*, Berkeley, CA, 1986, 1444–1451.

[4] G. Brassard, C. Crepeau: Quantum Bit Commitment and Coin Tossing Protocols. V *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, 49–61, 1991.

[5] I. Damgård: Commitment Schemes and Zero-Knowledge Protocols. V *1Proceeding Lectures on Data Security, Modern Cryptology in Theory and Practice*, 1998.

[6] I. Damgård, E. Fujisaki: A Statistically-Hiding Integer Commitment Scheme based on Groups with Hidden Order. V *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, 2002.

[7] I. Damgård, J. B. Nielson: Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. V *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, 581–596, 2002.

[8] M. Fischlin, R. Fischlin: Efficient Non-Malleable Commitment Schemes, Advances in Cryptology — Crypto 2000. *Lecture Notes in Computer Science*, vol. 1880, 414–432, Springer-Verlag, 2000.

[9] J. T. Gill, III. Computational complexity of probabilistic Turing machines. V *Proceedings of the sixth annual ACM symposium on Theory of computing*, 91–95, 1974.

[10] O. Goldreich, S. Micali, A. Widgerson: How to Prove All NP Statements in Zero-Knowledge and a Methodology of Cryptographic Protocol Design. *Advances in Cryptology - CRYPTO 86 Proceedings*, Springer-Verlag, 171–185, 1987.

[11] Y. Ishai, E. Kushilevitz, R. Ostrovsky: Sufficient conditions for collision-resistant hashing. V *Proceedings of the Second international conference on Theory of Cryptography (TCC 2005)*, 445–456, 2005.

[12] A. Juels, M. Wattenberg: A Fuzzy Commitment Scheme. V *Proceedings of the 6th ACM conference on Computer and communications security*, 28–36, 1999.

[13] M. Naor: Bit Commitment Using Pseudo-Randomness. V *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '89)*, Gilles Brassard (Ed.). Springer-Verlag, London, UK, 128–136.

[14] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung: Perfect Zero-Knowledge Arguments for NP Using Any One-Way Permutation. *Technical Report. Weizmann Science Press of Israel*, Israel, 1996.

[15] M. Naor. Bit Commitment Using Pseudorandom Generators. *Cryptology* 4(2): 151–158, 1991.

[16] C. Tang, D. Pei, Z. Liu, Z. Yao, M. Wang: Perfectly Hiding Commitment Scheme with Two-Round from Any One-Way Permutation. *IACR Cryptology* [ePrint Archive] 2008: 34.