Anonimnost in nevidnost podpisov brez možnosti zanikanja

Bor Brecelj

14. september 2021

Povzetek. Podpisi brez možnosti zanikanja se razlikujejo od navadnih digitalnih podpisov v tem, da mora pri preverjanju veljavnosti podpisa sodelovati tudi podpisnik. Ta lastnost da podpisniku moč, da se odloči, kdo lahko preveri njegov podpis in kdo ne. Kasneje so se razvile še razširitve takih podpisov, ki omogočajo pretvorbo v navadne digitalne podpise, ali pa, da lahko podpisnik dodeli pravico potrjevati podpise neki tretji osebi. Predstavili bomo podpise brez možnosti zanikanja in potrebne varnostne zahteve s poudarkom na anonimnosti in nevidnosti, za kateri velja, da sta pod določenimi pogoji ekvivalentni. Nato bomo opisali eno shemo, ki temelji na bilinearnih parjenjih. Zanimivo je, da ta shema zadošča le nevidnosti, ne pa tudi anonimnosti, kar v svojem članku analizirajo Loh, Heng, Tan in Kurosawa.

Ključne besede: podpisi brez možnosti zanikanja, varnost, bilinearno parjenje

1 UVOD

Predstavljajmo si podjetje, ki ustvarja programsko opremo. Ob izdaji aplikacije jo digitalno podpišejo, da se lahko uporabniki prepričajo, da ni bila spremenjena, preden je prišla do njih. Če želi podjetje aplikacijo prodajati, potem omogoči preverjanje podpisa le plačnikom. Sheme za podpis, ki to omogočajo, se imenujejo *podpisi brez možnosti zanikanja* (angl. undeniable signatures). Pri preverjanju podpisa mora sodelovati tudi podpisnik, ki se lahko odloči, ali bo preverjanje dovolil. Preverjanje veljavnosti podpisa je tako interaktiven protokol med podpisnikom Primožem in preverjevalko Vero. Ločimo *potrditveni protokol* (angl. confirmation protocol), kjer Primož dokazuje, da je podpis veljaven in *zavrnitveni protokol* (angl. disavowal protocol), kjer dokazuje, da podpis ni veljaven.

Drugi primer uporabe podpisov brez možnosti zanikanja so tajni dokumenti. Želimo, da lahko le tisti, ki imajo dostop, preverijo njihovo veljavnost. Če vemo, da bo dokument postal javen čez čas, lahko uporabimo razširitev podpisov brez možnosti zanikanja, ki omogoča prevedbo podpisa v normalen digitalen podpis, ki ga lahko kdorkoli preveri.

Leta 1989 sta Chaum in Van Antwerpen [3] predstavila prvo shemo za podpise brez možnosti zanikanja. Kasneje je bilo predlaganih veliko shem in razširitev. Razširitev, ki dovoljuje pretvorbo v navaden digitalen podpis, so predstavili Boyar, Chaum, Damgård in Pedersen [2]. Varnostno zahtevo nevidnosti so vpeljali Chaum, van Heijst in Pfitzmann [4]. Kasneje sta Galbraith in Mao [5] vpeljala še varnostno zahtevo anonimnosti in dokazala, da sta nevidnost in anonimnost pod določenimi pogoji ekvivalentni.

V naslednjem poglavju predstavimo bilinearna parjenja in predpostavke povezane z njimi ter programabilne zgoščevalne funkcije. Nato, v tretjem poglavju definiramo podpise brez možnosti zanikanja in varnostne zahteve zanje. Primer sheme za podpise brez možnosti zanikanja se nahaja v četrtem poglavju. V petem poglavju dokažemo njeno varnost.

2 OSNOVE

Shema, ki jo bomo predstavili v kasnejših poglavjih, temelji na bilinearnih parjenjih.

V tem poglavju predstavimo osnove, ki so potrebne za razumevanje sheme in njene varnostne analize. Najprej definiramo bilinearno parjenje ter opišemo predpostavke, na katerih temelji varnost sheme. Nato predstavimo še programabilne zgoščevalne funkcije, ki so pomembne v dokazih varnosti.

2.1 Bilinearno parjenje

Shema, ki jo bomo predstavili, je osnovana na bilinearnih parjenjih. Recimo, da imamo grupi G in G_T, obe reda p, kjer je p neko praštevilo. Naj bo g generator grupe G. Preslikava $e : G \times G \to G_T$ je *bilinearno parjenje* (angl. bilinear pairing), če veljajo naslednje lastnosti:

- *bilinearnost* (angl. bilinearity): za vsaka elementa $x, y \in G$ in vsaki dve števili $a, b \in \mathbb{Z}_p$ velja enakost $e\left(x^a, y^b\right) = e\left(x, y\right)^{ab}$,
- *nedegeneriranost* (angl. non-degeneracy): če je g generator grupe G, potem je $e(g, g)$ generator grupe G_T (iz tega sledi, da velja neenakost $e(g, g) \neq 1$),
- *izračunljivost* (angl. computability): obstaja učinkovit algoritem za izračun vrednosti $e(x, y)$ za

vsaka elementa $x, y \in G$.

Primer takega parjenja je Weilovo parjenje na eliptičnih krivuljah, ki sta ga uporabila Boneh in Franklin [1] za kriptosistem na osnovi identitete.

Spoznati moramo še nekaj predpostavk, na katerih temelji varnost sheme, ki jo bomo predstavili. Prva je *krepka Diffie-Hellmanova predpostavka*, oznaka q-SDH (angl. strong Diffie-Hellman assumption). Rečemo, da predpostavka q-SDH drži v grupi G, če ne obstaja verjetnostni polinomski algoritem $\mathcal{A}$, ki kot vhod prejme elemente

$$\left(g, g^x, g^{x^2}, ..., g^{x^q}\right) \in G^{q+1},$$

kjer je $x \in \mathbb{Z}_p$ neznano število, in vrne par

$$\left(g^{\frac{1}{x+s}}, s\right) \in G \times \mathbb{Z}_p$$

z nezanemarljivo verjetnostjo.

Druga predpostavka, ki jo bomo potrebovali, je *skrita krepka Diffie-Hellmanova predpostavka*, oznaka q-HSDH (angl. hidden strong Diffie-Hellman assumption), ki drži, če ne obstaja verjetnostni polinomski algoritem, ki kot vhod prejme elemente

$$\left(g, g^x, g^\beta, \left\{g^{\frac{1}{x+s_i}}, g^{s_i}, g^{\beta s_i}\right\}_{i=1}^{q}\right),$$

kjer je $s_i \in \mathbb{Z}_p$ $(1 \leq i \leq q)$, in vrne

$$\left(g^{\frac{1}{x+s}}, g^s, g^{\beta s}\right)$$

z nezanemarljivo verjetnostjo, kjer je $s \in \mathbb{Z}_p$ in velja $s \notin \{s_1, ..., s_q\}$.

Še zadnja je *odločitvena skrita krepka Diffie-Hellmanova predpostavka* z oznako q-DHSDH (angl. decisional hidden strong Diffie-Hellman assumption), ki velja natanko tedaj, ko ne obstaja algoritem, ki bi z verjetnostjo večjo od $\frac{1}{2}$ ločil element $g^{\frac{1}{x+s}}$ od naključno izbranega elementa $Z \in G$. Natančneje, algoritem kot vhod dobi elemente

$$\left(g, g^x, g^\beta, \left\{g^{\frac{1}{x+s_i}}, g^{s_i}, g^{\beta s_i}\right\}_{i=1}^{q}, g^{\beta s}\right)$$

in enega izmed $g^{\frac{1}{x+s}}$ ali Z ter vrne 0, če misli, da je dobil element $g^{\frac{1}{x+s}}$ in 1 sicer.

2.2 Programabilne zgoščevalne funkcije

V shemi, ki jo bomo predstavili, nastopajo *programabilne zgoščevalne funkcije* (angl. programmable hash functions), ki jih bomo krajše označili s PHF. Vpeljala sta jih Hofheinz in Kiltz [6]. Programabilne so v smislu, da lahko priredimo rezultat funkcije tako, da vsebuje znan ali neznan diskretni logaritem z določeno verjetnostjo. Uporabne so za dokaze varnosti, kjer jih bomo uporabili tudi mi.

Programabilno zgoščevalno funkcijo sestavljajo štirje verjetnostni polinomski algoritmi:

- algoritem za generiranje ključa $\kappa \leftarrow \texttt{PHF.Gen}(1^k)$,
- algoritem za izračun zgoščevalne funkcije H, ki za sporočilo M dolžine ℓ vrne vrednost $\texttt{PHF.Eval}(\kappa, M)$, kar na kratko zapišemo kot $H_\kappa(M) = \texttt{PHF.Eval}(\kappa, M)$,
- algoritem za generiranje ključa s stranskimi vrati $(\kappa', \tau) \leftarrow \texttt{PHF.TrapGen}(1^k, g, h)$, kjer sta g in h generatorja grupe G in τ stranska vrata,
- algoritem za izračun zgoščevalne funkcije s stranskimi vrati $(a_M, b_M) \leftarrow \texttt{PHF.TrapEval}(\tau, M)$, kjer velja
$$H_{\kappa'}(M) = g^{a_M} h^{b_M}$$
za vsaka generatorja g, h, vsak ključ κ' in vsako sporočilo $M \in \{0,1\}^\ell$.

Veljati mora še, da sta porazdelitvi ključev κ in κ' enaki.

Rečemo, da je funkcija (m, n)-*programabilna zgoščevalna funkcija*, če za vsake

$$X_1, X_2, ..., X_m, Z_1, Z_2, ..., Z_n \in \{0,1\}^\ell,$$

kjer $X_i \neq Z_j$ za vse i, j, velja

$$a_{X_1} = a_{X_2} = ... = a_{X_m} = 0 \quad \text{in}$$

$$a_{Z_1}, a_{Z_2}, ..., a_{Z_n} \neq 0$$

z dovolj veliko verjetnostjo. Vrednosti a_{X_i} in a_{Z_j} smo dobili s klici funkcije

$$(a_{X_i}, b_{X_i}) \leftarrow \texttt{PHF.TrapEval}(\tau, X_i) \quad \text{in}$$

$$(a_{Z_j}, b_{Z_j}) \leftarrow \texttt{PHF.TrapEval}(\tau, Z_j).$$

3 PODPISI BREZ MOŽNOSTI ZANIKANJA

Shema za podpis brez možnosti zanikanja (angl. undeniable signature scheme) je definirana kot četverica $(\texttt{KeyGen}, \texttt{Sign}, \texttt{Confirm}, \texttt{Disavow})$, kjer je

- `KeyGen` algoritem, ki zgenerira javni in zasebni ključ $(\texttt{pk}, \texttt{sk})$,
- `Sign` algoritem, ki kot vhod prejme sporočilo M in zasebni ključ `sk` ter vrne podpis σ sporočila M,
- `Confirm` protokol med podpisnikom Primožem in preverjevalko Vero, kjer Primož dokaže Veri, da je podpis veljaven, pri čemer privzamemo, da tako Primož kot Vera poznata Primožev javni ključ `pk`, podpis σ in pripadajoče sporočilo M: Primož preveri, da je podpis veljaven in če je, to dokaže Veri, sicer se protokol konča že prej,
- `Disavow` protokol med Primožem in Vero, kjer Primož dokaže, da podpis, ki ni njegov, tudi ni veljaven.

Spomnimo se primera podjetja, ki prodaja programsko opremo. Recimo, da podjetje želi izdati svojo programsko opremo brezplačno. Ali lahko podpis brez možnosti zanikanja pretvori v navaden digitalni podpis? Odgovor je pritrdilen le, če shema to dovoljuje. Taki razširitvi rečemo *pretvorljivost* (angl. convertibility). To

je možnost, da kadarkoli kasneje Primož pretvori podpis v normalen digitalen podpis. Tukaj imamo dve možnosti:

- *pretvorba vseh podpisov* (angl. universal convertibility): pri tej razširitvi lahko Primož razkrije skrivnost, ki omogoča, da lahko kdorkoli sam preveri vse njegove podpise, pri tem pa še vedno ne more nihče ponarediti njegovega podpisa,
- *pretvorba izbranega podpisa* (angl. selective convertibility): za poljuben svoj podpis lahko Primož razkrije skrivnost, ki omogoča, da ga lahko kdorkoli sam preveri, pri tem pa nihče ne more ponarediti podpisa ali sam preveriti veljavnosti drugih podpisov.

Kaj pa, če je podjetje tako veliko, da ne more vsem dokazati veljavnosti podpisa? V tem primeru eno od rešitev imenujemo *podpis z izbranim potrjevalcem* (angl. designated confirmer signature). Tukaj lahko veljavnost podpisa dokaže poleg podpisnika tudi tretja oseba, ki jo je vnaprej izbral podpisnik. Ta tretja oseba pa ne more podpisovati v njegovem imenu.

3.1 Varnostne zahteve

Pri navadnih digitalnih podpisih je potrebna le *neponaredljivost* (angl. unforgeability). To pomeni, da napadalec ne more ponarediti podpisa z nezanemarljivo verjetnostjo. Definiramo jo z igro med izzivalcem in nasprotnikom.

1) izzivalec zgenerira javni in zasebni ključ $(\mathtt{pk}, \mathtt{sk})$ ter nasprotniku pošlje javni ključ,
2) nasprotnik lahko opravi poljubno število poizvedb:
   - podpisna poizvedba, kjer nasprotnik izbere sporočilo M, katerega podpis σ dobi kot odgovor,
   - poizvedba za potrditveni ali zavrnitveni protokol, kjer nasprotnik izbere sporočilo M in podpis σ ter skupaj z izzivalcem izvedeta potrditveni ali zavrnitveni protokol,
3) na koncu nasprotnik vrne par (M^*, σ^*) in zmaga, če je σ^* veljaven podpis sporočila M^*, pri tem pa prej ni izvedel podpisne poizvedbe na sporočilu M^*.

Prednost (angl. advantage) nasprotnika v tej igri je definirana kot verjetnost, da zmaga. Za shemo rečemo, da je *neponaredljiva*, če ne obstaja nasprotnik z nezanemarljivo prednostjo v opisani igri.

Pri podpisih brez možnosti zanikanja mora poleg neponaredljivosti veljati še nekaj drugih lastnosti. Zelo pomembno je, da sta potrditveni in zavrnitveni protokol *brez razkritja znanja* (angl. zero-knowledge). Kaj to pomeni, si poglejmo na potrditvenem protokolu. Spomnimo, da s tem protokolom Primož dokaže, da je podpis veljaven. Da je protokol brez razkritja znanja morajo veljati naslednji pogoji:

- *polnost* (angl. completeness), kar pomeni, da če je podpis res veljaven, lahko Primož to vedno dokaže,
- *uglašenost* (angl. soundness), ki pravi, da če podpis ni veljaven, Primož ne more dokazati, da je veljaven,
- *brez razkritja znanja* (angl. zero-knowledge), ki pravi, da med izvajanjem tega protokola, Primož ne izda nobene druge informacije razen veljavnosti podpisa.

Iz teh treh pogojev sledi, da s protokolom, ki je brez razkritja znanja, Primož dokaže veljavnost podpisa le Veri. Tudi, če kdo vidi celotno komunikacijo med Vero in Primožem, se s tem ne more prepričati, da je podpis res veljaven.

Naslednja zahteva je *nevidnost* (angl. invisibility), ki pravi, da napadalec ne more določiti ali je podpis veljaven oziroma ni brez sodelovanja podpisnika. Natančneje, je nevidnost definirana z igro med izzivalcem in nasprotnikom.

1) izzivalec zgenerira javni in zasebni ključ $(\mathtt{pk}, \mathtt{sk})$ ter nasprotniku pošlje javni ključ,
2) nasprotnik lahko opravi poljubno število podpisnih poizvedb in poizvedb za potrditveni ali zavrnitveni protokol,
3) nasprotnik si izbere sporočilo M^* in ga pošlje izzivalcu, ki nato naključno izbere bit $b \in \{0, 1\}$: če je $b = 0$, izzivalec vrne veljaven podpis σ^* sporočila M^*, sicer naključno izbere podpis σ^* iz množice vseh možnih podpisov in ga vrne,
4) nasprotnik lahko naprej opravlja enake poizvedbe kot v 2. koraku, vendar ne sme klicati poizvedb na paru (M^*, σ^*).
5) Na koncu nasprotnik vrne bit b' in zmaga, če velja $b = b'$.

Prednost nasprotnika je definirana kot vrednost

$$|\Pr[b = b'] - \frac{1}{2}|.$$

Shema je *nevidna*, če ne obstaja nasprotnik s polinomsko časovno zahtevnostjo, ki bi imel nezanemarljivo prednost v zgoraj opisani igri.

Zadnja varnostna zahteva pa je anonimnost, ki je pomembna, ko opazujemo več uporabnikov hkrati. Ta pravi, da napadalec ne more samo iz podpisa določiti, komu ta podpis pripada. Definirana je s podobno igro kot nevidnost, le, da tukaj na začetku igre izzivalec zgenerira dva javna in zasebna ključa $(\mathtt{pk}_0, \mathtt{sk}_0)$ in $(\mathtt{pk}_1, \mathtt{sk}_1)$. Druga razlika pa je ta, da ko izzivalec dobi sporočilo M^* od nasprotnika in naključno izbere bit $b \in \{0, 1\}$, vrne veljaven podpis σ^* sporočila M^* podpisanega z zasebnim ključem $\mathtt{sk}_b$. Tako mora nasprotnik ugibati komu podpis pripada. Rečemo, da je shema *anonimna* (angl. anonymous), če ne obstaja nasprotnik s polinomsko časovno zahtevnostjo, ki bi imel nezanemarljivo prednost v opisani igri.

Leta 2003 sta Galbraith in Mao [5] dokazala, da sta lastnosti nevidnost in anonimnost ekvivalentni, če

in samo če imajo vsi uporabniki enake množice vseh možnih podpisov.

4 Shema Huanga in Wonga [7]

Za vpeljavo te sheme moramo definirati algoritem za generiranje ključa, algoritem za podpisovanje in potrditveni ter zavrnitveni protokol:

- algoritem za generiranje ključa `KeyGen`:
  1) inicializiraj PHF s klicem funkcije
  $$\kappa \leftarrow \texttt{PHF.Gen}(1^k),$$
  2) naključno izberi števili $x, y \in \mathbb{Z}_p^*$ in element $u \in G$,
  3) izračunaj $X = g^x$ in $Y = g^{\frac{1}{y}}$,
  4) zasebni ključ je $\texttt{sk} = (x, y)$ ter javni ključ je $\texttt{pk} = (g, X, Y, u, \kappa)$,
- algoritem za podpisovanje `Sign`, ki kot vhod prejme sporočilo M in podpisnikov zasebni ključ `sk`:
  1) naključno izberi število $s \in \mathbb{Z}_p^*$,
  2) izračunaj
  $$\delta = H_\kappa(M)^{\frac{1}{x+s}}, \quad \gamma = Y^s, \quad \theta = u^s,$$
  3) podpis sporočila M je $\sigma = (\delta, \gamma, \theta)$,
- potrditveni protokol `Confirm` in zavrnitveni protokol `Disavow`, kjer tako Vera kot Primož dobita kot vhod sporočilo M in podpis $\sigma = (\delta, \gamma, \theta)$:
  1) oba sodelujoča preverita, da podpis σ pripada množici vseh možnih podpisov
  $$S \in \{(\delta, \gamma, \theta) \in G^3 \,:\, e(Y, \theta) = e(\gamma, u)\},$$
  2) Primož preveri, ali je podpis veljaven: če velja enakost
  $$e(H_\kappa(M), g) = e(\delta, X \cdot \gamma^y),$$
  Primož začne s potrditvenim protokolom, sicer začne z zavrnitvenim protokolom,
  3) v primeru potrditvenega protokola, mora Primož brez razkritja znanja dokazati enakost dveh diskretnih logaritmov:
  $$\log_Y(g) = \log_{e(\delta,\gamma)}(\lambda),$$
  kjer je $\lambda = e(H_\kappa(M), g) \cdot e(\delta, X)^{-1}$, v primeru zavrnitvenega protokola pa mora dokazati razlikovanje omenjenih logaritmov
  $$\log_Y(g), \ \log_{e(\delta,\gamma)}(\lambda).$$

Za potrditveni in zavrnitveni protokol se moramo prepričati, da res delujeta. V prvem koraku oba sodelujoča preverita, da velja enakost $e(Y, \theta) = e(\gamma, u)$. Če je podpis veljaven, enakost drži, ker

$$e(Y, \theta) = e(Y, u)^s = e(\gamma, u).$$

V drugem koraku Primož preveri veljavnost podpisa. Ponovno, enakost velja natanko tedaj, ko je podpis veljaven:

$$\begin{aligned} e(\delta, X \cdot \gamma^y) &= e\left(H_\kappa(M)^{\frac{1}{x+s}}, g^x \cdot g^{\frac{sy}{y}}\right) \\ &= e(H_\kappa(M), g)^{\frac{x+s}{x+s}} \\ &= e(H_\kappa(M), g). \end{aligned}$$

Zdaj poglejmo še enakost logaritmov iz tretjega koraka. Za levo stran lahko hitro ugotovimo, da je enaka y, ker velja $g = Y^y$. Za desno stran bomo najprej preoblikovali argument logaritma

$$\begin{aligned} &e(H_\kappa(M), g) \cdot e(\delta, X)^{-1} = \\ &= e(H_\kappa(M), g) \cdot e(H_\kappa(M), g)^{-\frac{x}{x+s}} \\ &= e(H_\kappa(M), g)^{1-\frac{x}{x+s}} \\ &= e(H_\kappa(M), g)^{\frac{s}{x+s}}. \end{aligned}$$

Preoblikujmo še osnovo logaritma

$$e(\delta, \gamma) = e(H_\kappa(M), g)^{\frac{s}{y(x+s)}}.$$

Iz tega lahko opazimo, da je rezultat desne strani tudi y, če je podpis veljaven.

4.1 Razširitev

Predstavljena shema podpira obe možnosti pretvorbe v normalen digitalen podpis. Pri pretvorbi vseh podpisov Primož objavi vrednost y. Tako lahko kdorkoli preveri veljavnost podpisa $\sigma = (\delta, \gamma, \theta)$ sporočila M z enačbo

$$e(H_\kappa(M), g) = e(\delta, X \cdot \gamma^y),$$

ne more pa podpisovati, saj bi potreboval vrednost x.

Pri pretvorbi izbranega podpisa Primož objavi vrednost $\nu = \gamma^y$. Za preverjanje podpisa uporabimo enačbo

$$e(H_\kappa(M), g) = e(\delta, X \cdot \nu).$$

5 Varnostna analiza sheme

Dokažimo najprej neponaredljivost predstavljene sheme, kjer je dokaz povzet po [7].

Trditev 5.1 (neponaredljivost): Naj bo H $(m, 1)$-PHF in naj bo $\mathcal{F}$ nasprotnik iz igre za definicijo neponaredljivosti z nezanemarljivo prednostjo. Potem obstaja verjetnostni polinomski algoritem $\mathcal{A}_1$, ki je v protislovju s predpostavko q-SDH ali pa obstaja verjetnostni polinomski algoritem $\mathcal{A}_2$, ki je v protislovju s predpostavko q-HSDH.

Z M_i bomo označili i-to podpisno poizvedbo nasprotnika $\mathcal{F}$. Odgovor te poizvedbe je podpis

$$\sigma_i = (\delta_i, \gamma_i, \theta_i),$$

ki je zgeneriran z vrednostjo s_i (torej velja $\gamma_i = Y^{s_i}$ in $\theta_i = u^{s_i}$). Z (M, σ) bomo označili nasprotnikov poizkus ponarejenega podpisa, kjer je

$$\sigma = (\delta, \gamma, \theta) = (\delta, Y^s, u^s).$$

Ločili bomo dva tipa nasprotnika $\mathcal{F}$:

1) $\exists i, 1 \leq i \leq q, \gamma = \gamma_i$, iz česar sledi, da velja tudi $\theta = \theta_i$ in $s = s_i$. Takega nasprotnika bomo označili s $\mathcal{F}_1$.
2) $\forall i, 1 \leq i \leq q, \gamma \neq \gamma_i$, kar pomeni, da

$$s \notin \{s_1, ..., s_q\}.$$

Takega nasprotnika bomo označili s $\mathcal{F}_2$.

Najprej se bomo lotili le prvega tipa nasprotnika.

Lema 5.2: Recimo, da obstaja nasprotnik $\mathcal{F}_1$, ki s q podpisnimi poizvedbami ponaredi podpis z nezanemarljivo verjetnostjo. Potem obstaja verjetnostni polinomski algoritem $\mathcal{A}_1$, ki je v protislovju s predpostavko q-SDH.

Dokaz: Z drugimi besedami, želimo pokazati, da znamo hitro razbiti predpostavko q-SDH, če znamo ponarediti podpis. To bomo dokazali tako, da bomo sestavili algoritem $\mathcal{A}_1$, ki bo uporabil nasprotnika $\mathcal{F}_1$. Spomnimo, da algoritem $\mathcal{A}_1$ kot vhod dobi elemente $(\tilde{g}, \tilde{g}^x, \tilde{g}^{x^2}, ..., \tilde{g}^{x^q})$ in vrne par $(\tilde{g}^{\frac{1}{x+s}}, s)$.

Najprej si naključno izberemo vse vrednosti $s_1, s_2, ..., s_q$. Izberemo si neko vrednost $j \in \{1, 2, ..., q\}$. Algoritem $\mathcal{A}_1$ bo vrnil pravo vrednost le, če bo veljala enakost $s_j = s$. Zdaj moramo iz vhodnih podatkov inicializirati vse parametre sheme, da bomo lahko uporabili nasprotnika $\mathcal{F}_1$. Ker bi radi postopek malo spremenili, ne bomo uporabili algoritma `KeyGen`. Namesto normalne verzije bomo uporabili verzijo PHF s stranskimi vrati, kar pomeni, da moramo izbrati elementa $g, h \in G$.

Vpeljimo oznaki $S := \cup_{i=1}^{q} \{s_i\}$ in $S_j := S \setminus \{s_j\}$ in definiramo polinoma

$$p_j(x) = \Pi_{t \in S_j} (x + t) \quad \text{in} \quad p(x) = \Pi_{t \in S} (x + t).$$

Zdaj lahko definiramo vrednosti

$$g := \tilde{g}^{p_j(x)}, \quad h := \tilde{g}^{p(x)} \quad \text{in} \quad X := g^x = \tilde{g}^{x \cdot p_j(x)}.$$

Opozoriti moramo, da vrednosti x ne poznamo, ampak lahko vseeno izračunamo vrednosti g, h in X iz vhodnih podatkov $(\tilde{g}, \tilde{g}^x, \tilde{g}^{x^2}, ..., \tilde{g}^{x^q})$ algoritma $\mathcal{A}_1$. Zdaj lahko kličemo funkcijo

$$(\kappa, \tau) \leftarrow \texttt{PHF.TrapGen}(1^k, g, h),$$

s katero inicializiramo PHF. Naključno si izberemo še vrednosti $y \in \mathbb{Z}_p$ in $u \in G$ ter izračunamo $Y = g^{\frac{1}{y}}$. Dobili smo javni ključ (g, X, Y, u, κ) in zasebni ključ (x, y). Izmed teh vrednosti poznamo vse razen x.

Zdaj, ko imamo javni in zasebni ključ, lahko uporabimo ponarejevalca $\mathcal{F}_1$. Enostavno lahko odgovorimo vsaki poizvedbi za potrditveni in zavrnitveni protokol, saj poznamo vrednost y. Pri podpisni poizvedbi M_i brez problema izračunamo vrednosti γ_i in θ_i. Vrednost δ_i pa izračunamo po formuli:

$$\begin{aligned}
\delta_i &= H_\kappa(M_i)^{\frac{1}{x+s_i}} \\
&= \left(g^{a_{M_i}} h^{b_{M_i}}\right)^{\frac{1}{x+s_i}} \\
&= \left(\tilde{g}^{a_{M_i} \Pi_{t \in S_j}(x+t)} \tilde{g}^{b_{M_i} \Pi_{t \in S}(x+t)}\right)^{\frac{1}{x+s_i}} \\
&= \tilde{g}^{a_{M_i} \Pi_{t \in S^{i,j}}(x+t) + b_{M_i} \Pi_{t \in S^i}(x+t)},
\end{aligned}$$

kjer je $(a_{M_i}, b_{M_i}) \leftarrow \texttt{PHF.TrapEval}(\tau, M_i)$. Zgornjo formulo lahko uporabimo, če velja $s_i \neq s_j$. V primeru, da je $s_i = s_j$, tudi lahko uporabimo formulo, če je $a_{M_i} = 0$. Če formule ne moremo uporabiti ali če ne velja $s = s_j$, potem se algoritem $\mathcal{A}_1$ ustavi brez, da bi vrnil rezultat.

Če ponarejevalec $\mathcal{F}_1$ uspe vrniti ponarejen podpis (M, σ), potem moramo iz njega dobiti rezultat. Velja

$$\begin{aligned}
\delta &= H_\kappa(M)^{\frac{1}{x+s}} \\
&= \left(\tilde{g}^{a_M \Pi_{t \in S_j}(x+t)} \tilde{g}^{b_M \Pi_{t \in S}(x+t)}\right)^{\frac{1}{x+s}} \\
&= \tilde{g}^{\frac{a_M p_j(x)}{x+s}} \tilde{g}^{b_M p_j(x)} \\
&= \tilde{g}^{\frac{a_M p_j(x)}{x+s}} g^{b_M}.
\end{aligned}$$

Zdaj lahko izračunamo

$$\delta' = \left(\frac{\delta}{g^{b_M}}\right)^{\frac{1}{a_M}} = \tilde{g}^{\frac{p_j(x)}{x+s}}.$$

Ker velja $\gcd(x + s, p_j(x)) = 1$, lahko zapišemo

$$\frac{p_j(x)}{x+s} = \tilde{p}(x) + \frac{q_0}{x+s},$$

kjer je $\tilde{p}(x)$ nek polinom stopnje največ $q-2$ in $q_0 \neq 0$ neka konstanta. Izračunamo $g' = \tilde{g}^{\tilde{p}(x)}$. Zdaj lahko izračunamo:

$$\delta'' = \left(\frac{\delta'}{g'}\right)^{\frac{1}{q_0}} = \left(\tilde{g}^{\frac{p_j(x)}{x+s} - \tilde{p}(x)}\right)^{\frac{1}{q_0}} = \tilde{g}^{\frac{1}{x+s}}.$$

Par (δ'', s) je rešitev problema q-SDH. Tako smo sestavili algoritem $\mathcal{A}_1$, ki razbije predpostavko q-SDH z nezanemarljivo verjetnostjo. ■

Lema 5.3: Recimo, da obstaja nasprotnik $\mathcal{F}_2$, ki s q podpisnimi poizvedbami ponaredi podpis z nezanemarljivo verjetnostjo. Potem obstaja verjetnostni polinomski algoritem $\mathcal{A}_2$, ki je v protislovju s predpostavko q-HSDH.

Dokaz: Lemo bomo dokazali tako, da bomo sestavili algoritem $\mathcal{A}_2$, ki bo uporabil nasprotnika $\mathcal{F}_2$. Spomnimo, da algoritem $\mathcal{A}_2$ kot vhod dobi elemente

$$\left(g, g^x, g^\beta, \left\{g^{\frac{1}{x+s_i}}, g^{s_i}, g^{\beta s_i}\right\}_{i=1}^{q}\right)$$

in vrne

$$\left(g^{\frac{1}{x+s}}, g^s, g^{\beta s}\right),$$

kjer $s \notin \{s_1, ..., s_q\}$.

Sedaj želimo zgenerirati javni in zasebni ključ sheme. Ponovno bomo uporabili verzijo PHF s stranskimi vrati, zato bomo nastavili elemente $h := g^c$, $X := g^x$ in $u := g^\beta$, kjer je $c \in_R \mathbb{Z}_p$. Naključno si izberemo še $y \in \mathbb{Z}_p$ in izračunamo $Y = g^{\frac{1}{y}}$. Inicializiramo še PHF s klicem funkcije

$$(\kappa, \tau) \leftarrow \texttt{PHF.TrapGen}(1^k, g, h)$$

Dobili smo javni ključ $\texttt{pk} = (g, X, Y, u, \kappa)$ in zasebni ključ $\texttt{sk} = (x, y)$, kjer ne poznamo vrednosti x.

Zdaj uporabimo nasprotnika $\mathcal{F}_2$. Brez težav lahko odgovorimo na poizvedbe za potrditveni ali zavrnitveni protokol, saj poznamo vrednost y. Na podpisno poizvedbo M_i odgovorimo s formulami

$$\delta_i := \left(g^{\frac{1}{x+s_i}}\right)^{a_{M_i}+cb_{M_i}}, \ \gamma_i := (g^{s_i})^{\frac{1}{y}} \ \text{ in } \theta_i := u^{s_i},$$

kjer je $(a_{M_i}, b_{M_i}) \leftarrow \texttt{PHF.TrapEval}(\tau, M_i)$. Dobljen podpis je veljaven, ker veljajo enakosti

$$\delta_i = g^{\frac{a_{M_i}+cb_{M_i}}{x+s_i}} = \left(g^{a_{M_i}} h^{b_{M_i}}\right)^{\frac{1}{x+s_i}} = H_\kappa(M_i)^{\frac{1}{x+s_i}},$$
$$\gamma_i = \left(g^{\frac{1}{y}}\right)^{s_i} = Y^{s_i} \quad \text{in} \quad \theta_i = u^{s_i}.$$

Če za katerokoli podpisno poizvedbo M_i velja

$$a_{M_i} + cb_{M_i} \equiv 0 \pmod{p}$$

ali

$$a_M + cb_M \equiv 0 \pmod{p}$$

za rezultat nasprotnika $\mathcal{F}_2$, potem se algoritem $\mathcal{A}_2$ takoj konča in ne vrne nič. Če nasprotnik $\mathcal{F}_2$ konča, vrne sporočilo M in ponarejen podpis (δ, γ, θ), kjer velja

$$\delta = \left(g^{a_M} h^{b_M}\right)^{\frac{1}{x+s}} = \left(g^{a_M + cb_M}\right)^{\frac{1}{x+s}}.$$

Ker velja $a_M + cb_M \not\equiv 0 \pmod{p}$, lahko izračunamo

$$\delta' = \delta^{\frac{1}{a_M + cb_M}} = g^{\frac{1}{x+s}}.$$

Skupaj z $\gamma^y = g^s$ in $\theta = g^{\beta s}$ smo dobili rezultat $(\delta', \gamma^y, \theta)$ algoritma $\mathcal{A}_2$. ∎

Ko obe lemi sestavimo skupaj, dobimo dokaz trditve 5.1.

Naslednja varnostna zahteva je, da sta potrditveni in zavrnitveni protokol brez razkritja znanja, kar velja natanko tedaj, ko je dokaz enakosti, oziroma različnosti, diskretnih logaritmov brez razkritja znanja.

Dokažimo zdaj še nevidnost. Postopek je zopet povzet po [7].

Trditev 5.4 (nevidnost): Naj bo H $(m, 1)$-PHF in naj bo $\mathcal{D}$ nasprotnik iz igre za definicijo nevidnosti z nezanemarljivo prednostjo. Predpostavimo, da je shema neponaredljiva, in da sta potrditveni in zavrnitveni protokol brez razkritja znanja. Potem obstaja verjetnostni polinomski algoritem $\mathcal{A}$, ki razbije predpostavko $(q+1)$-DHSDH, kjer je q število podpisnih poizvedb nasprotnika $\mathcal{D}$.

Dokaz: Z M_i označimo i-to podpisno poizvedbo, z $(\delta_i, \gamma_i, \theta_i)$ pa odgovor nanjo, kjer bo odgovor zgeneriran z vrednostjo s_i. Naj bo M sporočilo, ki si ga izbere nasprotnik $\mathcal{D}$, in naj bo $\sigma = (\delta, \gamma, \theta)$ podpis, za katerega mora nasprotnik ugotoviti, ali je veljaven ali ne.

Sestavili bomo algoritem $\mathcal{A}$, ki bo uporabil nasprotnika $\mathcal{D}$. Algoritem $\mathcal{A}$ prejme kot vhod elemente

$$\left(g, g^x, g^\beta, \left\{g^{\frac{1}{x+s_i}}, g^{s_i}, g^{\beta s_i}\right\}_{i=1}^{q}, g^{\beta s}, Z\right)$$

in mora vrniti vrednost 1, če je $Z = g^{\frac{1}{x+s}}$, sicer pa 0.

Vpeljimo nove oznake $h := g^c$, $u := g^\beta$, $X := g^x$ in $Y := \left(g^\beta\right)^d$ za $c, d \in_R \mathbb{Z}_p$. Nato inicializiramo verzijo PHF s stranskimi vrati s klicem funkcije

$$(\kappa, \tau) \leftarrow \texttt{PHF.TrapGen}\left(1^k, g, h\right).$$

Zdaj lahko uporabimo nasprotnika $\mathcal{D}$. Ko le-ta naredi podpisno poizvedbo M_i, nanjo odgovorimo z vrednostmi izračunanimi po formulah

$$\begin{aligned}\delta_i &:= H_\kappa\left(M_i\right)^{\frac{1}{x+s_i}} \\ &= \left(g^{a_{M_i}} h^{b_{M_i}}\right)^{\frac{1}{x+s_i}} \\ &= \left(g^{\frac{1}{x+s_i}}\right)^{a_{M_i}+cb_{M_i}},\end{aligned}$$

$$\gamma_i := Y^{s_i} = (u^{s_i})^d \quad \text{in} \quad \theta_i := u^{s_i},$$

kjer je $(a_{M_i}, b_{M_i}) \leftarrow \texttt{PHF.TrapEval}(\tau, M_i)$. Če nasprotnik $\mathcal{D}$ naredi poizvedbo po potrditvenem protokolu za par (M', σ'), nanjo odgovorimo samo, če je prej naredil podpisno poizvedbo na tem sporočilu in kot odgovor dobil podpis σ'. Drugače algoritem $\mathcal{A}$ zaključi in ne vrne ničesar. Prav tako zaključimo in ne nič ne vrnemo, če nasprotnik $\mathcal{D}$ naredi poizvedbo za zavrnitveni protokol.

Ko nasprotnik $\mathcal{D}$ izbere sporočilo M, končamo in nič ne vrnemo, če velja $a_M + cb_M \equiv 0 \pmod{p}$, sicer izračunamo

$$\delta = Z^{a_M + cb_M}, \ \gamma \quad = Y^s = (u^s)^d \ \text{ in } \theta = u^s.$$

Če je $b = 0$, potem velja

$$\begin{aligned}\delta &= Z^{a_M + cb_M} \\ &= \left(g^{\frac{1}{x+s}}\right)^{a_M + cb_M} \\ &= \left(g^{a_M + cb_M}\right)^{\frac{1}{x+s}} \\ &= H_\kappa(M)^{\frac{1}{x+s}},\end{aligned}$$

kar pomeni, da je $\sigma = (\delta, \gamma, \theta)$ veljaven podpis sporočila M. Če pa je $b = 1$, potem je Z naključen element

iz grupe G, kar pomeni, da je $\sigma = (\delta, \gamma, \theta)$ naključen podpis iz množice vseh podpisov.

Rezultat nasprotnika $\mathcal{D}$ je tudi rezultat algoritma $\mathcal{A}$. S tem smo sestavili algoritem $\mathcal{A}$, ki razbije predpostavko $(q+1)$-DHSDH. ■

Anonimnost ne velja, kar je hitro vidno, ker iz podpisa $\sigma = (\delta, \gamma, \theta)$ lahko preverimo, če pripada uporabniku z javnim ključem $\mathtt{pk} = (g, X, Y, u, \kappa)$ tako, da preverimo, ali velja enakost

$$e(\gamma, u) = e(Y, \theta).$$

Iz tega sta Schuldt in Matsuura [9] sklepala, da shema tudi ni nevidna, kar pa ne drži, saj smo nevidnost že uspešno dokazali.

6 ZAKLJUČEK

V članku smo spoznali podpise brez možnosti zanikanja. Ker so taki podpisi kompleksnejši od navadnih digitalnih podpisov, potrebujejo tudi več varnostnih zahtev. Seznanili smo se z bilinearnimi parjenji in predstavili shemo, ki temelji na njih. Za predstavljeno shemo velja nevidnost, čeprav sta kasneje Schuldt in Matsuura [9] dokazala, da za shemo ne velja anonimnost zaradi različnih množic vseh možnih podpisov med uporabniki. Tako sta narobe sklepala, da shema tudi ni nevidna zaradi ekvivalence anonimnosti in nevidnosti. Loh et al. [8] so nato izpostavili, da ekvivalenca velja le, če so množice vseh možnih podpisov enake za vse uporabnike, kar za to shemo ne velja.

LITERATURA

[1] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.

[2] J. Boyar, D. Chaum, I. Damgård, and T. Pedersen. Convertible undeniable signatures. In *Conference on the Theory and Application of Cryptography*, pages 189–205. Springer, 1990.

[3] D. Chaum and H. Van Antwerpen. Undeniable signatures. In *Conference on the Theory and Application of Cryptology*, pages 212–216. Springer, 1989.

[4] D. Chaum, E. van Heijst, and B. Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In *Annual International Cryptology Conference*, pages 470–484. Springer, 1991.

[5] S. D. Galbraith and W. Mao. Invisibility and anonymity of undeniable and confirmer signatures. In M. Joye, editor, *Topics in Cryptology — CT-RSA 2003*, pages 80–97, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[6] D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. In *Annual International Cryptology Conference*, pages 21–38. Springer, 2008.

[7] Q. Huang and D. S. Wong. New constructions of convertible undeniable signature schemes without random oracles. *IACR Cryptol. ePrint Arch.*, 2009:517, 2009.

[8] J.-C. Loh, S.-H. Heng, S.-Y. Tan, and K. Kurosawa. A note on the invisibility and anonymity of undeniable signature schemes. In I. You, editor, *Information Security Applications*, pages 112–125, Cham, 2020. Springer International Publishing.

[9] J. C. Schuldt and K. Matsuura. An efficient convertible undeniable signature scheme with delegatable verification. In *International Conference on Information Security Practice and Experience*, pages 276–293. Springer, 2010.