Digitalni podpisi

Tomaž Klobučar

11. maj, 1998

1 Uvod

Digitalni podpis je nadomestek lastnoročnega podpisa v elektronski izmenjavi podatkov.

Primerjava digitalnega in navadnega (lastnoročnega) podpisa:

- navadni podpis je fizično del podpisanega dokumenta;
- navadni podpis preverjamo s primerjanjem, digitalnega z algoritmom, katerega rezultat je odvisen od ključa in dokumenta;
- kopija digitalnega podpisa je identična originalu;
- digitalni podpis je odvisen od dokumenta, ki ga podpisujemo.

Definicija 1 *Sistem za digitalno podpisovanje je peterka* $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, *kjer velja*

1. $\mathcal{P}$ *je končna množica sporočil*
2. $\mathcal{A}$ *je končna množica podpisov*
3. $\mathcal{K}$ *je končna množica ključev*
4. *Za vsak ključ* $K \in \mathcal{K}$ *obstaja algoritem za podpisovanje* $sig_K \in \mathcal{S}, sig_K : \mathcal{P} \rightarrow \mathcal{A}$ *in algoritem za preverjanje podpisa* $ver_K \in \mathcal{V}, ver_K : \mathcal{P} \times \mathcal{A} \rightarrow \{true, false\}$. *Funkciji* sig_K *in* ver_K *imata to lastnost, da za vsako sporočilo* $x \in \mathcal{P}$ *in vsak podpis* $y \in \mathcal{A}$ *velja*
$$ver_K(x, y) = \begin{cases} true & \text{če } y = sig_K(x) \\ false & \text{če } y \neq sig_K(x) \end{cases}$$

Opombe:

- algoritma sig_K in ver_K imata polinomsko časovno zahtevnost
- sig_K je znan le podpisniku
- ver_K je splošno znan
- računsko mora biti nemogoče ponarediti podpis

Primer: Algoritem RSA lahko uporabimo tudi za podpisovanje. Če je (n, d) skriti ključ, (n, e) pa javni, je:

$$\begin{aligned} sig_K(x) &= x^d \pmod n \\ ver_K(x, y) &= true \iff x \equiv y^e \pmod n \end{aligned}$$

Opomba: Z zgornjim algoritmom je mogoče ponarediti podpis naključnih sporočil. Ponarejevalec najprej izbere podpis y in nato izračuna $x \equiv y^e \pmod n$. Možnosti takega ponarejanja se izognemo z enosmernimi zgoščevalnimi funkcijami ali z zahtevo, da ima sporočilo x določen pomen.

Vrstni red šifriranja in digitalnega podpisovanja:

1. Najprej podpisovanje $x, sig_{Ana}(x) \rightarrow e_{Borut}((x, sig_{Ana}(x)))$

2. Najprej šifriranje

$$\begin{aligned} z = e_{Borut}(x), y = sig_{Ana}(z) &\quad \rightarrow (z, y) &\quad \rightarrow x = d_{Borut}(z), ver_{Ana}(z, y) \\ y' = sig_{Cene}(z) &\quad \rightarrow (z, y') &\quad \rightarrow x = d_{Borut}(z), ver_{Cene}(z, y') \end{aligned}$$

V primeru algoritma RSA je potrebno pri zaporednem podpisovanju in šifriranju paziti na velikosti modulov (*reblocking problem*). Če je $n_{Ana} > n_{Borut}$, se lahko zgodi, da Borut ne bo mogel razvozlati sporočila. Naj bo $(n_{Ana}, e_{Ana}, d_{Ana}) = (62894113, 5, 37726937)$ in $(n_{Borut}, e_{Borut}, d_{Borut}) = (55465219, 5, 44360237)$. Ana podpiše sporočilo $x = 1368797$ in podpis zašifrira:

1. $s = x^{d_{Ana}} \bmod n_{Ana} = 59847900$.

2. $y = s^{e_{Borut}} \bmod n_{Borut} = 38842235$.

Borut izračuna

1. $\widehat{s} = y^{d_{Borut}} \bmod n_{Borut} = 4382681$.

2. $\widehat{x} = \widehat{s}^{e_{Ana}} \bmod n_{Ana} = 54383568$.

Ker je $s > n_{Borut}$, je $x \neq \widehat{x}$. Verjetnost za to je $(n_{Ana} - n_{Borut})/n_{Ana}$.

Načini razdelitve shem za digitalno podpisovanje:

1. podpis je dodatek (ElGamal, DSA) sporočilu - sporočilo je možno rekonstruirati iz podpisa (RSA)
2. deterministični - nedeterministični
3. enkratni - večkratni

Različni sistemi za digitalno podpisovanje:

- RSA
- El Gamal, DSS (*Digital Signature Standard*)
- Enkratni podpisi (*one-time signatures*)
- Slepi podpisi (*blind signatures*)
- Podpisi brez možnosti zanikanja (*undeniable signatures*)
- Skupinski podpisi (*group signatures*)
- Fail-Stop podpisi

2 El Gamalov sistem za digitalno podpisovanje

Za razliko od algoritma RSA je El Gamalov sistem namenjen predvsem za digitalno podpisovanje, čeprav se ga da v posebnih primerih uporabiti tudi za šifriranje. Podpis je nedeterminističen (odvisen od naključnega števila).

Algoritem: Naj bo p takšno praštevilo, da je v Z_p težko najti diskretni algoritem, in $\alpha \in Z_p^*$ primitivni element. Naj bo še $\mathcal{P} = Z_p^*$, $\mathcal{A} = Z_p^* \times Z_{p-1}$ in

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}.$$

Število a je skrito, števila p, α in β pa so javno znana.

- **Podpisovanje:** podpisnik izbere naključno skrito število $k \in Z_{p-1}^*$ in določi

  $$sig_K(x, k) = (\gamma, \delta),$$

  kjer je

  $$\gamma \equiv \alpha^k \pmod{p}$$

  in

  $$\delta \equiv (x - a\gamma)k^{-1} \pmod{(p-1)}.$$

- **Preverjanje podpisa:**

  $$ver_K(x, \gamma, \delta) = true \iff \beta^\gamma \gamma^\delta \equiv \alpha^x \pmod{p}.$$

Primer: Naj bo $p = 467, \alpha = 2$ in $a = 127$. Potem je $\beta \equiv \alpha^a \bmod p = 132$. Recimo, da želimo podpisati $x = 100$, izbrali pa smo si tudi $k = 213$. Podpis je enak (γ, δ), kjer je

$$\gamma \equiv 2^{213} \bmod 467 = 29$$

in

$$\delta \equiv (100 - 127\ 29)\ 431 \bmod 466 = 51.$$

Pri preverjanju izračunamo

$$132^{29}\ 29^{51} \equiv 189 \pmod{467}$$

in

$$2^{100} \equiv 189 \pmod{467}.$$

Zadnji vrednosti se ujemata, zato je podpis pravi.

Varnost El Gamalovega sistema za podpisovanje: Kako bi lahko ponaredili podpis, ne da bi vedeli za vrednost skritega števila a?

1. Za dano sporočilo x je potrebno najti tak par (γ, δ), da bo veljalo $\beta^\gamma \gamma^\delta \equiv \alpha^x \pmod p$:

   - izberemo $\gamma : \delta = log_\gamma \alpha^x \beta^{-\gamma} \pmod p$
   - izbeemo δ: glede na γ je potrebno rešiti enačbo $\beta^\gamma \gamma^\delta \equiv \alpha^x \pmod p$
   - hkrati računamo γ in δ: za zdaj ni še nihče odkril hitrega postopka za reševanje zgornje enačbe

2. Za dan podpis (γ, δ) je potrebno najti ustrezno sporočilo x:

   - $x = log_\alpha \beta^\gamma \gamma^\delta \pmod p$

3. Hkratno računanje x, γ in δ: Naj bosta i in j takšni števili, da velja $0 \leq i, j \leq p-2$ in $\gcd(j, p-1) = 1$. Potem števila

$$\begin{aligned} \gamma &\equiv \alpha^i \beta^j \pmod p, \\ \delta &\equiv -\gamma j^{-1} \pmod{(p-1)}, \\ x &\equiv -\gamma i j^{-1} \pmod{(p-1)} \end{aligned}$$

   zadoščajo enačbi $\beta^\gamma \gamma^\delta \equiv \alpha^x \pmod p$.

   Primer: Če je $p = 467, \alpha = 2$ in $\beta = 132$, lahko z izbiro $i = 99$ in $j = 179$, dobimo veljaven podpis $(117, 41)$ za sporočilo 331.

4. Ali lahko pri veljavnem podpisu (γ, δ) za x najdemo še kakšen podpis za neko drugo sporočilo x'? Odgovor je lahko. Naj bodo h, i in j takšna števila, da za njih velja $0 \leq h, i, j \leq p-2$ in $gcd(h\gamma - j\delta, p-1) = 1$. Potem je par (λ, μ) veljaven podpis za x', kjer je

$$\begin{aligned} \lambda &\equiv \gamma^h \alpha^i \beta^j \pmod p \\ \mu &\equiv \delta\lambda(h\gamma - j\delta)^{-1} \pmod{(p-1)} \\ x' &\equiv \lambda(hx + i\delta)(h\gamma - j\delta)^{-1} \pmod{(p-1)} \end{aligned}$$

Nevarnosti pri napačni uporabi ElGamalovega sistema:

1. Če naključno število k ne ostane skrito, lahko izračunamo

$$a \equiv (x - k\delta)\gamma^{-1} \pmod{(p-1)}.$$

2. Število k lahko uporabimo le enkrat. V nasprotnem primeru ga je mogoče zlahka izračunati.

3 Digital Signature Standard

DSS je modifikacija ElGamalovega sistema za podpisovanje. Kot ameriški standard je bil predlagan l. 1991, sprejet pa l.1994.

Algoritem: Naj bo p praštevilo velikosti L bitov, kjer je $512 \leq L \leq 1024$ in $64|L$, q 160−bitno praštevilo, ki deli $p-1$, ter $\alpha \in Z_p^*$ $q-$ti koren enote modulo p. Definirajmo $\mathcal{P} = Z_p^*, \mathcal{A} = Z_q \times Z_q$ in

$$\mathcal{K} = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod p\}.$$

Vrednosti p, q, α in β so javne, število a pa skrito.

- **Podpisovanje:** podpisnik izbere naključno skrito število k, $1 \leq k \leq q-1$ in določi

  $$sig_K(x, k) = (\gamma, \delta),$$

  kjer je

  $$\gamma \equiv (\alpha^k \bmod p) \bmod q$$

  in

  $$\delta \equiv (x + a\gamma)\, k^{-1} \pmod q.$$

  Za število δ mora veljati $\delta \not\equiv 0 \pmod q$.

- **Preverjanje podpisa:** najprej izračunamo

  $$e_1 \equiv x\delta^{-1},$$

  $$e_2 \equiv \gamma\delta^{-1}.$$

  Potem je

  $$ver_K(x, \gamma, \delta) = true \iff (\alpha^{e_1} \beta^{e_2} \bmod p) \bmod q = \gamma.$$

Podobno kot pri ElGamalovi shemi je podpisovanje hitrejše od preverjanja (za razliko od RSA).

3.1 Prikrit kanal v algoritmu DSA

V algoritmu DSA obstaja prikrit kanal, ki omogoča:

- vključitev šifriranega sporočila v podpis, ki ga lahko prebere le tisti, ki pozna dodaten ključ;
- razkritje skritega ključa, brez vednosti njegovega lastnika.

Primer: Imejmo n skritih praštevil $p_1, \ldots, p_n$ in poskusimo v podpis skriti niz $b_1, \ldots, b_n$, $b_i \in \{0, 1\}$. Naključno število k izbiramo toliko časa, da za vsak $1 \leq i \leq n$ velja

$$b_i = 1 \Longrightarrow \gamma \text{ je kvadratični ostanek po modulu } p_i$$

$$b_i = 0 \Longrightarrow \gamma \text{ ni kvadratični ostanek po modulu } p_i,$$

kjer je $sig_K(x, k) = (\gamma, \delta)$.

4 Enkratni podpis

Z istim ključem lahko podpišemo le en dokument. Ponavadi algoritem temelji na enosmernih funkcijah.

Algoritem (Lamport): Naj bo k pozitivno število, $\mathcal{P} = \{0,1\}^k$ in $f : Y \to Z$ enosmerna funkcija. Naključno izberemo matriko $(y_{i,j}), 1 \leq i \leq k, j = 0, 1$ z elementi iz Y in določimo matriko enake velikosti z elementi $z_{i,j} = f(y_{i,j})$. Ključ K sestavljata obe matriki, prva je skrita, druga pa javna.

- **Podpisovanje:**
$$sig_K(x_1, \ldots, x_k) = (y_{1,x_1}, \ldots, y_{k,x_k}).$$
- **Preverjanje podpisa:**
$$ver_K(x_1, \ldots, x_k, a_1, \ldots, a_k) = true \iff f(a_i) = z_{i,x_i}, 1 \leq i \leq k.$$

Primer: Naj bo $f(x) = 3^x \pmod{7879}$, ključ pa sestavljen iz matrik

$$\begin{pmatrix} 5831 & 735 \\ 803 & 2467 \\ 4285 & 6449 \end{pmatrix} \text{ in } \begin{pmatrix} 2009 & 3810 \\ 4672 & 4721 \\ 268 & 5731 \end{pmatrix}.$$

Potem je podpis za $x = (1, 1, 0)$ enak $(y_{1,1}, y_{2,1}, y_{3,0}) = (735, 2467, 4285)$.

Pomanjkljivost omenjenega sistema je velikost podpisa (v zgornjem primeru dobimo za vsak bit število med 1 in 7878.

5 Slepi podpis

Želimo, da nam kdo podpiše dokument, hkrati pa nočemo, da bi podpisnik videl njegovo vsebino (npr. notarji, banke pri elektronskem denarju).

Algoritem (Chaum): Ana želi od Boruta podpis dokumenta x, $1 \leq x \leq n-1$, pri čemer je (n, e) Borutov javni ključ za algoritem RSA, d pa zasebni ključ.

1. Ana izbere takšno skrito naključno število k, da velja $0 \leq k \leq n-1$ in $\gcd(n, k) = 1$. Nato zastre dokument, t.j. izračuna $m \equiv xk^e \pmod{n}$, in ga pošlje Borutu.

2. Borut podpiše zastrt dokument: $s \equiv m^d \pmod{n}$.

3. Ana odstre podpis dokumenta: $y \equiv k^{-1}s \pmod{n}$.