Bos-Chaumova shema za enkratni podpis

$\mathcal{P} = \{0,1\}^{k\in\mathbb{N}}$, $n \in \mathbb{N}$ tak, da je $2^k \leq \binom{2n}{n}$.
B je množica z $2n$ elementi in

$$\phi : \{0,1\}^k \to \mathcal{B}$$

injekcija, kjer je $\mathcal{B}$ množica n-teric iz B.

Naj bo $f : Y \to Z$ enosmerna funkcija.

Naključno izberemo vektor $\mathbf{y}= (y_i) \in Y^{2n}$.

Naj bo ključ K tajni vektor $\mathbf{y}$ in javni vektor $(f(y_i))$.

$$\mathrm{sig}_K(x_1, \ldots, x_k) = \{y_j \,|\, j \in \phi(x_1, \ldots, x_k)\}.$$

in

$$\mathrm{ver}_K(x_1, \ldots, x_k, a_1, \ldots, a_n) = \mathrm{true}$$
$$\Updownarrow$$
$$\{f(a_i) \,|\, 1 \leq i \leq n\} = \{z_j \,|\, j \in \phi(x_1, \ldots, x_k)\}.$$

Uporabili smo $2^k \leq \binom{2n}{n}$. Ocenimo binomski koeficient in dobimo

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2}$$

oziroma z uporabo Stiringove formule $2^{2n}/\sqrt{(}\pi n)$.

Od tod dobimo

$$k \leq 2n - \frac{\log_2(n\pi)}{2}.$$

Asimptotično je torej n blizu $k/2$, zato smo dobili 50% redukcijo dolžine podpisa.

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): Anita želi od Bojana podpis dokumenta x, $1 \leq x \leq n-1$, pri čemer je (n, e) Bojanov javni ključ za algoritem RSA, d pa zasebni ključ.

1. *Anita izbere takšno skrito naključno število* k, *da velja* $0 \leq k \leq n-1$ *in* $D(n,k)=1$.

   Nato zastre dokument, tj. izračuna

   $$m = xk^e \bmod n,$$

   in ga pošlje Bojanu.

2. *Bojan podpiše zastrti dokument*

   $$s = m^d \bmod n.$$

3. *Anita odstre podpisani dokument*

   $$y = k^{-1}s \bmod n.$$

Podpisi brez možnosti zanikanja

Podpisa ni mogoče preveriti brez sodelovanja podpisnika, podpisnik pa tudi ne more zanikati, da bi že podpisani dokument res podpisal

(razen če odkloni sodelovanje pri podpisu, kar pa lahko pojmujemo kot priznanje, da je podpis v resnici ponarejen).

Primer algoritma (Chaum-van Antwerpen):

Naj bosta q in $p = 2q + 1$ praštevili, $\alpha \in \mathbb{Z}_p^*$ element reda q, $1 \leq a \leq q - 1$ in $\beta = \alpha^a \bmod p$.

Grupa G je multiplikativna podgrupa reda q grupe $\mathbb{Z}_p^*$ (G sestavljajo kvadratični ostanki po modulo p).

Naj bo $\mathcal{P} = \mathcal{A} = G$ in

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

Števila p, α in β so javna, vrednost a pa je skrita.

Podpisovanje (Bojan podpiše dokument $x \in G$):

$$y = \mathrm{sig}_K(x) = x^a \bmod p.$$

Preverjanje podpisa:

1. Anita izbere naključni števili $e_1, e_2 \in \mathbb{Z}_q^*$. Nato izračuna $c = y^{e_1}\beta^{e_2} \bmod p$ in ga pošlje Bojanu.
2. Bojan izračuna $d = c^{a^{-1} \bmod q} \bmod p$ in ga vrne Aniti.
3. Anita sprejme podpis kot veljaven, če je

$$d = x^{e_1}\alpha^{e_2} \bmod p.$$

Izrek. *Če je* $y \not\equiv x^a \pmod{p}$, *potem bo Anita sprejela* y *za veljaven podpis čistopisa* x *z verjetnostjo* $1/q$.

Poleg algoritmov za podpisovanje in preverjanje obstaja še algoritem (*disavowal protocol*), s katerim lahko podpisnik dokaže, da je ponarejen podpis res ponarejeni, hkrati pa ne more zanikati, da pravega podpisa ni napravil sam.

Primeri podpisov brez možnosti zanikanja

- *Entrusted undeniable signature*: *disavowal* protokol lahko izvede le za to določena ustanova, npr. sodišče.
- *Designated confirmer signature*: ob podpisu sami določimo, kdo bo namesto nas sodeloval pri preverjanjih podpisov. Podpišemo lahko še vedno le mi.
- *Convertible undeniable signature*: shema vsebuje skrito število. Do razkritja tega števila mora pri preverjanju podpisa sodelovati podpisnik.

  Po razkritju lahko kdorkoli preveri podpis sam

  (kot pri običajnem digitalnem podpisu).

Skupinski podpisi

Lastnosti:

- Dokumente lahko podpisujejo le člani določene skupine.
- Kdorkoli lahko preveri, da je dokument podpisal nekdo iz omenjene skupine, vendar ne more ugotoviti, kdo je to bil.
- V primeru spora je možno podpis “odpreti” in identificirati podpisnika.

Fail-stop podpisi

Če bi ponarejevalec z metodo grobe sile našel skriti ključ, bi lahko v večini sistemov za digitalne podpise podpis ponaredil. Fail-stop sistemi takšno možnost onemogočijo tako, da vsakemu javnemu ključu priredijo več skritih ključev.

Algoritem (van Heyst - Pedersen)

Generiranje ključa se razdeli med Anito in TTP (*Trusted Third Party*).

TTP izbere praštevili q in $p = 2q + 1$ (diskretni algoritem je težko izračunljiv), element $\alpha \in \mathbb{Z}_p^*$ reda q ter skrito naključno število a_0, $1 \leq a_0 \leq q-1$ in izračuna $\beta \equiv \alpha^{a_0} \pmod p$. Nato Anita pošlje četverko (p, q, α, β) in izbere skrita naključna števila $a_1, a_2, b_1, b_2 \in \mathbb{Z}_q$, ki predstavljajo njen skriti ključ, ter določi svoj javni ključ $(\gamma_1, \gamma_2, p, q, \alpha, \beta)$, kjer je

$$\gamma_1 = \alpha^{a_1}\, \beta^{a_2} \pmod p \text{ in } \gamma_2 = \alpha^{b_1}\, \beta^{b_2} \pmod p.$$

Podpisovanje: $y = \mathrm{sig}_K(x) = (y_1, y_2)$, kjer je

$$y_1 \equiv a_1 + x\, b_1 \pmod{q}$$

in

$$y_2 \equiv a_2 + x\, b_2 \pmod{q}.$$

Preverjanje podpisa:

$$\mathrm{ver}_K(x, y_1, y_2) = \mathrm{true} \iff \gamma_1 \gamma_2^x \equiv \alpha^{y_1} \beta^{y_2} \pmod{p}.$$

Opombe:

1. Natanko q^2 četverk (a'_1, a'_2, b'_1, b'_2), kjer so elementi iz $\mathbb{Z}_q$, da enaki vrednosti (γ_1, γ_2) v javnem ključu.
2. Teh q^2 četverk da pri istem dokumentu x q različnih podpisov.
3. Naj bo Q_1 množica q četverk, ki da pri x enak podpis. Potem da ta množica pri drugem dokumentu q različnih podpisov.

Varnost sistema

Recimo, da želi nekdo ponarediti Anitin podpis za sporočilo x'.

1. Če ponarejevalec pozna le skriti ključ, ki pripada javnemu, je verjenost $1/q$, da je njegov podpis enak Anitinemu.

2. Ponarejevalec ima dostop do drugega sporočila x in Anitinega podpisa (y_1, y_2). Po tretji opombi je verjetnost spet $1/q$.

7. poglavje

Zgoščevalne funkcije (Hash Functions)

- zgoščevalne funkcije brez trčenj
- verjetnost trčenja
- napad s pomočjo paradoksa rojstnih dnevov
- zgoščevalna funkcija z diskretnim logaritmom

Shema DSS (brez uporabe zgoščevalnih funkcij) podvoji dolžino podpisanega sporočila.

Resnejši problem nastane, ker je mogoče preurejati dele podpisanega sporočila ali pa nekatere celo izpustiti/dodati.

Celovitost podatkov ne more biti zagotovljena izključno s podpisovanjem majhnih delov dokumenta, zato vpeljemo **zgoščevalne funkcije** (angl. Hash Functions), ki poljubno dolgemu sporočilu priredijo kratko zaporedje bitov, ki jih potem podpišemo.

Zgoščevalne funkcije brez trčenj

(angl. Collision-free Hash Functions)

Naj bo (x, y) podpisano sporočilo, kjer je

$$y = \text{sig}_K(h(x)).$$

Preprost napad: izračunamo $z = h(x)$ in nato poiščemo tak od x različen x', da je $h(x') = h(x)$.

Def: Naj bo x sporočilo. Za zgoščevalno funkcijo h pravimo, da je **šibko brez trčenj** (angl. weakly collision-free), če v doglednem času ni možno najti (izračunati) tak od x različen x', da je $h(x) = h(x')$.

Še en napad: poiščemo taka x in x', da je $x \neq x'$ in $h(x') = h(x)$ ter prisilimo Bojana, da podpiše x. Potem je (x', y) poneverjen podpis.

Def: Za zgoščevalno funkcijo h pravimo, da je **krepko brez trčenj** (angl. strongly collision-free), če v doglednem času ni možno najti (izračunati) taka x in x', da je $x \neq x'$ in $h(x) = h(x')$.

Pa še en napad: recimo, da nam je uspelo ponarediti podpis naključnega števila z, nato pa poiščemo tak x, da je $z = h(x)$.

Ta napad preprečimo z enosmernimi funkcijami.

Dokazali bomo, da so funkcije brez trčenj enosmerne. To sledi iz trditve, da je možno algoritem za računanje obrata zgoščevalne funkcije uporabiti kot podprogram Las Vegas probabilističnega algoritma, ki išče trčenja.

Izrek: *Naj bo $h : X \longrightarrow Z$ zgoščevalna funkcija, $|X| < \infty$ in $|X| \geq 2|Z|$ ter naj bo* **A** *algoritem za računanje obrata zgoščevalne funkcije. Potem obstaja Las Vegas probabilistični algoritem, ki najde trčenja z verjetnostjo vsaj $1/2$.*

Dokaz: Naj bo **B** naslednji algoritem.

1. Izberi naključen element $x \in X$,
2. izračunaj $z := h(x)$,
3. izračunaj $x_1 := \mathbf{A}(z)$,
4. **if** $x_1 \neq x$ **then** x_1 in x trčita glede na h (uspeh)
   else QUIT(neuspeh).

Izračunajmo verjetnost za uspeh. Najprej definiramo ekvivalenčno relacijo

$$x \sim x' \iff h(x) = h(x').$$

Naj bo C množica ekvivalenčnih razredov, potem je $|C| \leq |Z|$. Velja tudi: $|Z| \leq |X|/2$.

$$P(\text{uspeh}) \;=\; \frac{1}{|X|}\sum_{x\in X}\frac{|[x]|-1}{|[x]|} \;=\; \frac{1}{|X|}\sum_{c\in C}\sum_{x\in c}\frac{|c|-1}{|c|}$$

$$= \frac{1}{|X|}\sum_{c\in C}(|c|-1) \;\geq\; \frac{|X|-|Z|}{|X|} \;\geq\; \frac{1}{2}\,. \;\blacksquare$$

Verjetnost trčenja

Naj bo $h : X \longrightarrow Z$ zgoščevalna funkcija,

$$s_z = |h^{-1}(z)|$$

in

$$N = |\{\{x_1, x_2\} \,|\, h(x_1) = h(x_2)\}|,$$

tj. N je število neurejenih parov, ki trčijo pri funkciji h.

Pokazali bomo, da obstaja od nič različna spodnja meja za verjetnost P, da je $h(x_1) = h(x_2)$, kjer sta x_1 in x_2 naključna (ne nujno različna) elementa iz X.

1. $\displaystyle\sum_{z\in Z} s_z = |X|$, tj. povprečje s_z-ov je $\overline{s} = |X|/|Z|$.

2. $\displaystyle N = \sum_{z\in Z} \binom{s_z}{2} = \frac{1}{2}\sum_{z\in Z} s_z^2 - \frac{|X|}{2}$.

3. $\displaystyle\sum_{z\in Z} (s_z - \overline{s})^2 = 2N + |X| - |X|^2/|Z|$.

4. $\displaystyle N \geq \frac{|X|}{2}\left(\frac{|X|}{|Z|} - 1\right)$, pri čemer velja enakost natanko tedaj, ko je $s_z = |X|/|Z|$ za vsak $z \in Z$.

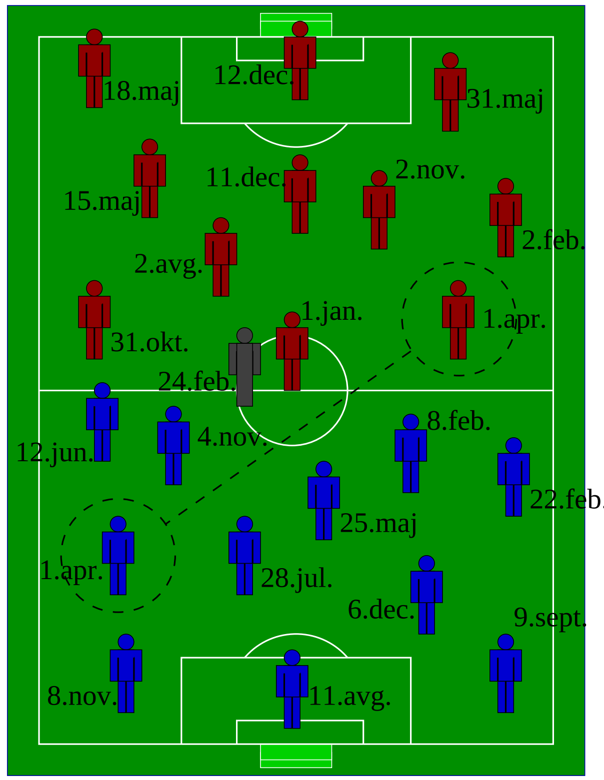
5. $P \geq 1/|Z|$, pri čemer velja enakost natanko tedaj, ko je $s_z = |X|/|Z|$ za vsak $z \in Z$.

Kakšno naključje!!! Mar res?

Na nogometni tekmi sta na igrišču dve enajsterici in sodnik, skupaj **23 oseb**.

Kakšna je verjetnost, da imata **dve osebi** isti rojstni dan?

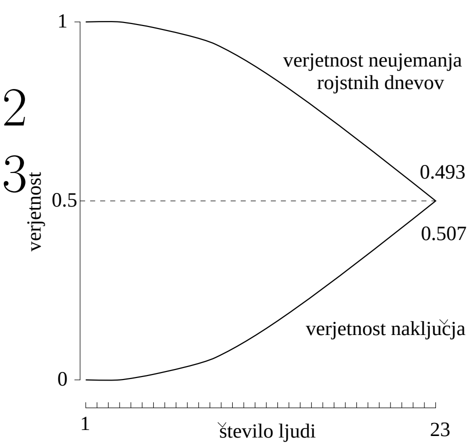
Ali je ta verjetnost lahko večja od 0.5?



Ko vstopi v sobo k-ta oseba, je verjetnost, da je vseh k rojstnih dnevov različnih enaka:

$$\frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \cdots \times \frac{365-k+1}{365}$$

$$= \begin{cases} 0.493, & \text{če je } k=22 \\ 0.507, & \text{če je } k=23 \end{cases}$$



V poljubni skupini 23-ih ljudi je verjetnost, da imata vsaj dva skupni rojstni dan $> 1/2$.

Čeprav je 23 majhno število, je med 23 osebami 253 različnih parov. To število je veliko bolj povezano z iskano verjetnostjo.

Testirajte to na zabavah z več kot 23 osebami.

Organizirajte stave in dolgoročno boste gotovo na boljšem, na velikih zabavah pa boste zlahka zmagovali.

Napad s pomočjo paradoksa rojstnih dnevov

(angl. Birthday Attack)

To seveda ni paradoks, a vseeno ponavadi zavede naš občutek.

Ocenimo še splošno verjetnost.

Mečemo k žogic v n posod in gledamo, ali sta v kakšni posodi vsaj dve žogici.

Poiščimo spodnjo mejo za verjetnost zgoraj opisanega dogodka.

Privzeli bomo, da je $|h^{-1}(x)| \approx m/n$, kjer je $n = |Z|$ in $m = |X|$ (v primeru, da velikosti praslik niso enake se verjetnost le še poveča).

$$\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{k-1}{n}\right)=\prod_{i=1}^{k-1}\left(1-\frac{i}{n}\right)$$

Iz Taylorjeve vrste

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots$$

ocenimo $1 - x \approx e^{-x}$ in dobimo

$$\prod_{i=1}^{k-1}\left(1 - \frac{i}{n}\right) \approx \prod_{i=1}^{k-1} e^{\frac{-i}{n}} = e^{\frac{-k(k-1)}{2n}}.$$

Torej je verjetnost trčenja

$$1 - e^{\frac{-k(k-1)}{2n}}.$$

Potem velja

$$e^{\frac{-k(k-1)}{2n}} \approx 1 - \varepsilon$$

oziroma

$$\frac{-k(k-1)}{2n} \approx \log(1 - \varepsilon)$$

oziroma

$$k^2 - k \approx 2n \log \frac{1}{1 - \varepsilon}$$

in če ignoriramo $-k$, dobimo končno

$$\boxed{k \approx \sqrt{2n \log \frac{1}{1 - \varepsilon}}\ .}$$

Za $\varepsilon = 0.5$ je

$$k \approx 1.17\sqrt{n},$$

kar pomeni, da, če zgostimo nekaj več kot $\sqrt{n}$ elementov, je bolj verjetno, da pride do trčenja kot da ne pride do trčenja.

V splošnem je k proporcionalen z $\sqrt{n}$.

Napad s pomočjo paradoksa rojstnih dnevov s tem določi spodnjo mejo za velikost zaloge vrednosti zgoščevalne funkcije.

40-bitna zgostitev ne bi bila varna, saj bi prišli do trčenja z nekaj več kot 2^{20} (se pravi milijon) naključnimi zgostitvami z verjetnostjo vsaj 1/2.

V praksi je priporočena najmanj 128-bitna zgostitev in shema DSS z 160-imi biti to vsekakor upošteva.