Digitálny odtlačok dokumentu

Dva príklady na začiatok:

1. Anagram je usporiadanie OT lexikograficky. Pre dostatočne dlhý text sú všetky anagramy približne zhodné.

Digitálny odtlačok dokumentu

Dva príklady na začiatok:

1. Anagram je usporiadanie OT lexikograficky. Pre dostatočne dlhý text sú všetky anagramy približne zhodné.
2. PopCo od Scarlett Thomas...Godelova funkcia:
   Frek. tabulka: e t a o i n s r h d l u c m f y w g p b v k x q j z
   Správa: M = articles
   Kód: $c(M) = 2^3 \times 3^8 \times 5^2 \times 7^5 \times 11^{13} \times 13^{11} \times 17^1 \times 19^7$
   $= 207346230290196365983579463986334514327755094 00$
   Odtlačok: $h(M) = 2073462302$

- identifikácii osôb - odtlačky prstov, dlane, nohy, dúhovky,tvár, DNA, ...

- identifikácii osôb - odtlačky prstov, dlane, nohy, dúhovky,tvár, DNA, ...
- reprezentácia elektronického dokumentu krátkym bitovým reťazcom – digitálnym odtlačkom

- identifikácii osôb - odtlačky prstov, dlane, nohy, dúhovky,tvár, DNA, ...
- reprezentácia elektronického dokumentu krátkym bitovým reťazcom – digitálnym odtlačkom
- potrebujeme teda nájsť funkciu $h : \{0,1\}^* \rightarrow \{0,1\}^m$, kde $\{0,1\}^*$ (funkcia h stratovo komprimuje binárny reťazec ľubovoľnej dĺžky na binárny reťazec pevne danej dĺžky)

Možné využitie napr.:

- efektívny prístup k údajom v databáze (vtedy nám stačí, aby každý prvok z oboru hodnôt funkcie h bol rovnako pravdepodobný),

Možné využitie napr.:

- efektívny prístup k údajom v databáze (vtedy nám stačí, aby každý prvok z oboru hodnôt funkcie h bol rovnako pravdepodobný),
- integrita prenášaných, resp. uložených údajov (tu nemáme na mysli ešte kryptologické zabezpečenie, skôr ide o zabezpečenie komunikácie vzhľadom na bežnú chybovosť komunikačného kanála(CRC),

Možné využitie napr.:

- efektívny prístup k údajom v databáze (vtedy nám stačí, aby každý prvok z oboru hodnôt funkcie h bol rovnako pravdepodobný),
- integrita prenášaných, resp. uložených údajov (tu nemáme na mysli ešte kryptologické zabezpečenie, skôr ide o zabezpečenie komunikácie vzhľadom na bežnú chybovosť komunikačného kanála(CRC),
- digitálny odtlačok dokumentu pre kryptologické aplikácie, napr. pri digitálnych podpisových schémach(tu však kladieme na funkciu h ďalšie požiadavky). Takéto funkcie sa nazývajú aj kryptografické hašovacie funkcie.

Hašovacia funkcia $h : \{0,1\}^* \rightarrow \{0,1\}^m$, m je pevne dané celé kladné číslo (128, 160, 196, 256 a 512), musí spĺňať nasledovné kritériá:

1. je jednocestná, t.j. pre daný digitálny odtlačok y je „ťažké" nájsť správu x, že $h(x) = y$ (nájdenie vzoru; angl. *„preimage resistance"*);

Hašovacia funkcia $h : \{0,1\}^* \to \{0,1\}^m$, m je pevne dané celé kladné číslo (128, 160, 196, 256 a 512), musí spĺňať nasledovné kritériá:

1. je jednocestná, t.j. pre daný digitálny odtlačok y je „ťažké" nájsť správu x, že $h(x) = y$ (nájdenie vzoru; angl. *„preimage resistance"*);
2. k danej správe x je „ťažké" nájsť správu x', že $x' \neq x$ a $h(x') = h(x)$ (nájdenie druhého vzoru; angl. *„second preimage resistance"*).

Hašovacia funkcia $h : \{0,1\}^* \to \{0,1\}^m$, m je pevne dané celé kladné číslo (128, 160, 196, 256 a 512), musí spĺňať nasledovné kritériá:

1. je jednocestná, t.j. pre daný digitálny odtlačok y je „ťažké" nájsť správu x, že $h(x) = y$ (nájdenie vzoru; angl. *„preimage resistance"*);
2. k danej správe x je „ťažké" nájsť správu x', že $x' \neq x$ a $h(x') = h(x)$ (nájdenie druhého vzoru; angl. *„second preimage resistance"*).

Vo všeobecnosti očakávame, že priemerná zložitosť nájdenia vzoru je $\mathcal{O}(\frac{1}{2}2^m)$. Rovnakú zložitosť očakávame aj pre nájdenie druhého vzoru.

Hašovacia funkcia odolná voči kolíziám je taká hašovacia funkcia $h : \{0,1\}^* \to \{0,1\}^m$, pre ktorú je „ťažké" nájsť dve rôzne správy x, x', že $h(x) = h(x')$ (angl. *„collision resistance"*).
Vzhľadom na narodeninový paradox, nájdenie kolízie, t.j. dvoch rôznych správ s rovnakým digitálnym odtlačkom, má zložitosť $\mathcal{O}(2^{m/2})$, preto je potrebné túto skutočnosť brať do úvahy pri voľbe veľkosti digitálneho odtlačku.

Podľa počtu vstupov možno hašovacie funkcie rozdeliť na:

- MDC (Manipulation Detection Code) - jediným vstupom je správa, ktorej digitálny odtlačok sa má vypočítať.
- MAC (Message Authentication Code) - okrem správy je vstupom aj (tajný) kľúč.

Väčšina známych hašovacích funkcií je založených na kompresnej funkcii so vstupmi pevnej veľkosti.

- Správa x sa rozdelí na bloky $x_1, x_2, \ldots, x_t$ fixnej dĺžky. Zvyčajne sa správa ešte špeciálne doplní niekoľkými bitmi podľa dopĺňacieho pravidla.

- Správa x sa rozdelí na bloky $x_1, x_2, \ldots, x_t$ fixnej dĺžky. Zvyčajne sa správa ešte špeciálne doplní niekoľkými bitmi podľa dopĺňacieho pravidla.
- Digitálny odtlačok $h(x)$ sa vypočíta iteratívnym spôsobom (dochádza k reťazeniu) použitím kompresnej funkcie f (Damgard-Merklova konštrukcia, CRYPTO 1989):
  $H_0 = IV$, $H_i = f(x_i, H_{i-1})$, $\quad i = 1, 2, \ldots, t$,
  $h(x) = g(H_t)$,
  kde IV je daný inicializačný vektor a g je výstupná funkcia, ktorá je v mnohých prípadoch identita.

- *IV* aj dopĺňacie pravidlo významne ovplyvňujú bezpečnosť hašovacej funkcie - súčasť špecifikácie.

- *IV* aj dopĺňacie pravidlo významne ovplyvňujú bezpečnosť hašovacej funkcie - súčasť špecifikácie.
- Dopĺňacie pravidlo má byť navrhnuté tak, aby neexistovali dve rôzne správy, ktoré by sa po doplnení zhodovali. Ako doplňovacie pravidlo sa často používa pridanie jedného jednotkového bitu a následne minimálneho potrebného počtu nulových bitov.

- *IV* aj dopĺňacie pravidlo významne ovplyvňujú bezpečnosť hašovacej funkcie - súčasť špecifikácie.
- Dopĺňacie pravidlo má byť navrhnuté tak, aby neexistovali dve rôzne správy, ktoré by sa po doplnení zhodovali. Ako doplňovacie pravidlo sa často používa pridanie jedného jednotkového bitu a následne minimálneho potrebného počtu nulových bitov.
- Damgard a Merkle navrhli ešte pridávať k správe aj jej dĺžku (zvyčajne ide o 64-bitové vyjadrenie pôvodnej dĺžky správy). Ak je f odolná voči kolíziám, tak je aj iterovaná hašovacia funkcia h.

Delenie h podľa konštrukcie:

- MDC založené na blokových šifrách,
- MDC založené na „ťažkých" matematických problémoch,
- špeciálne návrhy MDC.

Takmer všetky, s obľubou používané, hašovacie funkcie, ako napr. MD4, MD5, RIPEMD, HAVAL, SHA, a iné, patria do kategórie špeciálnych návrhov MDC.
V súčasnosti je celosvetovo štandardne používaná hašovacia funkcia SHA-3, príp. RIPEMD–160.

Útoky na hašovacie funkcie

- MD4, MD5
- SHA 1 1995 80 rounds 160bit 2005 – útok
- SHA 2 2001 64/80 224/256/384/512
- SHA 3 = Keccak
- Increased interest in cryptographic hash analysis during the SHA-3 competition produced several new attacks on the SHA-2 family 46, 52, 57. Only the collision attacks are of practical complexity; none of the attacks extend to the full round hash function. At FSE 2012, researchers at Sony gave a presentation suggesting pseudo-collision attacks could be extended to 52 rounds on SHA-256 and 57 rounds on SHA-512 by building upon the biclique pseudo-preimage attack.

Elektronický podpis

- Poskytovať autentizáciu správy (elektronického dokumentu), ktorá je dokázateľná pred treťou osobou v prípade sporu.

Elektronický podpis

- Poskytovať autentizáciu správy (elektronického dokumentu), ktorá je dokázateľná pred treťou osobou v prípade sporu.
- Poskytovať dôkaz, že správa je neporušená (tzv. integrita správy).

Elektronický podpis

- Poskytovať autentizáciu správy (elektronického dokumentu), ktorá je dokázateľná pred treťou osobou v prípade sporu.
- Poskytovať dôkaz, že správa je neporušená (tzv. integrita správy).

Situácia 1 Eva nakupuje od Viktora tovar. Napíše na počítači objednávku. Eva a Viktor sa poznajú a vymenili si spoločný tajný kľúč K. Eva objednávku týmto kľúčom zašifruje (a vytvorí MAC) a pošle Viktorovi. Viktor verí, že objednávka je od Evy, pretože iba tá pozná K.

Situácia 1 Eva nakupuje od Viktora tovar. Napíše na počítači objednávku. Eva a Viktor sa poznajú a vymenili si spoločný tajný kľúč K. Eva objednávku týmto kľúčom zašifruje (a vytvorí MAC) a pošle Viktorovi. Viktor verí, že objednávka je od Evy, pretože iba tá pozná K.

Medzitým ako Viktor pošle tovar, Eva nájde iného dodávateľa, ktorý je lacnejší a Viktorov tovar odmietne. Viktor nemá ako dokázať, že objednávka naozaj prišla od Evy a nenapísal ju on sám.

Situácia 2 Viktor sa poučil, a keď Peggy chce poslať elektronickú objednávku, žiada ju, aby ju „zašifrovala" jej asymetrickým súkromným kľúčom D_P. Keď objednávka príde, Viktor ju dešifruje verejným kľúčom E_P. Zašifrovanú objednávku M môže Viktor použiť ako dôkaz, keďže sa predpokladá, že iba Peggy pozná D_P a teda mohla túto objednávku vytvoriť.

Situácia 2 Viktor sa poučil, a keď Peggy chce poslať elektronickú objednávku, žiada ju, aby ju „zašifrovala" jej asymetrickým súkromným kľúčom D_P. Keď objednávka príde, Viktor ju dešifruje verejným kľúčom E_P. Zašifrovanú objednávku M môže Viktor použiť ako dôkaz, keďže sa predpokladá, že iba Peggy pozná D_P a teda mohla túto objednávku vytvoriť.

Na súde by však mohla vzniknúť celá sada otázok. Naozaj iba Peggy pozná kľúč D_P ? Nie je možné, aby ho nejakým spôsobom Viktor získal? Nemohol Viktor zašifrovanú objednávku M vyrobiť bez znalosti D_P ? Aby podobné otázky mohli byť zodpovedané, oblasť elektronického podpisu podlieha štandardom, resp. zákonným normám, ktoré vymedzujú určité požiadavky na jeho bezpečnú a *nespochybniteľnú* realizáciu.

Autentizácia pomocou RSA-algoritmu

Algoritmus 2.1 (RSA podpis)

parametre účastníkov n_A, e_A, d_A, resp. n_B, e_B, d_B;

digitálny podpis odosielateľa A elektronická podoba mena odosielateľa sa zakóduje ako X. Na konci správy pošle odosielateľ A adresátovi B dvojicu dát $Y_1 = X^{e_B} \pmod{n_B}$, $Y_2 = X^{d_A} \pmod{n_A}$

overenie podpisu adresátom B vypočíta $Y_1^{d_B} \pmod{n_B} = X_1$, z čoho zistí meno domnelého odosielateľa. Jeho verejné parametre e_A, n_A nájde vo *verejnom zozname* a použije ho na *dešifrovanie* správy Y_2: $Y_2^{e_A} \pmod{n_A} = X_2$. Jedine rovnosť $X_1 = X_2$ zabezpečuje pravosť podpisu.

autentizácia rovnaký postup použijeme na celú správu.

Takáto jednoduchá schéma má však zásadné **problémy:**

- možnosť vytvorenia kópie podpisu adresátom

Takáto jednoduchá schéma má však zásadné **problémy:**

- možnosť vytvorenia kópie podpisu adresátom
- možnosť vytvorenia nového podpisu z dvoch predchádzajúcich

Takáto jednoduchá schéma má však zásadné **problémy:**

- možnosť vytvorenia kópie podpisu adresátom
- možnosť vytvorenia nového podpisu z dvoch predchádzajúcich
- nároky na pamäť pri autentizácii celého textu

Všeobecná schéma e-podpisu - pätica $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, v ktorej

1. $\mathcal{P}$ je konečná množina možných správ;
2. $\mathcal{A}$ je konečná množina možných podpisov;
3. $\mathcal{K}$ je konečná množina možných kľúčov;
4. pre každé $k \in \mathcal{K}$ existuje podpisový algoritmus

$$sig_k \in \mathcal{S} : \mathcal{P} \to \mathcal{A}$$

a verifikačný algoritmus

$$ver_k \in \mathcal{V} : \mathcal{P} \times \mathcal{A} \to \{true, false\}$$

tak, že pre každé $x \in \mathcal{P}, y \in \mathcal{A}$ platí

$$ver_k(x, y) = \begin{cases} \text{true} & \text{if } y = sig_k(x) \\ \text{false} & \text{if } y \neq sig_k(x). \end{cases} \quad (1)$$

Dvojica (x, y) sa nazýva podpísaná správa.

Podpisová schéma musí spĺňať nasledovné vlastnosti:

1. pre každé k musia byť funkcie sig_k, ver_k ľahko vyčísliteľné;

Podpisová schéma musí spĺňať nasledovné vlastnosti:

1. pre každé k musia byť funkcie sig_k, ver_k ľahko vyčísliteľné;
2. pre dané x je výpočtovo nemožné, s výnimkou podpisovateľa (Alica), nájsť y tak, aby $ver_k(x, y) = true$. Pritom sa nevylučuje možnosť existencie viacerých y s touto vlastnosťou.

Podpisová schéma musí spĺňať nasledovné vlastnosti:

1. pre každé k musia byť funkcie sig_k, ver_k ľahko vyčísliteľné;
2. pre dané x je výpočtovo nemožné, s výnimkou podpisovateľa (Alica), nájsť y tak, aby $ver_k(x, y) = true$. Pritom sa nevylučuje možnosť existencie viacerých y s touto vlastnosťou.
3. Ak je Oskar schopný pre nepodpísanú správu x nájsť dvojicu (x, y) tak, že $ver_k(x, y) = true$, hovoríme o falzifikáte podpisu („forgery"), tj. o platnom podpise vytvorenom iným než Alicou.

Zjednodušená RSA-podpisová schéma by vyzerala nasledovne:

$$\mathcal{P} = \mathcal{A} = Z_{pq}, \mathcal{K} = \{(n, e, d); ed \equiv 1 \pmod{\varphi(n)}\}.$$

$$sig_k(x) \equiv x^d \pmod{n}, ver_k(x, y) = true \Leftrightarrow x \equiv y^e \pmod{n}.$$

Útoky na podpisové schémy:

útok iba s verejným kľúčom — „key-only attack" — Oskar má k dispozícii Alicin verejný kľúč, tj. môže počítať ver_k.

Útoky na podpisové schémy:

útok iba s verejným kľúčom — „key-only attack" — Oskar má k dispozícii Alicin verejný kľúč, tj. môže počítať ver_k.

ú. so známymi správami — „known message a." — Oskar má k dispozícii niekoľko pravých podpisov od Alice $y_i = sig_K(x_i), i = 1, 2 \ldots, r$ (nemôže ovplyvniť x_i).

Útoky na podpisové schémy:

útok iba s verejným kľúčom — „key-only attack" — Oskar má k dispozícii Alicin verejný kľúč, tj. môže počítať ver_k.

ú. so známymi správami — „known message a." — Oskar má k dispozícii niekoľko pravých podpisov od Alice $y_i = sig_K(x_i), i = 1, 2 \ldots, r$ (nemôže ovplyvniť x_i).

ú. s vybranými správami — „chosen message a." — Oskar má k dispozícii niekoľko pravých podpisov od Alice $y_i = sig_K(x_i), i = 1, 2 \ldots, r$ (môže ovplyvniť x_i).

Cieľ útoku na podpisové schémy:

totálne prelomenie — „total break" — Oskar je schopný určiť Alicin tajný kľúč, tj. počítať hodnoty funkcie sig_k. To znamená, že je schopný vytvoriť platný Alicin podpis.

Cieľ útoku na podpisové schémy:

totálne prelomenie — „total break" — Oskar je schopný určiť Alicin tajný kľúč, tj. počítať hodnoty funkcie sig_k. To znamená, že je schopný vytvoriť platný Alicin podpis.

falzifikát vybranej správy — „selective forgery" — Oskar je schopný ku danej správe x vytvoriť novú správu x' a podpis y tak, že s veľkou pravdepodobnosťou $ver_k(x', y) = true$.

Cieľ útoku na podpisové schémy:

totálne prelomenie — „total break" — Oskar je schopný určiť Alicin tajný kľúč, tj. počítať hodnoty funkcie sig_k. To znamená, že je schopný vytvoriť platný Alicin podpis.

falzifikát vybranej správy — „selective forgery" — Oskar je schopný ku danej správe x vytvoriť novú správu x' a podpis y tak, že s veľkou pravdepodobnosťou $ver_k(x', y) = true$.

existencia falzifikátu — „existential forgery" — Oskar je schopný na základe kontroly komunikácie vytvoriť aspoň jednu správu x a jej podpis y tak, že $ver_k(x, y) = true$.

K schéme existuje jednoduchá možnosť vytvorenia falzifikátu podpisu:

1. Oskar zvolí ľubovoľné $y \in \mathcal{A} = Z_n$ a vypočíta $x \equiv y^e \pmod{n}$.

K schéme existuje jednoduchá možnosť vytvorenia falzifikátu podpisu:

1. Oskar zvolí ľubovoľné $y \in \mathcal{A} = Z_n$ a vypočíta $x \equiv y^e \pmod{n}$.
2. Potom $sig_k(x) \equiv (y^e)^d \equiv y \pmod{n}$, tj. pozná podpis bez znalosti d.

K schéme existuje jednoduchá možnosť vytvorenia falzifikátu podpisu:

1. Oskar zvolí ľubovoľné $y \in \mathcal{A} = Z_n$ a vypočíta $x \equiv y^e \pmod{n}$.
2. Potom $sig_k(x) \equiv (y^e)^d \equiv y \pmod{n}$, tj. pozná podpis bez znalosti d.
3. Verifikácia $ver_k(x, y) = ver_k(y^e, y) = true$ prebehne úspešne, lebo platí: $(y^e)^d \equiv y \pmod{n}$.

Tento prípad ukazuje *existencia falzifikátu pri útoku iba s verejným kľúčom*.

Ak vytvoríme vďaka multiplikatívnej vlastnosti RSA z dvoch platných podpisov tretí $y_1 y_2 = sig_K(x_1 x_2)$ vytvoríme falzifikát *útokom so známymi správami*.

Ak vytvoríme vďaka multiplikatívnej vlastnosti RSA z dvoch platných podpisov tretí $y_1 y_2 = sig_K(x_1 x_2)$ vytvoríme falzifikát *útokom so známymi správami*.
Ak chce Oskar získať podpis pre správu x, môže nájsť dve iné správy x_1, x_2 tak, že $x \equiv x_1 x_2 \pmod{n}$, na ktoré získa od Alici dva platné podpisy y_1, y_2. Potom $y \equiv y_1 y_2 \pmod{n}$ je platný podpis pre správu x. To je príklad *falzifikátu vybranej správy* s použitím *útoku s vybranými správami*.

V prípade, že kombinujeme podpis so šifrovaním, ako je to v algoritme 1, preferuje sa podpísať správu pred šifrovaním. V opačnom prípade Oskar môže postupovať nasledovne:

1. Alica posiela správu x so svojim podpisom Bobovi:

$$z \equiv x^{e_B} \pmod{n_B}, y = sig_{d_A}(z) \equiv z^{d_A} \pmod{n_A}.$$

V prípade, že kombinujeme podpis so šifrovaním, ako je to v algoritme 1, preferuje sa podpísať správu pred šifrovaním. V opačnom prípade Oskar môže postupovať nasledovne:

1. Alica posiela správu x so svojim podpisom Bobovi:

$$z \equiv x^{e_B} \pmod{n_B}, y = sig_{d_A}(z) \equiv z^{d_A} \pmod{n_A}.$$

2. Oskar zachytí podpísanú správu (z, y), nahradí podpis y svojim vlastným

$$y' = sig_{d_O}(z) \equiv z^{d_O} \equiv x^{e_B d_O} \pmod{n_O}$$

a pošle Bobovi podpísanú správu (z, y').

V prípade, že kombinujeme podpis so šifrovaním, ako je to v algoritme 1, preferuje sa podpísať správu pred šifrovaním. V opačnom prípade Oskar môže postupovať nasledovne:

1. Alica posiela správu x so svojim podpisom Bobovi:

$$z \equiv x^{e_B} \pmod{n_B}, y = sig_{d_A}(z) \equiv z^{d_A} \pmod{n_A}.$$

2. Oskar zachytí podpísanú správu (z, y), nahradí podpis y svojim vlastným

$$y' = sig_{d_O}(z) \equiv z^{d_O} \equiv x^{e_B d_O} \pmod{n_O}$$

a pošle Bobovi podpísanú správu (z, y').

3. Bob overí $(y')^{e_O} \pmod{n_O} = x^{e_B} = z$, tj. $ver_{e_O}(z, y') = true$.

Podpisové schémy na báze DL

Algoritmus na elektronický podpis **DSA** je odvodený od ElGamalovho kryptosystému. Označenie *Digital Signature Algorithm* sa niekedy zamieňa s označením *Digital Signature Standard*, keďže je tento algoritmus štandardizovaný ako americký štandard FIPS186.

p, q, a g - doménové verejné parametre. Súkromný kľúč je parameter x a verejný kľúč je parameter y, náhodný parameter k, ktorý musí zostať tajný (po podpise sa zahodí).

1. p – prvočíselný modul veľkosti 1 024 bitov
2. q – prvočíselný deliteľ $p - 1$ veľkosti 160 bitov
3. g – prvok Z_p^* rádu q
4. x – náhodné číslo také, že $0 < x < q$
5. y – $y = g^x \mod p$

Generovanie podpisu
Podpis správy M je dvojica (r, s), kde:

$$r = (g^k \mod p) \mod q$$
$$s = (k^{-1}(\text{SHA3}(M) + xr)) \mod q$$

SHA3(M) označuje 160 bitový výstup z SHA-3, konvertovaný na celé číslo. V prípade, že $r = 0$ alebo $s = 0$, podpis neplatí a musí sa vytvoriť nový podpis, pri použití iného k.
Vygenerovaný podpis sa pripojí k správe a doručí adresátovi.

Overenie podpisu

Adresát okrem správy a podpisu potrebuje poznať aj doménové parametre p, q, g a odosielateľov verejný kľúč y.
Nech M', r' a s' sú prijaté verzie M, r a s. Adresát (overovateľ) v prvom rade skontroluje, či $0 < r' < q$ a $0 < s' < q$. Ak nie, správu zavrhne, inak vypočíta:

$$\begin{aligned} w &= (s')^{-1} \mod q \\ u_1 &= \mathrm{SHA3}(M') \cdot w \mod q \\ u_2 &= (r') \cdot w \mod q \\ v &= (g^{u_1} \cdot y^{u_2} \mod p) \mod q \end{aligned}$$

Ak $v = r'$, tak správa je overená a s pravdepodobnosťou blížiacou sa k istote bola poslaná osobou, ktorá pozná príslušné tajné x k verejnému y. V opačnom prípade správu zavrhneme ako falošnú.

Podpisové schémy na báze ECDL

V revízii štandardu FIPS 186-2 (a v ďalších štandardoch) je uvedená modifikácia ECDSA algoritmu DSA, ktorá využíva aritmetiku eliptických kriviek. Doménové parametre v prípade ECDSA tvorí vhodne zvolená eliptická krivka $\mathbb{E}$ definovaná nad konečným poľom F_q s charakteristikou p a jeden jej bod $G \in \mathbb{E}(F_q)$, ktorý je generátor dostatočne veľkej cyklickej podgrupy $\mathbb{E}(F_q)$ (s počtom bodov n).

Používateľka Alica generuje kľúčový pár nasledovne:

1. Zvolí náhodné číslo $x \in [1, n-1]$.
2. Vypočíta $Q = xG$
3. Verejný kľúč je Q a tajný kľúč je x.

Pri **podpise správy** m opäť volíme náhodné k, $1 \leq k \leq n-1$.
Potom postupujeme nasledovne:

1. Vypočítaj $R = kG = (x_1, y_1)$, $r = x_1 \mod n$.
2. Vypočítaj SHA3(m).
3. Vypočítaj $s = k^{-1}(SHA3(m) + xr) \mod n$.

Podpisom správy je opäť pár (r, s), kde r aj s musia byť nenulové.

Pri **overenie podpisu** počítame

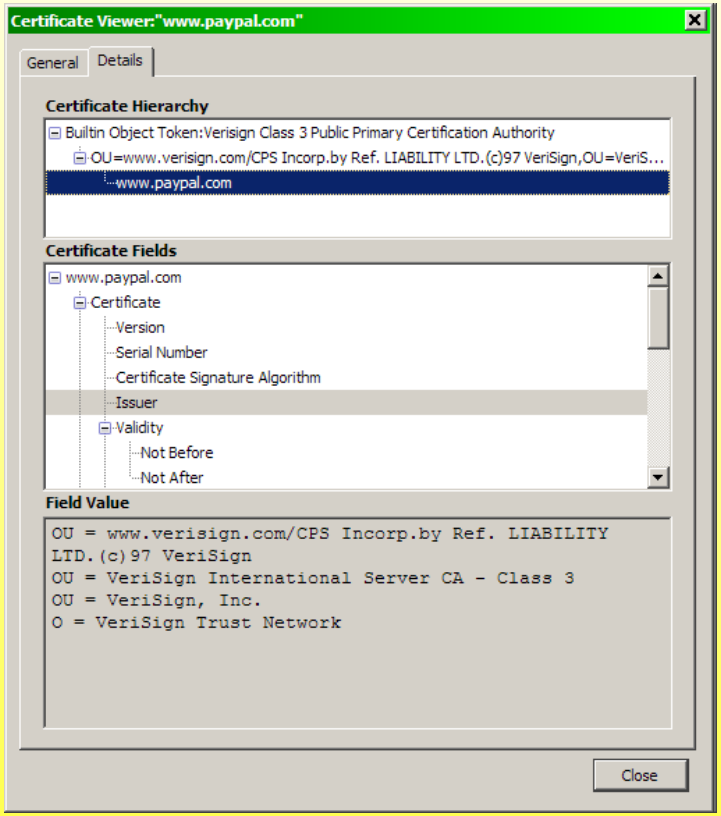
1. $w = s^{-1} \mod n$.
2. $u_1 = SHA3(\mathrm{m})w \mod n$
3. $u_2 = rw \mod n$.
4. $V = u_1 G + u_2 Q = (x_1, y_1)$.
5. Ak $V = \mathcal{O}$ zavrhni podpis. Inak $v = x_1 \mod n$.
6. Akceptuj podpis iba vtedy, keď $v = r$.

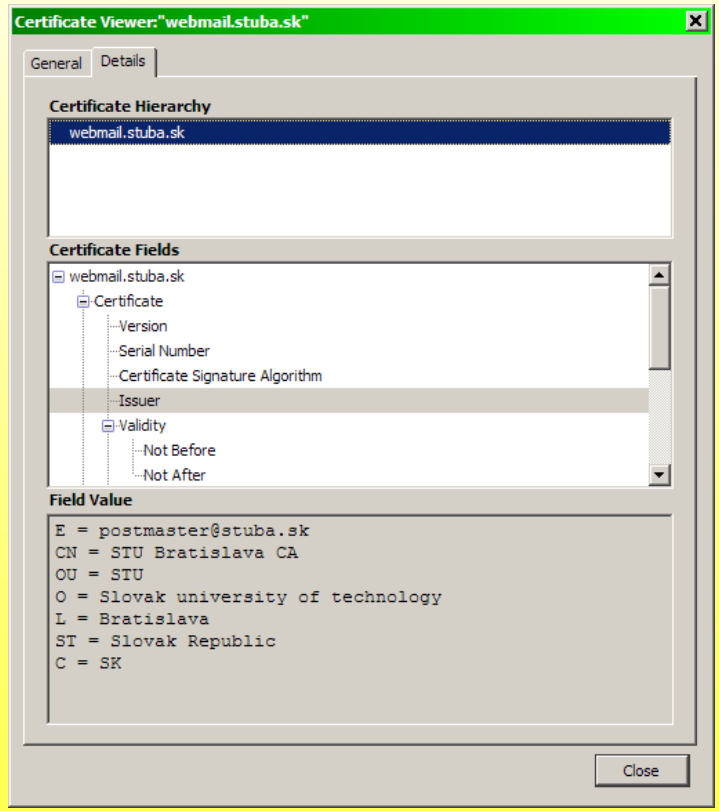
Vidíme, že generovanie a overenie podpisu je veľmi podobné DSA...

Autentifikácia a elektronický podpis v praxi

Elektronický podpis je založený na schéme s tromi základnými algoritmami — generovanie kľúčového páru, podpis správy a overenie podpisu správy.
Samotná schéma však nehovorí nič o identite osoby, ktorá kľúčový pár vytvorila.
Musíme použiť nejaký mechanizmus, ktorý aspoň do určitej miery dokáže previazať identitu osoby s kľúčovým párom. Jedno z riešení je napríklad také, že Alica po vygenerovaní kľúčového páru pošle všetkým ľuďom, s ktorými chce elektronicky komunikovať jej verejný kľúč, dôveryhodným kuriérom, resp. dá si poslať ich verejné kľúče.

Sústava PKI *Public Key Infrastructure* – infraštruktúra verejného kľúča. Jedná sa o sústavu mechanizmov, ktoré majú zabezpečiť používanie asymetrickej kryptografie v praxi, hlavne pre elektronický podpis, ale aj elektronické obchodovanie a podobne. V závislosti na rozsahu môžeme uvažovať o všeobecnej PKI na rôznych úrovniach – svetovej, národnej, či podnikovej.





Certifikát stránky zabezpečenej cez HTTPS. Certifikát vydaný svetovou certifikačnou autoritou. Všimnite si tri stupne v reťazi certifikátov. Napravo je certifikát ktorý si vyrobil sám majiteľ stránky. Prehliadač v tomto prípade varuje používateľa, lebo identitu stránky nevie potvrdiť.

Všetci používatelia PKI musia dôverovať CA a poznať jej verejný kľúč. Podobne ako existuje viac bánk, môže existovať aj viac CA. Verejný kľúč CA1 môže byť podpísaný inou CA2, a ak overujeme platnosť verejného kľúča E podpísaného CA1 a máme kľúč CA2 musíme postupne overovať všetky certifikáty v tzv. reťazi certifikátov.
V SR je cieľom národná sústava PKI s jedinou, tzv. koreňovou CA, ktorú spravuje štát cez NBÚ. Táto vystavuje certifikáty (a licencie) komerčným CA po preskúmaní ich dôveryhodnosti a splnení zákonných podmienok.

Sústava PKI musí riešiť aj problém, keď niekoho kľúč je kompromitovaný, tzn. útočník (pravdepodobne) získal súkromný kľúč. V tomto prípade je certifikát odvolaný a zverejnený v tzv. CRL — *Certificate Revocation List*. Pri overovaní podpisu je potrebné sa pozrieť, či certifikát, ktorý bol použitý, nie je zverejnený v CRL, pretože v tom prípade podpis neplatí.

Riešenie PGP

Keď Alica a Bob sú chudobní študenti a nechcú platiť zväčša drahé certifikáty a komerčné produkty na tvorbu elektronického podpisu, obrátia sa na otvorené riešenie — napríklad nekomerčnú verziu programu PGP, alebo jeho open-source klon GPG.

Distribúcia kľúčov prebieha na princípe priateľ môjho priateľa je môj priateľ – tranzitívna relácia. Podľa zverejnených výskumov dokážu ľubovoľný dvaja ľudia vo svete elektronicky nadviazať komunikáciu cez priemerne 6 medzičlánkov.

Schémy na zdieľanie tajomstva

Prahová schéma: Nech $t \leq w$ sú kladné čísla. $(t, w)-$ prahová schéma je rozdelenie kľúča k medzi w účastníkov tak, aby ľubovoľných t účastníkov vedelo vypočítať kľúč, ale žiadna podskupina $t-1$ účastníkov to nedokázala.
Distribúciu častí tajomstva bude vykonávať díler D.

Shamirova schéma

1. D zvolí w rôznych nenulových prvkov Z_p, označených $x_i, 1 \leq i \leq w$, a pridelí účastníkovi P_i číslo x_i.
2. Nech $k \in Z_p$ je zvolený kľúč. D zvolí $t-1$ prvkov Z_p, označených $a_i, 1 \leq i \leq t-1$.
3. Pre $1 \leq i \leq w$ D vypočíta $y_i = a(x_i)$, kde
$$a(x) = k + \sum_{j=1}^{t-1} a_j x^j \bmod p.$$
4. Pre $1 \leq i \leq w$, D pridelí časť y_i účastníkovi P_i.

Shamirova schéma

1. D zvolí w rôznych nenulových prvkov Z_p, označených $x_i, 1 \leq i \leq w$, a pridelí účastníkovi P_i číslo x_i.
2. Nech $k \in Z_p$ je zvolený kľúč. D zvolí $t-1$ prvkov Z_p, označených $a_i, 1 \leq i \leq t-1$.
3. Pre $1 \leq i \leq w$ D vypočíta $y_i = a(x_i)$, kde
$$a(x) = k + \sum_{j=1}^{t-1} a_j x^j \bmod p.$$
4. Pre $1 \leq i \leq w$, D pridelí časť y_i účastníkovi P_i.

Díler zvolil $a(x) = a_0 + a_1 x + \ldots + a_{t-1} x^{t-1}$, keď sa zíde t osôb, tak vedia určiť polynóm a $a_0 = k$. Ak ich bude menej, tak neurčia ten polynóm.

Shamirova schéma

1. D zvolí w rôznych nenulových prvkov Z_p, označených $x_i, 1 \leq i \leq w$, a pridelí účastníkovi P_i číslo x_i.
2. Nech $k \in Z_p$ je zvolený kľúč. D zvolí $t-1$ prvkov Z_p, označených $a_i, 1 \leq i \leq t-1$.
3. Pre $1 \leq i \leq w$ D vypočíta $y_i = a(x_i)$, kde
$$a(x) = k + \sum_{j=1}^{t-1} a_j x^j \bmod p.$$
4. Pre $1 \leq i \leq w$, D pridelí časť y_i účastníkovi P_i.

Díler zvolil $a(x) = a_0 + a_1 x + \ldots + a_{t-1} x^{t-1}$, keď sa zíde t osôb, tak vedia určiť polynóm a $a_0 = k$. Ak ich bude menej, tak neurčia ten polynóm.
Špeciálne (t,t) schéma: $y_i \in Z_m, 1 \leq i \leq t-1$ sú ľubovoľné a
$$y_t = k - \sum^{t-1} y_i \bmod m$$

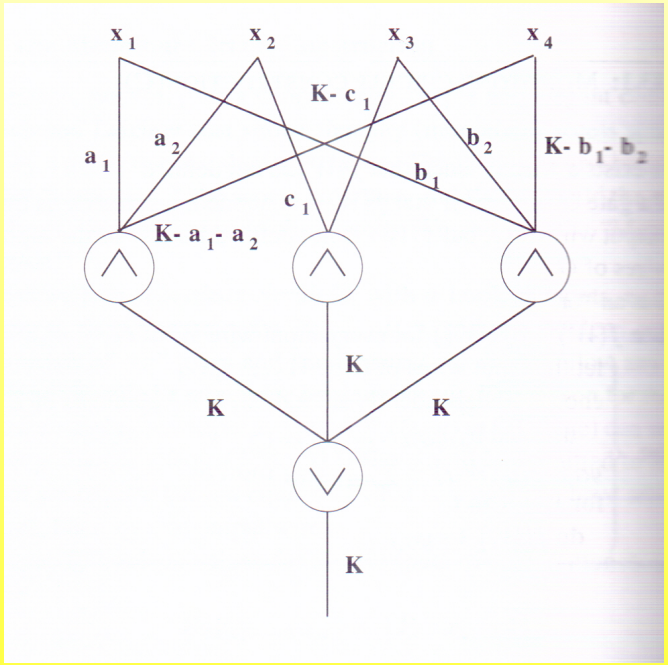
- $\mathcal{P}$ = množina účastníkov
- $A \subset \mathcal{P}$ = autorizovaná množina
- Γ = systém autorizovaných množín = PS
- K = tajomstvo = kľúč
- Schéma je perfektná ak
  1. každá $A \in \Gamma$ vie určiť K
  2. každá $B \notin \Gamma$ nevie povedať nič o K

- PS Γ má vlastnosť monotónnosti: $A \in \Gamma, A \subset C \subset \mathcal{P} \Rightarrow C \in \Gamma$
- Existujú minimálne prístupové množiny – báza pre Γ, ozn. Γ_0
- Príklad: $\mathcal{P} = \{P_1, P_2, P_3, P_4\}, \Gamma_0 = \{\{P_1, P_2, P_4\}, \{P_1, P_3, P_4\}, \{P_2, P_3\}\}$, potom

$$\Gamma = \Gamma_0 \cup \{\{P_1, P_2, P_3\}, \{P_2, P_3, P_4\}, \{P_1, P_2, P_3, P_4\}\}$$

- Realizácia cez obvody;

Príklad 13.4/489: $(P_1 \bigwedge P_2 \bigwedge P_4) \bigvee (P_1 \bigwedge P_3 \bigwedge P_4) \bigvee (P_2 \bigwedge P_3)$



P_1	a_1	b_1	
P_2	a_2		c_1
P_3		b_2	$K - c_1$
P_4	$K - a_1 - a_2$	$K - b_1 - b_2$	

- $D =$ dealer a jeho úloha
- $S =$ množina častí (shares) K
- $f : \mathcal{P} \to S, f(P_i) =$ časť tajomstva pridelená P_i
- Distribučné pravidlá f sú verejne známe
- $S(P_i) =$ množina častí K, ktoré môže dostať P_i pri rôznych f_K
- Prenosový pomer pre $P_i : \varrho_i = \log_2 |K| / \log_2 |S(P_i)|$
- Pre schému: $\varrho = \min \varrho_i$

f je perfektná, ak

1. pre každé $A \in \Gamma$ je $H(K/A) = 0$
2. pre každé $B \notin \Gamma$ je $H(K/B) = H(K)$
3. S pomocou entropie sa dokáže $\varrho \leq 2/3$.