Kryptografie založená na problému diskrétního logaritmu

Andrew Kozlík

KA MFF UK

Diffieho-Hellmanův protokol ustanovení klíče (1976)

- Alice a Bob chtějí mezi sebou dohodnout tajný klíč prostřednictvím veřejného komunikačního kanálu.
- Před zahájením protokolu se ustanoví veřejně známé parametry:
  - Konečná grupa $(G, \cdot)$, např. $\mathbb{Z}_p^*$, kde p je prvočíslo.
  - Prvek $g \in G$ řádu q.
- Protokol:

1.	Alice:	$a \in_{\mathcal{R}} \{0, 1, \ldots, q-1\}$,	$A := g^a$
	Bob:	$b \in_{\mathcal{R}} \{0, 1, \ldots, q-1\}$,	$B := g^b$
2.	Alice $\rightarrow$ Bob:	A	
	Bob $\rightarrow$ Alice:	B	
3.	Alice:	$k_1 := B^a$	
	Bob:	$k_2 := A^b$	

- Platí $k_1 = g^{ab} = k_2$.

Příklad

- Máme-li $G = \mathbb{Z}_{11}$ a $g = 2$, pak řád g je $q = 10$.
- Alice si náhodně zvolí tajnou hodnotu $a = 9$ a spočte $A = 2^9 = 6$.
- Bob si náhodně zvolí tajnou hodnotu $b = 8$ a spočte $B = 2^8 = 3$.
- Strany si vymění hodnoty A a B.
- Alice spočte $B^a = 3^9 = 4$.
- Bob spočte $A^b = 6^8 = 4$.

Bezpečnost

- Výměna hodnot A a B probíhá po kanálu, který
  - nemusí zajišťovat utajení, ale
  - musí zajišťovat autentizaci původců zpráv (systém by jinak byl zranitelný útokem typu man-in-the-middle).

- Cíl útočníka:
  Ze znalosti g, g^a a g^b určit g^{ab}.
  (Diffieho-Hellmanův problém).

- Problém diskrétního logaritmu (DL-problém):
  Ze znalosti $g \in G$ a g^s určit $s \in \{0, 1, \ldots, \mathrm{ord}(g) - 1\}$.

- Útočník, který umí efektivně řešit DL-problém, umí efektivně řešit Diffieho-Hellmanův problém.

K čemu je Diffie-Hellman dobrý?

- Alice chce poslat Bobovi zprávu x šifrovaně, ale nemá s ním dohodnutý klíč symetrické šifry.
- Alice zná Bobův veřejný RSA klíč k_v.
- Možnost první (RSA šifrování + AES):
  - Alice náhodně zvolí klíč symetrické šifry k.
  - Alice pošle Bobovi $(\mathrm{RSA}_{k_v}(k),\ \text{AES-CBC}_k(x))$.
  - Po skončení účastníci zapomenou k.
- Nevýhoda: Postrádá perfect forward secrecy.
  - Útočník odposlechne komunikaci.
  - Útočník v budoucnu ukradne Bobovi soukromý klíč k_s.
  - Útočník zpětně dešifruje k a následně x.

Perfect forward secrecy

- Možnost druhá (Diffie-Hellman + RSA podpis + AES):
  - Bob náhodně zvolí b, spočítá $B = g^b$ a podpis $\mathrm{RSA}_{k_\mathrm{s}}(B)$.
  - Bob pošle Alici $(B,\ \mathrm{RSA}_{k_\mathrm{s}}(B))$.
  - Alice ověří podpis.
  - Alice náhodně zvolí a, spočítá $A = g^a$ a spočítá $k = B^a$.
  - Alice pošle Bobovi $(A,\ \text{AES-CBC}_k(x))$.
  - Po skončení účastníci zapomenou a, b a k.
- Klíče a a b se generují pro každou relaci znovu.
  Označují se anglicky jako *ephemeral* (efemérní, prchlavé).
- Bob používá RSA klíče k_v a k_s dlouhodobě, jsou *statické*.
- Nevýhoda: Vyžaduje součinnost Boba.

Definice

Říkáme, že protokol ustanovení klíče zaručuje *perfect forward secrecy*, jestliže odhalení dlouhodobého klíče neohrozí utajení klíčů ustanovených v dřívějších relacích.

Volba grupy G a prvku g

- Používáme například $G = \mathbb{Z}_p^*$, kde p je prvočíslo.
- Typicky volíme p o velikosti alespoň 2048 bitů.
- Z bezpečnostních důvodů volíme prvočíslo p ve tvaru $p = qa + 1$, kde q je velké prvočíslo a a je celé číslo.
- Prvek g volíme tak, aby jeho řád byl roven q.
- Grupu generovanou prvkem g nazýváme *Schnorrova grupa*.
- Tato volba parametrů znemožňuje efektivní použití Pohligova-Hellmanova algoritmu k výpočtu
  - diskrétního logaritmu nebo
  - částečné informace o diskrétním logaritmu.

Sestrojení prvočísla p a prvku g

- Algoritmus pro sestrojení prvočísla p a prvku g řádu q:
  1. Zvol velké prvočíslo q náhodně.
  2. Zvol $a \in \mathbb{N}$ tak, aby $p := aq + 1$ bylo prvočíslo.
  3. Zvol $h \in \mathbb{Z}_p^*$ tak, aby $g := h^a \neq 1$.

- Podle čeho víme, že prvek g vrácený algoritmem je řádu q?
  - Podle malé Fermatovy věty je $g^q = h^{aq} = h^{p-1} = 1$.
  - Řád prvku g tedy dělí q.
  - Dělitelé q jsou pouze 1 a q, přičemž g řádu 1 jsme vyloučili.

Princip Pohligova-Hellmanova algoritmu

- Nechť $p = 31$ a $g = 3$.
- Máme $\varphi(p) = 30 = 2 \cdot 3 \cdot 5 = p_1 \cdot p_2 \cdot p_3$.
- V $\mathbb{Z}_p^*$ lze sestrojit následující tabulku:

s	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	29
g^s	1	3	9	27	19	26	16	17	20	29	25	13	8	24	10	30	...	21
$(g^s)^{15}$	1	30	1	30	1	30	1	30	1	30	1	30	1	30	1	30	...	30
$(g^s)^{10}$	1	25	5	1	25	5	1	25	5	1	25	5	1	25	5	1	...	5
$(g^s)^6$	1	16	8	4	2	1	16	8	4	2	1	16	8	4	2	1	...	2

- To, že poslední 3 řádky tabulky vykazují periodické chování, není náhoda.
- Zapíšeme-li totiž s ve tvaru $s = q_i p_i + r_i$, kde $r_i = s \bmod p_i$, pak

$$(g^s)^{\varphi(p)/p_i} = g^{(q_i p_i + r_i)\varphi(p)/p_i} = g^{q_i \varphi(p) + r_i \varphi(p)/p_i} = (g^{r_i})^{\varphi(p)/p_i}.$$

Princip Pohligova-Hellmanova algoritmu

- Vyřešte $g^s = 23$. [Hrubou silou 30 možností pro s.]
- Vyřešíme 3 jednodušší úlohy
  - $23^{15} = 30 = (g^{15})^{r_1} \rightarrow r_1 = 1,$ [2 možnosti pro r_1]
  - $23^{10} = 1 = (g^{10})^{r_2} \rightarrow r_2 = 0,$ [3 možnosti pro r_2]
  - $23^{6} = 8 = (g^{6})^{r_3} \rightarrow r_3 = 2.$ [5 možností pro r_3]
- Potom vyřešíme soustavu kongruencí:

$$\left.\begin{array}{ll} s \equiv 1 & (\mathrm{mod}\ 2) \\ s \equiv 0 & (\mathrm{mod}\ 3) \\ s \equiv 2 & (\mathrm{mod}\ 5) \end{array}\right\} \quad \rightarrow \quad s \equiv 27 \quad (\mathrm{mod}\ 30)$$

- Poučení: Bezpečnost diskrétního logaritmu v grupě je určena nejmenším prvočinitelem jejího řádu.

ElGamalův šifrovací systém (1985)

- Generování klíčů:
  1. Zvol velké prvočíslo p, generátor g grupy $\mathbb{Z}_p^*$ a náhodnou tajnou hodnotu $s \in \mathbb{Z}_{p-1}$.
  2. Spočti $v = g^s \bmod p$.
  3. Označ $k_s = (s, p, g)$ a $k_v = (v, p, g)$.
- Šifrování zprávy x:
  1. Zvol tajnou nonci $r \in \mathbb{Z}_{p-1}$.
  2. Spočti $y_1 = g^r \bmod p$ a $y_2 = x v^r \bmod p$.
  3. Šifrový text je $y = (y_1, y_2)$.
- Dešifrování:
  - Spočti $x = y_2 {y_1}^{-s}$.
- Důkaz správnosti:

$$y_2 {y_1}^{-s} \equiv (x v^r)(g^r)^{-s} \equiv x g^{sr} g^{-rs} \equiv x \pmod{p}.$$

- Použije-li se k šifrování dvou zpráv x_1 a x_2 stejná hodnota r, pak lze z šifrových textů odhalit $x_1 {x_2}^{-1} \pmod{p}$.

ElGamalovo podpisové schéma (1985)

- Generování klíčů:
  1. Zvol velké prvočíslo p, generátor g grupy $\mathbb{Z}_p^*$ a náhodnou tajnou hodnotu $s \in \mathbb{Z}_{p-1}$.
  2. Spočti $v = g^s \bmod p$.
  3. Označ $k_s = (s, p, g)$ a $k_v = (v, p, g)$.
- Podepsání zprávy:
  1. Spočti otisk zprávy $h = \text{hash}(x) \in \mathbb{Z}_{p-1}$.
  2. Zvol tajnou nonci $r \in \mathbb{Z}_{p-1}^*$.
  3. Spočti $y_1 = g^r \bmod p$ a $y_2 = (h - sy_1)r^{-1} \bmod (p-1)$.
  4. Pokud $y_2 = 0$, vrať se ke kroku 1.
  5. Podpis je $y = (y_1, y_2)$.
- Ověření podpisu zprávy:
  - Ověř, že $v^{y_1} y_1^{y_2} \equiv g^h \pmod p$ a $1 \leq y_1 < p$.
- Důkaz úplnosti:

$$v^{y_1} y_1^{y_2} \equiv (g^s)^{y_1} (g^r)^{(h-sy_1)r^{-1}} \equiv g^{sy_1+h-sy_1} \equiv g^h \pmod p.$$

Bezpečnost ElGamalova podpisového schématu

- Použije-li se schéma bez hashovací funkce, pak lze docílit existenčního podvržení podpisu:
  - Zvol $a \in \mathbb{Z}_p$ libovolně.
  - Spočti $y_1 = g^a v \bmod p$
  - Spočti $y_2 = -y_1 \bmod (p-1)$
  - Potom (y_1, y_2) je podpis zprávy $x = a y_2 \bmod (p-1)$.

- Z podepsané zprávy a z hodnoty r lze s vysokou pravděpodobností vypočítat soukromý klíč.

- Použije-li se k podpisu dvou různých zpráv stejná hodnota r, pak lze z podpisů odhalit soukromý klíč.

- Proti ověřovateli, který neověří, že $1 \leq y_1 < p$, lze provést univerzální podvržení podpisu.

Další schémata založená na DL-problému

- Spousta schémat využívá Schnorrovu grupu, např.:
  - Schnorrovo identifikační schéma,
  - Schnorrovo podpisové schéma,
  - Digital Signature Algorithm (DSA).
- Nechť p a q jsou velká prvočísla, kde $q \mid p - 1$.
- Typicky např. p velikosti 2048 bitů a q velikosti 224 bitů.
- Nechť $g \in \mathbb{Z}_p^*$ je prvek řádu q.
- Grupu generovanou prvkem g nazýváme *Schnorrova grupa*.
- Generování klíčů:
  1. Zvol náhodnou tajnou hodnotu $s \in \mathbb{Z}_q^*$.
  2. Spočti $v = g^s \bmod p$.
  3. Označ $k_s = (s, p, q, g)$ a $k_v = (v, p, q, g)$.

Schnorrovo identifikační schéma (1989)

- Alice chce prokázat svou totožnost Bobovi.
- Bob zná veřejný klíč Alice $k_v = (v, p, q, g)$.

- Protokol:

1.	A:	Zvolí tajnou nonci $r \in \mathbb{Z}_q^*$.	
	A:	Spočte $R := g^r \bmod p$.	
	A $\to$ B:	Pošle R.	[závazek]
2.	B:	Zvolí nepředvídatelně náhodně $e \in \mathbb{Z}_q^*$.	
	B $\to$ A:	Pošle e.	[výzva]
3.	A:	Spočte $y := (r - se) \bmod q$.	
	A $\to$ B:	Pošle y.	[odpověď]
4.	B:	Ověří, že $v^e g^y \bmod p = R$.	[ověření]

- Důkaz úplnosti:

$$v^e g^y \equiv (g^s)^e g^{r-se} \equiv g^r \equiv R \pmod{p}$$

Bezpečnost Schnorrova identifikačního schématu

Pozorování:

- Pokud útočník předem ví, jakou hodnotu e Bob zvolí, tak se může vůči Bobovi vydávat za Alici:
  - Útočník zvolí libovolné $y \in \mathbb{Z}_q^*$ a použije $R = v^e g^y \bmod p$.
- Pokud útočník odposlechne komunikaci a zároveň odhalí r, pak může z y a e vypočítat soukromý klíč s:
  - $s \equiv (r - y)e^{-1} \pmod{q}$
- Pokud Alice použije ve dvou instancích protokolu stejnou hodnotu r, pak lze vypočítat její soukromý klíč s.
  - Útočník vyřeší soustavu dvou rovnic o dvou neznámých r a s nad tělesem $\mathbb{Z}_q$:

$$y \equiv r - se \pmod{q}$$
$$y' \equiv r - se' \pmod{q}$$

Bezpečnost Schnorrova identifikačního schématu

Útočník, který zná algoritmus, jímž se umí efektivně vydávat za libovolnou osobu, umí stejně efektivně řešit DL-problém:

- Mějme útočníka, který zná k_v Alice, ale nezná k_s.
- Předpokládejme, že existuje efektivní algoritmus, kterým:
  - útočník zvolí závazek R a
  - na výzvu e určí správnou odpověď y,

  aby se mohl vydávat za Alici.
- Tento algoritmus může útočník použít tak, že
  - nechá algoritmus vygenerovat R,
  - uloží stav algoritmu do proměnné S,
  - nechá algoritmus spočítat odpověď y na výzvu $e = 1$,
  - vrátí algoritmus do stavu S
  - nechá algoritmus spočítat odpověď y' na výzvu $e' = 2$,
  - určí s vyřešením soustavy rovnic (viz předcházející slajd).

Schnorrovo podpisové schéma (1989)

- Jedná se o úpravu identifikačního schématu, kde výzva e je simulována hashem závazku R a zprávy x.
- Existují dvě varianty podpisu, průběh výpočtu je ale shodný.
- Podepsání zprávy:
  1. Zvol tajnou nonci $r \in \mathbb{Z}_q^*$ a spočti $R = g^r \bmod p$.
  2. Spočti $e = \text{hash}(R \| x) \in \mathbb{Z}_q$.
  3. Spočti $y = (r - se) \bmod q$.
  4. (a) Podpis je (e, y).
     (b) Podpis je (R, y).
- Ověření podpisu zprávy:

  (a) Ověř, že $\text{hash}\big((v^e g^y \bmod p) \,\big\|\, x\big) = e$.

  (b) Ověř, že $v^{\text{hash}(R\|x)} g^y \bmod p = R$.

  V obou případech se ověřuje, že $v^e g^y \bmod p = R$ a $e = \text{hash}(R \| x)$.

Digital Signature Algorithm (DSA, 1991)

- Podepsání zprávy:
  1. Spočti otisk $h = \text{hash}(x)$.
  2. Pokud je h delší než $\log_2 q$ bitů, odřež bity vpravo.
  3. Zvol tajnou nonci $r \in \mathbb{Z}_q^*$.
  4. Spočti $y_1 = (g^r \bmod p) \bmod q$.
  5. Spočti $y_2 = (h + s y_1) r^{-1} \bmod q$.
  6. Pokud $y_1 = 0$ nebo $y_2 = 0$, vrať se ke kroku 3.
  7. Podpis je $y = (y_1, y_2)$.
- Ověření podpisu zprávy:
  1. Ověř, že $1 \leq y_1 < q$ a $1 \leq y_2 < q$.
  2. Označ y_2' inverzní prvek k y_2 v $\mathbb{Z}_q^*$.
  3. Ověř, že $((g^h v^{y_1})^{y_2'} \bmod p) \bmod q = y_1$.
- Důkaz úplnosti:

$$((g^h v^{y_1})^{y_2'} \bmod p) \bmod q = ((g^h g^{s y_1})^{y_2'} \bmod p) \bmod q =$$
$$(g^{(h+s y_1) y_2'} \bmod p) \bmod q = (g^r \bmod p) \bmod q = y_1$$

Vlastnosti DSA – cvičení

- Ukažte, že z podepsané zprávy a z hodnoty r lze vypočítat soukromý klíč.
- Ukažte, že použije-li se k podpisu dvou různých zpráv stejná hodnota r, pak lze z podpisů odhalit soukromý klíč.
- Ukažte, že použije-li se schéma bez hashovací funkce, tj. $h = x$, pak lze provést existenční podvržení podpisu zvolením $y_1 = (g^a v^b \bmod p) \bmod q$ pro libovolné $a \in \mathbb{Z}_q$ a $b \in \mathbb{Z}_q^*$ takové, že $y_1 \neq 0$. (Stačí dopočítat h a y_2.)
- Ukažte, že lze provést selektivní podvržení podpisu pro zprávu x takovou, že $\text{hash}(x) = 0$.

Vlastnosti DSA – cvičení

- Ukažte, že nekontroluje-li ověřovatel podmínky $1 \leq y_1 < q$ a $1 \leq y_2 < q$, pak lze vůči němu provést univerzální podvržení podpisu na základě znalosti k_v.

  - Pokud ověřovatel nekontroluje první podmínku, pak zvolíme libovolné $y_2 \in \mathbb{Z}_q^*$ a dopočítáme y_1 tak, aby
    1. $y_1 \equiv 0 \pmod{q}$ a
    2. $y_1 \equiv g^{h y_2'} \pmod{p}$.

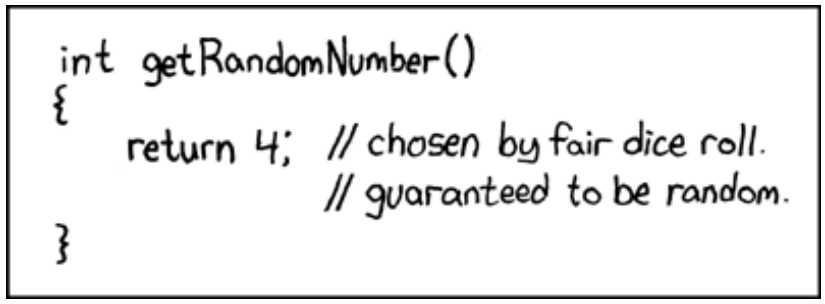
    Takové y_1 existuje podle Čínské věty o zbytcích.

  - Jestliže y_2 není invertibilní modulo q, pak některé algoritmy pro počítání inverzu mohou vrátit $y_2' = 0$.

  - Pokud ověřovatel nekontroluje druhou podmínku a používá takový algoritmus, tak zvolíme $y_1 = 1$ a $y_2 \equiv 0 \pmod{q}$.

Generování tajné nonce

- Všechna uvedená schémata podpisu vyžadují vygenerování tajné nonce r pro každou podepisovanou zprávu.
- V roce 2010 firmě Sony unikl soukromý klíč pro podepisování spustitelných souborů pro PlayStation3, protože při podepisování opakovaně použili stejnou nonci r.



Zdroj: `https://xkcd.com/221/`

- Nonci je dobré generovat deterministicky.
  - Pro DSA například algoritmus popsaný v RFC 6979.
  - Vstupem algoritmu je soukromý klíč k_s a zpráva x.
  - Není třeba spoléhat na generátor náhodných čísel.
  - Kdokoliv může ověřit, že podepisovací zařízení se chová správně, tj. např. že výrobce nezabudoval zadní vrátka.

Úroveň bezpečnosti

V závislosti na požadované úrovni bezpečnosti je doporučeno volit následující velikosti parametrů (v bitech):

Klíč symetrické šifry	Délka hashe	DH, ElGamal, DSA, Schnorr		RSA
		p	q	N
80	160	1024	160	1024
112	224	2048	224	2048
128	256	3072	256	3072
192	384	7680	384	7680
256	512	15360	512	15360

Zdroj: NIST Special Publication 800-57 Part 1 Revision 4

Srovnání délky podpisu

- ElGamal: $2 \log_2 p$ bitů
- Schnorr (a): $2 \log_2 q$ bitů
- Schnorr (b): $\log_2 p + \log_2 q$ bitů
  - Výhodou je, že existuje efektivní algoritmus pro hromadné ověřování podpisů typu (b).
- DSA: $2 \log_2 q$ bitů
- RSA: $\log_2 N$ bitů

Srovnání složitosti výpočtu a ověření podpisu

- Průměrný počet operací násobení modulo p, resp. N, zanedbáme-li malé aditivní konstanty a hashování:
  - ElGamal podepsání: $1{,}5 \cdot \log_2 p$
  - ElGamal ověření: $2 \cdot 1{,}5 \cdot \log_2 p + 1{,}5 \cdot \log_2 q$
  - Schnorr/DSA podepsání: $1{,}5 \cdot \log_2 q$
  - Schnorr/DSA ověření: $2 \cdot 1{,}5 \cdot \log_2 q$
  - RSA podepsání: $1{,}5 \cdot \log_2 N$
  - RSA ověření: 17
- Srovnání pro 128bitovou úroveň bezpečnosti:

Typ	Délka podpisu (bajty)	Složitost podepsání	Složitost ověření
ElGamal	768	4608	9600
Schnorr (a)	64	384	768
Schnorr (b)	416	384	768
DSA	64	384	768
RSA	384	4608	17

Slepý podpis založený na Schnorrově schématu

- Vytvoření slepého DSA podpisu je poměrně složité.
- U Schnorrova podpisového schématu to jde o dost snadněji:
  1. Požádáme podepisujícího, aby si zvolil tajnou nonci $r \in \mathbb{Z}_q^*$ a aby nám poskytl $R' = g^r \bmod p$.
  2. Zvolíme tajné α, $\beta \in \mathbb{Z}_q$ a spočteme:
     $R = R'\, g^{\alpha} v^{\beta} \bmod p$,
     $e = \mathrm{hash}(R \| x) \in \mathbb{Z}_q$,
     $e' = e - \beta \bmod q$.
  3. Podepisujícímu pošleme e'.
  4. Podepisující nám vrátí y' takové, že $v^{e'} g^{y'} \bmod p = R'$.
  5. Dopočítáme $y = y' + \alpha \bmod q$.
  6. Potom (e, y), resp. (R, y), je platný podpis zprávy x.
- Pro ověření správnosti stačí ověřit, že $v^e g^y \bmod p = R$:

$$v^e g^y \equiv v^{e'+\beta} g^{y'+\alpha} \equiv v^{e'} g^{y'} g^{\alpha} v^{\beta} \equiv R'\, g^{\alpha} v^{\beta} \equiv R \bmod p.$$

Lamportův jednorázový podpis

- Generování klíčů:
  1. Zvol 512 tajných 256-bitových hodnot $s_{i,j} \in \{0,1\}^{256}$, kde $i = 0, \ldots, 255$ a $j = 0, 1$.
  2. Spočti $v_{i,j} = \text{hash}(s_{i,j})$ pro všecna i a j.
  3. Vrať veřejný klíč $(v_{0,0}, v_{0,1}, \ldots, v_{255,0}, v_{255,1})$.
- Podepsání zprávy:
  1. Spočti otisk zprávy $h = \text{hash}(x)$.
  2. Vrať $(y_0, \ldots, y_{255}) = (s_{0,h_0}, \ldots, s_{255,h_{255}})$, kde h_i je i-tý bit otisku zprávy.
- Ověření podpisu zprávy:
  - Ověř, že $\text{hash}(y_i) = v_{i,h_i}$ pro $i = 0, \ldots, 255$.
- Problémy
  - Lze bezpečně podepsat jen jednu zprávu.
  - Veřejný klíč je obrovský, $512 \times 32 = 16384$ bajtů.
  - Podpis je obrovský, $256 \times 32 = 8192$ bajtů.