Pokročilá kryptologie

Asymetrická kryptografie

prof. Ing. Róbert Lórencz, CSc., Ing. Josef Kokeš

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra informační bezpečnosti

Obsah přednášky

- RSA
- Principy kryptografie s veřejným klíčem (VK)
- Kryptografické systémy s VK
- El Gamal
- Digitální podpis El Gamal
- Srovnání RSA a El Gamal

RSA (1)

Úvod

- Zabezpečení utajené komunikace v síti $\Rightarrow$ každá komunikující dvojice musí používat šifrovací klíč
- Pokud je šifrovací klíč známý je dešifrovací klíč vygenerovatelný s použitím malého počtu operací.
- Šifrovací systém veřejného klíče (VK) je řešení problému s přidělovaním klíče pro utajenou komunikaci.
- Šifrovací systém VK má šifrovací klíč veřejný VK a tajný SK.
  - Vypočítat dešifrovací transformaci ze šifrovací je problém.
  - Použitím VK je zřízena utajená komunikace v síti s několika subjekty.
  - Každý subjekt má VK a SK pro daný šifrovací systém.
  - Subjekt si ponechává určité utajené soukromé informace vnesené do konstrukce šifrovací transformace pomocí SK.
- Seznam klíčů $VK_1, VK_2, \ldots, VK_n$ je veřejný.

RSA (2)

- Subjekt 1 vysílá zprávu *m* subjektu 2:
  - Zpráva $\rightarrow$ blok (obvykle 1) určité délky; bloku OT *m* odpovídá blok ŠT, písmena $\rightarrow$ numerické ekvivaleny.
  - Subjekt 2 s použitím dešifrovací transformace dešifruje blok ŠT.
- Podmínka: dešifrovací transformace nemůže být nalezena v rozumném čase někým jiným než subjektem 2 $\Rightarrow$ neautorizované subjekty komunikace nemůžou dešifrovat zprávu bez znalosti klíče.

Princip RSA šifrovacího systému

- Uveden Rivestem, Shamirem a Adlemanem v roce 1970.
- RSA je šifrovací systém VK a je založený na modulárním umocňování.
- Dvojice (e, n) je *VK* klíče; e - exponent a n - modul.
- $n \rightarrow$ součin dvou prvočísel p a q, tj. $n = pq$ a $\gcd(e, \Phi(n)) = 1$.

RSA (3)

- Zašifrování OT: písmena $\rightarrow$ numerické ekvivalenty, vytváříme bloky s největší možnou velikostí (se sudým počtem číslic).
- Pro zašifrování zprávy m na ŠT c použijeme vztah:

$$E(m) = c = |m^e|_n, \quad 0 < c < n.$$

- K dešifrování požadujeme znalost inverze d čísla e modulo $\Phi(n)$, $\gcd(e, \Phi(n)) = 1 \Rightarrow$ inverze existuje. Pro dešifrování bloku c platí:

$$D(c) = |c^d|_n = |m^{ed}|_n = |m^{k\Phi(n)+1}|_n = |(m^{\Phi(n)})^k m|_n = |m|_n,$$

kde $ed = k\Phi(n) + 1$ pro nějaké celé číslo k ($|ed|_{\Phi(n)} = 1$) a z Eulerovy věty platí $|m^{\Phi(n)}|_n = 1$, kde $\gcd(m, n) = 1$.

Pravděpodobnost, že m a n nejsou nesoudělná je extrémně malá. Ale co se stane, když m a n jsou soudělná?!

RSA (4)

Důkaz

- Platí, že $\gcd(m, n) = \gcd(m, pq) \neq 1$
- Protože p a q jsou prvočísla platí buď:

  $$m = \alpha p \text{ nebo } m = \beta q,$$

  kde α a β jsou celá čísla, pro které platí $\alpha < q$ a $\beta < p$.
- Předpokladejme, že $m = \alpha p$ a teda $\gcd(m, \beta) = 1$. Dále platí

  $$1 \equiv 1^k \equiv (m^{\Phi(q)})^k \pmod{q},$$

  kde k je celé kladné číslo.
- Výraz $(m^{\Phi(n)})^k$ můžeme potom psát:

  $$(m^{\Phi(n)})^k \equiv (m^{(p-1)(q-1)})^k \equiv ((m^{\Phi(q)})^k)^{(p-1)} \equiv 1^{(p-1)} \equiv 1 \pmod{q}$$
- Potom $(m^{\Phi(n)})^k = 1 + \gamma q$, kde γ je celé číslo. Vynásobením této rovnice hodnotou m dostávame:

  $$m(m^{\Phi(n)})^k = m + m\gamma q = m + (\alpha p)\gamma q = m + \alpha\gamma(pq) = m + \alpha\gamma n$$

RSA (5)

Důkaz 2

- Z toho plyne

$$m(m^{\Phi(n)})^k \equiv m \pmod{n} \text{ a platí}$$

$$D(c) = |c^d|_n = |m^{ed}|_n = |m^{k\Phi(n)+1}|_n = |(m^{\Phi(n)})^k m|_n = |m|_n,$$

Square-and-Multiply algoritmus pro modulární umocňování

Vstup: základní element m, exponent $H = \sum_{i=0}^{t} h_i 2^i$, kde $h_i \in \{1, 0\}$ a $h_t = 1$ a modul n.
Výstup: $m^H \bmod n$
Inicializace: $r = m$
Algoritmus:

1. FOR $i = t - 1$ DOWNTO 0
   1. $r = r^2 \bmod n$
   2. IF $h_i = 1$ THEN $r = rm \bmod n$
2. RETURN (r)

RSA (6)

Příklad

- Šifrovací modul je součinem dvou prvočísel 43 a 59. Potom dostáváme $n = 43 \cdot 59 = 2537$ jako modul.
- $e = 13$ je exponent, kde platí $\gcd(e, \Phi(n)) = \gcd(13, 42 \cdot 58) = 1$.
- Dále plati $\Phi(2537) = (43 - 1) \cdot (59 - 1) = 42 \cdot 58 = 2436$.
- Pro zašifrování zprávy

  PUBLIC KEY CRYPTOGRAPHY,
- převedeme OT do číselných ekvivalentů písmen textu $\Rightarrow$ vytvoříme bloky o délce 4 číslic (n je 4ciferné!) a dostáváme:

  1520 0111 0802 1004 2402 1724 1519 1406 1700 1507 2423,

  Písmeno X = 23 je výplň (padding).
- Pro šifrování bloku OT do bloku ŠT použijme vztah $c = |m^{13}|_{2537}$. Šifrovaním prvního bloku OT 1520 dostáváme blok ŠT

$$c = |(1520)^{13}|_{2537} = 95.$$

RSA (7)

- Zašifrováním všech bloků OT dostáváme:
  0095 1648 1410 1299 0811 2333 2132 0370 1185 1457 1084.
- Pro dešifrování zprávy, která byla zašifrována RSA šifrou, musíme najít inverzi $e = |13^{-1}|_{\Phi(n)}$, kde $\Phi(n) = \Phi(2537) = 2436$.
- S použitím Euklidova algoritmu získáme číslo $d = 937$, které je multiplikativní inverzí čísla 13 modulo 2436.
- K dešifrování bloku c ŠT použijeme vztah:

$$m = |c^{937}|_{2537}, \;\; 0 \leq m \leq 2537,$$

který platí, protože

$$|c^{937}|_{2537} = |(m^{13})^{937}|_{2537} = |m \cdot (m^{2436})^5|_{2537} = m,$$

kde jsme použili Eulerovu větu

$$|m^{\Phi(2537)}|_{2537} = |m^{2436}|_{2537} = 1,$$

když platí $\gcd(m, 2537) = 1$, a to je splněno pro každý blok/zprávu m OT.

RSA (8)

RSA - definece systému klíčů

Výstup: veřejný klíč VK = (n, e), exponent soukromého klíče = (d)

1. Vyhledáme dva velké prvočísla p a q.
2. Vypočítáme $n = pq$ a $\Phi(n) = (p-1)(q-1)$.
3. Zvolíme veřejný exponent $e \in \{1, 2, \ldots \Phi(n) - 1\}$takové, že platí
$$\gcd(e, \Phi(n)) = 1$$
4. Spočítáme SK d tak, že
$$de \equiv \pmod{\Phi(n)}.$$

Dvojici $VK = (n, e)$ prohlásíme za veřejný klíč (a zveřejníme), dvojici $SK = (n, d)$ prohlásíme za soukromý klíč.

Podmínka $\gcd(e, \Phi(n)) = 1$ zaručuje, že inverze e modulo $\Phi(n)$ existuje a je to číslo d exponent privátní části klíče.

RSA (9)

Exponent d můžeme vypočítat použitím EEA s použitím vstupních hodnot n a e kde platí vztah:

$$\gcd(\Phi(n), e) = s\Phi(n) + te.$$

V případě, že $\gcd(\Phi(n), e) = 1$ víme, že e je platný veřejný klíč. Také víme, že parametr t je vypočítaný pomocí EEA vyjadřuje inverzi e, pro kterou platí

$$d = t \bmod \Phi(n).$$

Pokud e a $\Phi(n)$ jsou soudělná čísla zvolíme nové e a výpočet $\gcd(\Phi(n), e)$ opakujeme.

RSA (10)

Postup pro genrování *VK* a *SK*:

- Každý subjekt najde 2 velká náhodná prvočísla p a q se 100 dekadickými číslicemi za rozumnou dobu.
- Z věty o prvočíslech plyne, že pravděpodobnost toho, že takto vybraná čísla jsou prvočísla, $\approx 2/\log(10^{100})$.
- Pro nalezení prvočísla potřebujeme v průměru $1/(2/\log(10^{100})) \approx 115$ testů takových celých čísel.
- Ke zjištění, jestli jsou takto náhodně vybraná lichá celá čísla prvočísly, použijeme Rabinův-Millerův pravděpodobnostní test.
- 100číslicové celé liché číslo je testováno Rabin-Millerovým testem pro 100 "svědků".
- Pravděpodobnost, že testované číslo je složené je $\approx 10^{-60}$.
- Každý subjekt provádí daný výpočet pouze dvakrát.

RSA (11)

- Jakmile jsou prvočísla p a q nalezena $\Rightarrow$ je vypočítán šifrovací exponent e (platí $\gcd(e, \Phi(pq)) = 1$).
- Doporučení: zvolit e jako nějaké prvočíslo $> p$ a q.
- Pokud $2^e > n = pq \Rightarrow$a znemožnění odkrytí bloku otevřeného textu m následným jednoduchým umocňováním celého čísla c, kde $c = |m^e|_n, 0 < c < n$, bez provedení redukce modulo n.
- Podmínka $2^e > n$ zaručí, že každý blok otevřeného textu m, kde je zašifrovaný umocněním a následnou redukcí modulo n.

Bezpečnost RSA

- Modulární umocňování potřebné k šifrování zprávy s použitím RSA může být provedeno při *VK* a *m* o velikosti $\approx$ 200 dekadických číslic za několik málo sekund počítačového času.
- Se znalostí *p* a *q* ($\Phi(n) = \Phi(pq) = (q-1)(r-1)$) a s použitím Euklidova algoritmu lze najít dešifrovací klíč *d*, kde $|de|_{\Phi(n)} = 1$,
- K objasnění proč znalost šifrovacího klíče (e, n), který je veřejný, nevede lehce k nalezení dešifrovacího klíče (d, n) je důležité si uvědomit, že k nalezení dešifrovacího klíče *d* jako inverzi šifrovacího klíče *e* modulo $\Phi(n)$ vyžaduje znalost hodnot *p* a *q*, které umožní snadný výpočet $\Phi(pq) = (p-1)(q-1)$.

V případě, kdy nepoznáme hodnoty *p* a *q*, je nalezení $\Phi(n)$ podobně složité jako faktorizace celého čísla *n*.

RSA (13)

Problem faktorizace a RSA (1)

- Pokud p a q jsou 100číslicová prvočísla $\Rightarrow$ n je 200číslicové.
- Nejrychlejší známé algoritmy pro faktorizaci potřebují $\approx 10^6$ roků počítačového času k faktorizaci takových celých čísel.
- Naopak, pokud známe d, ale neznáme $\Phi(n)$, je možné lehce faktorizovat n, protože víme, že $ed - 1$ je násobkem $\Phi(n)$.
- Pro takovou úlohu existují speciální algoritmy faktorizace celého čísla n s použitím nějakého násobku $\Phi(n)$.
- Dosud nebylo prokázáno dešifrování zprávy zašifrované s použitím RSA bez faktorizace n!
- $\Rightarrow$ pokud neexistuje žádná metoda pro dešifrování RSA bez provedení faktorizace modulu n je RSA šifrovací systém metodou používající faktorizaci!

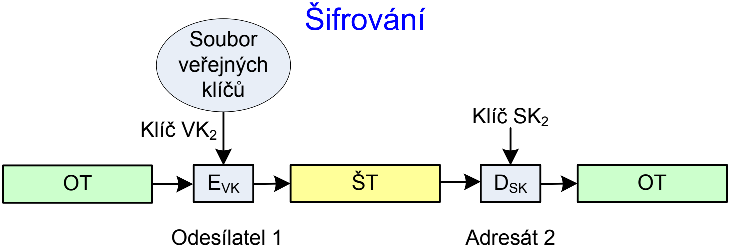
RSA (14)

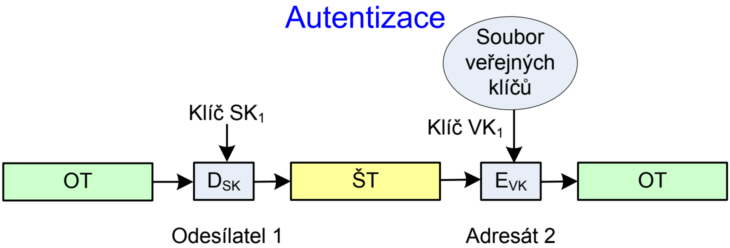
Problem faktorizace a RSA (2)

Výpočetní náročnost je tím větší, čím větší je modul.

- Zprávy šifrované s použitím RSA systému se stávají zranitelné proti útokům v tom okamžiku, když se faktorizace n stane proveditelnou v "reálných podmínkách"!
- Znamená to zvýšenou pozornost při výběru a používání prvočísel p a q k zajištění ochrany utajení zpráv, které mají být utajeny na desítky a stovky let.
- Ochrana proti speciálním, rychlým technikám pro faktorizaci $n = pq$. Například obě hodnoty $p - 1$ a $q - 1$ by měly mít velký prvočíselný faktor, tedy $\gcd(p-1, q-1)$ by mělo být malé a p a q by měly mít rozdílnou desítkovou reprezentaci v délce několika málo číslic.

Schémata kryptografie veřejného klíče





Digitální podpis a RSA (1)

- Šifrovací systém RSA lze použít pro vyslání podepsané zprávy.
- Při použití podpisu se příjemce zprávy může ujistit, že zpráva přišla od oprávněného odesílatele a že tomu tak je na základě nestranného a objektivního testu.
- Takové ověření je potřebné pro elektronickou poštu, elektronické bankovnictví, elektronický obchod atd.

Princip

- Nechť subjekt 1 vysílá podepsanou zprávu m subjektu 2.
- Subjekt 1 spočítá pro zprávu m OT

$$S = D_{SK_1}(m) = |m^{d_1}|_{n_1},$$

kde $SK_1 = (d_1, n_1)$ je tajný dešifrovací klíč pro subjekt 1.

- Když $n_2 > n_1$, kde $VK_2 = (e_2, n_2)$ je veřejný šifrovací klíč pro subjekt 2, subjekt 1 zašifruje S pomocí vztahu

$$c = E_{VK_2}(S) = |S^{e_2}|_{n_2}, \quad 0 < c < n_2 \ .$$

Digitální podpis a RSA (2)

- Když $n_2 < n_1$ subjekt 1 rozdělí S do bloků o velikosti menší než n_2 a zašifruje každý blok s použitím šifrovací transformace E_{VK_2}.
- Pro dešifrování subjekt 2 nejdříve použije soukromou dešifrovací transformaci D_{SK_2} k získání S, protože

$$D_{SK_2}(c) = D_{SK_2}(E_{VK_2}(S)) = S\ .$$

- K nalezení OT m předpokládejme, že byl vyslán subjektem 1, subjekt 2 dále použije veřejnou šifrovací transformaci E_{VK_1}, protože

$$E_{VK_1}(S) = E_{VK_1}(D_{SK_1}(m)) = m.$$

  Zde jsme použili identitu $E_{VK_1}(D_{SK_1}(m)) = m$, která plyne z faktu, že

$$E_{VK_1}(D_{SK_1}(m)) = |(m^{d_1})^{e_1}|_{n_1} = |m^{d_1 e_1}|_{n_1} = m,$$

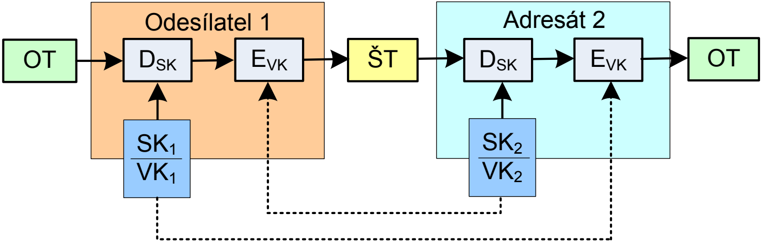
  protože

$$|d_1 e_1|_{\Phi(n_1)} = 1.$$

Digitální podpis a RSA (3)

- Kombinace OT *m* a podepsané verze *S* přesvědčí subjekt 2, že zpráva byla vyslána subjektem 1.
- Také subjekt 1 nemůže odepřít, že on vyslal danou zprávu, protože žádný jiný subjekt než 1 nemůže generovat podepsanou zprávu *S* z originálního textu zprávy *m*.

Digitální podpis



RSA-CRT (1)

Urychlení šifrování

- Pro urychlení šifrování je normou doporučena množina šifrovacích exponentů e.
- Exponenty se vyznačují malou Hammingovou váhou $\Rightarrow$
- šifrování probíhá rychle v několika krocích, viz modulární umocňování.
- Například $e = 11_2, 1011_2, 10001_2, 2^{16} + 1, \ldots$

VK e	e v binární podobě	#MUL + #SQ
3	11	3
17	10001	5
$2^{16} + 1$	1 0000 0000 0000 0001	17

RSA-CRT (2)

Urychlení dešifrování

- Pro urychlení dešifrování se využívá rozklad pomocí Čínské věty o zbytcích - RSA-CRT
- Potom se při dešifrování počítá s čísly poloviční délky $\Rightarrow$ zrychlení 4 až 8 násobné oproti původnímu dešifrovacímu výpočtu.

Definice RSA-CRT

- Nechť p a q jsou prvočísla.
- Vypočítáme $n = pq$, $\Phi(n) = (p-1)(q-1)$.
- Zvolíme e, $1 < e < n$, $\gcd(e, \Phi(n)) = 1$ a spočítáme $d = \left|e^{-1}\right|_{\Phi(n)}$.
- Vypočítáme $d_p = |d|_{p-1}$, $d_q = |d|_{q-1}$, $q_{inv} = \left|q^{-1}\right|_p$.
- Dvojici $VK = (n, e)$ prohlásíme za veřejný klíč (a zveřejníme), šestici $SK = (n, p, q, d_p, d_q, q_{inv})$ prohlásíme za soukromý klíč.

RSA-CRT (3)

Šifrování a dešifrování

- Pro šifrování platí stejný vztah jako pro RSA: $c = |m^e|_n$.
- Pro dešifrování v RSA-CRT musí platit pro d_p a d_q následující kongruence:

$$
\begin{aligned}
ed_p &\equiv 1 \bmod (p-1) \\
ed_q &\equiv 1 \bmod (q-1)
\end{aligned}
$$

RSA-CRT (4)

Dešifrování

1. Vypočteme
$$m_1 = |c^{d_p}|_p$$
$$m_2 = |c^{d_q}|_q$$
2. Vypočteme
$$h = \left|\left|q^{-1}\right|_p (m_1 - m_2)\right|_p$$
3. Vypočteme
$$m = m_2 + hq$$

- Krok 1 je výpočetně nejnáorčnější. Počítáme ale s polovičními délkami čísel než v případě RSA.
- Výpočet m_1 a m_2 je datově nezávislý a lze ho provádět paralelně.
- V kroku 2 je nenáročné násobení rozdílu m_1 a m_2 s předpočítanou konstantou $\left|q^{-1}\right|_p$s a následnou redukcí modulo p.
- Poslední krok představuje nejméně náročné násobení a sčítaní.

El Gamal (1)

Úvod

- El Gamal - Taher ElGamal
- Algoritmus pro kryptografii s veřejným klíčem
- Založen na Diffie-Hellmanově výměně klíče, resp. problému diskrétního logaritmu
- Podobně jako RSA i El Gamal umožňuje šifrování i digitální podpis

El Gamal (2)

Diffie-Hellman připomenutí

- Alice (A) a Bob (B) si veřejně dohodnou prvočísla g a m nesoudělná, $1 < g < m$ (přesněji: grupu řádu m a její generátor g)
- A si náhodně zvolí číslo x takové, že $0 < x < m$, spočítá $c = |g^x|_m$ a odešle ho B.
- B si náhodně zvolí číslo y takové, že $0 < y < m$, spočítá $d = |g^y|_m$ a odešle ho A.
- A i B spočítají sdílený klíč $k = |d^x|_m = |(g^y)^x|_m = |g^{xy}|_m = |(g^x)^y|_m = |c^y|_m$
- Útočník nedokáže z $|g^x|_m$ a $|g^y|_m$ spočítat $k = |g^{xy}|_m$ (tzv. Diffie-Hellmanův problém, DHP)
- DHP není složitější než problém diskrétního logaritmu (DLP): Kdyby útočník uměl vyřešit DLP, dokázal by z $|g^x|_m$ spočítat x a následně z x a $|g^y|_m$ triviálně spočítá $|g^{xy}|_m$
- Je DHP jednodušší než DLP? Nevíme jistě, ale zdá se, že ne.

El Gamal (3)

El Gamal - příprava klíče

- El Gamal vzniká úpravou DH:
- Alice (A) ~~a Bob (B) si veřejně dohodnou~~ zvolí prvočísla g a m nesoudělná, $1 < g < m$ (přesněji: grupu řádu m a její generátor g)
- A si náhodně zvolí číslo x takové, že $0 < x < m$, spočítá $c = |g^x|_m$ ~~a odešle ho B~~.
- A zveřejní uspořádanou trojici (m, g, c) jako svůj veřejný klíč. x je jejím soukromým klíčem.

El Gamal (4)

El Gamal - šifrování

- Bob chce Alici poslat zprávu p:
- B si náhodně zvolí číslo y takové, že $0 < y < m$, spočítá $d = |g^y|_m$ ~~a odešle ho A~~.
- ~~A i B spočítají~~B spočítá sdílený klíč $k = |d^x|_m = |(g^y)^x|_m = |g^{xy}|_m = |(g^x)^y|_m = |c^y|_m$
- B zašifruje zprávu p pomocí vztahu $e = |p \cdot k|_m$
- B odešle Alici uspořádanou dvojici (d, e).

El Gamal (5)

El Gamal - šifrování - příklad

- $m = 2543, g = 5, c = |g^x|_m = 505$ (pro $x = 10$, které ale B nezná).
- $y = 123$ (náhodná volba)
  $\rightarrow d = |g^y|_{2543} = 308, k = |c^y|_{2543} = 1883$!! Zjednodušení pro demonstrační účely, v praxi by to byla hrubá chyba, viz poznámky níže!!
- p = "`ELGAMAL RULES`"→ `0511, 0701, 1301, 1118, 2111, 0519`
- $e = |p \cdot k|_{2543} \rightarrow$ `0959, 0166, 0874, 2133, 0304, 0765`
- B odesílá A dvojice $(308, 959), (308, 166), (308, 874)$...

El Gamal (6)

El Gamal - dešifrování

- Alice dostala od Boba zprávu (d, e)
- A si spočítá sdílený klíč $k = |d^x|_m = |(g^y)^x|_m = |g^{xy}|_m$.
- A si spočítá $|k^{-1}|_m$ (Eukleidův rozšířený algoritmus)
- A dešifruje zprávu $p' = |e \cdot k^{-1}|_m = |p \cdot k \cdot k^{-1}|_m = |p|_m = p$

El Gamal (7)

El Gamal - dešifrování - příklad

- Alice dostala od Boba zprávu $(308, 959)$, tzn $d = 308$, $e = 959$
- A si spočítá sdílený klíč $k = |d^x|_m = |308^{10}|_{2543} = 1883.$
- A si spočítá $|1883^{-1}|_{2543} = 1337$ (Eukleidův rozšířený algoritmus)
- A dešifruje zprávu $p' = |959 \cdot 1337|_{2543} = 511 \rightarrow$ "EL"
- Obdobně pro další bloky zprávy

El Gamal - poznámky

- Místo operace násobení lze v $e = |p \cdot k|_m$ použít i jiné invertovatelné operace, např. sčítání mod m nebo xor. To může být výhodné např. z hlediska rychlosti (šifrování i dešifrování).
- Všimněte si, že že zašifrovaná zpráva je dvakrát delší než původní otevřený text p.
- Dešifrovat komunikaci mezi A a B je nejvýš tak složité jako vyřešit DHP, protože kdybychom dokázali vyřešit DHP, můžeme spočítat $k = |g^{xy}|_m$ ze znalosti $c = |g^x|_m$ a $d = |g^y|_m$.
- Je dešifrování komunikace mezi A a B jednodušší než vyřešit DHP? Nevíme, ale asi ne.

El Gamal (9)

El Gamal - poznámky

- El Gamal není bezpečný pro útok typu chosen ciphertext attack, protože útočník může ze znalosti platné šifrované zprávy (d, e) pro i neznámou zprávu p spočítat jinou platnou šifrovanou zprávu (d, e') např. pomocí vztahu $e' = |2 \cdot e|_m$, kde (neznámé) $p' = |2 \cdot p|_m$ (uvažujte pro případ, kdy p představuje částku nebo číslo účtu).
- Pro každé šifrování je nutné zvolit jiný dočasný (ephemeral) klíč y. Pokud použijeme pro dvě různé zprávy p_1, p_2 stejné y, pak $|\frac{e_1}{e_2}|_m = |\frac{p_1}{p_2}|_m$ ... vydělením dvou šifrovaných textů dostaneme podíl otevřených textů, klíč zcela zmizel! Ze znalosti jedné nešifrované zprávy pak jde dopočítat druhou. Vyzkoušejte na příkladu výše, kde bylo použito konstantní y.

Digitální podpis a El Gamal (1)

El Gamal - podepisování

- Alice chce podepsat zprávu p tak, aby kdokoliv mohl podpis ověřit
- Veřejný klíč je totožný jako pro šifrování, tzn. trojice $(m, g, c = |g^x|_m)$
- Alice náhodně zvolí y takové, že $0 < y < m - 1$ a neopakovalo se, a spočítá:
  - $r = |g^y|_m$
  - $s = |(p - x \cdot r) \cdot y^{-1}|_{m-1}$
- Alice tyto kroky opakuje, dokud $s \neq 0$.
- Uspořádaná dvojice (r, s) tvoří podpis zprávy p.

Digitální podpis a El Gamal (2)

El Gamal - podepisování - příklad

- Z veřejného klíče: $m = 2543, g = 5, x = 10, c = 505$
- Podepisovaná zpráva: $p = 1234$
- Alice náhodně zvolí $y = 1111$, které nikdy dříve nebylo použito
- $|y^{-1}|_{2542} = 1835$
- $r = |g^y|_m = |5^{1111}|_{2543} = 1567$
- $s = |(p - x \cdot r) \cdot y^{-1}|_{m-1} = |(1234 - 10 \cdot 1567) \cdot 1835|_{2542} = 122$
- Uspořádaná dvojice $(1567, 122)$ tvoří podpis zprávy 1234.

Digitální podpis a El Gamal (3)

El Gamal - ověření podpisu

- Pro platný podpis platí: $|g^p|_m = |c^r \cdot r^s|_m$
- Proč:
  - Úpravou vztahu pro s dostaneme $p = |xr + sy|_{m-1}$
  - Z malé Fermatovy věty: $|a|_{m-1} = |b|_{m-1} \Rightarrow |c^a|_m = |c^b|_m$ pro všechna c. Proto:
  - $|g^p|_m = |g^{xr} \cdot g^{sy}|_m = |(g^x)^r \cdot (g^y)^s|_m = |c^r \cdot r^s|_m$
- m, g, c známe z veřejného klíče, p, r, s dostáváme jako zprávu a její podpis.
- Nikdo jiný než Alice nemůže podpis vytvořit, protože nikdo jiný nezná x ani y.

Digitální podpis a El Gamal (4)

El Gamal - ověření podpisu - příklad

- Z veřejného klíče: $m = 2543, g = 5, x = 10, c = 505$
- Podepisovaná zpráva: $p = 1234$
- Podpis: $r = 1567, s = 122$
- Levá strana: $|g^p|_m = |5^{1234}|_{2543} = 2009$
- Pravá strana: $|c^r \cdot r^s|_m = |505^{1567} \cdot 1567^{122}|_{2543} = 2009$
- Podpis souhlasí

Digitální podpis a El Gamal (5)

El Gamal - poznámky

- Je nutné, aby y nebylo použito dvakrát: Útočník může z n zpráv $p_1 ... p_n$ a odpovídajících podpisů $r_1 ... r_n, s_1 ... s_n$ sestavit n rovnic tvaru $p_i = |x \cdot r_i + y_i \cdot s_i|_{m-1}$ s $n + 1$ neznámými (jednou x, n-krát y_i). Tato soustava má velmi mnoho řešení. Kdyby bylo některé y použito dvakrát, má tato soustava právě jedno řešení a útočník získá soukromý klíč x.
- Snaha o vyjádření x, y z $|g^p|_m = |c^r \cdot r^s|_m$ odpovídá nalezení diskrétního logaritmu, protože jak x tak y vystupuje v exponentu.

El Gamal - poznámky

- Zfalšování podpisu (nalezení vhodného r, s pro $|g^p|_m = |c^r \cdot r^s|_m$) odpovídá nalezení diskrétního logaritmu, protože levá strana je konstantní, c^r je dané (libovolnou) volbou r a s, které potřebujeme dopočítat, vystupuje v exponentu. Volba s s úmyslem dopočítat r vede na rovnici tvaru $A = |r^s \times B^r|_m$, o které se domníváme, že je stejně složitá jako DLP.
- Existuje útok (společný i pro další podpisová schemata), který dokáže ze znalosti platné trojice (p, r, s) generovat další platné trojice (p', r', s'), neumožňuje ale zvolit si p' (tzn. lze podepsat falešnou zprávu, ale útočník nedokáže zajistit, aby ta zpráva měla obsah, který on chce). Podrobnosti viz [1].

Digitální podpis a El Gamal (7)

El Gamal - příklad dvakrát použitého y

- Neznámé hodnoty pro podpis: $x = 10, y = y_1 = y_2 = 1111$.
- Známé hodnoty z veřejného klíče: $m = 2543, g = 5, c = 505$
- Podepsané zprávy (p_i, r_i, s_i): $(1234, 1567, 122)$, $(2323, 1567, 425)$
- Všimněte si, že pro $y_1 = y_2$ nutně platí $r_1 = r_2$ (protože $r_i = |g^{y_i}|_m$), tzn. útočník snadno pozná, že bylo dvakrát použito stejné y.
- Víme, že $p = |xr + sy|_{m-1}$. Tudíž v našem případě: $1234 = |1567x + 122y|_{2542}$, $2323 = |1567x + 425y|_{2542}$. Odečtením první rovnice od druhé: $1089 = |303y|_{2542}$. $|303^{-1}|_{2542} = 797$, z toho $y = |797 \cdot 1089|_{2542} = 1111$. Dosazením do první rovnice $1567x = |1234 - 122 \cdot 1111|_{2542} = 418$. Protože $|1567^{-1}|_{2542} = 73$, dostaneme $x = |73 \cdot 418|_{2542} = 10$. Získali jsme Alicin soukromý klíč!!
- Pozn.: V případě, že neexistují příslušné inverze, je řešení složitější, ale stále dosažitelné.

Digitální podpis a El Gamal (8)

El Gamal - poznámky

- Zfalšování podpisu (nalezení vhodného r, s pro $|g^p|_m = |c^r \cdot r^s|_m$) odpovídá nalezení diskrétního logaritmu, protože levá strana je konstantní, c^r je dané (libovolnou) volbou r a s, které potřebujeme dopočítat, vystupuje v exponentu. Volba s s úmyslem dopočítat r vede na rovnici tvaru $A = |r^s \cdot B^r|_m$, o které se domníváme, že je stejně složitá jako DLP.
- Existuje útok (společný i pro další podpisová schemata), který dokáže ze znalosti platné trojice (p, r, s) generovat další platné trojice (P, R, S), neumožňuje ale zvolit si P (tzn. lze podepsat falešnou zprávu, ale útočník nedokáže zajistit, aby ta zpráva měla obsah, který on chce). Podrobnosti viz [1] a další slajd.

Digitální podpis a El Gamal (9)

El Gamal - generování falešných podpisů

- Máme platně podepsanou zprávu (p, r, s), tzn. $|g^p|_m = |c^r \cdot r^s|_m$.
- Zvolíme libovolné A, B, C tak, aby $gcd(A \cdot r - C \cdot s, m-1) = 1$.
- Buď $R = |r^A \cdot g^B \cdot c^C|_m$, $S = |\frac{s \cdot R}{A \cdot r - C \cdot s}|_{m-1}$, $P = |\frac{R \cdot (A \cdot p + B \cdot s)}{A \cdot r - C \cdot s}|_{m-1}$
- Pak (P, R, S) je také platný podpis:

$$
\begin{aligned}
|c^R R^S|_m &= |c^R (R^A g^B c^C)^{\frac{sR}{Ar-Cs}}|_m & (1)\\
&= |(c^{R(Ar-Cs)+CsR} r^{AsR} g^{BsR})^{\frac{1}{Ar-Cs}}|_m & (2)\\
&= |(c^{RAr} r^{AsR} g^{BsR})^{\frac{1}{Ar-Cs}}|_m & (3)\\
&= |((c^r r^s)^{AR} g^{BsR})^{\frac{1}{Ar-Cs}}|_m & (4)\\
&= |((g^p)^{AR} g^{BsR})^{\frac{1}{Ar-Cs}}|_m & (5)\\
&= |g^{\frac{pAR+BsR}{Ar-Cs}}|_m & (6)\\
&= |g^P|_m & (7)
\end{aligned}
$$

- Pro $A = 0$ můžeme generovat podpisy, niž bychom měli nějakou

Srovnání RSA a El Gamal (1)

Shodné vlastnosti

- RSA i ElGamal nám umožňují operace šifrování, dešifrování, podepisování, ověření podpisu.
- RSA i ElGamal jsou založeny na jednosměrných funkcích. V případě RSA jde o tzv. "trapdoor function"(jednosměrná funkce, kterou lze invertovat, pokud známe speciální informaci - v tomto případě rozklad $\Phi(n)$), v případě ElGamal jde o funkci se speciální vlastností ($|(g^a)^b|_m = |(g^b)^a|_m$).
- RSA i ElGamal lze volně použít, nejsou zatíženy patenty nebo licencemi (RSA až od r. 2000).

Srovnání RSA a El Gamal (2)

Příprava klíčů

- RSA: Vyžaduje vygenerování dvou silných prvočísel, volbu šifrovacího exponentu, výpočet dešifrovacího exponentu.
- ElGamal: Vyžaduje volbu vhodné grupy a ideálně (ne nezbytně) i nalezení jejího generátoru.
- Závěr: ElGamal má přípravu jednodušší.

Velikost klíčů

- RSA: Veřejný klíč je dvojice (n, e), soukromý klíč je (d).
- ElGamal: Veřejný klíč je $(m, g, c = g^x)$, soukromý klíč je (x).

Srovnání RSA a El Gamal (3)

Bezpečnost

- RSA: Bezpečnost je založena na problému faktorizace.
- ElGamal: Bezpečnost je založena na problému diskrétního logaritmu.
- Závěr: Obě šifry jsou považovány za bezpečné, ale obě jsou prolomeny kvantovým počítačem a pro obě existují výkonné ne-kvantové algoritmy (GNFS pro RSA, Index Calculus pro ElGamal).
- Při stejné délce klíče a správně zvolených parametrech je ElGamal bezpečnější, protože v množině dané velikosti má více přípustných hodnot (typicky $n - 1$ pro ElGamal vs. $\frac{n}{\log n}$ pro RSA).

Srovnání RSA a El Gamal (4)

Operace

- RSA: Všechny čtyři operace jsou realizovány modulárním mocněním (s odlišnými exponenty). Šifrování/verifikaci podpisu lze urychlit vhodnou volbou e, dešifrování/podpis lze urychlit pomocí CRT.
- ElGamal: Operace používají odlišné postupy, kromě umocňování se používá i násobení a inverze. Nelze vhodně volit exponent, ale lze předpočítat některé údaje, případně zvolit jiné (rychlejší) vztahy (XOR místo násobení apod.).
- Závěr: RSA je z programátorský jednodušší a rychlejší.

Otevřený a šifrový text (PT, CT)

- RSA: $CT = PT^e$
- ElGamal: $CT = (g^y, c^y \cdot PT)$ kde y je nonce (neopakuje se).
- Závěr: ElGamal má dvakrát delší šifrový text než RSA.
- Ale zároveň: RSA je náhodné orákulum, ElGamal dá pro stejnou zprávu pokaždé jiný šifrový text.

Srovnání RSA a El Gamal (5)

Citlivé hodnoty

- RSA: $PT = 0$ a $PT = 1$ vedou na $CT = PT$. Nízké hodnoty PT v komninaci s malým e mohou být snáze dešifrovatelné (obyčejná odmocnina místo modulární).
- ElGamal: Porušení požadavku na jedinečnost y umožňuje u podpisu odhalit soukromý klíč, u šifrování dešifrovat obě zprávy. Totéž pro známou nebo předvídatelnou volbu y.

Použité zdroje

El Gamal - použité zdroje

- ElGamal, Taher: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Advances in cryptology: Proceedings of CRYPTO 84. Lecture Notes in Computer Science 196. Santa Barbara, California, United States: Springer-Verlag. pp. 10–18. Dostupné z: `http://groups.csail.mit.edu/cis/crypto/classes/6.857/papers/elgamal.pdf`