Y36BEZ – Bezpečnost přenosu a zpracování dat

Róbert Lórencz

8. přednáška

RSA, kryptografie s veřejným klíčem

http://service.felk.cvut.cz/courses/Y36BEZ
lorencz@fel.cvut.cz

Obsah přednášky

- RSA
- Princípy kryptografie s veřejným klíčem (VK)
- Kryptografické systémy s VK

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 $|p^{\Phi(n)}|_n = 1$, kde $\gcd(p, n) = 1$.

Pravděpodobnost, že m a n nejsou nesoudělná je extrémně malá. Dvojice (d, n) je dešifrovací klíč - tajná část klíče SK.

RSA (4)

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 (5)

- 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:

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

  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 (6)

Definice RSA

- 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 $d = \left|e^{-1}\right|_{\Phi(n)}$.
- Dvojici $VK = (n, e)$ prohlásíme za veřejný klíč (a zveřejníme), dvojici $SK = (n, d)$ prohlásíme za soukromý klíč.

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.

RSA (7)

- 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.
- 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.

RSA (8)

Bezpečnost RSA

- Modulární umocňování potřebné k šifrování zprávy s použitím RSA může být provedeno při VK a m o velikosti ≈ 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.

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!

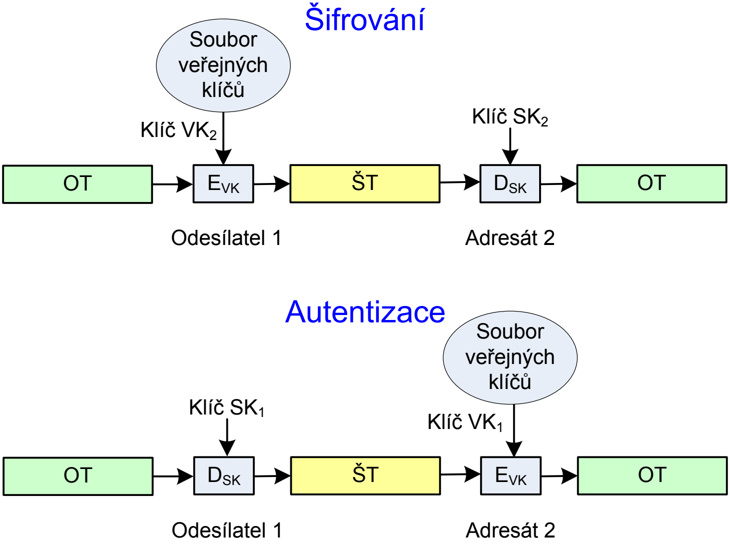
RSA (10)

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

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

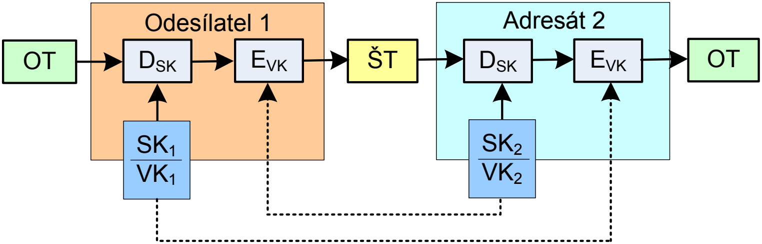
Digitální podpis a RSA (2)

- 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



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 $e = 11_2, 1011_2, 10001_2, 2^{16} + 1, \ldots$

Urychlení dešifrování

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

RSA-CRT (2)

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.

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 (3)

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.

Asymetrická a symetrická kryptografie (1)