Bezpečnost v IT – 3. přednáška

Založeno na materiálech dr. Bartla

Radek Janoštík

Univerzita Palackého v Olomouci

29. 2. 2024

Outline

- Aktuální (kyber)bezpečnostní situace
- Asymetrická kryptografie
- RSA
- Bezpečnost RSA
- Doporučená četba

Aktuální (kyber)bezpečnostní situace

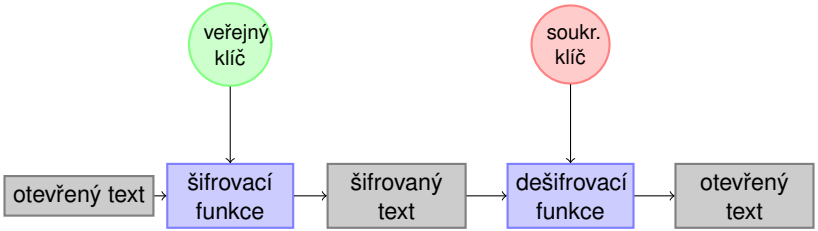
- Sledujete zprávy z dění v kyberprostoru? Jaké?
- Chyby ve wordPress pluginech (SQL injection a eskalace práv
- Odměna 15 milionů dolarů za dopadení vůdců ransomware LockBit (+ Operace Cronos - UK
- Pokuta 16 milionů dolarů pro Avast za prodej citlivých dat
- Rozpoznávání obličeje prodejním automatem
- Omezení PWA na iOS

Asymetrické šifrování

- klíče pro šifrování a dešifrování nejsou stejné
- šifrovací klíč – *veřejný* klíč
- dešifrovací klíč – *soukromý* klíč
- existuje vztah mezi soukromým a veřejným klíčem
  - soukromý $\rightarrow$ veřejný – jednoduše vypočítatelné
  - veřejný $\rightarrow$ soukromý – jednoduše nelze odvodit (jednosměrnost)
- soukromý klíč je držen v tajnosti, veřejný klíč ale může být veřejně distribuován
  - mluvíme o *šifrování s veřejným klíčem* (*public-key cryptography*)
- příklad: šifrovaní založené na zavazadlovém problému, diskrétním logaritmu, RSA, eliptických křivkách, atd.

Asymetrické šifrování

- postup:
  1. příjemce vytvoří soukromý klíč a veřejný klíč
  2. soukromý klíč uschová příjemce v tajnosti, veřejný klíč může být zveřejněn
  3. odesilatel zašifruje zprávu pomocí veřejného klíče
  4. šifrovaná zpráva může být poslána příjemci
  5. příjemce dešifruje zprávu pomocí soukromého klíče



Asymetrické šifrování

Výhody:

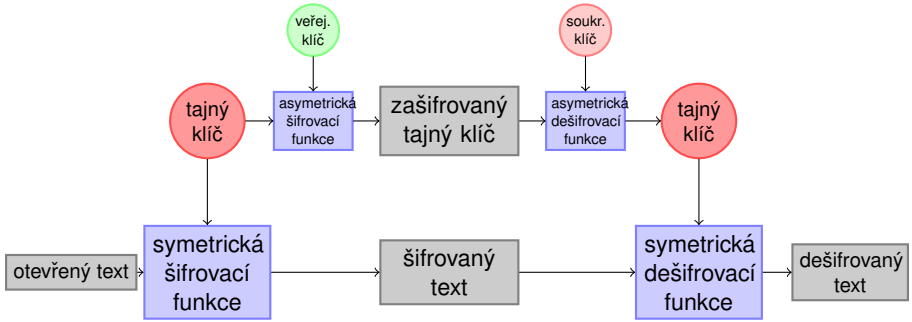
- bezpečnější
- není složitá správa klíčů

Nevýhoda:

- asymetrické šifrování je pomalejší než symetrické
- Např. na 2GHz CPU je propustnost cca 204 kbit/s (RSA 2048)
- vs. $\approx$ 400Mbit/s pro symetrickou šifru AES

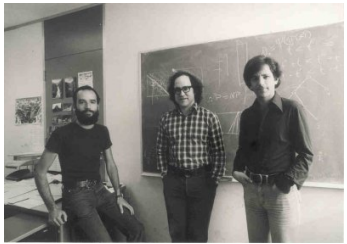
Hybridní šifrování

- kombinuje symetrické a asymetrické šifrování
- odstraňuje nevýhody obou předešlých metod
- šifrování je ve své podstatě symetrické (rychlost), asymetrické šifrování je použito při distribuci tajného klíče (bezpečnost)
- hybridní šifrování je například použito v protokolech SSL a TLS



Algoritmus RSA

- RSA (**R**ivest, **S**hamir a **A**dleman – tvůrci RSA)
- asymetrická šifra, která je považována za velmi bezpečnou (dle délky klíče)



- algoritmus publikován roku 1977
- tentýž rok patentován (patent platil pouze pro USA do roku 2000)

Algoritmus RSA

- ale: už roku 1973 vyvinul ekvivalentní systém Clifford Christopher Cocks
- skutečnost odtajněna až roku 1998

Použití RSA

1. bezpečná výměna tajných klíčů při symetrickém šifrování (dříve uvedené hybridní šifrování; protokoly SSL a TSL využívají RSA)
2. digitální podpis

Modulární aritmetika (připomenutí)

Inverzní prvky v $(\mathbb{Z}_n, \cdot)$:

- platí: prvek a má inverzi v $(\mathbb{Z}_n, \cdot)$ právě tehdy, když $\gcd(a, n) = 1$

Eulerova funkce:

- Eulerova funkce φ: $\varphi(n)$ je počet přirozených čísel menších než n a nesoudělných s n
- vlastnosti Eulerovy funkce:

(i) $\varphi(p) = p - 1$ pro libovolné prvočíslo p

(ii) $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$ pro libovolná nesoudělná čísla a, b

(iii) jestliže $a = \prod_{i=1}^{m} p_i^{e_i}$ je rozklad čísla a na prvočísla p_i, pak

$$\varphi(a) = \prod_{i=1}^{m} (p_i^{e_i} - p_i^{e_i - 1})$$

- např.: $60 = 2^2 \cdot 3^1 \cdot 5^1$, takže $\varphi(60) = (4 - 2) \cdot (3 - 1) \cdot (5 - 1) = 16$

Algoritmus RSA

Generování soukromého a veřejného klíče (příjemce)

- zvolí se dvě různá prvočísla p a q (přibližně stejně velká, typická velikost 1024 až 3072 bitů nebo i více)
- vypočítá se součin $n = p \cdot q$, platí $\varphi(n) = (p-1) \cdot (q-1)$
- náhodně se zvolí $e \in \{1, 2, \ldots, \varphi(n) - 1\}$ tak, aby $\gcd(e, \varphi(n)) = 1$ (e se nazývá veřejný exponent), často se volí $e = 3$
- pomocí rozšířeného Euklidova algortimu vypočteme inverzi e modulo $\varphi(n)$, označíme $d = e^{-1} \bmod \varphi(n)$
- $k_e = \langle e, n \rangle$ reprezentuje veřejný klíč
- $k_d = d$ reprezentuje soukromý klíč
- čísla p, q se mohou odložit(nejsou potřeba), ale nikdy se nesmí zveřejnit

Algoritmus RSA

Šifrování a dešifrování

- otevřený text $x \in \mathcal{M}$ se rozdělí na **číselné bloky** x_i tak, aby $x_i < n$
- šifrování:

$$e(x_i, k_e) = x_i^e \mod n$$

- dešifrování:

$$d(y_i, k_d) = y_i^d \mod n$$

Prolomení RSA

- invertovat funkci $e(x_i, k_e)$ znamená vyřešit RSAP:

Problém RSA (RSA Problem – RSAP)

Nechť n je velké složené číslo a e je nesoudělné s $\varphi(n)$ a $y \in \{1, 2, \ldots, n-1\}$. Problém RSA je problém nalezení takového x, pro které platí

$$y \equiv x^e \pmod{n}.$$

- Není znám efektivní algoritmus pro řešení (stejně jako problém faktorizace dvou

Algoritmus RSA

Příklad:

- otevřený text $m = 688232687966668003$
- $p = 47$, $q = 71$, $n = p \cdot q = 3337$
- zvolíme $e = 79$, 79 není soudělné s $(p-1) \cdot (q-1) = 3220$
- d vypočítáme jako inverzní prvek k 79 v $(\mathbb{Z}_{3320}, \cdot)$, tzn. $d = 1019$
- otevřený text m rozdělíme do bloků

$$m_1 = 688$$
$$m_2 = 232$$
$$m_3 = 687$$
$$m_4 = 966$$
$$m_5 = 688$$
$$m_6 = 003$$

tak, aby $m_i < n = 3337$

Algoritmus RSA

- šifrování 1. bloku: $c_1 = 688^{79} \mod 3337 = 1570$
- podobně pro další bloky
- celkem získáme šifrovanou zprávu

$$c = 1570\ 2756\ 2091\ 2276\ 2423\ 158$$

- dešifrování 1. bloku: $m_1 = 1570^{1019} \mod 3337 = 688$
- podobně pro další bloky

Algoritmus RSA

Věta

Algoritmus RSA pracuje správně, tzn., platí

$$d(e(x, k_e), k_d) = x$$

pro libovolný blok x otevřeného textu.

Modulární aritmetika (připomenutí)

Eulerova věta (Euler-Fermatova věta)

- Eulerova věta: $a^{\varphi(n)} \equiv 1 \pmod{n}$ pro libovolná nesoudělná čísla a, n
- speciálním případem je Fermatova malá věta (použijeme nyní v souvislosti s šifrou RSA): $a^{p-1} \equiv 1 \pmod{p}$ pro libovolné a a provočíslo p
- Eulerova věta se dá použít pro hledání inverzního prvku k číslu a v $\mathbb{Z}_n$, platí totiž:

$$a \cdot a^{\varphi(n)-1} \equiv 1 \pmod{n}$$

- jestliže $\gcd(a, n) = 1$, pak pro inverzní prvek a^{-1} k prvku a v $\mathbb{Z}_n$ platí:

$$a^{-1} = a^{\varphi(n)-1}$$

- problém je ve výpočtu $\varphi(n)$: můžeme použít vztah vztah

$$\varphi(n) = \prod_{i=1}^{m} (p_i^{e_i} - p_i^{e_i - 1}),$$

musíme ale znát faktorizaci n, tj. $n = \prod_{i=1}^{m} p_i^{e_i}$

Algoritmus RSA

Věta

Algoritmus RSA pracuje správně, tzn., platí

$$d(e(x, k_e), k_d) = x$$

pro libovolný blok x otevřeného textu.

Důkaz:

- chceme dokázat $d(e(x, k_e), k_d) \equiv x \pmod{n}$
- tzn. $d(e(x, k_e), k_d) \equiv (x^e)^d \equiv x^{ed} \equiv x \pmod{n}$
- d je inverze k e modulo $\varphi(n)$, tzn. $ed \equiv 1 \pmod{\varphi(n)}$
- můžeme psát: $ed = t \cdot \varphi(n) + 1$, kde $t \in \mathbb{Z}$
- tzn.
$$x^{ed} \equiv x^{t\varphi(n)+1} \equiv x^{t\varphi(n)} \cdot x^1 \equiv (x^{\varphi(n)})^t \cdot x \pmod{n}$$
- potřebujeme tedy dokázat, že $(x^{\varphi(n)})^t \cdot x \equiv x \pmod{n}$

Algoritmus RSA

(a): $\gcd(x, n) = 1$

- z Eulerovy věty: $(x^{\varphi(n)})^t \cdot x \equiv 1 \cdot x \equiv x \pmod{n}$

(b): $\gcd(x, n) \neq 1$

- x a n mají společného dělitele; protože $x < n$, musí být tímto dělitelem buď p nebo q
- to znamená: buď $x = r \cdot p$ nebo $x = s \cdot q$, kde $r, s \in \mathbb{N}$ taková, že $r < q$, $s < p$
- předpokládejme, že $x = r \cdot p$ (druhý případ se dokáže analogicky)
- platí tedy $\gcd(x, q) = 1$
- můžeme tedy psát

$$(x^{\varphi(n)})^t \equiv (x^{(q-1)(p-1)})^t \equiv ((x^{\varphi(q)})^t)^{p-1} \equiv 1^{p-1} = 1 \pmod{q}$$

- právě dokázanou kongruenci $(x^{\varphi(n)})^t \equiv 1 \pmod{q}$ můžeme přepsat ve tvaru:

$$(x^{\varphi(n)})^t = k \cdot q + 1,$$

kde $k \in \mathbb{N}$

Algoritmus RSA

(b): $\gcd(x, n) \neq 1$ (pokračování)

- rovnici vynásobíme x a využijeme předpokladu $x = r \cdot p$:

$$\begin{aligned}(x^{\varphi(n)})^t \cdot x &= k \cdot q \cdot x + x = r \cdot p \cdot k \cdot q + x \\ &= r \cdot k \cdot n + x\end{aligned}$$

- celkem tedy $(x^{\varphi(n)})^t \cdot x = (r \cdot k) \cdot n + x$
- jinými slovy $(x^{\varphi(n)})^t \cdot x \equiv x \pmod{n}$

RSA – praktické aspekty

Generování soukromého klíče – volba velkých prvočísel

- pro výpočet modulu n potřebujeme dvě velká prvočísla ($n = p \cdot q$)
- např. pro n délky 1024 bitů potřebujeme dvě prvočísla délky asi 512 bitů

Princip:

- náhodně vygenerujeme číslo příslušné délky
- provedeme test prvočíselnosti

Problém

1. kolik čísel musíme v průměru vygenerovat, abychom narazili na prvočíslo?
   - prvočísel s délkou ubývá: 2,3,5,7,11,13,17,19,23,29,31,37,...
   - známý výsledek teorie čísel: náhodně vygenerované číslo p mezi 1 a N je prvočíslem s pravděpodobností přibližně $\frac{1}{\ln N}$
2. jak efektivně provést test prvočíslenosti?
   - pravděpodobnostní algoritmy
   - Fermatův test, Miller-Rabin, AKS, Solovay-Strassn

RSA – praktické aspekty

Rychlé umocnění

- výhoda symetrických šifer: počítá se s malými čísly
- modul n je velké číslo (typicky 1024 až 3072 bitů), pokud mají exponenty e a d plnou bitovou délku, pak se jedná o obrovská čísla
- porovnejme: pro e délky 1024 bitů je pro výpočet $x^e \mod n$ potřeba provést 2^{1024} násobení; odhadovaný počet atomů ve vesmíru je 2^{300}
- jedná se o zcela zásadní problém!
- bez rychlého umocnění by bylo šifrování RSA nepoužitelné

RSA – praktické aspekty

Rychlé umocnění

- pro urychlení se používá vhodná kombinace násobení (MUL) a výpočtu druhé mocniny (SQ)
- např. výpočet x^9: $x \overset{\mathrm{SQ}}{\to} x^2 \overset{\mathrm{SQ}}{\to} x^4 \overset{\mathrm{SQ}}{\to} x^8 \overset{\mathrm{MUL}}{\to} x^9$
- jak vypočítat $x^e \mod n$ obecně?
- budeme uvažovat binární zápis exponentu:

$$e = \sum_{i=0}^{t} h_i 2^i,$$

  kde $h_i \in \{0, 1\}$, $h_t = 1$
- MUL vkládá v binárním zápisu na pozici nejméně významného bitu jedničku
- SQ posouvá jedničku v binárním zápisu doleva a na pozici nejméně významného bitu vkládá nulu
- příklad: $x^{26} = x^{11010}$: $x^1 \overset{\mathrm{SQ}}{\to} x^{10} \overset{\mathrm{MUL}}{\to} x^{11} \overset{\mathrm{SQ}}{\to} x^{110} \ldots$

Bezpečnost RSA

- bezpečnost RSA je založena na předpokladu, že problém faktorizace IFP je pro velké moduly obtížný
- nevíme však přesně, do jaké složitostí třídy tento problém spadá (v tento okamžik se samozřejmě bavíme o rozhodovací variantě IFP – má modul n mezi faktory číslo menší než dané číslo m?)
- předpokládá se, že je v NP-complete
- s jistotou však pouze víme, že je v NP a co-NP
- docela zajímavé je, že příbuzný problém PRIMES je v P

Bezpečnost RSA

Faktorizace

- RSA Security Inc. vyhlásila soutěž ve faktorizaci
- z dosavadních údajů bylo vytvořeno několik aproximací, jak bude faktorizace pokračovat
- např. Silvermanova aproximace:

$$k = 4,23 \cdot (r - 1970) + 23,$$

kde k je počet cifer faktorizovaného čísla, r je rok faktorizace

Bezpečnost RSA

Silvermanova aproximace a plánované odměny za faktorizaci (soutěž však byla ukončena roku 2007)

počet bitů	rok faktorizace	odměna [USD]
640	2010	20 000
704	2015	30 000
768	2019	50 000
896	2028	75 000
1024	2038	100 000
1536	2074	150 000
2048	2110	200 000

- Reálné roky faktorizace
  `https://en.wikipedia.org/wiki/RSA_Factoring_Challenge`
- číslo n o délce 256 bitů a kratší může být dnes faktorizováno na obyčejném osobním počítači
- dnes se používá délka modulu 1024–3072 bitů

Bezpečnost RSA

Útok pomocí postranních kanálů

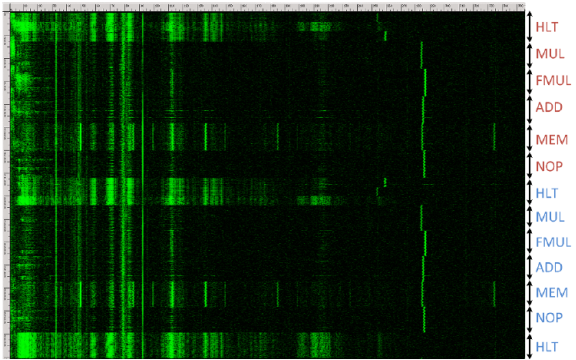
- při implementaci vznikají postranní kanály
- postranní kanál . . . nežádoucí způsob výměny informací mezi zařízením nebo programem implementujícím šifru a jeho okolím, např.:
  - časový postranní kanál
  - chybový postranní kanál (Daniel Bleichenbacher - 1998)
- postranní kanály se dají použít k prolomení RSA, aniž bychom se pokoušeli o faktorizaci

Útoky založené na postranních kanálech HW

- Máme-li fyzický přístup k počítači, ale ne ke klíči
- Při šifrování v HW mohou být postranní kanály (spotřeba, zvuk, teplo, . . .)
- Tyto projevy můžeme změřit a na jejich základě odhadnou část soukromého klíče
- Často založeny na rozdílných projevech při algoritmu `Square-and-Multiplication`

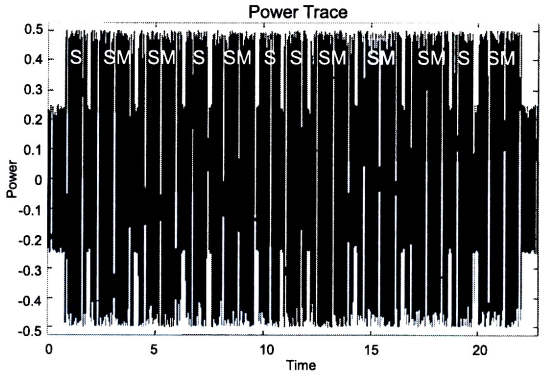
Poslouchání mikrofonem

- D. Genkin, A. Shamir, and E. Tromer. RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. *Advances in Cryptology – CRYPTO 2014* 444–461. 2014.
  - Nahrávání mikrofonem nízkých zvukových projevů počítačů
  - K nahrávání používali i pouze mobilní telefon
  - Originální přednáška: `https://www.youtube.com/watch?v=DU-HruI7Q30`



Měření spotřeby

- Podobně lze měřit spotřebu (odebraný proud) procesoru, který provádí dešifrování



- Z posloupnosti operací lze odhadnout, že část klíče je `011010011101`

Doporučená literatura

- Singh S. 2003. *Kniha kódů a šifer – Tajná komunikace od starého Egypta po kvantovou kryptografii*. Dokořán.
- Schneier B. *Applied Cryptography Second Edition*. John Wiley & Sons, 1996. ISBN 0-471-12845-7.