Bezpečnost v IT

2. přednáška – založeno na materiálech dr. Bartla

Radek Janoštík

Univerzita Palackého v Olomouci

1. 3. 2024

Outline

- Aktuální (kyber)bezpečnostní situace
- Kryptografie
  - úvod, historie
  - Základní typy šifer
- Symetrické šifrování
- Asymetrické šifrování
- RSA

Aktuální (kyber)bezpečnostní situace

- Sledujete zprávy z dění v kyberprostoru? Jaké?
- Securence (antispam as a service) uniklo velké množství emailů
- Keytrap chyba v resolverech DNSSEC (nadměrná zátěž CPU)
- Eskalace práv v antiviru ESET – mazání souborů běžným uživatelem
- Pokuta 16 milionů dolarů pro Avast za prodej citlivých dat
- Chyby ve wordPress pluginech (SQL injection a eskalace práv
- Něco dalšího?

Kryptografie

Scénář

- odesilatel chce poslat příjemci zprávu
- požaduje však bezpečnost odeslání:

Obsah zprávy je schopen přečíst pouze příjemce a nikdo jiný

Co znamená ”krypto”?

- z řečtiny ”kruprós”– skryté, utajené
- v angličtině ”crypto”asi od roku 1760

Kryptografie

- *kryptografie* . . . věda o utajování zpráv
- kryptografické metody zajišťují:
  - důvěrnost dat – utajení obsahu komunikace (nikoliv komunikace samotné, tím se zabývá steganografie)
  - autentičnost (původnost, hodnověrnost) zprávy – příjemce má možnost zjistit původ zprávy
  - neodmítnutelnost – odesilatel nemůže popřít, že zprávu odeslal
  - integritu zprávy – příjemce má možnost zjistit, jestli během přenosu nedošlo je změně zprávy (úmyslné změně nebo vlivem technické poruchy)

Kryptografie vs. kryptologie

- *kryptoanalýza* . . . věda o luštění šifrovaných zpráv
- *kryptologie* . . . věda zahrnující kryptografii a kryptoanalýzu
- termín kryptografie se však často používá ve významu termínu kryptologie

Kryptografie – terminologie

- *otevřený text* (message, plaintext) – zpráva určená k odeslání
- *šifrování* – proces úpravy otevřeného textu, který ukryje jeho obsah; převede ho do tvaru, který *není srozumitelný*
- *zašifrovaný text, kryptogram* (ciphertext) – výsledek aplikace šifrování na otevřený text
- *dešifrování* – opačný proces k šifrování ion) – matematická funkce provádějící šifrování
- *dešifrovací funkce* (decryption function) – matematická funkce provádějící dešifrování
- *šifra* (cipher) – společné označení pro šifrovací a dešifrovací funkci
- *kanál* (channel) – komunikační spoj, např. Internet, LAN, apod.

Kryptografie

Terminologie:

- Alice – odesilatel (sender)
- Bob – příjemce (receiver)
- Eva (někdy také Oskar) – útočník, protivník (**eave**sdropper, adversary, bad guy)

- *kryptografický modul* – zařízení nebo program zajišťující šifrování, dešifrování, podpisování apod.
  - kryptografický modul zamýšleným způsobem komunikuje se svým okolím prostřednictvím vstupně/výstupních kanálů
  - činností kryptografického modulu vznikají postranní kanály – nežádoucí způsob výměny informací mezi modulem a okolím

Něco málo z historie

- kryptografie je v současnosti spojována s moderními komunikačními technologiemi, je však velmi starým oborem

Např.:

- 2000 let př.n.l., starověký Egypt – tajné hieroglyfy
- starověké Řecko – Řecká skytalé

Něco málo z historie

- 100-44 př.n.l., starověký Řím – Caesarova šifra
- římský historik Gaius Suetonius Tranquillus píše:

  „Existují také Caesarovy dopisy Cicerovi o známých věcech, ve kterých psal tajným písmem, pokud něco muselo být důvěrně sděleno. Změnil pořadí písmen tak, že nešlo zjistit jediné slovo. Pokud někdo chtěl toto rozluštit a poznat obsah, musel dosadit čtvrté písmeno abecedy, tedy D, za A, a podobně toto provést se zbývajícími písmeny."
- Caesarova šifra je jednoduchá monoabecední posouvací šifra
- nemění četnosti výskytu znaků
- jednoduché prolomit hrubou silou

Něco málo z historie

- 15. století, Leon Battista Alberti – italský architekt, historik umění a matematik
- vynalezl polyabecední šifru
- zkonstruoval šifrovací zařízení – *Formula* (používal se téměř 500 let)
- dva kotouče na jedné ose
- vnější kotouč (*Stabilis*) – abeceda otevřeného textu
- vnitřní kotouč (*Mobilis*) – abeceda šifrovaného textu
- pracuje jako jednoduchá posouvací šifra, nebo složitěji jako polyabecední šifra

Něco málo z historie

Něco málo z historie

- 16. století, Blaise de Vigenère – francouzský diplomat, rozvinutí polyabecední šifry
- roku 1586 vyšla jeho knihu *Traicté des Chiffres*, ve které popsal všechny doposud známé šifry
- přelom 19. a 20. století, Arthur Scherbius – německý vynálezce, elektrifikovaná verze Albertiho šifrovacího stroje: Enigma

Enigma

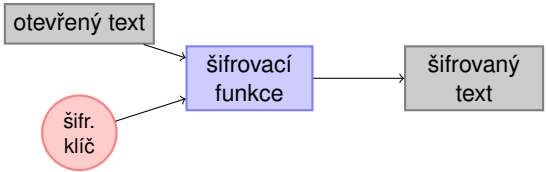
Omezené šifry

- bezpečnost šifrování je založena na utajení způsobu, jakým šifra (šifrovací a dešifrovací funkce) pracuje
- na první pohled dobrý nápad: utajíme šifrovací algoritmus, zvýšíme bezpečnost šifry
- z praktického pohledu nevýhodné
- uvažujeme skupinu uživatelů nějaké omezené šifry, pak
  - může dojít k odhalení principu činnosti šifry
  - odchodem jednoho člena skupiny je nutno šifru změnit
  - nemožnost normalizace – každá skupina si musí vytvořit svoje vlastní hardwarové a softwarové nástroje
  - nemožnost kontroly kvality – pokud ve skupině není skutečně dobrý kryptograf, skupina si nemůže být jistá kvalitou šifry
- dnes se nevyužívají (Enigma) $\Rightarrow$ memorandum:
  „Při posuzování bezpečnosti šifrovacího systému vycházíme z předpokladu, že nepřítel má přístroj k dispozici“

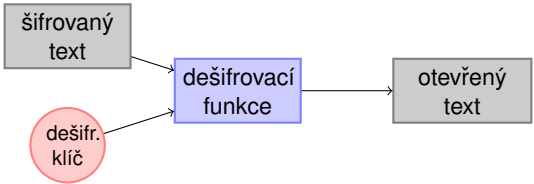
Šifry založené na klíči

- Kerckhoffův princip: bezpečnost šifry závisí pouze na utajení klíče, nikoliv na utajení šifrovací a dešifrovací funkce
- princip šifry může být zveřejněn (a tedy i standardizován)

Šifrovací proces:



Dešifrovací proces:



Šifry založené na klíči

- $\mathcal{M}$ ... konečná množina všech zpráv
- $\mathcal{C}$ ... konečná množina všech zašifrovaných zpráv
- $\mathcal{K}$ ... konečná množina všech klíčů
- $e : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$ ... šifrovací funkce
- $d : \mathcal{C} \times \mathcal{K} \to \mathcal{M}$ ... dešifrovací funkce

Definice (Claude E. Shannon)

Šifra založená na klíči je pětice $\langle \mathcal{M}, \mathcal{C}, \mathcal{K}, e, d \rangle$ taková, že pro libovolný šifrovací klíč $k_e \in \mathcal{K}$ a jemu odpovídající dešifrovací klíč $k_d \in \mathcal{K}$ platí

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

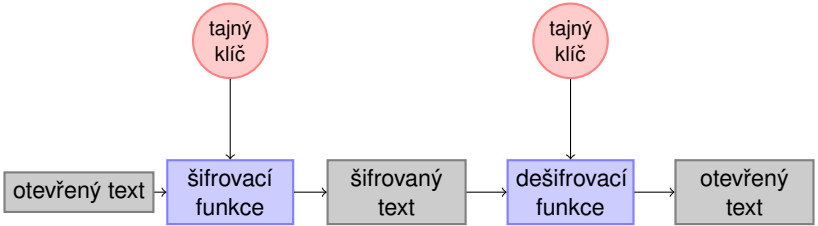
pro všechna $x \in \mathcal{M}$. Krátce budeme mluvit o šifře.

Symetrické šifry

- klíče pro šifrování a dešifrování jsou identické nebo mezi nimi existuje jednoduchý vztah (obousměrný)
- Alice a Bob tedy sdílí stejnou znalost (klíč) a oba umí šifrovat i dešifrovat
- jejich vztah je tedy symetrický, proto *symetrická* šifra
- příklad: všechny klasické šifry (např. posouvací, Vigenèrova), Enigma; z nových šifer např. RC2, DES a jeho varianty (např. Triple DES), AES, Blowfish, IDEA

Symetrické šifry

- postup:
  1. Alice a Bob se domluví na klíči
  2. Alice zašifruje zprávu pomocí klíče
  3. šifrovaná zpráva může být poslána Bobovi přes nezabezpečený komunikační kanál
  4. Bob dešifruje zprávu pomocí klíče
- odesilatel i příjemce musí udržovat klíč v tajnosti → často se také mluví o *šifrování s tajným klíčem* (*private-key cryptography*)



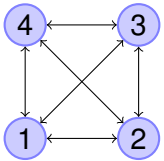
Symetrické šifry

Výhoda:

- šifrování i dešifrování je velmi rychlé

Nevýhody:

- bezpečnost
  - tajný klíč musí být distribuován mezi komunikujícími uživateli
  - nebezpečí odhalení tajného klíče třetí stranou
- velký počet klíčů, složitý key management
  - počet klíčů = počet všech komunikačních kanálů
  - jak rychle roste počet klíčů v závislosti na počtu uživatelů?



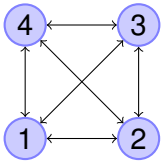
Symetrické šifry

Výhoda:

- šifrování i dešifrování je velmi rychlé

Nevýhody:

- bezpečnost
  - tajný klíč musí být distribuován mezi komunikujícími uživateli
  - nebezpečí odhalení tajného klíče třetí stranou
- velký počet klíčů, složitý key management
  - počet klíčů = počet všech komunikačních kanálů
  - počet klíčů roste kvadraticky, n uživatelů potřebuje $\frac{n \cdot (n-1)}{2}$ klíčů



- problémy řeší asymetrické šifrování

Jak vypadají šifrovací a dešifrovací funkce?

- otevřenou zprávu zakódujeme do posloupnosti čísel, např. s využitím ASCII
- šifrovací a dešifrovací funkce budou využívat aritmetické operace: sčítání, odčítání, násobení, atd.
- jak tyto operace definovat na konečných číselných množinách?
- využijeme modulární aritmetiku – anglická abeceda, 26 písmen
- písmena abecedy jsou kódovány: $\mathtt{A} \to 0$, $\mathtt{B} \to 1$, $\ldots$, $\mathtt{Z} \to 25$
- počítáme tedy v $\mathbb{Z}_{26}$
- všechny šifry se ovšem dají zobecnit na abecedy s n symboly ($\mathbb{Z}_n$)
- dva typy klasických šifer:
  - monoabecední (posouvací, afinní, substituční šifra) – prvky množin $\mathcal{M}$ a $\mathcal{C}$ jsou jednotlivé symboly abecedy, tzn. symbol abecedy je vždy mapován šifrovací funkcí na jediný symbol
  - polyabecední (Viegenèrova šifra) – prvky množin $\mathcal{M}$ a $\mathcal{C}$ jsou posloupnosti symbolů abecedy určité délky, tzn. symbol abecedy je mapován na jeden z několika symbolů

Klasické šifry (1 / 3)

- Posouvací šifra – monoabecední šifra; písmeno abecedy je mapováno na jiné písmeno téže abecedy

Definice

Nechť $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$. Pro $k \in \mathcal{K}$ definujeme šifrovací funkci a dešifrovací funkci

$$e(x, k) = x + k$$

$$d(y, k) = y - k.$$

- Afinní šifra – monoabecední šifra

Definice

Nechť $\mathcal{M} = \mathcal{C} = \mathbb{Z}_{26}$, $\mathcal{K} = \{\langle a, b\rangle \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} \mid \gcd(a, 26) = 1\}$. Pro $k = \langle a, b\rangle \in \mathcal{K}$ definujeme šifrovací funkci a dešifrovací funkci

$$e(x, k) = ax + b,$$

$$d(y, k) = a^{-1}(y - b).$$

Klasické šifry (2 / 3)

- Substituční šifra – monoabecední šifra; písmeno abecedy je mapováno na jiné písmeno téže abecedy podle zvolené permutace této abecedy

Definice

Nechť $\mathcal{M} = \mathcal{C} = \mathbb{Z}_{26}$, $\mathcal{K} = \{\pi \mid \pi \text{ je permutace } \mathbb{Z}_{26}\}$. Pro $\pi \in \mathcal{K}$ definujeme šifrovací funkci a dešifrovací funkci

$$e(x, \pi) = \pi(x),$$

$$d(y, \pi) = \pi^{-1}(y).$$

Klasické šifry (3 / 3) – Vigenèrova šifra

- polyabecední šifra:
  - prvky $\mathcal{M}$ jsou m-tice písmen abecedy
  - klíčem je také m-tice písmen abecedy; mluvíme o klíčovém slově (keyword)
  - písmeno abecedy může být mapováno na jedno z m písmen téže abecedy (pokud předpokládáme, že je klíč složen z m různých písmen)

Definice

Nechť $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}^m$, $m \in \mathbb{N}$. Pro klíč $\mathbf{k} = \langle k_1, \ldots, k_m \rangle \in \mathcal{K}$ definujeme šifrovací funkci

$$e(\mathbf{x}, \mathbf{k}) = \langle x_1 + k_1, \ldots, x_m + k_m \rangle,$$

a dešifrovací funkci

$$d(\mathbf{y}, \mathbf{k}) = \langle y_1 - k_1, \ldots, y_m - k_m \rangle$$

pro všechna $\mathbf{x} = \langle x_1, \ldots, x_m \rangle \in \mathcal{M}$, $\mathbf{y} = \langle y_1, \ldots, y_m \rangle \in \mathcal{C}$.

DES a AES

- soudobé symetrické šifry
- jedná se o iterační blokové šifry
- operace zvyšující bezpečnost kryptosystémů (C. Shannon)
  - *konfúze* – operace, která se snaží skrýt vztah mezi kryptogramem a použitým klíčem; v šifrách DES a AES je realizována operací substituce
  - *difúze* – znak otevřeného textu by měl ovlivňovat co nejvíce znaků kryptogramu; moderní šifry: změna jednoho bitu otevřeného textu vede v průměru ke změně poloviny bitů v kryptogramu
- DES (Data Encryption Standard) je založená na *Feistelově šifře*
- AES (Advanced Encryption Standard) založená na šifře *Rijndael*

Historie zavedení DES

- 70. léta, USA – nástup elektronických bankovních systémů, požadavek na bezpečný přenos informací mezi bankami
- 1973 – ministerstvo obchodu USA vyhlašuje soutěž, organizuje ji National Institute for Standards and Technologies (NIST)
- požadavky soutěže:
  - jednoduchý a bezpečný kryptosystém (bezpečnost neměla záviset na utajení šifrovacího algoritmu)
  - jednoduchá hardwarová implementace (kryptografický modul na malém a levném čipu)
- 1974 – opakování soutěže
- vítěz: firma IBM předložila kryptosystém založený na vlastním algoritmu Lucifer (využívá Feistelovu šifru)
- NIST, IBM a NSA se domlouvají na standardizaci tohoto systému
- vzniká DES (Data Encryption Standard), který může být bezplatně používán na území USA

Historie zavedení AES

- začátkem devadesátých let se stává zřejmým, že DES nebude do budoucna dostatečně bezpečná
- NIST vyhlašuje novou soutěž
- 5 finalistů:
  - MARS (IBM)
  - RC6 (RSA Security)
  - Twofish (UC Berkeley)
  - Serpent (spolupráce univerzit v Izraeli, Norsku a UK)
  - Rijndael [rejndál] (belgičtí kryptologové Joan Daemen a Vincent Rijmen)
- rok 2001, vítěz Rijndael
- délka klíče výrazně zvětšena (volitelně 128, 192, 256 bitů)
- AES (Advanced Encryption Standard) – standard založený na šifře Rijndael

AES v kostce

- šifra AES je založená na šifře Rijndael
- implementace použitím polynomiální aritmetiky nad prvočíslenými tělesy
- volitelná délka klíče: 128, 192 nebo 256 bitů $\Rightarrow$ délka bloku
- bajtově orientovaná
- šifrování probíhá iterativně v tzv. *rundách* ($r = 10, 12, 14$), počet dle délky klíče
- z hlavního klíče je odvozeno $r + 1$ klíčů

Implementace AES

- šifra navržena pro snadnou implementaci na 8bitových procesorech
  - využití např. ve *smart cards*
  - naivní implementace na 32 a 64bit procesorech pomalá → optimalizace, hacky, lookup tables, ...
- v dnešním HW jsou speciální chipy, FPGA, ASIC (síťové karty)
- paralelizace (pipeline), blok za blokem

AES: dnešní využití

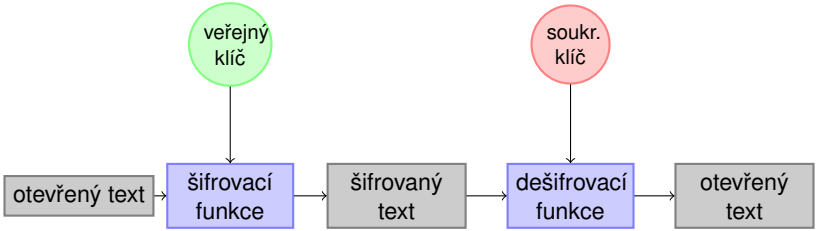
- WiFi encryption standard IEEE 802.11, IEEE 802.11i (WPA2)
  - 2018 – WPA3 použití AES-256
- SSH (hybridní šifrování)
- Skype
- SSL/TLS (hybridní šifrování)
- Šifrovací algoritmy pro pevné disky
- IPSec
- Signal Protocol
- Implementace přímo v HW (x86-64, Arm)

Asymetrické šifrování

- klíče pro šifrování a dešifrování nejsou stejné
- šifrovací klíč – *veřejný* klíč
- dešifrovací klíč – *soukromý* klíč
- stále existuje vztah mezi soukromým a veřejným klíčem, ale soukromý klíč nemůže být v rozumně krátké době odvozen z veřejného (jednosměrnost)
- soukromý klíč je držen v tajnosti, veřejný klíč ale může být veřejně distribuován – často se proto mluví 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:

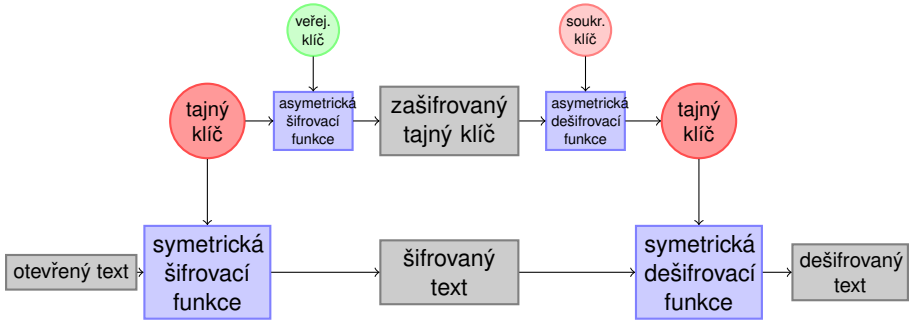
- bezpečnější
- není složitá správa klíčů

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

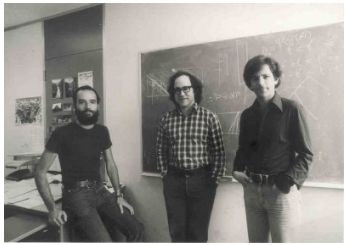
Hybridní šifrování

- 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



- algoritmus publikován roku 1977
- tentýž rok patentován (patent platil pouze pro USA do roku 2000)
- už roku 1973 vyvinul ekvivalentní systém Clifford Christopher Cocks (odtajněno 1998)

Modulární aritmetika

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

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

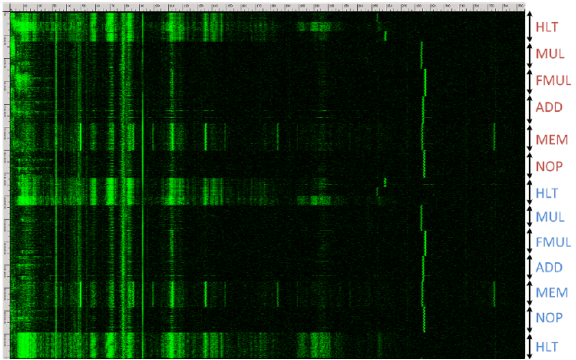
- při implementaci vznikají postranní kanály
- postranní kanál . . . nežádoucí způsob výměny informací mezi zařízením nebo programem implementujícím šifru a jeho okolím, např.:
  - časový postranní kanál
  - chybový postranní kanál (Daniel Bleichenbacher - 1998)
- postranní kanály se dají použít k prolomení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

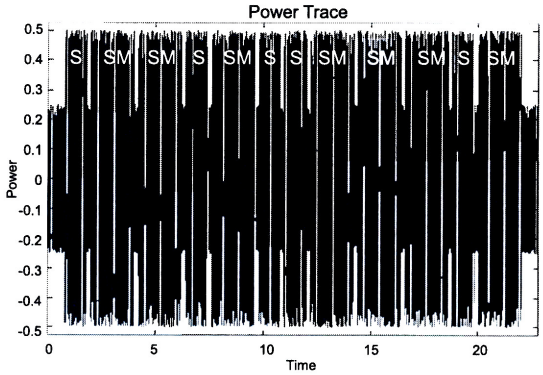
Poslouchání mikrofonem

- 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 četba

- Singh S. 2003. *Kniha kódů a šifer – Tajná komunikace od starého Egypta po kvantovou kryptografii*. Dokořán.
- Paar C. and Pelzl J. 2010. *Understanding Cryptography. A Textbook for Students and Practitioners*. Springer-Verlag.
  - AES – str. 87-117
  - RSA – str. 173-187
  - Generování prvočísel – str. 187-192