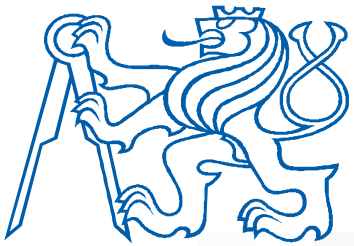
České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra telekomunikační techniky

A7B32KBE 6. přednáška

Asymetrické kryptosystémy I

Ing. Tomáš Vaněk, Ph.D. tomas.vanek@fel.cvut.cz



"Encryption software is expensive...so we just rearranged all the letters on your keyboard."

Osnova

Kryptosystémy veřejného klíče

- obecné informace
- Merkle-Hellmann (Knapsack)
- El-Gamal
- Diffie-Hellman
- RSA
- LUC

Příště:

- ECC
- srovnání IFP / DLP / ECDLP

Šifrovací algoritmy - rozdělení

symetrické

- blokové (DES, 3DES, IDEA, RC6, Blowfish, CAST, AES - Rijndael….)
- proudové (Enigma, SEAL, RC4, A5,…)

asymetrické

- IFP (RSA, Rabin-Williams, Lucas,…)
- DLP (DSA, Diffie-Hellman, El-Gamal, …)
- ECDLP (ECDSA, PSEC,…)

Šifrovací algoritmy - srovnání

Symetrické algoritmy

- dnes velmi rychlé algoritmy hodící se k šifrování velkých objemů dat
- malé (relativně) délky klíčů (128-bit, 256-bit)
- obě dvě strany musí znát stejný tajný klíč
- problém distribuce/změny klíčů v případě velkého množství komunikujících stran
- nelze prokázat totožnost autora zašifrovaných dat (data mohla zašifrovat kterákoliv strana vlastnící klíč)

Šifrovací algoritmy - srovnání

Asymetrické algoritmy

- pomalé (100-1000x)
- velké délky klíčů (512-bit, 1024-bit, 2048-bit)
- různé klíče pro šifrování a dešifrování - klíče spolu vzájemně souvisejí

Hybridní systémy

- využívají kladné vlastnosti obou předchozích skupin.
- zpráva je zašifrována symetrickým algoritmem s náhodně vygenerovaným klíčem a tento klíč je zašifrován asymetrickým algoritmem
- příklad - PGP

Šifrovací algoritmy - srovnání

Výhody symetrických algoritmů

- rychlost
- bezpečnost – u dobře navržených a implementovaných algoritmů je jediným možným útokem útok hrubou silou
- délky klíčů

Nevýhody symetrických algoritmů

- distribuce klíčů
- lze použít pouze k šifrování

Šifrovací algoritmy - srovnání

Výhody asymetrických algoritmů

- řeší problém s distribucí klíčů
- možnost realizovat i jiné činnosti než šifrování

Nevýhody asymetrických algoritmů

- rychlost
- délky klíčů
- obecně náchylné na „choosen-plaintext attack“

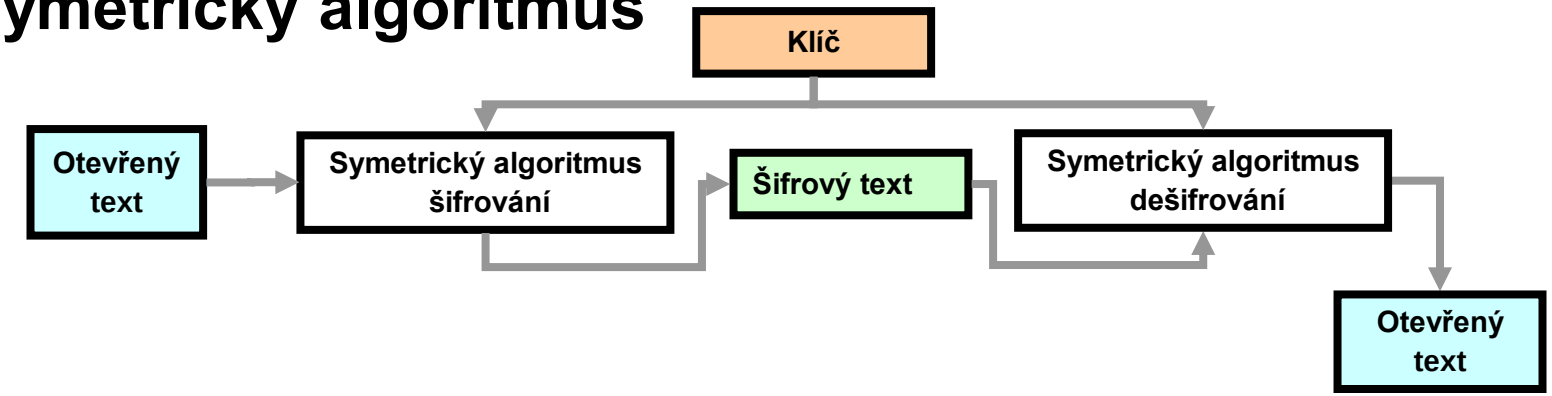
Použití veřejných kryptosystémů

Tři možné oblasti použití:

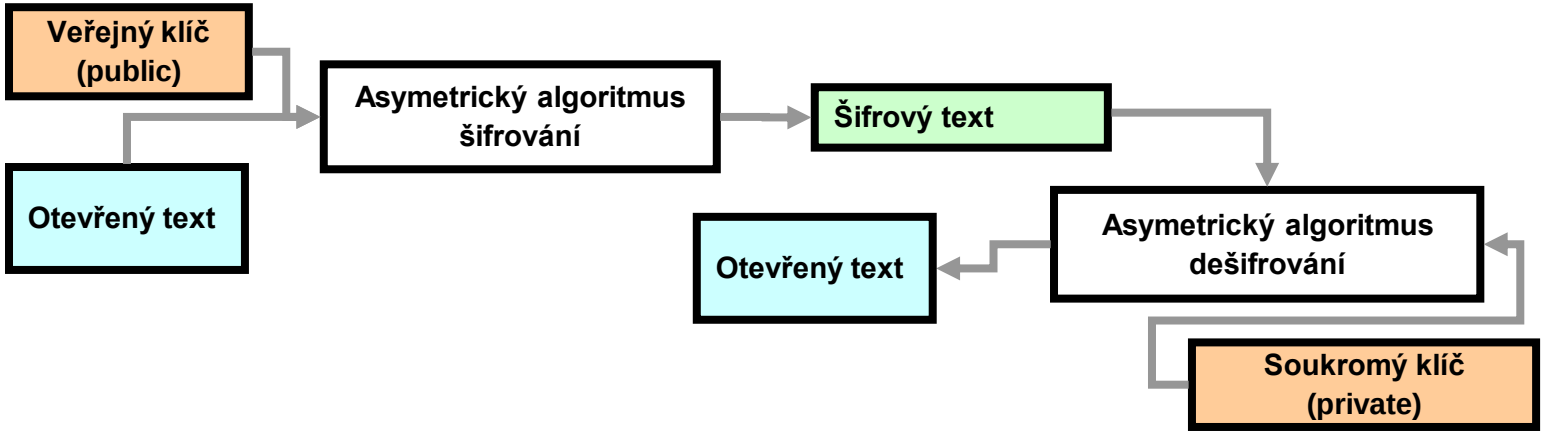
- **šifrování /dešifrování**
- **digitální podpisy**
- **výměna klíčů**

Některé algoritmy patří do všech třech kategorií, jiné pouze do jedné.

Symetrický algoritmus



Asymetrický algoritmus



Šifrování s asymetrickým algoritmem

1) Vygeneruji pár klíčů – veřejný (šifrovací) soukromý (dešifrovací).
   - Veřejný klíč může být volně dostupný.
   - Lze ho zveřejnit např. na Internetu.
   - Není žádný důvod ho tajit.
   - Soukromý (dešifrovací) klíčje potřeba uchovávat v tajnosti.
2) Pokud mi někdo chce poslat tajnou zprávu, zašifruje ji mým veřejným klíčem.
3) Pouze já (jakožto držitel tajného klíče) mohu dešifrovat zprávu svým soukromým (privátní) klíčem.

Šifrování s asymetrickým algoritmem

1) Vygeneruji dva klíče – soukromý (pro vytváření podpisu) a veřejný (pro ověřování podpisu).

- Veřejný klíč může být volně dostupný.
- Není žádný důvod ho tajit.
- Soukromý klíč je potřeba uchovávat v tajnosti.

2) Pokud chci podepsat zprávu, zašifruji ji svým soukromým klíčem.

3) Pokud chce někdo ověřit autorství zprávy, dešifruje ji mým veřejným klíčem. Protože pouze držitel tajného klíče mohl zprávu zašifrovat…

Certifikáty

Je potřeba vyřešit otázku :

„Jak poznám, že veřejný klíč, který někde získám (např. na Internetu) patří skutečně osobě, která to o sobě tvrdí?“

Řešením jsou tzv. certifikáty a certifikační autority (CA).

Detailní informace viz. přednáška o elektronickém podpisu.

Nezaměňovat podepisování a šifrování !

- Od roku 1976 bylo představeno mnoho kryptosystémů veřejného klíče
- Mnoho z nich není bezpečných, nebo jsou velmi nepraktické (např.: vyžadují příliš velké klíče, nebo ŠT je výrazně větší než OT)
- Pouze několik málo algoritmů je současně jak bezpečných tak i praktických.
- Některé algoritmy lze použít jen šifrování, některé jen k podepisování a některé k obojímu.
- Pouze tři algoritmy je možné použít jak pro šifrování, tak pro podepisování: RSA, ElGamal, a Rabinův.

Využití asymetrických kryptosystémů

- PGP – Pretty Good Privacy
- GPG – implementace OpenPGP
- SSH – Secure Shell
- SSL/TLS – Secure Socket Layer/Transport Layer Security
- IKE (Internet Key Exchange) – součást Ipsec
- ESP – Encapsulated Secure Payload – součást Ipsec
- …

Merkle-Hellmanův zavazadlový algoritmus

- 1978
- jeden z nejstarších asymetrických kryptosystémů
- objevil jej *Ralph Merkle* a *Martin Hellman*
- lze ho použít pouze na šifrování
- základem je „zavazadlový problém“ , který je NP-úplný
- původní Merkle-Hellmanův algoritmus byl patentován v USA i jinde ve světě
- patent vypršel 19.4.1997
- dnes se nepoužívá

Zavazadlový algoritmus

- Nechť je dána množina čísel $M_1, M_2, \ldots, M_n$, a suma S, spočtená podle vzorce:

  $$S=b_1M_1+b_2M_2+\ldots+b_nM_n \quad (b_i = 0 \text{ nebo } 1)$$

- 1→ dané číslo bude součástí zavazadla
- 0→ dané číslo nebude součást zavazadla

Příklad :

váhy jsou : 1, 5, 6, 11, 14, 20

chceme vytvořit zavazadlo o váze 22

Řešení: 5, 6, 11

Zavazadlový algoritmus

- Existují dva typy zavazadel. Pro jeden z nich je problém řešitelný v lineárním čase (superrostoucí zavazadlo), pro druhý v nepolynomiálním (normální zavazadlo).
  - superrostoucí posloupnost je taková, kde každý prvek je větší než součet všech předchozích prvků

Příklad:

(1,3,6,13,27,52) není superrostoucí posloupnost.

(1,2,4,9,19,45) je superrostoucí posloupnost.

Pokud je seznam vah superrostoucí posloupnost , pak je velmi jednoduché zjistit výsledné složení zavazadla.

Zavazadlový algoritmus

Příklad:

Mějme jednotlivé váhy =(2,3,6,13,27,52), a chci zavazadlo s celkovou váhou S = 70

Krok1. S=70

největší váha je 52<S, takže 52 tam bude

S=70-52=18, další váha v pořadí je 27

Krok2. S=18

další váha je 27>S, takže 27 není součást zavazadla

S=18, další váha je 13

Krok3. S=18

největší váha je 13<S, takže 13 je součástí zavazadla.

S=18-13=5

Zavazadlový algoritmus

Krok4. S=5

dalši váha je 6>S, takže 6 není součástí zavazadla..

S=5, další váha je 3

Krok5. S=5

další váha je 3<S, takže 3 je součástí zavazadla.

S=5-3=2, další váha je 2

Krok6. S=2

další váha je, 2=2, takže 2 je součástí zavazadla.

S=2-2=0→znamená, že jsme našli řešení.

Výsledkem je číslo: **10101**

Superrostoucí zavazadlo

Merkle-Hellmanův algoritmus:

- soukromý klíč je sekvence vah pro superrostoucí zavazadlo
- veřejný klíc je sekvence vah pro normální zavazadlo se stejnou váhou jako má odpovídající superrostoucí

Získání veřejného klíče ze soukromého

Abychom získali posloupnost pro normální zavazadlo (veřejný klíč) vezmeme superrostoucí zavazadlo (soukromý klíč), a vynásobíme všechny členy číslem *n* mod m.

$$M_i' = (M_i \times n) \bmod m$$

Modul m musí být větší než je součet všech čísel v posloupnosti

$$m > \Sigma M_i$$

Násobky by neměli mít žádné společné dělitele s modulem

$$\gcd(m,n) = 1$$

Zavazadlový algoritmus

Příklad :

superrostoucí posloupnost {2,3,6,13,27,52}.

zvolme m=105 > 103=2+3+6+13+27+52

n=31 , pro (105, 31)=1

pak 2 *31 mod 105 = 62

3 *31 mod 105 = 93

6 *31 mod 105 = 81

13 *31 mod 105 = 88

27 *31 mod 105 = 102

52 *31 mod 105 = 37

Normální posloupnost: {62,93,81,88,102,37}, která odpovídá vstupní posloupnosti {2,3,6,13,27,52}.

Zavazadlový algoritmus - šifrování

Příklad:

zpráva otevřeného textu: 011000 110101 101110

veřejný klíč ={62,93,81,88,102,37}

- 011000 odpovídá 93 + 81 = 174
- 110101 odpovídá 62 + 93 + 88 + 37 = 280
- 101110 odpovídá 62+81+88+102=333

Šifrový text bude (174,280,333)

Zavazadlový algoritmus - dešifrování

- Příjemce musí nejprve spočítat multiplikativní inverzi n^{-1} tak, že $n \cdot n^{-1} = 1 \pmod m$.
- Poté násobí každé číslo ŠT hodnotou $n^{-1} \bmod m$.
  $(\text{ŠT} \times n^{-1}) \bmod m$
- V našem případě superrostoucí zavazadlo=(2,3,6,13,27,52), m=105, a n=31. ŠT=174,280,333
  $n^{-1} = 61$

Řešení:

174*61 mod 105 = 9 = 3 + 6, což odpovídá 011000

280*61 mod 105 = 70 = 2 + 3 + 13 + 52, což odpovídá 110101

333*61 mod 105 = 48 = 2 + 6 + 13 + 27, což odpovídá 101110

Získaný OT má tvar = 011000 110101 101110.

Zavazadlový algoritmus - bezpečnost

- Shamir a Zippel nalezli bezpečnostní díru, v procesu transformace obyčejného zavazadla na superrostoucí.
- Podařilo se jim z ŠT získat první a poslední bit OT.
- Poté Shamir ukázal, že tento kryptosystém může být za určitých okolností prolomen (obyčejné zavazadlo vygenerované ze superrostoucího není „obyčejné“ ale pouze tak vypadá a jedná se o speciální případ superrostoucího)

ElGamal

Obecná definice problému

- mějme dánu konečnou cyklickou grupu ***G*** řádu *r*, její generátor α a prvky α^a, α^b pro neznámé hodnoty *a*, *b*
- cílem je najít prvek $\beta = \alpha^{ab}$
- ab je diskrétní logaritmus o základu α z β

- bezpečnost závisí na obtížnosti výpočtu diskrétního logaritmu v konečném poli GF
- není chráněn patenty
- existují dva různé (!) algoritmy podobného jména
  - ElGamal signature scheme
    - varianta DSA
  - ElGamal encryption scheme
    - založeno na DH protokolu
    - používáno v PGP

ElGamal - podepisování

Generování klíčů:

Veřejný klíč: trojice čísel (p,q,y)

p: prvočíslo (může být sdíleno mezi skupinou uživatelů)

g: náhodně zvolené číslo g < p (může být sdíleno mezi skupinou uživatelů)

y: $y \equiv g^x \bmod p$

Soukromý klíč: náhodně zvolené číslo x, x < p

ElGamal - podepisování

Podepisování:

1) vygenerujeme náhodné *k*

   - toto číslo je nutné držet v tajnosti

   - gcd(k,p-1)=1

2) spočteme $a \equiv g^k \bmod p$

3) pomocí rozšířeného Euklidova algoritmu vyřešíme rovnici $M \equiv (xa + kb) \bmod (p-1)$ pro číslo *b.*

   $b \equiv (M - xa)k^{-1} \bmod (p-1)$

4) podpisem jsou čísla *a* a *b*

Ověření podpisu:

Podpis je platný, pokud $y^a a^b \bmod p = g^M \bmod p$

ElGamal - šifrování

Příprava klíčů

Strana A vygeneruje multiplikativní cyklickou grupu G, řádu q s generátorem g; náhodné číslo x $\in\{0,..q-1\}$ a spočítá $h = g^x$

Veřejný klíč - $\{G,q,g,h\}$

Soukromý klíč - x

Šifrování

1) strana B si zvolí náhodné číslo y $\in\{0,..q-1\}$
2) spočítá: $c_1=g^y$
3) spočítá sdílené tajemství $s=g^y$

ElGamal - šifrování

4) převede zprávu m na m', která leží v G

5) spočítá $c_2 = m' \cdot s$

6) ŠT je $(c_1; c_2)$

Dešifrování

1) strana A spočítá $s = c_1^x$
2) spočítá $m' = c_2 \cdot s^{-1}$

ElGamal – bezpečnost, efektivita

- bezpečnost závisí na vlastnostech podložní grupy *G* způsobu doplňování bloků OT.
- ElGamal není odolný proti útokům se znalostí ŠT
  - má vlasnost označovanou jako „poddajnost / tvárnost“ (malleability)
  - pro daný ŠT $(c_1;c_2)$ lze bez znalosti původního OT_1 sestrojit dvojici $(c_1;2c_2)$ patřící OT_2, kde $OT_2=2OT_1$
  - toto se týká i jiných asym. kryptosystémů
- proto je nutné použít vhodné výplňové schema (padding)
- není příliš efektivní – ŠT je dvakrát větší než OT
- šifrování potřebuje dvě umocňování – lze předpočítat
- dešifrování vyžaduje pouze jedno umocňování

Diffie-Helmannův protokol

- objevil ho M.J. Williamson (GCHQ) a nezávisle *Whitfield Diffie* a *Martin Hellman* (1976) a navíc ještě nezávisle na nich *Ralph Merkle*

- zcela první kryptosystém veřejného klíče
- nelze použít k šifrování / podepisování
- vhodný pouze pro „výměnu klíčů“
- používá se k ustanovení sdíleného (symetrického) klíče
- bezpečnost závisí na obtížnosti řešení **diskrétního logaritmu**

Diffie-Helmannův protokol

Nechť *p* je prvočíslo a *g* je **generátor**

- pro každé $x \in \{1,2,\ldots,p-1\}$ existuje n takové, že $x = g^n \bmod p$

1. Alice vygeneruje tajné náhodné číslo a
2. Bob vygeneruje tajné náhodné číslo b
3. Alice odešle Bobovi $g^a \bmod p$
4. Bob pošle Alici $g^b \bmod p$
5. Oba si spočítají sdílenou tajnou hodnotu

$(g^b \bmod p)^a \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p$

6. Takto získané číslo lze použít jako sdílený tajný klíč

Diffie-Helmannův protokol

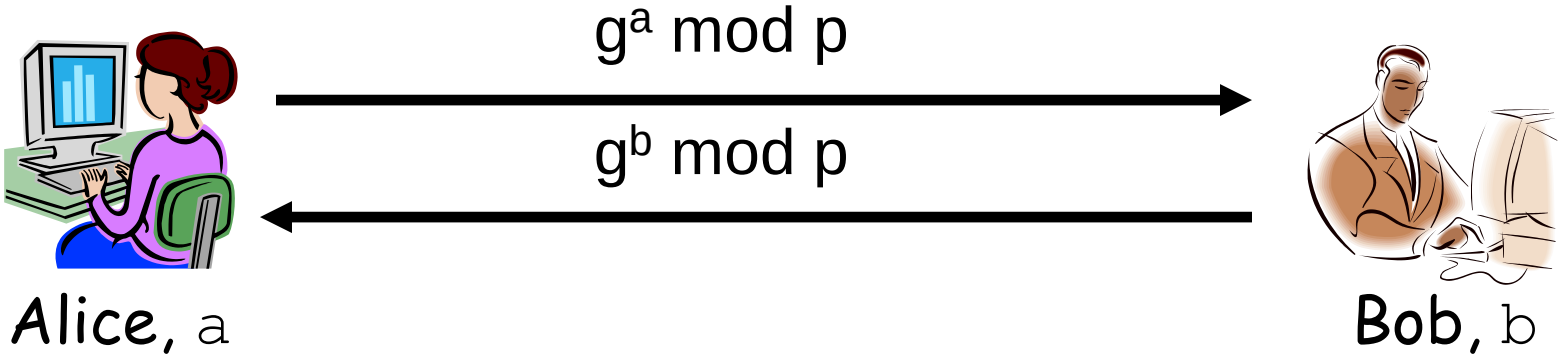
- Bob a Alice použijí $g^{ab} \bmod p$ jako symetrický klíč
- Útočník může zjistit $g^{a} \bmod p$ a $g^{b} \bmod p$

Poznámka: $g^{a}\, g^{b} \bmod p = g^{a+b} \bmod p \neq g^{ab} \bmod p$

- Systém je prolomen, pokud útočník zjistí hodnotu a nebo b
- Systém je také možné prolomit vyřešením diskrétního logaritmu

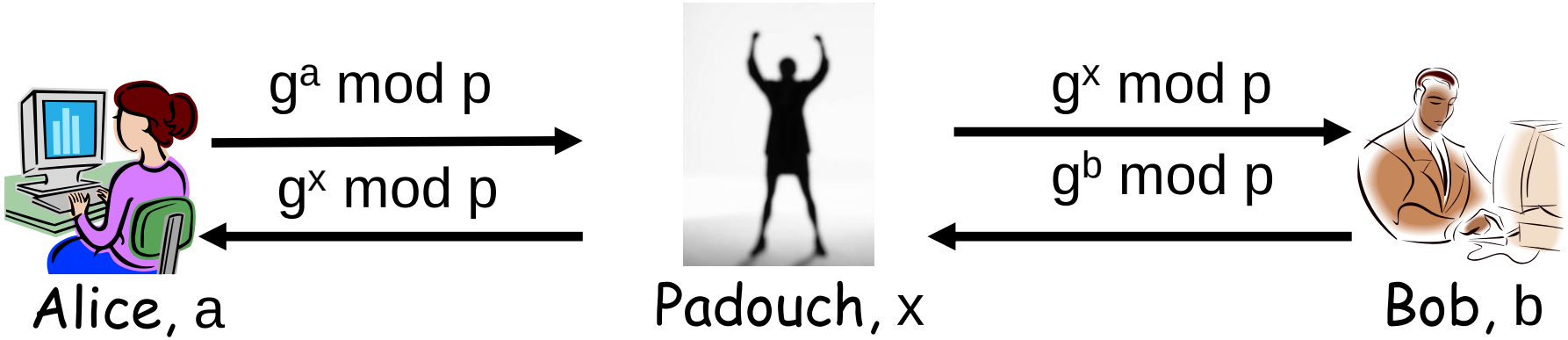
Diffie-Helmannův protokol

- **Veřejný klíč:** g, p
- **Tajný klíč:** pro Alici a, pro Boba b



- Alice spočítá $(g^b)^a = g^{ba} \bmod p = g^{ab} \bmod p$
- Bob spočítá $(g^a)^b = g^{ab} \bmod p$
- $K = g^{ab} \bmod p$ je možné použít jako klíč pro symetrickou šifru
- potencionální útočník může zachytit pouze g,p, g^a g^b

DH protokol a útok Man-in-the-middle



- Padouch sdílí tajemství „g^{ax} mod p“ s Alicí
- Padouch sdílí tajemství „g^{bx} mod p“ s Bobem
- Alice ani Bob nemají ani potuchy, že nekomunikují přímo ale prostřednictvím Padoucha

DH protokol a útok Man-in-the-middle

Jak zabránit útoku MiM ?

- Šifrovat DH výměnu pomocí sdíleného tajemství
- Šifrovat DH výměnu pomocí veřejného klíče
- Podepsat DH výměnu pomocí soukromého klíče
- Jinak ?

V případě použití protokolu DH je nutné počítat s možností MitM útok.

RSA

Pojmenován po svých třech vynálezcích:

Ron Rivest

Adi Shamir

Leonard Adleman

- 1976 Rivest, Shamir a Adleman způsobili doslova revoluci v kryptografii
- zřejmě nejznámější kryptosystém veřejného klíče
- pomocí RSA je možné jak šifrovat, tak i podepisovat

- v roce 1997 vyšlo najevo, že v již roce 1973 objevil Clifford Cocks (GCHQ - Government Communications Headquarters) možnost konstrukce kryptosystémů veřejného klíče na stejném principu

RSA

- založen na umocňování celých čísel modulo prvočíslo
- používá velká prvočísla (více než 100 dekadických míst ~1024 bitů)
- dvě prvočísla *p* a *q*, které mají *x* a *y* dekadických čísel je možné podle vynásobit s obtížností $O(log_2 x \cdot log_2 y)$ bitových operací
- opačná úloha tj. pro dané $n = p \cdot q$ najít prvočíselné dělitele je **výrazně** těžší.
- toto tvrzení však dodnes nebylo dokázáno (tj. je možné, že k vyřešení IFP není nutné provádět faktorizaci)
- složitost při šifrování je $O((\log n)^3)$
- faktorizace pokusným dělením zabírá $O(e^{\log n \cdot \log \cdot \log n})$ operací
- nejefektivnější faktorizační metoda (Brent-Pollardova ρ-metoda) vyžaduje $O(\frac{e^{\sqrt{2 \ln p \ln \ln p}}}{\ln p})$

RSA – generování klíčových párů

- Nechť p a q jsou velká a náhodně zvolená prvočísla
- Číslo $N = pq$ nazveme modul
- Zvolíme e, které je relativním prvočíslem k $\varphi(N)=(p-1)(q-1)$
- Nalezneme d takové, že $ed = 1 \bmod \varphi(N)$

- Veřejný klíč je dvojice (N,e)

- Soukromý klíč je dvojice (N,d)

RSA – šifrování a dešifrování

Šifrování zprávy M: $C = M^e \bmod N$

Dešifrování zprávy M: $M = C^d \bmod N$

N, e jsou veřejně známé.

Pokud útočník je schopný najít rozklad N, pak je jednoduché najít *d*, protože ed = 1 mod (p-1)(q-1)

Rozklad modulu je cesta k prolomení RSA.
Není jisté, zda-li je faktorizace jediná možnost jak prolomit RSA.

Proč RSA funguje?

- Dáno $C = M^e \bmod N$

  Chceme dokázat, že $M = C^d \bmod N = M^{ed} \bmod N$

- Použijeme Eulerův teorém

  Pokud *x* je k *n* relativní prvočíslo, pak platí, že

  $$x^{\varphi(n)} = 1 \bmod n$$

- Fakta:

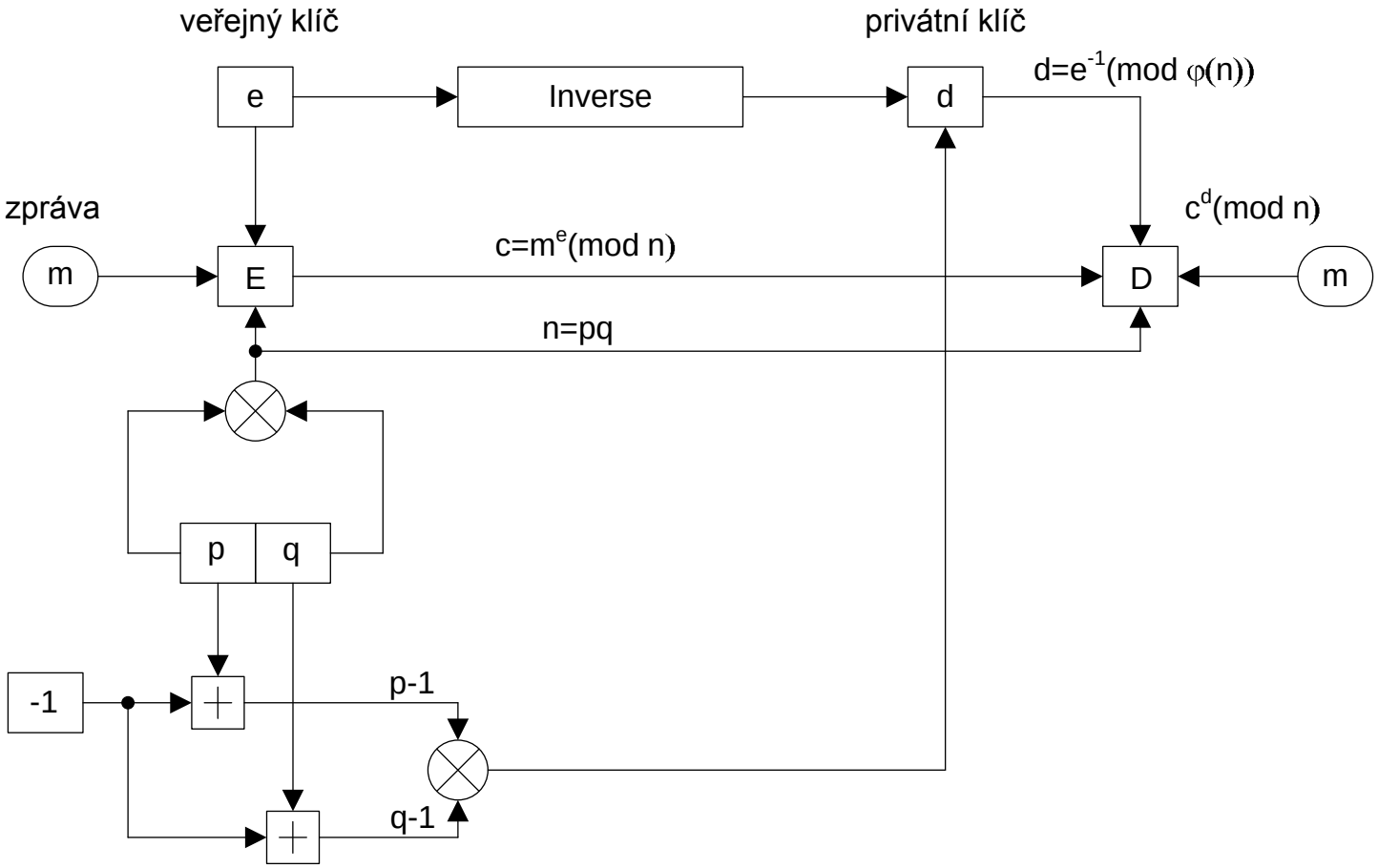
  $$ed = 1 \bmod (p-1)(q-1)$$

  $\varphi(N) = (p-1)(q-1) \Rightarrow ed - 1 = k\varphi(N)$ pro nějaké k

- Pak

  $$M^{ed} = M^{(ed-1)+1} = M \cdot M^{ed-1} = M \cdot M^{k\varphi(N)} = M \bmod N$$

Algoritmus RSA



Příklad šifrování pomocí RSA

Zvolíme dvě prvočísla: $p=17$ & $q=11$

1. Spočteme $n = pq = 17\ 11 = 187$
2. Spočteme $\varphi(n)=(p-1)(q-1)=16\ 10=160$
3. Zvolíme e: $\gcd(e,160)=1$; např.: $e=7$
4. Spočteme d: $de=1 \bmod 160$ a $d < 160$
   Hodnota $d=23$, protože $23\ 7=161=10\ 160+1$
5. Veřejný klíč $KU=\{7,187\}$
6. Tajný klíč $KR=\{23,178\}$

RSA - příklad

- Máme zprávu $M = 88$ (platí, že $88<187$)
- šifrování:

  $C = 88^7 \bmod 187 = 11$
- dešifrování:

  $M = 11^{23} \bmod 187 = 88$

RSA

- před šifrováním je potřeba upravit text – formátovací pravidla - PKCS#1 (http://www.rsasecurity.com/rsalabs/node.asp?id=2124)
- jinak hrozí, že pro vybrané vstupní texty M bude zašifrovaný text lehce prolomitelný
- bezpečnost RSA závisí na obtížnosti rozkladu modulu N

Na stejném zařízení je

- při HW implementaci :
  - RSA cca 1000x pomalejší nežli DES.
- při SW implementaci :
  - RSA cca 100x pomalejší nežli DES.

RSA - SW urychlení

Veřejný klíč :

- RSA šifrování je výrazně rychlejší, když si zvolíte vhodnou hodnotu e.
- nejčastější volby jsou 3, 17, a 65537 ($2^{16} + 1$).
- na druhou stranu toto snižuje bezpečnost (když tu samou zprávu pošlu více lidem, lze využít CRT a získat tak soukromý klíč)
- Pro útok pomocí CRT je potřeba minimálně *e* identických textů zašifrovaných stejným klíčem
- Obrana - přidávání různého množství náhodných dat různým příjemcům

RSA - SW urychlení

Soukromý klíč :

- Operace se soukromým klíčem je možné urychlit pomocí CRT (čínská věta o zbytcích)
- Dopředu si vypočtu a uložím p, q, d mod (p-1), d mod (q-1), a q^{-1} mod p
- pokud je soukromý klíč malý do cca ¼ n a současně $e < n$, pak je možné ho získat → volte velké soukromé klíče

Efektivita RSA

RSA s velkými moduly je neefektivní

- Se vzrůstající délkou modulu roste bezpečnost pomalu, ale výpočetní nároky rychle.
  - složitost RSA-1024 je srovnatelná se symetrickou šifrou s klíčem délky 80bitů
  - složitost RSA-2048 přibližně odpovídá klíči délky 112 bitů (3-DES)
  - složitost RSA-3072 přibližně odpovídá klíči délky 128 bitů (např. AES-128)
  - složitost RSA-7680 přibližně odpovídá klíči délky 192 bitů (např. AES-192)
  - složitost RSA-15,380 přibližně odpovídá klíči délky 256 bitů (např. AES-256)

Efektivita RSA

- Výkon RSA s velkými moduly je velmi nízký
- Čas nutný k podepisování roste s třetí mocninou délky klíče
  - K výpočtu podpisu pomocí RSA-2048 je potřebuje přibližně 8x delší čas než s RSA-1024
    - Např. – 60ms pro RSA-1024 se změní na 480 ms pro RSA-2048
    - RSA-15,360 by zabrala 3375-ta násobek trvání RSA-1024 (~ 200 sekund)
- Existuje řešení pro bezpečné délky klíčů a rozumné výpočetní nároky – algoritmy ze skupiny “Suite B”

RSA - faktorizace

Německý tým faktorizoval RSA-640 - MathWorld Headline News

http://mathworld.wolfram.com/news/2005-11-08/rsa-640/

- realizováno pomocí GNFS (general number field sieve)
- tři měsíce byly získávány potřebné vztahy
- 45 dní zabralo řešení vzniklé soustavy
- výpočet prováděl cluster 80 počítačů (AMD Opteron 2,2 GHz)

Číslo	Počet číslic	Odměna	Faktorizováno
RSA-100	100		04/1991
RSA-110	110		04/1992
RSA-120	120		06/1993
RSA-129	129	$100	04/1994
RSA-130	130		10.4.1996
RSA-140	140		2.2. 1999
RSA-150	150		16. 4. 2004
RSA-155	155		22.8. 1999
RSA-160	160		1.4. 2003
RSA-200	200		9.5. 2005
RSA-576	174	$10,000	3. 12. 2003
RSA-640	193	$20,000	4.11. 2005
RSA-704	212	$30,000	Doposud se
RSA-768	232	$50,000	12.12.2009
RSA-896	270	$75,000	nepodařilo
RSA-1024	309	$100,000	faktorizovat
RSA-1536	463	$150,000	Čeká možná
RSA-2048	617	$200,000	na Vás!

Největší faktorizovaný RSA modul

RSA-768 = 123018668453011775513049495838496272
077285356959533479219732245215172640050726365751
874520219978646938995647494277406384592519255732
630345373154826850791702612214291346167042921431
160222124047927473779408066535141959745985690214
3413

RSA-768 = 334780716989568987860441698482126908
177047949837137685689124313889828837938780022876
147116525317430877378144679994893674604366679959042824463379962795263227915816434308764267603228381573966651127923337341714339681027009279873630891 7

Metody faktorizace prvočíselných modulů

- **Pokusné dělení** (Trial Division)
- Metoda řetězových zlomků (Continued Fraction Method)
- Metoda p-1
- Metoda p+1
- **Pollardova ρ metoda**
- Kvadratické síto (Quadratic Sieve)
- Síto číselného pole (Number Field Sieve)
- **Zobecněné síto číselného pole (General NFS)**
- Metoda eliptických křivek

Rabinův kryptosystém

- Bezpečnost závisí na obtížnosti nalezení kvadratických kořenů mod velké složené číslo
- První asymetrický kryptosystém u kterého se povedlo prokázat, že jeho „problém“je stejně těžký jako IFP

- nepoužívá se
- nevhodný pro šifrování náhodných dat

Rabinův kryptosystém

Implementace :

p,q : prvočísla taková, že $p \equiv 3 \pmod 4$, $q \equiv 3 \pmod 4$

soukromý klíč : *p* a *q*

veřejný klíč: $n = p \cdot q$

Šifrování : zpráva M (<n)

$C = M^2 \bmod n$

Dešifrování :

$m_1 = C^{(p+1)/4} \bmod p$

$m_2 = (p - C^{(p+1)/4}) \bmod p$

$m_3 = C^{(q+1)/4} \bmod q$

$m_4 = (q - C^{(q+1)/4}) \bmod q$

Rabinův kryptosystém

zvolíme $a = q(q^{-1} \bmod p)$

$b = p(p^{-1} \bmod p)$

4 možná řešení :

$M_1=(am_1+bm_3) \bmod n$

$M_2=(am_1+bm_4) \bmod n$

$M_3=(am_2+bm_3) \bmod n$

$M_4=(am_2+bm_4) \bmod n$

Pokud je text normální zpráva, není problém ji identifikovat.

Pokud by byla zašifrována náhodná data (např. pro symetrickou šifru) není možné rozpoznat, které řešení je správné

LUC

- Zobecnění RSA, které používá různé permutace polynomů místo umocňování.
- Skupina novozélandských kryptologů si nechala v roce 1993 patentovat toto šifrovací schéma
- Základem jsou Lucasova čísla – posloupnost podobná Fibonacciho posloupnosti

n-té Lucasovo číslo $V_n(P,1)=PV_{n-1}(P,1)-QV_{n-2}(P,1)$, kde P,Q jsou nesoudělná čísla

LUC

Generování páru klíčů (soukromý / veřejný)

1) zvolíme prvočísla p,q
2) spočteme n=pq
3) šifrovací klíč e: zvolíme náhodně, ale tak aby byl nesoudělný s p-1,q-1,p+1,q+1
4) Čtyři možné dešifrovací klíče:

$d = e^{-1} \bmod (\mathrm{lcm}((p+1),(q+1)))$

$d = e^{-1} \bmod (\mathrm{lcm}((p+1),(q-1)))$

$d = e^{-1} \bmod (\mathrm{lcm}((p-1),(q+1)))$

$d = e^{-1} \bmod (\mathrm{lcm}((p-1),(q-1)))$

LUC

- Veřejný klíč: d,n
- Soukromý klíč: e,n

Šifrování:

$C = V_e(P,1) \pmod n$ P…zpráva, která se šifruje

Dešifrování:

$P = V_d(P,1) \bmod n$, spolu se správným d

LUC je přinejlepším stejně bezpečný jako RSA.

McEliece

- objeven Robertem McEliece v roce 1978
- založen na algebraické teorii kódování
- používá třídu opravných kódů pojmenovaným **Goppovi kódy**
- hlavní myšlenka: sestrojit Goppa kód a převést ho na obecný lineární kód
- princip podobný jako u Merkle-Hellmanova zavazadla
- není příliš praktický:
  - veřejný klíč je velmi velký: 2^{19} bitů !!!
  - ŠT je dvakrát větší než OT

http://en.wikipedia.org/wiki/Goppa_code
http://entropy.stop1984.com/files/ctcmcel.html

Dotazy