Seznámení s asymetrickou kryptografií, díl 1.

Ing. Tomáš Rosa
ICZ a.s., Praha
Katedra počítačů, FEL, ČVUT v Praze
tomas.rosa@i.cz

Osnova přednášky

- Základní principy
  - pojem *bezpečnost*
  - související (snad) složité úlohy
  - role jednosměrných funkcí
  - schéma vs. transformace
- RSA
  - základní popis
  - využití Čínské věty o zbytku - CRT
  - kódování šifrovaných zpráv
- D-H protokol
  - dohoda na klíči
- ElGamal
  - základní popis
  - souvislost s D-H protokolem
- Příklad praktického nasazení
  - SSL/TLS - příklad ustavení spojení

O pojmu *bezpečnost*

Poznámka o hodnocení kryptografické bezpečnosti

- Nepodmíněná bezpečnost
  - lze dokázat, že schéma nelze prolomit (prolomení definujeme jako ztrátu některé z proklamovaných vlastností – zajištění důvěrnosti, integrity, ...) bez ohledu na prostředky útočníka
  - zvláštní třída: *absolutní bezpečnost* (u šifer *perfect secrecy*)
    - útočník nezíská interakcí se systémem žádnou novou užitečnou informaci
- Podmíněná bezpečnost
  - důkaz o neprolomitelnosti se opírá o omezení prostředků útočníka
  - nejčastěji se užívá výpočetně-složitostní přístup: útočník je omezen co do velikosti a růstu své výpočetní síly
  - *prokazatelná bezpečnost* - existuje důkaz o tom, že schopnost provést útok znamená schopnost řešit problém, u kterého se obecně uznává jeho vysoká složitost
  - pojem *složitost* lze zobecnit na libovolné fyzické prostředky

K bezpečnosti asymetrické kryptografie

- Asymetrická kryptografie nemůže být bezpodmínečně bezpečná
  - veřejný klíč nese dostatek informace pro určení klíče privátního
  - patrně platí i pro kvantovou asymetrickou kryptografii (zatím nepříliš rozvinuta)
    - nacházíme například podmínky omezující počet kopií veřejného klíče
- Nejlepší výsledek: *prokazatelná bezpečnost*
  - je dokázána *výpočetní ekvivalence* úlohy luštění s jinou úlohou, u které se obecně uznává vysoká složitost
  - příklad: RSA
    - snaha o prokazatelnou bezpečnost vzhledem k úloze faktorizace
    - taková bezpečnost nebyla dosud dokázána
  - zvláštní význam v kontextu *Random Oracle Model* (ROM)

Problém faktorizace

- Obecný případ
  - mějme dáno celé číslo n, cílem je nalézt jeho zápis ve tvaru: $n = p_1^{e1} p_2^{e2} \ldots p_k^{ek}$, kde p_i je prvočíslo a e_i je celé číslo
- Případ RSA (n zde nazýváme *modul*)
  - speciální případ, $k = 2$ (*splitting*)
  - *multi-prime RSA*: $k > 2$, avšak obvykle je známo, kolik faktorů modul obsahuje
- Řešení
  - metoda NFS
  - TWINKLE, TWIRL (teoreticky)
  - kvantové počítače (teoreticky)

  } Metody typu ***random square***

Problém RSA

- Pro praktické účely definován na okruhu $\boldsymbol{Z}_n$, kde n je velké celé složené číslo
- Cíl: Pro daná n, $c \in \boldsymbol{Z}_n$ a e, splňující $\gcd(e, \phi(n)) = 1$, nalézt $m \in \boldsymbol{Z}_n$ takové, aby:
  - $m^e \equiv c \pmod n$
- Dosud nebylo prokázáno, že by tato úloha byla ekvivalentní úloze faktorizace modulu n
- Jiná úloha: Pro výše zadané hodnoty n a e nalézt celé číslo d takové, že:
  - $ed \equiv 1 \pmod{\phi(n)}$
  - tato úloha **je** (pravděpodobnostně) ekvivalentní úloze faktorizace n
  - často nesprávně považována za samotný problém RSA, ekvivalence zde ovšem **nebyla** dosud dokázána

Problém diskrétního logaritmu

- Obecná definice
  - mějme dánu konečnou cyklickou grupu $\boldsymbol{G}$ řádu r, její generátor α a prvek $\beta \in \boldsymbol{G}$
  - cílem je najít celé číslo x, $0 \le x \le r-1$, takové, že $\beta = \alpha^x$
  - píšeme také $x = \log_\alpha \beta$
- Speciální případy používané v kryptografii
  - $\boldsymbol{G} = \boldsymbol{Z}_p^*$, označujeme jej jako DLP
  - $\boldsymbol{G} = E(\boldsymbol{F}_q)$, označujeme jej jako ECDLP
- Řešení
  - metoda NFS (jako rozšíření algoritmu Index-Calculus)
  - Pollardovy metody ró a lambda, Pohligův-Hellmanův algoritmus
  - TWINKLE, TWIRL (teoreticky)
  - kvantové počítače (teoreticky)

Problém Diffieho-Hellmana

- Obecná definice
  - mějme dánu konečnou cyklickou grupu $\boldsymbol{G}$ řádu r, její generátor α a prvky α^a, α^b pro neznámé hodnoty a, b
  - cílem je najít prvek $\beta = \alpha^{ab}$
- Speciální případy
  - $\boldsymbol{G} = \boldsymbol{Z}_p^*$, označujeme jej jako DHP
  - $\boldsymbol{G} = E(\boldsymbol{F}_q)$, označujeme jej jako ECDHP
- Dosud nebylo prokázáno, že by tato úloha byla ekvivalentní úloze diskrétního logaritmu na grupě $\boldsymbol{G}$

Aby problém byl PROBLÉM

- Faktorizace, RSA problém
  - prvočísla tvořící modul n musí být generována nezávisle a zhruba stejně velká
  - $d > n^{1/2}$
- (EC)DLP, (EC)DHP
  - řád grupy musí obsahovat alespoň jedno velké prvočíslo (160 bitů a více)
  - další specifické požadavky vznikají při použití eliptických křivek

Kryptografické systémy vs. problémy

- RSA: faktorizace, **neprokázáno**
- DSA: DLP, **neprokázáno**
- ECDSA: ECDLP, **neprokázáno**
- D–H protokol: (EC)DLP, **neprokázáno**
- El Gamal: (EC)DLP, **neprokázáno**
- Rabinův systém: faktorizace, **prokázáno**
  - prakticky se nepoužívá, mimo jiné pro silnou náchylnost k útokům s voleným šifrovým textem
    - obecně však takovou náchylnost musíme u prokazatelně bezpečných systémů očekávat
    - čelí se jí způsobem použití – formátování zpráv, …

Asymetrická kryptografie -elementární principy- (1)

- Jednosměrná funkce
  - f: $X \to Y$ nazveme jednosměrnou funkcí, když:
    - pro každé $x \in X$ je výpočetně snadné spočítat hodnotu $y = f(x)$
    - pro náhodně zvolený obraz $y \in Y$ je výpočetně neschůdné najít $x \in X$ tak, že $f(x) = y$

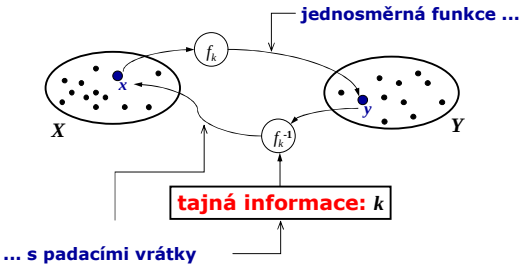
Asymetrická kryptografie -elementární principy- (2)

- Jednosměrná funkce s padacími vrátky
  - f_k: $X \to Y$ nazveme jednosměrnou funkcí s padacími vrátky (k), když f_k je jednosměrná s takovou vlastností, že při znalosti určité dodatečné informace (k) je pro každý obraz $y \in Y$ výpočetně snadné najít $x \in X$ takové, že $f_k(x) = y$, tedy $x = f_k^{-1}(y)$
  - v kryptografii představuje dodatečnou informaci k nejčastěji *privátní klíč* nebo jeho přímý derivát

Asymetrická kryptografie
-elementární principy- (3)

- Předpokládejme
  - funkce f_k je spojena s určitým konkrétním subjektem - uživatelem *A*
    - je to jeho veřejný klíč
  - informace o padacích vrátkách *k* je známa pouze subjektu *A*
    - je to jeho privátní klíč

Asymetrická kryptografie
-elementární principy- (4)



Asymetrická kryptografie
-elementární principy- (5)

- Z vlastností f_k plyne
  - z pouhé znalosti f_k nelze najít výpočetně schůdnou (parciálně) inverzní funkci f_k^{-1}
  - čili speciálně: ze znalosti veřejného klíče nelze nalézt klíč privátní
  - tedy nakonec prakticky: ten, kdo zná pouze veřejný klíč, **dokáže zašifrovat** libovolnou zprávu, ale **nedokáže** náhodně vybraný šifrový text sám **odšifrovat**

Asymetrická schémata
-inicializace instance-

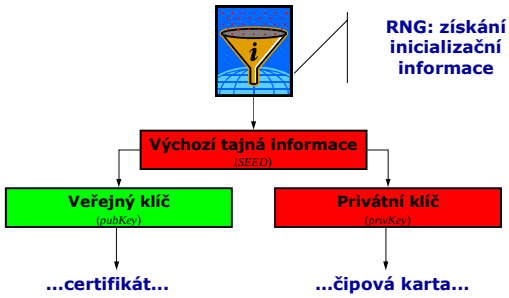
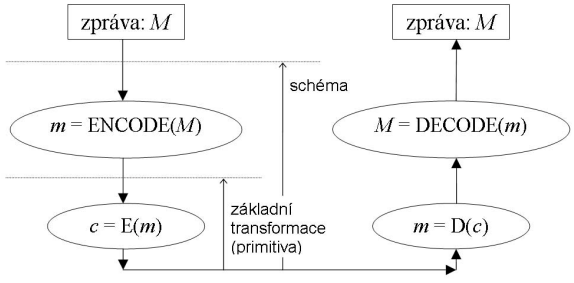


Schéma vs. transformace



RSA (1)

- Inicializace schématu
  - vygenerujme nezávisle dvě velká (zhruba stejně) prvočísla p, q, $p \neq q$
  - spočtěme $n = pq$, $\lambda = \text{lcm}(p\text{-}1, q\text{-}1)$
  - zvolme náhodné číslo e, $1 < e < \lambda$, $\gcd(e, \lambda) = 1$
    - někdy se volí e pevně (zejména $e = 3$, 65537)
  - spočtěme d splňující: $1 < d < \lambda$, $ed \equiv 1 \pmod{\lambda}$
    - použijeme rozšířený Eukleidův algoritmus
  - veřejným klíčem budiž dvojice (n, e)
    - n nazýváme modul a e veřejný exponent RSA
  - privátním klíčem budiž dvojice (n, d)
    - d nazýváme privátní exponent RSA
    - pro bezpečnost je nutné ošetřit integritu dvojice (n, d)

RSA (2)

- Šifrovací transformace: RSAEP((*n*, *e*), *m*)
  - vstup: veřejný klíč RSA (*n*, *e*), zformátovaná zpráva pro šifrování *m*, $0 \le m \le n-1$
  - výpočet:
    - RSAEP((*n*, *e*), *m*) = $m^e \bmod n$
- Ověřovací transformace: RSAVP((*n*, *e*), *s*)
  - vstup: veřejný klíč RSA (*n*, *e*), ověřovaný podpis *s*, $0 \le s \le n-1$
  - výpočet:
    - RSAVP((*n*, *e*), *s*) = RSAEP((*n*, *e*), *s*)

RSA (3)

- Odšifrovací transformace: RSADP((*n*, *d*), *c*)
  - vstup: privátní klíč RSA (*n*, *d*), šifrový text pro odšifrování *c*, $0 \le c \le n-1$
  - výpočet:
    - RSADP((*n*, *d*), *c*) = $c^d \bmod n$
- Podepisovací transformace: RSASP((*n*, *d*), *m*)
  - vstup: privátní klíč RSA (*n*, *d*), zformátovaná zpráva pro podpis *m*, $0 \le m \le n-1$
  - výpočet:
    - RSASP((*n*, *d*), *m*) = RSADP((*n*, *d*), *m*)

RSA (4)
-s využitím CRT-

- Inicializace schématu
  - vygenerujme nezávisle dvě velká (zhruba stejně) prvočísla *p*, *q*, $p \ne q$
  - spočtěme $n = pq$, $\lambda = \text{lcm}(p-1, q-1)$
  - zvolme náhodné číslo *e*, $1 < e < \lambda$, $\gcd(e, \lambda) = 1$
    - někdy se volí *e* pevně (zejména *e* = 3, 65537)
  - spočtěme d_p, d_q:
    - $1 < d_p < p-1$, $ed_p \equiv 1 \pmod{(p-1)}$
    - $1 < d_q < q-1$, $ed_q \equiv 1 \pmod{(q-1)}$
  - spočtěme q_{inv}: $0 < q_{inv} < p$, $qq_{inv} \equiv 1 \pmod{p}$
  - veřejným klíčem budiž dvojice (*n*, *e*)
    - *n* nazýváme modul a *e* veřejný exponent RSA
  - privátním klíčem budiž pětice (p, q, d_p, d_q, q_{inv})
    - pro bezpečnost je nutné ošetřit integritu celé pětice

RSA (5)
-s využitím CRT-

- Odšifrovací transformace: RSADP((p, q, d_p, d_q, q_{inv}), *c*)
  - vstup: privátní klíč RSA (p, q, d_p, d_q, q_{inv}), šifrový text pro odšifrování *c*, $0 \le c \le pq - 1$
  - výpočet:
    1. $m_1 = c^{dp} \bmod p$
    2. $m_2 = c^{dq} \bmod q$
    3. $h = (m_1 - m_2)q_{\text{inv}} \bmod p$
    4. $m = m_2 + hq$
    5. výsledek budiž hodnota *m*

RSA (6)
-s využitím CRT-

- Základní důvod použití: rychlost
  - dosahuje se průměrně zhruba 4násobného zrychlení odšifrovací (podepisovací) transformace
  - výhodné zejména pro čipové karty
- Další zrychlování: využití více než dvou prvočísel pro konstrukci modulu *n*
  - multi-prime RSA (patentováno, patent platí)
    - US Patent #5,848,159, Jan. 1997
  - asymptoticky lze koeficient zrychlování oproti dvoufaktorovému RSA (s využitím CRT) odhadnout jako $b^2/4$, kde *b* je počet prvočíselných faktorů v modulu *n*
  - při zvětšování *b* je třeba hlídat možnosti útoků založených na použití nízkých faktorů

RSA (7)
-základní vlastnosti-

- Multiplikativní vlastnost (homomorfismus) RSA
  - pro všechna $x_1, x_2 \in \mathbb{Z}$ platí
    - $\text{RSADP}(x_1 * x_2) = \text{RSADP}(x_1) * \text{RSADP}(x_2) \bmod n$
    - $\text{RSAEP}(x_1 * x_2) = \text{RSAEP}(x_1) * \text{RSAEP}(x_2) \bmod n$
- Tvrzení o individuálních bitech RSA
  - zhruba napsáno: schopnost invertovat RSAEP pro individuální bity znamená schopnost invertovat RSAEP zcela

RSA (8)
-schémata vyšší úrovně-

- Schéma šifrovací
  - vystavěno na transformacích RSAEP(.) a RSADP(.)
  - důležitým novým prvkem je kódování šifrované zprávy
    - *m* = ENCODE(*M*), kde *M* je vlastní zpráva v otevřeném tvaru a *m* je její tvar zpracovávaný v RSAEP(.) a RSADP(.)
    - *M* = DECODE(*m*) – transformace inverzní k ENCODE
  - kvalita kódování je rozhodující pro bezpečnost celého schématu
  - EME-PKCS1-v1_5, EME-OAEP, KEM

Standard PKCS#1 v. 1.5

příklad 1024bitového modulu *n*				
11001110	.....	.....	.....	.....
kódování EME-PKCS1-v1_5				
00000000	**00000010**	...???...	**00000000**	***M***

Standard PKCS#1 v. 2.1

příklad 1024bitového modulu n				
11001110	.....	.....	.....	.....
kódování EME-PKCS1-v1_5 - postranní kanál: Bleichenbacher, 1998, cca 2 miliony dotazů				
00000000	**00000010**	...???...	**00000000**	***M***

OAEP

00000000	...???...	...???...	...???...	...???...

EME-OAEP

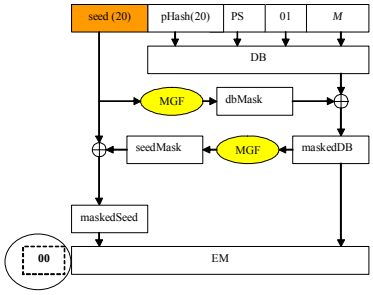


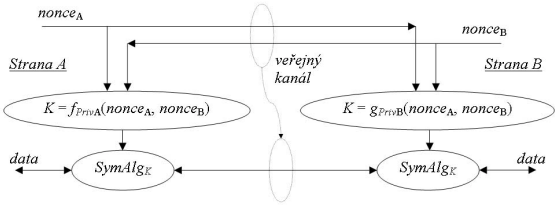
Schéma RSA-KEM

formátování RSA-KEM: žádná struktura (náhodné r, $r < n$)				
...???...	...???...	...???...	...???...	...???...

náhodné r, $r < n$
žádné formátování
$K = \mathrm{KDF}(r)$
$C_0 = \mathrm{RSAEP}(PubKey, r)$
$C_1 = \mathrm{Encrypt\_Sym\_šifra}(K, M)$
šifrový text $C = C_0 \parallel C_1$

šifrový text $C = C_0 \parallel C_1$
$r = \mathrm{RSADP}(PrivKey, C_0)$
žádná kontrola formátování r
$K = \mathrm{KDF}(r)$
$M = \mathrm{Decrypt\_Sym\_šifra}(C_1, K)$

Dohoda na klíči
-obecný princip-



- Dohodnuté tajemství (*K*) závisí na příspěvcích obou stran ($nonce_{A,B}$)
  - žádná ze stran nemá výsledek dohody pod svou výhradní kontrolou

Diffie-Hellman (1)

- Inicializace schématu
  - vygenerujme prvočíslo p a najděme α jako generátor $\mathbf{Z}_p^*$
    - p-1 musí mít alespoň jeden velký prvočíselný faktor (případně p-1 = 2t, kde t je prvočíslo)
    - p, α budou společné pro všechny uživatele
  - uživatel A si zvolí privátní klíč x_A, $0 < x_A < p$-1, a vypočte svůj veřejný klíč y_A:
    - $y_A = \alpha^{xA} \bmod p$
    - je nutné zajistit integritu trojic (p, α, y_A), (p, α, x_A)
  - uživatel B si zvolí privátní klíč x_B , $0 < x_B < p$-1, a vypočte svůj veřejný klíč y_B:
    - $y_B = \alpha^{xB} \bmod p$
    - je nutné zajistit integritu trojic (p, α, y_B), (p, α, x_B)

Diffie-Hellman (2)

- Postup dohody na klíči:
  - uživatel A získá z důvěryhodného zdroje veřejný klíč y_B
    - spočítá $K_A = y_B^{xA} \bmod p$
  - uživatel B získá z důvěryhodného zdroje veřejný klíč y_A
    - spočítá $K_B = y_A^{xB} \bmod p$
  - pokud vše proběhlo správně, platí $K_A = K_B$
  - oba uživatelé sdílí společné tajemství $K = K_A = K_B$
    - s ohledem na pevnou hodnotu K (do změny alespoň jednoho veřejného klíče) je vhodné zavést do následného odvození symetrických klíčů dodatečný náhodný faktor

Diffie-Hellman (3)

Half-certified, Ephemeral, ElGamal key agreement

- Základní myšlenka: pouze jeden z uživatelů použije svůj veřejný klíč (certifikovaný)
  - uživatelé nemusí nadále sdílet stejné hodnoty p, α
  - jeden z uživatelů ani nemusí mít pevný veřejný klíč

Diffie-Hellman (4)

Half-certified, Ephemeral, ElGamal key agreement

- Příklad komunikace: Pouze uživatel B použije svůj veřejný klíč
  - uživatel A – zahajuje komunikaci s uživatelem B
    1. získá (p, α, y_B)
       - například z certifikátu uživatele B
    2. vygeneruje tajné náhodné číslo a, $0 < a < p$-1
    3. vypočte $y = \alpha^a \bmod p$, $K_A = y_B^a \bmod p$
    4. hodnotu y zašle uživateli B
  - uživatel B – přijímá komunikaci od uživatele A
    - vypočte $K_B = y^{xB} \bmod p$
  - při správném postupu uživatelé A, B sdílí tajemství $K = K_A = K_B$
    - uživatel A má možnost diversifikovat K svou volbou hodnoty a, přesto je však nutné do odvození (symetrických) klíčů zavést ještě náhodný faktor od uživatele B

ElGamal (1)

- Inicializace schématu
  - vygenerujme prvočíslo p a najděme α jako generátor $\mathbf{Z}_p^*$
    - viz nároky uvedené u D-H protokolu
  - zvolme privátní exponent x, $0 < x < p$-1
  - vypočtěme veřejný klíč y, $y = \alpha^x \bmod p$
  - veřejné parametry jsou (p, α)
    - někdy je veřejný klíč uváděn ve tvaru (p, α, y)
  - privátní klíč je trojice (p, α, x)
    - je nutné zajistit integritu trojice (p, α, x)

ElGamal (2)

- Šifrovací transformace: ElGamalEP((p, α, y), m)
  - vstup: veřejné parametry a klíč (p, α, y), zformátovaná zpráva pro šifrování m, $0 \le m \le p$-1
  - výpočet:
    1. vygenerujme tajné náhodné číslo k, $0 < k < p$-1
    2. vypočtěme $c_0 = \alpha^k \bmod p$, $c_1 = m * y^k \bmod p$
    3. šifrový text c budiž $c = (c_0, c_1)$
- Odšifrovací transformace: ElGamalDP((p, α, x), c)
  - vstup: privátní klíč (p, α, x), šifrový text $c = (c_0, c_1)$
  - výpočet:
    1. vypočtěme $z = c_0^{p-1-x} \bmod p$
    2. výsledkem odšifrování budiž m, $m = z * c_1 \bmod p$

poznámka: $z * c_1 \equiv \alpha^{k(p-1)-kx} y^k m \equiv \alpha^{k(p-1)-kx} \alpha^{kx} m \equiv m \pmod p$

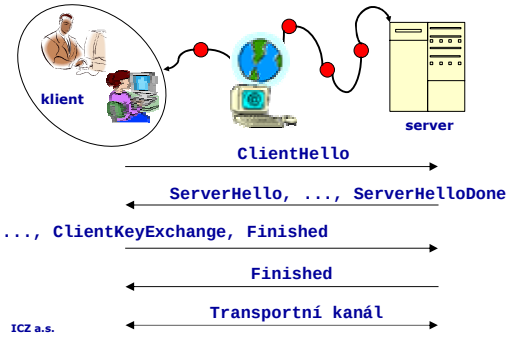
ElGamal (3)
-schémata vyšší úrovně-

- Schéma šifrovací
  - vystavěno na transformacích ElGamalEP(.) a ElGamalDP(.)
  - důležitým novým prvkem je kódování šifrované zprávy
    - m = ENCODE(M), kde M je vlastní zpráva v otevřeném tvaru a m je její tvar zpracovávaný v ElGamalEP(.) a ElGamalDP(.)
    - M = DECODE(m) – transformace inverzní k ENCODE
  - kvalita kódování je rozhodující pro bezpečnost celého schématu
  - kódovací postupy nejsou sjednoceny; lze se inspirovat u RSA (OAEP a jeho obecná vylepšení jako je OAEP+ apod.)

Poznámka o zobecnění
-Diffie-Hellman a ElGamal-

- Obě schémata mají řadu společných algebraických vlastností
- Obě schémata lze dále rozvíjet (zvyšovat bezpečnost, měnit použitou algebraickou strukturu, apod.)
  - použití podgrupy prvočíselného řádu
    - minimalizuje užitečnou informaci o privátním klíči, která je obsažena v klíči veřejném
    - snižuje efektivitu útoků (či je zcela eliminuje)
  - použití eliptických křivek - $E(\boldsymbol{F}_q)$
    - slibuje kratší délky privátního klíče

Příklad: SSL/TLS
-ustavení spojení-



Doporučená literatura

- Obecně
  - archiv (zejména českých) článků o kryptologii
    - http://www.decros.cz/bezpecnost/_kryptografie.html
  - Menezes, A. J., van Oorschot, P. C. and Vanstone, S. A.: *Handbook of Applied Cryptography*, CRC Press, 1996
    - http://www.cacr.math.uwaterloo.ca/hac/
  - IEEE P1363: *IEEE Standard Specifications for Public-Key Cryptography*, August 29, 2000.
- RSA
  - dokumenty PKCS, zejména PKCS#1, PKCS#3
    - http://www.rsasecurity.com/rsalabs/pkcs/index.html
- Diffie-Hellman
  - PKCS#3
  - RFC 2631
- TLS - jako příklad kombinace asymetrických a symetrických metod
  - RFC 2246