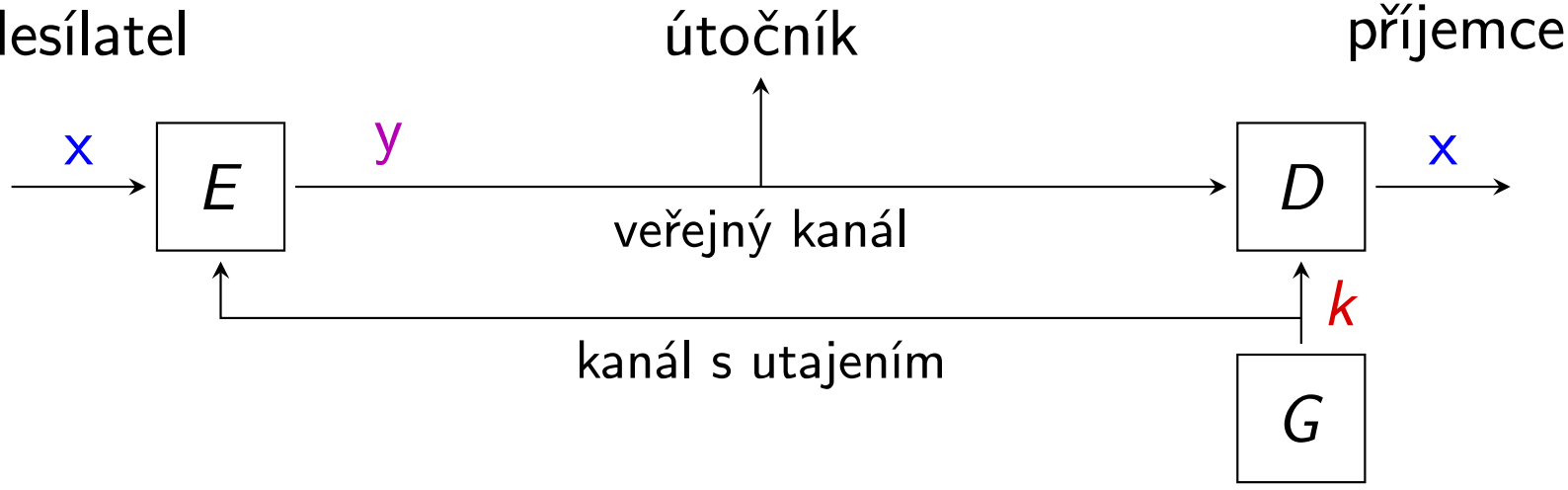
Kryptografie s veřejným klíčem

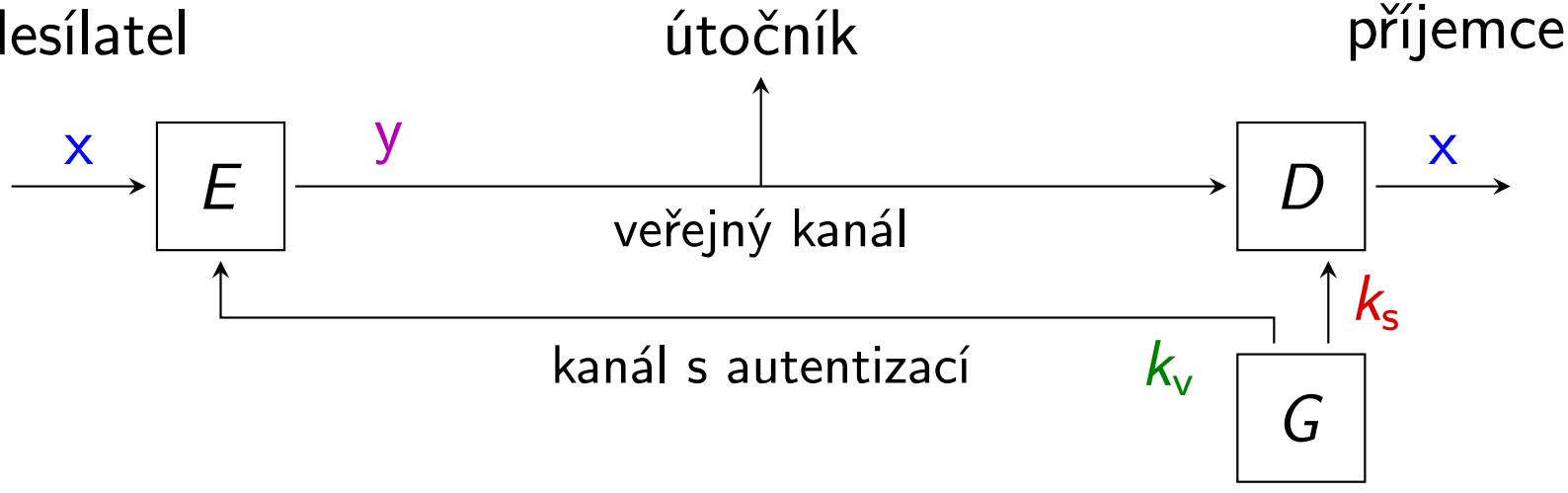
Andrew Kozlík

KA MFF UK

Symetrická kryptografie



Asymetrická kryptografie



Kryptografie s veřejným klíčem

- S nápadem přišli Whitfield Diffie a Martin Hellman v článku *New Directions in Cryptography* (1976).
- Jednosměrná funkce s „padacími vrátky“ (trapdoor one-way function).
- Algoritmy využívají dvojici klíčů (k_s, k_v), kde
  - k_s je *soukromý klíč* a
  - k_v je příslušný *veřejný klíč*.
- K šifrování slouží veřejný klíč: $y = E_{k_v}(x)$.
- K dešifrování slouží soukromý klíč: $x = D_{k_s}(y)$.
- Určit otevřený text x ze znalosti šifrového textu y a veřejného klíče k_v je problém přesahující výpočetní možnosti útočníka.
- Kryptografický systém, který využívá veřejný a soukromý klíč, nazýváme *asymetrický*.

Kryptografický systém RSA

- Autoři: Ron Rivest, Adi Shamir a Leonard Adleman (1977).
- RSA popsal pracovník GCHQ Clifford Cocks už v roce 1973.
- Založený na problému prvočíselného rozkladu celých čísel:
  - Je snadné vygenerovat číslo se známým rozkladem.
  - Je však obtížné určit rozklad libovolného čísla.

Kryptografický systém RSA

- Algoritmus generování klíčů pro RSA:
  1. Zvol $e \geq 3$.
  2. Zvol náhodně dvě různá prvočísla p a q tak, aby e bylo nesoudělné s $\varphi(pq) = (p-1)(q-1)$.
  3. Spočti $N = pq$.
  4. Spočti d takové, že $de \equiv 1 \pmod{\varphi(N)}$.
  5. Vrať soukromý klíč $k_s = (d, N)$ a veřejný klíč $k_v = (e, N)$.
- Šifrování zprávy $x \in \mathbb{Z}_N$:

$$E_{(e,N)}(x) = x^e \bmod N$$

- Dešifrování šifrového textu $y \in \mathbb{Z}_N$:

$$D_{(d,N)}(y) = y^d \bmod N$$

Poznámky

Korektnost algoritmu:

- Z Eulerovi věty plyne, že $(x^e)^d \bmod N = x$ pro $x \in \mathbb{Z}_N^*$.
- Tvrzení na dalším slajdu dokazuje, že RSA funguje nejen pro $x \in \mathbb{Z}_N^*$, ale pro všechna $x \in \mathbb{Z}_N$.

Dešifrovací exponent:

- Multiplikativní inverz $d \equiv e^{-1} \pmod{\varphi(N)}$ je určen jednoznačně v intervalu $[1, \varphi(N) - 1]$.
- V intervalu $[1, \varphi(N) - 1]$ však existuje víc exponentů, kterými lze úspěšně dešifrovat.
- Často se jako dešifrovací exponent používá

$$d \equiv e^{-1} \pmod{\mathrm{NSN}(p-1, q-1)}.$$

Korektnost RSA

Tvrzení

Nechť $p-1 \mid ed-1$ a zároveň $q-1 \mid ed-1$, kde p a q jsou nesoudělná prvočísla.
Pak pro každé $x \in \mathbb{Z}_N$ platí $(x^e)^d \bmod N = x$, kde $N = pq$.

Důkaz.

Jestliže x je nesoudělné s p, pak podle Fermatovi malé věty

$$x^{ed} = x^{1+(p-1)\frac{ed-1}{p-1}} = x\left(x^{p-1}\right)^{\frac{ed-1}{p-1}} \equiv x \pmod{p}.$$

Jestliže x je soudělné s p, pak $x \equiv 0 \pmod{p}$, čili opět

$$x^{ed} \equiv x \pmod{p}.$$

Analogicky i modulo q dostaneme pro všechna $x \in \mathbb{Z}_N$

$$x^{ed} \equiv x \pmod{q}.$$

Jelikož jsou p a q nesoudělné, podle čínské věty o zbytcích

$$x^{ed} \equiv x \pmod{N}.$$

□

Jednoznačnost dešifrovacího exponentu

Tvrzení

Dešifrovací exponent RSA je určen jednoznačně modulo $\mathrm{NSN}(p-1, q-1)$.

Důkaz.

- Mějme d a $d' \in \mathbb{N}$ takové, že $y^d \equiv y^{d'} \pmod{N}$ pro všechna $y \in \mathbb{Z}_N^*$.
- Potom $y^{d-d'} \equiv 1 \pmod{p}$ pro všechna $y \in \mathbb{Z}_p^*$.
- Existuje $y \in \mathbb{Z}_p^*$ řádu $p-1$, čili $p-1 \mid d-d'$.
- Stejně tak $q-1 \mid d-d'$.
- Odtud $d \equiv d' \pmod{\mathrm{NSN}(p-1, q-1)}$.

□

Volba parametrů p, q a e

- Prvočísla p a q volíme tak, aby každé z nich mělo $\frac{1}{2}s$ bitů v binární reprezentaci, kde s je bezpečnostní parametr.
- Typicky volíme $1024 \leq s \leq 4096$.
- Největší RSA modulus N, který se dosud podařilo faktorizovat měl $s = 829$ bitů.
- Veřejný exponent se zpravidla volí $e = 2^{16} + 1 = 65537$.
  - Malá hodnota e vede k rychlejšímu šifrování.
  - Příliš malá hodnota otevírá cestu některým útokům.
  - S exponenty tvaru $2^u + 1$ se dobře počítá.
  - Jestliže e je prvočíslo, pak je pravděpodobnější, že bude nesoudělné s $\varphi(N)$.

Bezpečnost

- Cíl útočníka:
  Ze znalosti šifrového textu $y = x^e \bmod N$ a veřejného klíče (e, N) určit otevřený text x (tzv. RSA problém).
- Útočník, který umí efektivně faktorizovat RSA modulus N, umí efektivně řešit RSA problém.
- Existují indície, že obrácená implikace obecně neplatí.
  (To je dobře, protože jinak by byl systém zranitelný útokem typu chosen-ciphertext attack.)

Sdílení modulu

- Ze znalosti čísel N, e a d lze určit prvočíselný rozklad N.
- Dvě entity proto nesmění sdílet stejný modulus N.

Časová složitost RSA

Generování klíčů v čase $\mathcal{O}(s^4)$

1. V čase $\mathcal{O}(s)$ vygenerujeme náhodné liché číslo p.
2. V čase $\mathcal{O}(s^3)$ zjistíme Millerovým-Rabinovým testem, zda p je prvočíslo.
3. Jestliže p není prvočíslo, zvýšíme p o 2 a vrátíme se ke kroku 2.
   Vzhledem k hustotě rozložení prvočísel je počet opakování kroku 2 v průměru $\mathcal{O}(\log p)$, čili $\mathcal{O}(s)$.
4. Stejným způsobem vygenerujeme prvočíslo q.
5. V čase $\mathcal{O}(s^2)$ spočítáme součin p a q.
6. Zvolíme e buď náhodně, anebo $e = 2^{16} + 1$.
7. V čase $\mathcal{O}(s^2)$ spočítáme d Eukleidovým algoritmem.

Časová složitost RSA

Šifrování v čase $\mathcal{O}(s^3)$ nebo $\mathcal{O}(s^2)$

- V čase $\mathcal{O}(s^3)$ provedeme modulární mocnění algoritmem square-and-multiply.
- Je-li e voleno jako malá konstanta, pak například pro $e = 2^{16} + 1$ stačí 17 operací modulárního násobení, které provedeme v čase $\mathcal{O}(s^2)$.

Dešifrování v čase $\mathcal{O}(s^3)$

- V čase $\mathcal{O}(s^3)$ provedeme modulární mocnění algoritmem square-and-multiply.

Rychlejší dešifrování podle čínské věty o zbytcích

- Pro tento postup musí příjemce kromě soukromého klíče (d, N) uchovávat také prvočísla p a q.
- Postup:
  1. Pro šifrový text y spočítáme
  $$x_1 = y^{d \bmod (p-1)} \bmod p, \qquad x_2 = y^{d \bmod (q-1)} \bmod q.$$
  2. Podle čínské věty o zbytcích lze z hodnot x_1 a x_2 určit otevřený text x.
- Při výpočtu x_1 umocňujeme y na $\frac{s}{2}$-bitové číslo modulo $\frac{s}{2}$-bitové číslo. Časová složitost mocnění tímto klesá na $\frac{1}{8}$ oproti počítání s s-bitovými čísly.
- Mocnění se však provádí dvakrát (modulo p a q). Dále je třeba z x_1 a x_2 určit x v čase $\mathcal{O}(s^2)$.
- Pro velké s je časová složitost 2. kroku zanedbatelná. Celkem tak dosahujeme zrychlení blížící se 400 %.

ASN.1 typ RSAPrivateKey

Privátní klíč se běžně ukládá jako struktura s položkami:

- N modulus
- e veřejný exponent
- d soukromý exponent
- p první prvočinitel čísla N
- q druhý prvočinitel čísla N
- $d_{\mathrm{P}} = d \bmod (p-1)$
- $d_{\mathrm{Q}} = d \bmod (q-1)$
- $q_{\mathrm{Inv}} \equiv q^{-1} \pmod{p}$

Rychlé dešifrování pomocí Garnerova algoritmu:

- Spočteme $x_1 = y^{d_{\mathrm{P}}} \bmod p$ a $x_2 = y^{d_{\mathrm{Q}}} \bmod q$.
- Potom $x = x_2 + q \cdot \big((x_1 - x_2) \cdot q_{\mathrm{Inv}} \bmod p\big)$.

Použití asymetrických šifer

Problém:

- Šifrovací a dešifrovací funkce asymetrických šifer jsou relativně výpočetně náročné ve srovnání se symetrickými šiframi.
- Srovnání softwarových implementací RSA-1024 a AES-128:
  - Při šifrování vyžaduje RSA 60krát více času než AES.
  - Při dešifrování vyžaduje RSA 1000krát více času než AES.

Řešení:

- Odesílatel náhodně vygeneruje klíč k pro symetrickou šifru.
- Zpráva se zašifruje symetrickou šifrou s klíčem k.
- Klíč k se zašifruje asymetrickou šifrou a pošle se společně se zašifrovanou zprávou: $(\mathrm{RSA}_{k_v}(k), \mathrm{AES}_k(x))$.

Håstadův útok na malý veřejný exponent

- Jeden neznámý otevřený text x byl zašifrován třemi různými veřejnými klíči $(3, N_1)$, $(3, N_2)$ a $(3, N_3)$ jako

$$y_1 = x^3 \bmod N_1, \quad y_2 = x^3 \bmod N_2 \quad \text{a} \quad y_3 = x^3 \bmod N_3.$$

- Můžeme předpokládat, že N_1, N_2 a N_3 jsou nesoudělné.
- Ze šifrových textů y_1, y_2 a y_3 lze potom spočítat $y \in \mathbb{Z}_{N_1 N_2 N_3}$ takové, že $y \equiv x^3 \pmod{N_1 N_2 N_3}$.
- Jelikož $x < N_i$ pro všechna i, je $x^3 < N_1 N_2 N_3$.
- Otevřený text lze tedy spočítat jako $x = \sqrt[3]{y}$ v celých číslech.

Kódování zprávy (resp. klíče symetrické šifry)

- Před aplikováním šifrovací operace RSA se zpráva (resp. klíč symetrické šifry) nejdříve zakóduje.
- Používá se buď starší kódování EME-PKCS1-v1_5 nebo novější EME-OAEP.
- Obě kódování jsou definovaná ve standardu PKCS #1.
- EME-PKCS1-v1_5 kódování má tvar

$$\texttt{0x00} \| \texttt{0x02} \| PS \| \texttt{0x00} \| k,$$

kde

  - k je klíč symetrické šifry;
  - PS je řetězec pseudonáhodných nenulových bajtů délky $\text{len}(N) - \text{len}(k) - 3$ bajtů.

- Proti EME-PKCS1-v1_5 existuje Bleichenbacherův chosen-ciphertext útok. Upřednostňujte EME-OAEP.

Cvičení

Nechť N je RSA modulus. Ukažte, že ze znalosti N a $\varphi(N)$ lze snadno určit prvočísla p a q.

Nápověda: p a q najdete jako kořeny vhodně zvoleného polynomu.

Digitální podpis

- Co požadujeme od digitálního podpisu?
  - Zajištění integrity zprávy.
  - Autentizaci původce zprávy.
  - Nepopiratelnost původu zprávy (podepsaná osoba nemůže prokázat, že podpis mohla vytvořit jiná osoba).
- Algoritmy digitálního podpisu využívají dvojici klíčů (k_s, k_v),
  - k_s je *soukromý klíč* a
  - k_v je příslušný *veřejný klíč*.
- Podepisovanou zprávu budeme značit x a její podpis y.
- K podepisování slouží soukromý klíč:

$$y = \mathrm{sig}_{k_s}(x).$$

- K ověřování podpisu slouží veřejný klíč:

$$\mathrm{ver}_{k_v}(x, y) \in \{\mathrm{true}, \mathrm{false}\}.$$

Typy útoků na podpisová schémata

- **Key-only útok**
  Útočník zná pouze k_v.

- **Known message útok**
  Útočník zná k_v a podpisy $y_1, \ldots, y_n$ zpráv $x_1, \ldots, x_n$.

- **Chosen message útok**
  Útočník zná k_v, zvolí zprávy $x_1, \ldots, x_n$ a dozví se příslušné podpisy $y_1, \ldots, y_n$.

Cíle útoků na podpisová schémata

- **Odhalení soukromého klíče**
- **Univerzální podvržení**
  Vytvoření podpisu k libovolné zprávě.
- **Selektivní podvržení**
  Vytvoření podpisu k libovolné zprávě ležící v určité podmnožině množiny všech zpráv. Tato podmnožina je nezávislá na klíči.
- **Existenční podvržení**
  Vytvoření nové dvojice (x, y), na zprávu x nejsou kladeny žádné požadavky.

Vyrobení podpisového schématu z asymetrické šifry

- Podpis se vytvoří „dešifrováním“ zprávy

$$y = \mathrm{sig}_{k_s}(x) := D_{k_s}(x).$$

- Ověření se provede „zašifrováním“ podpisu

$$\mathrm{ver}_{k_v}(x, y) = \mathrm{true} \quad \Longleftrightarrow \quad E_{k_v}(y) = x.$$

- Výhoda: Zprávu x není nutné posílat společně s podpisem, protože ji lze z podpisu y vypočítat jako $x = E_{k_v}(y)$.
- Útok: Lze docílit existenčního podvržení podpisu. (Zvolíme y libovolně a spočítáme $x = E_{k_v}(y)$.)
- Nevýhoda: Asymetrické šifry jsou výpočetně náročné. Postup není vhodný pro podepisování velkého množství dat.

Příklad: Podpisové schéma RSA

- Máme RSA klíče $k_s = (d, N)$ a $k_v = (e, N)$ a algoritmy
  - $\mathrm{sig}_{(d,N)}(x) = x^d \bmod N$,
  - $\mathrm{ver}_{(e,N)}(x, y) = \text{true} \iff x = y^e \bmod N$.

- Útok: Lze docílit selektivního podvržení podpisu na množině zpráv $\{1, -1\}$.

- Multiplikativní vlastnost RSA:
  - Mějme dvě podepsané zprávy (x_1, y_1) a (x_2, y_2).
  - Platí $(x_1 \cdot x_2)^d \equiv {x_1}^d \cdot {x_2}^d \equiv y_1 \cdot y_2 \pmod N$.
  - Můžeme tedy vytvořit novou podepsanou zprávu $(x_1 \cdot x_2 \bmod N,\ y_1 \cdot y_2 \bmod N)$.

- Multiplikativní vlastnost RSA dává další možnost existenčního podvržení podpisu.

Podpisová schémata vyrobená z asymetrických šifer

- Uvedené útoky a nevýhody se řeší tím, že místo zprávy se podepisujeme pouze její hash.

- Srovnání softwarových implementací RSA-1024 a SHA-256:
  - Při podepisování vyžaduje RSA 1000krát více času než SHA.
  - Při ověřování vyžaduje RSA 60krát více času než SHA.

- Hashovací funkce musí být kolizivzdorná.
  - V případě že zprávy x_1 a x_2 kolidují, mají obě zprávy stejný podpis.
  - Útočník může přesvědčit podepisující osobu, aby podepsala zprávu x_1, ale výsledný podpis připojí ke zprávě x_2.

Kódování otisku zprávy

- Před podepsáním se otisk zprávy nejdříve zakóduje.
- Používá se buď starší kódování EMSA-PKCS1-v1_5 nebo novější EMSA-PSS.
- Obě kódování jsou definovaná ve standardu PKCS #1.
- EMSA-PKCS1-v1_5 kódování má tvar

$$\texttt{0x00} \| \texttt{0x01} \| PS \| \texttt{0x00} \| T,$$

kde

  - T obsahuje identifikátor hashovací funkce a otisk $h(x)$ (konkrétně T je DER kódovaná ASN.1 hodnota typu DigestInfo);
  - PS je řetězec bajtů `0xff` délky $\text{len}(N) - \text{len}(T) - 3$ bajtů.

Slepý podpis založený na RSA

- Princip slepého podpisu:
  - Autor zprávy a podepisující jsou různé osoby.
  - Podepisující osobě je obsah zprávy utajen.
  - Když se podepisující setká s odtajněnou podepsanou zprávou, nesmí poznat, kdy a komu ji podepsal.
- Příklady využití:
  - Elektronické peníze. (Vydavatel podepisuje a proplácí poukázky. Chceme před ním utajit, za co jsme je utratili.)
  - Elektronické volby. (Volební komisař podepisuje hlasovací lístky. Chceme před ním utajit, jak jsme hlasovali.)
- Slepé podepsání zprávy x:
  - Zvolíme náhodnou tajnou zaslepovací hodnotu $\alpha \in \mathbb{Z}_N^*$.
  - Vytvoříme zaslepenou zprávu $x' \equiv x\alpha^e \pmod{N}$.
  - Předložíme x' k podpisu a obdržíme $y' \equiv x'^d \pmod{N}$.
  - Spočítáme $y \equiv y'\alpha^{-1} \pmod{N}$.
  - Platí $y \equiv (x\alpha^e)^d\alpha^{-1} \equiv x^d \pmod{N}$.