8 Asymetrické šifry

8.1 RSA

Dnes asi nejznámější asymetrickou šifrou je algoritmus RSA publikovaný v roce 1977 Ronem Rivestem, Adi Shamirem a Leonardem Adlemanem. Tentýž algoritmus popsal už v roce 1973 Clifford Cocks zaměstnaný britskou špionážní službou GCHQ, která ho ovšem až do roku 1997 tajila.

Definice: Klíče pro RSA generujeme takto:

- Vygenerujeme dvě náhodná velká prvočísla p a q.
- Spočítáme *modul* $n = pq$.
- Spočítáme Eulerovu funkci $\varphi(n) = (p-1)(q-1)$.
- Zvolíme *šifrovací exponent* $e \perp \varphi(n)$.
- Spočítáme *dešifrovací exponent* d tak, aby $ed \equiv_n 1$ (d je multiplikativní inverze e).
- Dvojici (e, n) prohlásíme za *šifrovací klíč* a (d, n) za *dešifrovací klíč*.

Šifrujeme a dešifrujeme takto:

- *Zprávy* jsou prvky $\mathbb{Z}_n$.
- *Šifrovací funkce* je $E(x) = x^e \bmod n$.
- *Dešifrovací funkce* je $D(y) = y^e \bmod n$.

V závislosti na použití je bud šifrovací klíč *veřejný* a dešifrovací *soukromý* (každý může Alici poslat šifrovanou zprávu, ale jen ona ji může dešifrovat), nebo naopak šifrovací soukromý a dešifrovací veřejný (jen Alice může zašifrovat zprávu a kdokoliv ji dešifruje – to se chová jako podpis).

Pozor na to, že kdykoliv by se někdo dozvěděl prvočísla p a q, dokáže si spočítat $\varphi(n)$, a tím pádem vyrobit z veřejného klíče soukromý klíč.

Nyní dokážeme, že RSA je korektní:

Věta: Pro všechna $x \in \mathbb{Z}_n$ platí $D(E(x)) = x$.

Důkaz: Počítejme modulo n: $D(E(x)) \equiv (x^e)^d \equiv x^{ed} \equiv x^{k\varphi(n)+1}$. Pro $x \in \mathbb{Z}_n^*$ můžeme použít Eulerovu větu, podle níž $x^{\varphi(n)} \equiv 1$, takže $x^{k\varphi(n)+1} \equiv (x^{\varphi(n)})^k \cdot x \equiv 1^k \cdot x \equiv x$.

Pokud $x \notin \mathbb{Z}_n^*$, znamená to, že x je soudělné s $n = pq$, tedy dělitelné p nebo q. Kdybychom takovou zprávu vyslali, mohl by útočník spočítat $\gcd(x, n)$ a tím se dozvědět buď p nebo q. Jelikož ale p a q volíme náhodně, je velmi nepravděpodobné, že tento případ nastane.

Pro úplnost dokážeme, že i pro $x \notin \mathbb{Z}_n^*$ je RSA korektní. Korektnost dokážeme zvlášť modulo p a q, z čehož podle Čínské zbytkové věty poplyne i korektnost modulo n. Bez újmy na obecnosti je x násobkem p. Pak modulo p platí $x^{ed} \equiv 0^{ed} \equiv 0 \equiv x$. Modulo q můžeme použít Malou Fermatovu větu: $x^{ed} \equiv x^{k\varphi(n)+1} \equiv x^{k(p-1)(q-1)} \cdot x \equiv (x^{q-1})^{k(p-1)} \cdot x \equiv 1^{k(p-1)} \cdot x \equiv x$. □

Rychlost RSA

Generování klíčů je randomizované a pracuje v průměrně polynomiálním čase. Šifrování i dešifrování je modulární umocňování, o kterém víme, že se dá spočítat v kubickém čase. Dokonce o něco rychleji, když použijeme chytřejší algoritmy na násobení, ale ani tak nedosáhneme hezké lineární složitosti symetrických šifer.

Operace s veřejným klíčem se často zrychlují tak, že veřejný exponent zafixujeme jako malé celé číslo – typicky 3 nebo 17. V obou případech k umocnění postačí malý počet modulárních násobení.

Na soukromé straně tuto optimalizaci nemůžeme použít (malý soukromý exponent by útočník mohl uhodnout), ale zase si můžeme pamatovat p a q, operace provádět zvlášť modulo p a modulo q, a pak výsledky zkombinovat pomocí Čínské zbytkové věty. Rozbor složitosti ponecháváme jako cvičení.

Vlastnosti RSA

Pokud dva klíče sdílí tentýž modul, RSA je *komutativní šifra* – šifrujeme-li nejdříve jedním klíčem a pak výsledek druhým, výsledek nezáleží na pořadí klíčů. Vskutku: $(x^e)^{e'} \equiv x^{ee'} \equiv (x^{e'})^e$. Časem ovšem uvidíme, že používat stejný modul vícekrát je problematické.

Jelikož korektnost RSA potřebuje o e a d jenom to, že jsou navzájem multiplikativními inverzemi modulo $\varphi(n)$, můžeme e a d prohodit a RSA bude stále fungovat. Až po vygenerování klíčů si tedy můžeme vybrat, který z nich použijeme na šifrování a který na dešifrování. Není ovšem zdravé kombinovat obě možnosti v jednom protokolu.

Také si všimneme, že RSA je *multiplikativní* – pro každé $a, b \in \mathbb{Z}_n$ platí $E(ab) \equiv E(a)E(b)$. Této vlastnosti se rovněž říká, že RSA je *homomorfní,* jelikož E funguje jako homomorfismus ze $(\mathbb{Z}_n, \cdot)$ do $(\mathbb{Z}_n, \cdot)$.

Slepé podpisy

Multiplikativnost RSA dokáže být jak jeho prokletím, tak požehnáním, protože umožňuje zajímavé útoky i zajímavé aplikace. Třeba tu následující.

Představme si, že Alice je ochotna podepsat libovolnou zprávu tím, že ji zašifruje svým soukromým klíčem (e, n), načež ji kdokoliv může dešifrovat veřejným klíčem (d, n). Bob

by si chtěl nechat podepsat svou zprávu $x \in \mathbb{Z}_n$, ale nerad by, aby se Alice dozvěděla její obsah. Tomu se říká *slepý podpis.*[1]

Bob si vedle zprávy x zvolí nějaké číslo $r \in \mathbb{Z}_n$ a spočítá jeho inverzi r^{-1} modulo n. Kdyby nastal nepravděpodobný případ, kdy r vyjde soudělné s n, Bob rovnou umí faktorizovat n a získat soukromý klíč.

Bob pošle Alici k podepsání $xr^d \bmod n$, ona mu vrátí podpis $(xr^d)^e \equiv x^e r^{ed} \equiv x^e r$. Bob podpis vynásobí r^{-1} a získá $x^e r r^{-1} \equiv x^e \cdot 1 \equiv x^e$, což je podpis tajné zprávy x.

Co umí Alice zjistit o x? Jelikož r je rovnoměrně náhodný invertibilní prvek a šifrovací funkce je na $\mathbb{Z}_n^*$ bijekce, musí být r^e také rovnoměrně náhodný prvek $\mathbb{Z}_n^*$. Pokud x je také prvkem $\mathbb{Z}_n^*$ (jinak faktorizujeme n), násobíme v grupě zadaný prvek rovnoměrně náhodným prvkem, a jak už víme víme z rozboru jednorázových klíčů, vznikne opět rovnoměrně náhodný prvek. Ten jistě nenese žádnou informaci o x.

Cvičení

1. Odhadněte, kolikrát se RSA zrychlí s optimalizací přes Čínskou zbytkovou větu.

8.2 Bezpečnost RSA

Na RSA existuje řada útoků. Začneme těmi triviálními a postupně se propracujeme k sofistikovanějším.

Předně pokud budeme šifrovat číslo x, které menší než $n^{1/e}$, při výpočtu $x^e \bmod n$ se modulo vůbec neprojeví. K dešifrování tedy stačí spočítat celočíselnou e-tou odmocninu, což lze provést v polynomiálním čase binárním vyhledáváním. Pozor, že pro typickou volbu $e = 3$ tento problém nastane i pro docela velká x.

Také si všimneme, že znalost $\varphi(n)$ sama o sobě stačí k faktorizaci n. Víme totiž, že $\varphi(n) = (p-1)(q-1) = pq - p - q + 1 = n - p - q + 1$. To nám dává soustavu rovnic $pq = n$, $p + q = \varphi(n) - n - 1$, kterou lze v polynomiálním čase vyřešit.

TODO: Je-li $d < n^{1/4}$, lze ho spočítat z e [Wiener 1990].

TODO: Z e a d lze spočítat $\varphi(n)$ (viz Stinson & Paterson)

[1] Může být realistické, aby Alice byla ochotna podepisovat cokoliv? Ano, například pokud každý den generuje nový klíč. Pak její podpis slouží jako *časové razítko,* které potvrzuji, že zpráva existovala v daný den.

Setkání na půli cesty

I na RSA existují „narozeninové“ útoky. Nechť známe zašifrovanou zprávu x^e a chceme zjistit x. Budeme budovat množiny $U = \{u^e \bmod n\}$ $V = \{x(v^{-1})^e \bmod n\}$ pro dostatečně mnoho různých $u, v \in \mathbb{Z}_n^*$, dokud neobjevíme nějaký prvek v jejich průniku. Pak víme, že pro nějaká u, v je $u^e \equiv x(v^{-1})^e$. To znamená, že $x \equiv (uv)^e$ a máme dešifrováno.

Jak velké U a V potřebujeme? Pokud u i v volíme rovnoměrně náhodně, budou i prvky U a V rovnoměrně náhodné. Tuto situaci známe už z útoku „setkáním na půli cesty“ v oddílu **??**. Tam jsme dokázali, že pro pravděpodobnost úspěchu $1/2$ potřebujeme, aby U i V byly velké řádově $\sqrt{n}$.

To nám pro b-bitový modul n omezuje úroveň bezpečnosti na $b/2$. Stejnou složitost má ovšem triviální útok faktorizující n hledáním všech dělitelů do $\sqrt{n}$.

Podobné zprávy

Co se stane, když někdo stejným klíčem zašifruje dvě podobné zprávy x a $x' = x + \delta$? Představme si, že známe zašifrované zprávy $y = x^e$ a $y' = (x + \delta)^e$ a také rozdíl δ (ten můžeme postupně zkoušet). Tehdy můžeme x najít jako společný kořen polynomů $f(t) = t^e - y$ a $f'(t) = (x + \delta)^e - y'$. Abychom tyto polynomy sestrojili, potřebujeme, aby šifrovací exponent e byl nízký.

Jelikož Euklidův algoritmus funguje i pro polynomy, můžeme spočítat polynom $h = \gcd(f, g)$. Kořeny polynomu h jsou průnikem kořenů f s kořeny g. S velkou pravděpodobností ovšem vyjde, že h je lineární polynom, takže jeho kořen triviálně zjistíme.

Podobná prvočísla

Podobně problematické je, když jsou podobná prvočísla p a q. Představme si, že $q = p + 2d$ a že uhodneme d. Potom víme, že $n = pq = p^2 + 2pd$. Přičtením d^2 k oběma stranám získáme $n + d^2 = (p + d)^2$. Stačí tedy zkoušet malá d a zkoušet odmocňovat $n + d^2$, abychom získali p.

Tomuto útoku se obvykle předchází tak, že vygenerujeme p délky $b/2 - 1$ bitů, zatímco q délky $b/2 + 1$, obě s nejvyšším bitem zafixovaným na 1. Pak je rozdíl $q - p$ určitě větší než $2^{b/2-1}$.

Stejná zpráva zašifrovaná více klíči

TODO: Více klíčů se stejným modulem?

Také není bezpečné tutéž zprávu zašifrovat více klíči. Princip útoku předvedeme pro šifrovací exponent $e = 3$. Tehdy potřebujeme tři různé klíče $(3, n_1)$, $(3, n_2)$, $(3, n_3)$. Označíme-li

x původní zprávu a y_1 až y_3 její zašifrované podoby, řešíme kongruence

$$\begin{aligned} x^3 &\equiv y_1 \pmod{n_1} \\ x^3 &\equiv y_2 \pmod{n_2} \\ x^3 &\equiv y_3 \pmod{n_3} \end{aligned}$$

Jistě je $x < \min(n_1, n_2, n_3)$. Proto zvolíme-li $N = n_1 n_2 n_3$, bude platit $x^3 < N$. Podle Čínské zbytkové věty ovšem existuje právě jedno číslo ze $\mathbb{Z}_N$, které splňuje dané tři kongruence. Stačí ho tedy celočíselně odmocnit (v $\mathbb{Z}$, nikoliv modulárně).

Chyby při výpočtu

Na závěr ukážeme zajímavý útok na zrychlenou implementaci RSA, která šifruje pomocí pomocí Čínské zbytkové věty (CRT). Představme si, že Alice spočítá $y_p = x^e \bmod p$ a $y_q = x^e \bmod q$ a pak tyto výsledky složí do výsledného y.

Co kdyby v jednom z výpočtů udělala chybu a spočítala místo správného y_p nějaké jiné y'_p? Tehdy by z CRT vyšlo y', které se od správného y liší o násobek q. Podobně pro chybu v y_q.

Útočník tedy může Alici nechat jednu zprávu zašifrovat (podepsat) mnohokrát a jakmile nalezne dva rozdílné výsledky y a y', spočítá $\gcd(y - y', n)$, což bude buď p nebo q.

TODO: Někde zmínit aktivní ovlivňování výpočtu.

Jak používat RSA bezpečně

TODO: Co jsou rizika. Doporučená délka klíče.

TODO: Padding.

8.3 Sémantická bezpečnost

8.4 Rabinova šifra

8.5 Diffieho-Hellmanův protokol

8.6 ElGamalova šifra

8.7* Eliptické křivky