1 RSA

RSA je určite najznámejším a jedným z najpoužívanejších asymetrických šifrovacích systémov. Názov je vytvorený zo začiatočných písmen mien jeho autorov – Ronald L. Rivest, Adi Shamir a Leonard Adleman, ktorí ho zverejnili v roku 1978.

1.1 Inicializácia

Používateľ vytvorí vlastnú inštanciu RSA, teda vytvorí svoj súkromný a verejný kľúč. Popíšme podrobne postup inicializácie.

1. Používateľ zvolí dve (dostatočne veľké) prvočísla p a q ($p \neq q$). Položí $n = p \cdot q$.
2. Používateľ vyberie prirodzené číslo e také, že $1 < e < \phi(n)$ a $\mathrm{nsd}(e, \phi(n)) = 1$, kde $\phi(n) = (p-1)(q-1)$ je Eulerova funkcia a nsd označuje najväčší spoločný deliteľ svojich argumentov. Teda e je nesúdeliteľné s $\phi(n)$.
3. Používateľ vypočíta d také, že $e \cdot d \equiv 1 \pmod{\phi(n)}$.

Aké prvočísla považujeme za veľké, závisí od efektívnosti metód faktorizácie a miere bezpečnosti, ktorú od nášho systému požadujeme. V súčasnosti sa za veľké považujú aspoň 512 bitové prvočísla, aby sme po ich vynásobení dostali aspoň 1024 bitov dlhý modulus n. Bezpečnosti RSA sa budeme podrobnejšie venovať neskôr.

Verejný kľúč je dvojica čísel (e, n). Súkromný kľúč tvorí hodnota d. Parameter d sa niekedy nazýva súkromný/tajný exponent a parameter e verejný exponent. Prvočísla p, q nie sú pre činnosť RSA potrebné, používateľ môže na ne po inicializácii svojej inštancie zabudnúť. Je však dôležité, ako uvidíme neskôr, aby sa prvočísla nedostali do rúk potenciálnych útočníkov.

Priestor správ (otvorených aj šifrových textov) je množina $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$.

Šifrovanie: $E(m) = m^e \bmod n$.

Dešifrovanie: $D(c) = c^d \bmod n$.

Pri výpočte dešifrovacej transformácie potrebujeme okrem súkromného kľúča d poznať aj hodnotu n. Tá je súčasťou verejného kľúča a teda všeobecne známa.

Príklad.
1. Nech $p = 5$ a $q = 11$. Prirodzene, tieto prvočísla nie sú dostatočne veľké pre praktické použitie, slúžia pre náš ilustratívny príklad. Teda $n = 5 \cdot 11 = 55$.
2. Máme $\phi(n) = (5-1)(11-1) = 40$. Zvoľme $e = 3$, nesúdeliteľné s $\phi(n)$.
3. Zodpovedajúce $d = 27$, lebo $d \cdot e \equiv 27 \cdot 3 \equiv 81 \equiv 1 \pmod{40}$.

Priestor správ je množina $\{0, \ldots, 54\}$. Verejný kľúč je $(3, 55)$ a súkromný kľúč 27. Nech je napríklad správa $m = 14$. Potom $E(m) = 14^3 \bmod 55 = 49$ a $D(49) = 49^{27} \bmod 55 = 14$.

Pokiaľ chceme šifrovať text, ktorý je dlhší ako n (máme na mysli zápis textu, napr. bitový), rozdelíme text na bloky a tieto šifrujeme samostatne. Takže ak n je 1024 bitové číslo, tak môžeme text deliť na 1023 bitové bloky a tieto samostatne šifrovať – šifrový text potom pozostáva z blokov dĺžky najviac 1024 bitov.

1.2 Korektnosť

V tejto časti ukážeme matematickú korektnosť RSA. To znamená, že po dešifrovaní šifrového textu dostaneme opäť pôvodný otvorený text.

Veta 1 (Korektnosť RSA). *Pre ľubovoľnú inštanciu RSA systému platí:*

$$\forall m \in \mathbb{Z}_n : \; D(E(m)) = m.$$

Dôkaz. Nech e a d sú verejný a súkromný exponent v inštancii RSA systému s $n = p \cdot q$. Potrebujeme ukázať, že $(m^e \bmod n)^d \bmod n = m$, pre všetky $m \in \mathbb{Z}_n$.

Špeciálny prípad je $m = 0$. Vtedy $E(m) = 0$ a $D(0) = 0$, preto tvrdenie platí. Pre $m \in \mathbb{Z}_n \setminus \{0\}$ budeme uvažovať dva prípady: $\mathrm{nsd}(n, m) = 1$ a $\mathrm{nsd}(n, m) \neq 1$. Vieme, že $ed \equiv 1 \pmod{\phi(n)}$. Teda $\exists k \in \mathbb{N} : \; ed = 1 + k\phi(n)$.

1. $\mathrm{nsd}(n, m) = 1$. Počítajme:

$$\begin{aligned} D(E(m)) &= (m^e \bmod n)^d \bmod n \\ &= m^{ed} \bmod n \\ &= m^{1+k\phi(n)} \bmod n \\ &= m \cdot (m^{\phi(n)})^k \bmod n \\ &= m \bmod n = m. \end{aligned}$$

   Predposledná rovnosť vyplýva z Eulerovej vety.

2. $\mathrm{nsd}(n, m) \neq 1$. Potom buď $p \mid m$ alebo $q \mid m$ (nie však obe súčasne, lebo $0 < m < n$). Bez ujmy na všeobecnosti budeme predpokladať, že $m = l \cdot p^s$, kde $s \geq 1$ a $\mathrm{nsd}(l, n) = 1$ ($s, l \in \mathbb{N}$). Potom

$$\begin{aligned} D(E(m)) &= m^{ed} \bmod n \\ &= (lp^s)^{1+k\phi(n)} \bmod n \\ &= l \cdot (p^{1+k\phi(n)})^s \bmod n. \end{aligned} \qquad (1)$$

Podľa Fermatovej (Eulerovej) vety $p^{q-1} \equiv 1 \pmod q$. Odtiaľ:

$$\begin{aligned} p^{(q-1)(p-1)} &\equiv 1 \pmod q \\ p^{k\phi(n)} &= 1 + aq \qquad \text{pre nejaké } a \geq 1 \\ p^{k\phi(n)+1} &= p + apq = p + an \\ p^{k\phi(n)+1} &\equiv p \pmod n. \end{aligned}$$

Po dosadení do (1) dostaneme:

$$D(E(m)) = lp^s \bmod n = m.$$

□

1.3 Realizovateľnosť

Postupne prejdeme jednotlivými krokmi inicializácie RSA a rozoberieme ich realizovateľnosť.

1.3.1 Generovanie prvočísel

V prvom kroku inicializácie RSA potrebujeme vygenerovať dve veľké prvočísla. Generovanie prebieha nasledovne:

1. Zvolíme náhodné nepárne číslo požadovanej veľkosti.
2. Otestujeme, či je prvočíslom. Ak nie, opakujeme postup.

Prvým problémom pri tomto postupe je, koľkokrát musíme v priemernom prípade voliť nepárne číslo, kým narazíme na prvočíslo. Odpoveďou je Prvočíselná veta (Prime Number Theorem):

$$\lim_{n\to\infty} \frac{\pi(n)}{n/\log n} = 1,$$

kde $\pi(n)$ označuje počet prvočísel nie väčších ako n. Pre veľkosť hodnoty $\pi(n)$ platí napríklad nasledujúci odhad[1]:

$$\ln n - \frac{3}{2} < \frac{n}{\pi(n)} < \ln n - \frac{1}{2}, \qquad \text{pre } n \geq 67.$$

Takže ak uvažujeme o generovaní 512 bitového prvočísla (≈ 154 ciferné číslo v desiatkovej sústave), tak pravdepodobnosť, že náhodne zvolené nepárne 512 bitové číslo je prvočíslom je väčšia ako

$$2 \cdot \left(\frac{2^{513}}{\ln 2^{513} - \frac{1}{2}} - \frac{2^{512}}{\ln 2^{512} - \frac{3}{2}} \right) / 2^{512} \doteq 0.005605.$$

Teda priemerne treba voliť približne 178 krát.

Druhý problém je testovanie prvočíselnosti zvoleného čísla. Prakticky používané algoritmy sú pravdepodobnostné. To znamená, že s určitou nenulovou pravdepodobnosťou, ktorú však vieme ovplyvniť, môžu vypočítať chybný výsledok. Existuje viacero takýchto algoritmov, my si ukážeme Millerov-Rabinov test prvočíselnosti.

Millerov-Rabinov test

vstup:
n – nepárne číslo, pričom $n - 1 = 2^s \cdot t$,
kde t je nepárne
k – počet opakovaní testu

for ($i = 0$; $i < k$; i++) {
  zvolíme náhodne $b \in_R \mathbb{Z}_n^*$;
  if ($b^t \equiv \pm 1 \pmod n$) continue; [2]
  else {
    $l = b^t \bmod n$;
    for ($j = 1$; $j < s \ \wedge \ l \neq -1$; j++)
      $l = l^2 \bmod n$;
    if ($l \neq -1$) return „n je zložené číslo“;
  }
}
return „n je prvočíslo“;

Matematicky sa dá vyjadriť fakt, že pre dané n a zvolené b ($b \in \mathbb{Z}_n^*$) prejde n testom (jedna iterácia) s výsledkom „n je prvočíslo“ takto:

$$\begin{aligned} &b^t \bmod n = \pm 1 \quad \vee \\ &\exists j \in \{1, \ldots, s-1\} : \ b^{t \cdot 2^j} \bmod n = -1. \end{aligned} \tag{2}$$

Pri splnení vlastnosti (2) sa zvyčajne hovorí, že n je silným pseudoprvočíslom vzhľadom na bázu b. Millerov-Rabinov test je pravdepodobnostný algoritmus, takže niekedy môže dať nesprávnu odpoveď. Vlastnosti tohto algoritmu vyplývajú z nasledujúcej vety[3].

Veta 2. *Ak n je zložené nepárne číslo, tak vlastnosť* (2) *platí pre najviac* 25% *všetkých* b *($b \in \mathbb{Z}_n^*$).*

Millerov-Rabinov test dáva takéto výsledky:

- ak je n prvočíslo – odpoveď je vždy správna: „n je prvočíslo“;
- ak je n zložené nepárne číslo – pravdepodobnosť omylu, t.j. odpovede „n je prvočíslo“, je najviac $\frac{1}{4^k}$.

Pravdepodobnosť omylu s rastúcim k veľmi rýchlo (exponenciálne) klesá. Môžeme si stanoviť požadovanú úroveň omylu (nenulovú) a vypočítať hodnotu parametra k tak, aby sme ju dosiahli.

[1] R.L. Graham, D.E. Knuth, O. Patashnik: *Concrete Mathematics: A Foundation for Computer Science*, 2nd Edition, Addison-Wesley, 1994, strana 111
[2] hodnota -1 je v modulárnej aritmetike $\bmod n$ rovná $n - 1$
[3] N. Koblitz: *A Course in Number Theory and Cryptography*, Springer-Verlag, 1987, strana 117
[4] M. Agrawal, N. Kayal, N. Saxena: *PRIMES is in P*, 2002

Poznámka. V roku 2002 bol nájdený deterministický polynomiálny algoritmus na testovanie prvočíselnosti[4]. Jeho časová zložitosť je však pomerne vysoká. Preto je zatiaľ na testovanie prvočíselnosti vhodnejšie použiť niektorý z pravdepodobnostných algoritmov.

1.3.2 Modulárne umocňovanie

Pri inicializácii RSA ako aj pri samotnom šifrovaní a dešifrovaní sa používa operácia umocňovania v modulárnej aritmetike. To znamená, že potrebujeme efektívny algoritmus, ktorý počíta $v = a^b \bmod n$, pričom čísla a, b, n môžu mať niekoľko sto alebo tisíc bitov.

Nech $b = (b_k \dots b_1 b_0)_2$ je reprezentácia čísla b v binárnej sústave. Najjednoduchší algoritmus na modulárne umocňovanie (prirodzene, deterministický a polynomiálny), vyzerá nasledovne:

Modulárne umocňovanie

$v = 1$; $p = a$;
for ($i = 0$; $i \leq k$; i++) {
  if ($b_i == 1$) $v = p \cdot v \bmod n$;
  $p = p^2 \bmod n$;
}
return v;

Operácie mod a prevod do binárnej sústavy je tiež možné robiť efektívne. Upravený algoritmus s prevodom do binárnej sústavy ukrytým vo vnútri:

Upravené modulárne umocňovanie

$v = 1$; $p = a$;
while ($b > 0$) {
  if (b je nepárne) $v = p \cdot v \bmod n$;
  $p = p^2 \bmod n$;
  $b = b/2$; // celočíselné delenie
}
return v;

1.3.3 Konštrukcia parametrov e a d

Verejný exponent e môžeme konštruovať niekoľkými spôsobmi. Môžeme e voliť ako náhodné nepárne číslo a testovať, či $\mathrm{nsd}(e, \phi(n)) = 1$, pričom postup opakujeme, kým vhodné e nenájdeme. Zvyčajne sa však verejný exponent vyberá cielene, s ohľadom na efektívnosť šifrovania (t.j. výpočet funkcie E). Vhodnými kandidátmi sú čísla s krátkou binárnou reprezentáciou alebo s binárnou reprezentáciou v špeciálnom tvare. Napríklad malé prvočísla 3, 7, 11, 13, alebo číslo $65537 = (10000000000000001)_2$.

Súkromný exponent d sa vypočíta z e pomocou rozšíreného Euklidovho algoritmu. Rozšírený Euklidov algoritmus pre $a > b$ počíta $u, v \in \mathbb{Z} : ub + va = \mathrm{nsd}(a, b)$. Ak položíme $a = \phi(n)$ a $b = e$, tak dostaneme u, v: $ue + v\phi(n) = \mathrm{nsd}(e, \phi(n))$. To znamená, že $ue \equiv 1 \pmod{\phi(n)}$. Tajný exponent d dostaneme úpravou u – pripočítaním vhodného celočíselného násobku $\phi(n)$ (napr. ak je u záporné).

1.3.4 Rýchly výpočet dešifrovacej funkcie

Snaha o urýchlenie šifrovacej/dešifrovacej funkcie je vzhľadom na modulárne operácie s dlhými číslami prirodzená. Urýchlenie šifrovacej funkcie sa (okrem iného) dosahuje voľbou malého verejného exponentu e. Keďže súkromný exponent d sa počíta z e a $\phi(n)$, jeho štruktúru nemôžeme príliš ovplyvniť (v skutočnosti môžeme tým, že najskôr zvolíme d s požadovanou štruktúrou a dopočítame e, čo sa však z bezpečnostných dôvodov neodporúča).

Urýchlenie výpočtu $D(c)$ spočíva zvyčajne vo využití Čínskej zvyškovej vety. Používateľ si ako súkromnú informáciu pamätá nielen d, ale aj prvočísla p, q. Dešifrovanie šifrového textu $c \in \mathbb{Z}_n$ prebieha tak, že používateľ vypočíta $a_1 = c^d \bmod p$ a $a_2 = c^d \bmod q$. Otvorený text m dostane takto (pozri dôkaz Čínskej zvyškovej vety):

$$m = a_1 q(q^{-1} \bmod p) + a_2 p(p^{-1} \bmod q) \bmod n. \quad (3)$$

Na prvý pohľad sa môže zdať, že jedno modulárne umocnenie sme nahradili dvoma ($c^d \bmod p$ a $c^d \bmod q$), takže o urýchlení nemôže byť reč. Opak je však pravdou. Dve realizované umocnenia majú kratšie moduly, preto prebehnú rýchlejšie ako jedno umocnenie s dlhým modulom n. Následná lineárna kombinácia (3) je už len elementárnym výpočtom. Poznamenajme, že hodnoty $q^{-1} \bmod p$ a $p^{-1} \bmod q$ môžu byť predvypočítané a nie je potrebné ich pri každom výpočte dešifrovacej funkcie znova počítať.