Y36BEZ – Bezpečnost přenosu a zpracování dat

Róbert Lórencz

6. přednáška

Čínská věta o zbytcích, testy prvočíselnosti, proudové šifry, RC4

http://service.felk.cvut.cz/courses/Y36BEZ
lorencz@fel.cvut.cz

Obsah přednášky

- Čínská věta o zbytcích
- Kvadratická residua
- Generátory
- Rozklad složených čísel
- Proudové šifry, RC4

Čínská věta o zbytcích (1)

Věta 30 – Čínská věta o zbytcích

Nechť $m_1, m_2, \ldots, m_r$ jsou vzájemně nesoudělná kladná celá čísla. Potom systém kongruencí

$$
\begin{aligned}
x &\equiv a_1 \pmod{m_1} \\
x &\equiv a_2 \pmod{m_2} \\
x &\equiv a_3 \pmod{m_3} \\
\vdots & \qquad \vdots \\
x &\equiv a_r \pmod{m_r}
\end{aligned}
$$

má jediné řešení modulo $M = m_1 m_2 \cdots m_r$.

Příklad: Mějme zbytky čísla x:
$|x|_3 = 1, |x|_5 = 2, |x|_7 = 3 \Rightarrow M = 3{\cdot}5{\cdot}7 = 105.$
Nechť $M_3 = 7{\cdot}5 = 35$, $M_5 = 3{\cdot}7 = 21$ a $M_7 = 3{\cdot}5 = 15 \Rightarrow$ pro x platí:
$x = \Big| |x|_3 M_3 y_3 + |x|_5 M_5 y_5 + |x|_7 M_7 y_7 \Big|_M = |1{\cdot}35{\cdot}y_3 + 2{\cdot}21{\cdot}y_5 + 3{\cdot}15{\cdot}y_7|_{105},$

Kvadratická residua (1)

kde $35y_3 \equiv 1 \pmod 3), 21y_5 \equiv 1 \pmod 5)$ a $15y_7 \equiv 1 \pmod 7)$.
Řešním těchto kongruencí dostávamé pro $y_3 = 2, y_5 = 1$ a $y_7 = 1$ a $\Rightarrow$ pro x platí: $x = |1{\cdot}35{\cdot}2 + 2{\cdot}21{\cdot}1 + 3{\cdot}15{\cdot}1|_{105} = |157|_{105} = 52$

Kvadratická residua

Definice – Kvadratické residuum

Pokud m je kladné celé číslo $\Rightarrow$ celé číslo a je kvadratické residuum modulo m, když $\gcd(a, m) = 1$ a kongruence $x^2 \equiv a \pmod{m}$ má nějaké řešení.
Když tato kongruence nemá žádné řešení $\Rightarrow$ a je kvadratické non-residuum modulo m.

Věta 31 – Kvadratické residuum modulo prvočíslo

Nechť p je liché prvočíslo a a je celé číslo nedělitelné p. Potom kongruence $x^2 \equiv a \pmod{m}$ má buď přesně 2 vzájemně nekongruentní řešení modulo p nebo nemá žádné řešení.

Kvadratická residua (2)

Příklad: Pro $p = 7$ (prvočíslo) jsou kvadratická residua čísla 1,2 a 4:

$$\begin{aligned}
|1^2|_7 &= |1|_7 = |1|_7 \\
|2^2|_7 &= |4|_7 = |4|_7 \\
|3^2|_7 &= |9|_7 = |2|_7 \\
|4^2|_7 &= |16|_7 = |2|_7 \\
|5^2|_7 &= |25|_7 = |4|_7 \\
|6^2|_7 &= |36|_7 = |1|_7
\end{aligned}$$

Každé kvadratické residuum se vyskytuje $2\times$. Přitom neexistuje žádná hodnota x, která by vyhovovala následujícím kongruencím:

$$\begin{aligned}
|x^2|_7 &= |3|_7 \\
|x^2|_7 &= |5|_7 \\
|x^2|_7 &= |6|_7
\end{aligned}$$

Kvadratická nonresidua

Kvadratická nonresidua jsou čísla, která nejsou kvadratickými residui. Čísla 3,5 a 6 jsou kvadratickými nonresidui pro případ modula $p = 7$.

Kvadratická residua (3)

Věta 32 – Kvadratického residuum složeného modulu

Nechť $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, je kanonický rozklad n, kde $2 < p_1 < p_2 < \cdots < p_k$ jsou prvočísla a $\alpha_1, \ldots, \alpha_k$ jsou přirozená čísla. Potom a je kvadratickým residuem modulo $n \Leftrightarrow a$ je kvadratické residuum modulo $p_1^{\alpha_1}, p_2^{\alpha_2}, \ldots, p_k^{\alpha_k}$ (platí z činské věty o zbytcích).

Příklad: $x^2 \equiv a \pmod{15}, 15 = 5 \cdot 3$

$$
\begin{array}{lllllll}
|1^2|_{15} & = & |1|_{15} & = & |1|_{15} & \rightarrow & \text{1. kořen} \quad |1^2|_5 = |1|_5,\ |1^2|_3 = |1|_3 \\
|2^2|_{15} & = & |4|_{15} & = & |4|_{15} & \rightarrow & \text{1. kořen} \quad |2^2|_5 = |4|_5,\ |2^2|_3 = |1|_3 \\
|3^2|_{15} & = & |9|_{15} & = & |9|_{15} & \rightarrow & \gcd(9, 15) = 3 \\
|4^2|_{15} & = & |16|_{15} & = & |1|_{15} & \rightarrow & \text{2. kořen} \quad |4^2|_5 = |1|_5,\ |4^2|_3 = |1|_3 \\
|5^2|_{15} & = & |25|_{15} & = & |10|_{15} & \rightarrow & \gcd(10, 15) = 5 \\
|6^2|_{15} & = & |36|_{15} & = & |6|_{15} & \rightarrow & \gcd(6, 15) = 3 \\
|7^2|_{15} & = & |49|_{15} & = & |4|_{15} & \rightarrow & \text{2. kořen} \quad |7^2|_5 = |4|_5,\ |7^2|_3 = |1|_3
\end{array}
$$

- $1, 4$ jsou 4-násobná kvadratická residua.
- $9, 10, 6$ nesplňují podmínku $\gcd(a, n) = 1$.
- $2, 3, 5, 7, 8, 11, 12, 13, 14$ jsou kvadratická nonresidua.

Kvadratická residua (4)

- Pro liché prvočíslo p existuje $\frac{p-1}{2}$ kvadratických residuí modulo p a stejný počet kvadratických nonresiduí modulo p.
- Když je a kvadratickým residuem modulo prvočíslo $p \Rightarrow$ existují přesně 2 kořeny odmocniny výrazu $|x^2|_p$ a to:
    1. číslo a v intervalu $(0, \frac{p-2}{2})$ a
    2. číslo $|-a|_p$ v intervalu $(\frac{p-1}{2}, p-1)$.
- V případě n, které je součinem 2 lichých prvočísel p a q, bude počet kvadratických residuí mod n roven $\frac{(p-1)(q-1)}{4}$.
- V tomto případě vytváří 4 kvadratická residua tzv. úplnou odmocninu modulo n (perfect square mod n).
- Aby kvadratické residuum bylo "odmocninou" modulo n, musí být odmocninou také kvadratická residua modulo p a modulo q.
- Pro číslo $n = 5 \cdot 7 = 35$ je 6 kvadratických residuí 1, 4, 9, 11, 16, 29. Každé má přesně 4 kořeny odmocniny.

Kvadratická residua (5)

Definice – Legendreův symbol

Nechť p je liché prvočíslo, dále mějme celé číslo a a platí $p \nmid a \Rightarrow$ definujeme
Legendreův symbol následovně:

$$\left(\frac{a}{b}\right) = \begin{cases} 1 & \text{když } a \text{ je kvadratickým residuem.} \\ -1 & \text{když } a \text{ je kvadratickým nonresiduem.} \end{cases}$$

Příklad:

$$\left(\frac{1}{5}\right) = \left(\frac{4}{5}\right) = 1 \qquad \text{a} \qquad \left(\frac{2}{5}\right) = \left(\frac{3}{5}\right) = -1$$

Věta 33 – Eulerovo kritérium

Nechť p je liché prvočíslo, a je celé kladné číslo a platí $p \nmid a \Rightarrow$

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}.$$

Kvadratická residua (6)

Důkaz: Předpokládejme, že $\left(\frac{a}{p}\right) = 1 \Rightarrow$ kongruence $x^2 \equiv a \pmod{p}$ má řešení např. $x = x_0$. Použitím Malé Fermatovy věty dostáváme $a^{\frac{p-1}{2}} = (x_0^2)^{\frac{p-1}{2}} = x_0^{p-1} \equiv 1 \pmod{p}$. Odsud, když $\left(\frac{a}{p}\right) = 1 \Rightarrow$ víme, že $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}$.

V případě, že $\left(\frac{a}{p}\right) = -1$ kongruence $x^2 \equiv a \pmod{p}$ nemá řešení $\Rightarrow$ pro celé číslo $i \in \langle 1, p-1 \rangle$ existuje jediné celé číslo $j \in \langle 1, p-1 \rangle$ tak, že $ij \equiv a \pmod{p}$ (dá se dokázat). Protože $x^2 \equiv a \pmod{p}$ nemá řešení $\Rightarrow$ můžeme seskupit čísla $1, 2, \ldots, p-1$ do $\frac{p-1}{2}$ součinových párů rovnající se a. Vynásobením těchto párů dostáváme $(p-1)! \equiv a^{\frac{p-1}{2}} \pmod{p}$. S použitím Wilsonovy věty: $(p-1)! \equiv -1 \pmod{p}$, $\Rightarrow$ dostáváme $-1 = \left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}$.

Kvadratická residua (7)

Zjednodušující předpisy pro výpočet Legendreovy funkce

1. Když $a = 1 \Rightarrow \left(\frac{a}{p}\right) = 1$.
2. Když a je sudé $\Rightarrow \left(\frac{a}{p}\right) = \left(\frac{\frac{a}{2}}{p}\right) \cdot (-1)^{\frac{p^2-1}{8}}$.
3. Když $a > 1$ je liché $\Rightarrow \left(\frac{a}{p}\right) = \left(\frac{|p|_a}{a}\right) \cdot (-1)^{\frac{(a-1)(p-1)}{4}}$

- Pomocí těchto zjednodušujících předpisů lze efektivněji určit zda a je kvadratickým residuem modulo p, kde p je prvočíslo.
- Tyto předpisy se opírají o řadu vět z teorie čísel, které lze najít i s důkazy v [6].
- Zobecněním Legendreovy funkce pro složené moduly je Jacobiho funkce, popis které lze také nalézt v [6].
- Vlastnosti kvadratických residuí se využívají v testech pro vyhledávaní prvočísel.

Kvadratická residua (8)

Jacobiho symbol – funkce

- Je zobecněním Legendreovy funkce pro složené moduly.
- Definuje se pro celé číslo a a lichý celočíselný modul n.

Definice – Jacobiho symbol

Nechť n je liché celé číslo s kanonickým rozkladem $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, kde $p_1 < p_2 < \cdots < p_k$ jsou prvočísla a $\alpha_1, \ldots, \alpha_k$ jsou přirozená čísla. Dále nechť a je celé číslo nesoudělné s $n \Rightarrow$ Jacobiho symbol platí:

$$\left[\frac{a}{n}\right] = \left[\frac{a}{p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}}\right] = \left(\frac{a}{p_1}\right)^{\alpha_1} \left(\frac{a}{p_2}\right)^{\alpha_2} \cdots \left(\frac{a}{p_k}\right)^{\alpha_k}.$$

Příklady: 1. $\left[\frac{2}{45}\right] = \left[\frac{2}{3^2 \cdot 5}\right] = \left(\frac{2}{3}\right)^2 \left(\frac{2}{5}\right) = (-1)^2(-1) = -1.$

2. $\left[\frac{2}{15}\right] = \left(\frac{2}{3}\right)\left(\frac{2}{5}\right) = (-1)(-1) = 1 \Rightarrow$ Jaké x pro $x^2 \equiv 2 \pmod{15}$?

Generátory (1)

Definice – Generátor

Pokud p je prvočíslo a celé číslo g je menší než p a bude-li dále pro každé číslo $b \in \langle 1, p-1 \rangle$ existovat nějaké číslo a takové, že platí $g^a \equiv b \pmod{p} \Rightarrow$ číslo g je generátor modulo p, tj. g je k p primitivní.

Příklad: Mějme $= 7 \Rightarrow$ číslo 3 je generátorem modulo p:

$$\begin{aligned}
|3^6|_7 &= 1\\
|3^2|_7 &= 2\\
|3^1|_7 &= 3\\
|3^4|_7 &= 4\\
|3^5|_7 &= 5\\
|3^3|_7 &= 6
\end{aligned}$$

Každé číslo od 1 do 6 se dá vyjádřit jako $|3^a|_7$. Pro $p = 7$ jsou generátory čísla 3 a 5. Čísla 2, 4 a 6 nejsou generátory.

Generátory (2)

Hledání generátorů je obecně obtížný problém. Generátory modulo prvočíslo p hledáme tak, že náhodně zvolíme číslo z intervalu $\langle 2, p-1 \rangle$ a testujeme je. V případě znalosti kanonického rozkladu čísla $(p-1)$ je testování jednodušší.

Věta 34 – Hledání generátorů

Nechť p je prvočíslo a $p-1 = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, je kanonický rozklad $p-1$, kde $p_1 < p_2 < \cdots < p_k$ jsou prvočísla a $\alpha_1, \ldots, \alpha_k$ jsou přirozená čísla. Celé číslo g je generátorem modulo p, pokud pro všechny hodnoty $p_1, p_2, \ldots, p_k$ platí $|g^{\frac{p-1}{p_i^{\alpha_i}}}|_p \neq 1$.

Příklad: Mějme $p = 13$, potom $p - 1 = 12 = 3 \cdot 2^2$. Pro testování čísla 2 jako generátoru vypočítáme: $|2^{\frac{12}{4}}|_{13} = 8$ a $|2^{\frac{12}{3}}|_{13} = 3$. Žádný z výsledků se nerovná 1 $\Rightarrow$ číslo 2 je generátorem modulo 13. Pro číslo 3 dostáváme $|3^{\frac{12}{4}}|_{13} = 1$ a $\Rightarrow$ 3 nemůže být generátorem modulo 13.

Rozklad složených čísel (1)

- Rozklad čísel na prvočinitele $\in$ nejstarší problémy teorie čísel.
- Rozklad není obtížný, je ale časově náročný.

Známé algoritmy pro rozklad čísel

- Síto číselného pole – Number Field Sieve (NFS), rychlý algoritmus pro rozklad 110 a víc místních čísel.
- Kvadratické síto – Quadratic Sieve (QS), rychlý pro čísla do 110 dekadických čísel.
- Eliptická metoda – Ecliptic Curve Method (ECM), do 43 místních čísel.
- Pollardův algoritmus Monte Carlo a Algoritmus řetězových zlomků (Continued Fraction Algorithm) jsou méně používanými algoritmy.
- Zkusmé dělení – Trial Division (TD), nejstarší algoritmus založený na testování každého prvočísla menšího nebo rovného odmocnině testovaného čísla.

Rozklad složených čísel (2)

Pokud n je součin 2 prvočísel $\Rightarrow$ výpočet kořenů odmocniny modulo n je z hlediska náročnosti výpočtu rovna faktorizaci n. Pokud známe prvočíselný rozklad $n \Rightarrow$ lze snadno spočítat kořeny odmocniny modulo n, jinak je výpočet obtížný, jako rozklad čísla na prvočinitele.

- $\pi(2^{512}) \approx 10^{151}$. Vesmír má $\approx 10^{77}$ atomů. Kdyby každý atom spotřeboval od počátku vzniku vesmíru až do dnes každou μs 1 miliardu prvočísel $\Rightarrow$ by to bylo dohromady 10^{109} prvočísel.
- Postup generování prvočísel nezačíná jejich náhodným generováním a následným rozkladem na prvočinitele.
- Správný postup je testování vygenerovaných čísel na prvočíselnost.
- Testy na prvočíselnost určí s danou pravděpodobností skutečnost, že vygenerované číslo je prvočíslo.

Testy prvočíselnosti (1)

Solovay-Strassenův test

Test čísla p na prvočíselnost:

1. Výběr náhodného $a < p$.
2. Když $\gcd(a, p) \neq 1 \quad \Rightarrow \quad p$ není prvočíslo.
3. Vypočítáme $j = |a^{\frac{p-1}{2}}|_p$.
4. Když $j \neq \left[\frac{a}{p}\right] \Rightarrow p$ určitě není prvočíslo.
5. Když $j = \left[\frac{a}{p}\right] \Rightarrow$ pravděpodobnost, že p je složené, je $\leq 50\%$.

- a, které dosvědčí, že p není prvočíslo říkáme svědek – Witness.
- Když p je složené $\Rightarrow$ pravděpodobnost vystupování náhodného čísla a jako svědka je $\geq 50\%$.
- Opakováním testu t krát pokaždé s jinou hodnotou a docílíme, že pravděpodobnost toho, že složené p projde všemi testy jako prvočíslo je menší než 2^{-t}.

Testy prvočíselnosti (2)

Lehmannův test

Test čísla p na prvočíselnost:

1. Výběr náhodného $a < p$.
2. Vypočítáme $j = |a^{\frac{p-1}{2}}|_p$.
3. Když $j \not\equiv \pm 1 \pmod{p} \Rightarrow p$ určitě není prvočíslo.
4. Když $j \equiv \pm 1 \pmod{p} \Rightarrow p$ pravděpodobnost, že p je složené, je $\leq 50\%$.

- Je jednodušší test na prvočíselnost.
- Opět, když p je složené $\Rightarrow$ pravděpodobnost vystupování náhodného čísla a jako svědka je $\geq 50\%$.
- Opakováním testu t krát vždy s jinou hodnotou a je pravděpodobnost toho, že složené p projde všemi testy jako prvočíslo, menší než 2^{-t}. Přitom se musí vyskytnou minimálně jednou hodnota -1 (krok 2 až 4).

Testy prvočíselnosti (3)

Rabin-Millerův test

Zvolíme náhodně p a spočítáme b a m tak, aby platilo: $p = 1 + 2^b m \Rightarrow$

1. Výběr náhodného $a < p$.
2. Nechť $j = 0$ a $z \leftarrow |a^m|_p$.
3. Když $z = 1 \Rightarrow p$ může být prvočíslem, k další iteraci
4. Dokud $z \neq p - 1 \wedge j \leq b - 2 \Rightarrow$ opakuj $z \leftarrow |z^2|_p, j \leftarrow j + 1$.
5. Když $z \neq p - 1 \Rightarrow p$ určitě není prvočíslo

- Zjednodušená verze testu na prvočíselnost doporučeného normou DSS.
- Pravděpodobnost průchodu testem složeného čísla jako prvočísla klesá rychleji než u předchozích testů
- O $\frac{3}{4}$ hodnot a lze tvrdit, že mohou vystupovat v roli svědků.
- Znamená to, že složené číslo nepronikne t testy častěji než 4^{-t}.

Testy prvočíselnosti (4)

Rady pro generování prvočísel

1. Vygenerování náhodného čísla p s požadovanou délkou bitů n.
2. MSB $=$ LSB $=$ 1: MSB $= 1 \Rightarrow$ záruka požadované délky, LSB $= 1$ $\Rightarrow$ liché číslo.
3. Prověření, zda vygenerované prvočíslo není dělitelné malými prvočísly (prvočísla < 1000). Testování lichého čísla p na prvočíselnost čísly 3, 5 a 7 vyloučí 54% složených čísel a testování s prvočísly < 256 vyloučí 80% složených čísel.
4. Provedení Rabin-Millerova testu na opakovaném generování náhodných čísel a. Volíme menší a. Test opakujeme minimálně 5-krát. V případě, že p v některém testu nevyhoví, vygenerujeme jiné p.
5. Implementace takové metody trvá v závislosti na délce prvočísla řádově sekundy až desítky sekund.

Testy prvočíselnosti (5)

Silná prvočísla

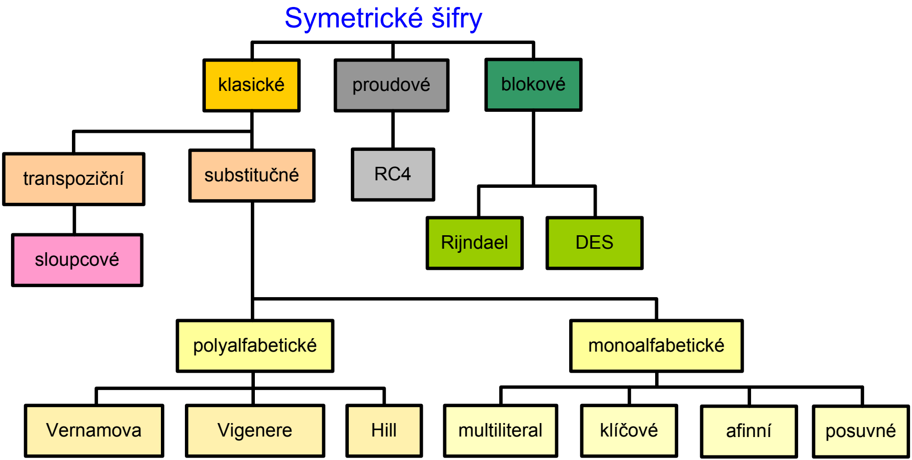
- Když n má být součinem 2 silných prvočísel p a $q \Rightarrow$
- Silná prvočísla mají mít vlastnosti ztěžující rozklad čísla n na prvočinitele $\Rightarrow$
- GCD čísel $(p-1)$ a $(q-1)$ má být malý.
- $(p-1)$ a $(q-1)$mají mít velké prvočinitele p' a q'.
- $(p'-1), (q'-1), (p'+1)$ a $(q'+1)$ mají mít velké prvočinitele.
- $(p-1)/2$ a $(q-1)/2$ mají být prvočísla.

Argumenty proti používaní silných prvočísel:

- Použití silných prvočísel je předmětem diskusí.
- Délka prvočísel je důležitější než jejich struktura.
- Struktura může být na škodu nahodilosti.

Proudové šifry (1)

Rozdělení symetrických šifer



Proudové šifry (2)

Proudové šifry

- Z hlediska použití klíče ke zpracování OT rozeznáváme dva základní druhy symetrických šifer - proudové a blokové.
- Nechť OT používá vstupní abecedu A o q symbolech. Proudová šifra šifruje zvlášť jednotlivé znaky abecedy, zatímco bloková šifra zpracovává najednou bloky (řetězce) délky t znaků.
- Podstatné na blokových šifrách však je, že všechny bloky jsou šifrovány (dešifrovány) stejnou transformací $E_k(D_k)$, kde k je šifrovací klíč.
- Ale proudové šifry nejprve z klíče k vygenerují posloupnost $h_1, h_2, \ldots$ a každý znak otevřeného textu šifrují jinou transformací E_{h_i}.
- Proudové šifry by mohly být chápány i jako blokové šifry s blokem délky $t = 1$, ale u proudových šifer je každý tento "blok" zpracováván jiným způsobem, jinou substitucí.

Proudové šifry (3)

Definice symetrické proudové šifry

- Nechť A je abeceda q symbolů, nechť $M = C$ je množina všech konečných řetězců nad A a nechť K je množina klíčů.
- Proudová šifra se skládá z transformace (generátoru) G, zobrazení E a zobrazení D.
- Pro každý klíč $k \in K$ generátor G vytváří posloupnost hesla $h_1, h_, \ldots$ přičemž prvky h_i reprezentují libovolné substituce $E_{h_1}, E_{h_2}, \ldots$ nad abecedou A.
- Zobrazení E a D každému klíči $k \in K$ přiřazují transformace zašifrování E_k a odšifrování D_k.
- Zašifrování OT $m = m_1, m_2, \ldots$ probíhá podle vztahu $c_1 = E_{h_1}(m_1), c_2 = E_{h_2}(m_2), \ldots$
- Dešifrování ŠT $c = c_1, c_2, \ldots$ probíhá podle vztahu $m_1 = D_{h_1}(c_1), m_2 = D_{h_2}(c_2), \ldots$, kde $D_{h_i} = E_{h_1}^{-1}$.

Proudové šifry (4)

- Z historických důvodů nazýváme G generátor hesla, neboť $h_1, h_2, \dots$ bývá proud znaků abecedy A a substituce E_{h_i} posunem v abecedě A o h_i pozic, tj. $c_i = |m_i + h_i|_q$.
- Proudové šifry jsou příkladem historických tzv. heslových systémů. V anglické literatuře se heslo $h_1, h_2, \dots$ nazývá running-key nebo key-stream (keystream), tj. proud klíče, i když se jedná o derivát originálního klíče k.
- Pokud se proud hesla začne od určité pozice opakovat, říkáme, že jde o periodické heslo a periodickou šifru (Vigenèrova šifra).
- Moderní proudové šifry pracuji nad abecedou $A = 0, 1$, tj. $q = 2$. Sčítání/odčítaní modulo 2 je binárním sčítáním/odčítaní. Platí $|a + b|_2 = |a - b|_2$ a vyjadřuje diferenci bitů. Označuje se zkratkou xor nebo operátorem $\oplus$.
- Jedinou smysluplnou substituci E_{h_i} nad bitem abecedy m_i je transformace $E_{h_i}(m_i) = m_i + h_i$ nebo $E_{h_i}(m_i) = m_i + h_i + 1$.

Proudové šifry (5)

- U moderních proudových šifer ŠT vzniká tak, že jednotlivé bity proudu hesla jsou postupně slučovány s jednotlivými bity proudu OT binárním sčítáním.
- Vzhledem k rovnosti binárního sčítání a odčítání je transformace pro zašifrování a odšifrování také stejná.
- Jako u všech symetrických šifer, odesílatel i příjemce musí mít k dispozici tentýž klíč (heslo).
- Heslo může být vytvořeno zcela náhodně jako u Vernamovy šifry nebo může být vygenerováno deterministicky nějakým šifrovacím algoritmem na základě šifrovacího klíče.

Vernamova šifra

- Vernamova šifra používala náhodné heslo stejně dlouhé jako OT $\Rightarrow$ se heslo ničilo (nikdy nebylo použito k šifrování 2 různých OT).
- Na OT (5b na písmeno v 32znakovém Baudotově kódu) se bit po bitu binárně načítá náhodná posloupnost bitů klíče (děrná páska).

Proudové šifry (6)

Vernamova šifra má vlastnost absolutní bezpečnosti, tj. dokonalého utajení. ŠT nenese žádnou informaci o OT – definice absolutně bezpečné šifry.

Algoritmické proudové šifry

- Heslo se "vypočítá" na základě tajného klíče (distribuuje se).
- Aby klíč nemusel být měněn příliš často $\Rightarrow$ princip náhodně se měnícího inicializačního vektoru (IV).
- IV je pro každou zprávu vybírán náhodně a je přenášen před ŠT v otevřené podobě.
- IV (za účasti tajného klíče nebo bez něj) nastavuje příslušný algoritmus (konečný automat, šifrátor) vždy do jiného (náhodného) počátečního stavu $\Rightarrow$ i při stejném tajném klíči je generována pokaždé jiná heslová posloupnost.
- Za různost hesla zodpovídá IV, za utajenost zodpovídá tajný šifrovací klíč. (podobný princip se využívá i u blokových šifer).

Proudové šifry (7)

Proudová šifra RC4

- Jedna z nejpoužívanějších šifer na internetu (Rivest – 1987).
- Nevyužívá IV $\Rightarrow$ na každé spojení generuje náhodně nový tajný klíč (pomocí asymetrické metody).
- Její popis nebyl oficiálně publikován, i když je znám.
- RC4 zveřejněna neznámým hackerem v roce 1994 (získán disassemblováním z programu BSAFE společnosti RSA).
- Šifra RC4 = "Arcfour" (z důvodu ochrany autorských práv)
- RC4 používá mnoho protokolů a standardů (S/MIME a SSL).
- RC4 umožňuje volit délku klíče, 40b a 128b jsou nejpoužívanější.
- Šifrovací klíč se používá pouze k vygenerování tajné substituce {0, ..., 255} $\rightarrow$ {0, ..., 255}, tedy substituci bajtu za bajt.
- Pomocí tabulky S se pak konečným automatem generují jednotlivé bajty hesla $h_0, h_1, \ldots,$ které se xorují na OT nebo ŠT.

Proudové šifry (8)

Proudová šifra — princip generování náhodné permutace

1. Naplníme identickou permutací: $P_i = i$ pro $i = 0, 1, 2, \ldots, 255$.
2. Pomocí náhodné posloupnosti r promícháme permutaci P.
3. Míchání provádíme postupně tak, že v každém kroku i ($i = 0, 1, 2, \ldots, 255$) v permutaci P vyměníme hodnoty na pozicích i a r_i, tj. hodnoty P_i a P_{r_i} vzájemně vyměníme.

```
P(0)=0, P(1)=1,..., P(255)=255
for i = 0 to 255
{
  vyměň mezi sebou hodnoty P(i) a P(r(i))
  i = i + 1
}
```

- P zůstává stále permutací.
- Výměna postihne každou její pozici.
- Výsledek je nová permutace závislá na náhodné posloupnosti r.

Proudová šifra RC4 — inicializace permutace S

1. Naplníme identickou permutací: $S_i = i$ pro $i = 0, 1, 2, \ldots, 255$.
2. Pomocí posloupnosti bajtů klíče k (délky n) promícháme permutaci S.
3. Míchání provádíme postupně tak, že v každém kroku i ($i = 0, 1, 2, \ldots, 255$) v permutaci S vyměníme hodnoty na pozicích podle následujícího předpisu.

```
S(0)=0, S(1)=1,..., S(255)=255
j = 0
for i = 0 to 255
{
   j = |j + S(i) + k(|i|n)|256
   vyměň mezi sebou hodnoty S(i) a S(j)
}
```

Proudové šifry (10)

Proudová šifra RC4 — princip tvorby hesla RC4

- Počítaní s bajty $\Rightarrow$ redukce modulo 256.
- *i* se systematický zvyšuje modulo 256.
- *j* je náhodný klíčově závislý index .
- Hodnota h_{index} obsahuje heslovou posloupnost generovanou tímto algoritmem.

```
i = j = 0
for index = 0 to n
{
   i = |i + 1|256
   j = |j + S(i)|256
   vyměň mezi sebou hodnoty S(i) a S(j)
   h(index) = |S( S(i) + S(j))|256
}
```

Proudová šifra RC4 — princip generování náhodné posloupnosti