19. 10. 2017

Naším cílem bude konstrukce private-key encryption scheme, které je computationally secure. Víme, že OTP je nepraktický, místo toho chceme jednoduché a bezpečné schéma + krátký klíč. Chceme tedy z krátkého klíče sestrojit pseudonáhodný "pad" (tím xorujeme zprávu).

Pseudonáhodné generátory *PRG*

Vlastnosti, které očekáváme od pseudonáhodného řetězce délky n:

- jakýkoliv bit je rozdělený rovnoměrně, tj. je *unbiased* = *nestranný*, tedy $\Pr[x_i = 1] \approx \Pr[x_i = 0]$,
- nejdelší posloupnost samých jedniček je délky $\mathcal{O}(\log n)$,
- nelze efektivně komprimovat. Kolmogorova složitost se v krypto moc nepoužívá, spíš chceme aby byl nerozlišitelný od náhodného stringu.

Návrhy PRG v praxi:

- stream ciphers:
  - ortogonální k tomu, o čem se budeme bavit,
  - prakticky se věří, že jsou PRG, ale neumí se to dokázat → heuristické
- lineární kongruenční generátory
  - náhodně zvolená čísla $a, b, s_0 \leftarrow \mathbb{Z}_m$, pak volíme $s_i = s_{i-1}a + b (\text{mod } m)$,
  - dávají posloupnosti, které mají hodně korelací,
  - musíme být opatrní i při použití pro simulace,
  - dnes prolomeny

Definice: Řekneme, že $G\colon \{0,1\}^* \to \{0,1\}^*$ je *pseudonáhodný generátor* (*PRG*), pokud platí:

(efektivita) G je *deterministický* algoritmus pracující v polynomiálním čase

(expanze) $|G(x)| = \ell(|x|)$ pro $\ell\colon \mathbb{N} \to \mathbb{N}$ t.ž. $\forall n \in \mathbb{N}\colon \ell(n) > n$

(pseudonáhodnost) pro všechny PPT distinguishery[1] D existuje negligible ε takové, že

$$\forall n \in \mathbb{N}\colon |\Pr[D(G(U_n)) = 1] - \Pr[D(U_{\ell(n)}) = 1]| \leq \varepsilon(n)$$

Kde U_m značí uniformně náhodné bitové řetězce délky m a pravděpodobnosti jsou přes tyto řetězce a náhodné bity D.

Kdyby $\ell(n) = n + 1$, pak PRG generuje jen polovinu ze všech možných výstupů, obecně generuje jen $2^{n-\ell(n)}$ zlomek prostoru, ale přesto chceme neodlišitelnost. Pro naše aplikace chceme polynomiální stretch (o kolik se string protáhne).

Historicky vstupům pseudonáhodných generátorů říkáme *seed*.

Pokud je A PPT algoritmus a místo náhodných bitů dostává výstup PRG, pak se chová skoro stejně jako na skutečně náhodném vstupu.

Definice: Computational OTP Nechť G je PRG, definujme encryption scheme (G_{Enc}, E, D):

(G_{Enc}) $\mathcal{K} \leftarrow \{0,1\}^n$ (seed pro G)

(E) $E_{\mathcal{K}}(m) = G(\mathcal{K}) \oplus m$ pro $m \in \{0,1\}^{\ell(n)}$

(D) $D_{\mathcal{K}}(m) = G(\mathcal{K}) \oplus c$

[1] ne adversary A

Tvrzení: Pokud G je PRG, pak (G_{Enc}, E, D) splňuje computational indistinguishability ciphertextů.
Důkaz: Nechť A je PPT adversary, $m_0, m_1 \in \mathcal{M}$ zprávy. Naším cílem je ukázat, že distribuce ciphertextů pro m_0, m_1 nelze efektivně rozlišit. Budeme porovnávat ideální a reálný svět, definujme proto následující pravděpodobnostní distribuce:

$\mathbf{Real}_0 = E_{\mathcal{K}}(m_0) = G(\mathcal{K}) \oplus m_0$

$\mathbf{Real}_1 = E_{\mathcal{K}}(m_1) = G(\mathcal{K}) \oplus m_1$

$\mathbf{Ideal}_0 = \mathcal{K}' \oplus m_0$ kde $\mathcal{K}' \leftarrow \{0,1\}^{\ell(n)}$

$\mathbf{Ideal}_1 = \mathcal{K}' \oplus m_1$ kde $\mathcal{K}' \leftarrow \{0,1\}^{\ell(n)}$

Chceme dokázat, že $\mathbf{Real}_0$ nelze rozlišit od $\mathbf{Real}_1$. Víme, že $\mathbf{Ideal}_0$ a $\mathbf{Ideal}_1$ jsou totožné distribuce (dle důkazu bezpečnosti OTP). Dokažme, že neumíme rozlišit $\mathbf{Real}_0$ od $\mathbf{Ideal}_0$ (a tím tedy ani nemůžeme rozlišit $\mathbf{Real}_0$ od $\mathbf{Real}_1$). Nechť máme pro spor $D_{R,I}$ pro distribuce $\mathbf{Real}_0$ a $\mathbf{Ideal}_0$, který je efektivní a platí

$$|\Pr[D_{R,I}(\mathbf{Real}_0) = 1] - \Pr[D_{R,I}(\mathbf{Ideal}_0) = 1]| \geq \frac{1}{p(n)}$$

kde p je nějaký polynom.

Potom definujeme distinguishera D pro vstup x:

- $c = x \oplus m_0$ kde $x \leftarrow G(U_n)$
- $b' = D_{R,I}(c)$
- return b'

Vidíme, že $D(x)$ simuluje $\mathbf{Real}_0$. Pokud $x \leftarrow U_{\ell(n)}$, pak $D(x)$ simuluje $\mathbf{Ideal}_0$. Tedy $D(x)$ rozliší s pravděpodobností aspoň $\frac{1}{2} + \frac{1}{p(n)}$. □

Optimálně bychom chtěli vědět, že pokud $P \neq NP$, pak existují PRG. Umíme dokázat, že pokud existují jednosměrné funkce (OWF), pak existují PRG (Johan Håstad, Russell Impagliazzo, Leonid A. Levin, Michael Luby).

Neznáme moc pseudonáhodných generátorů, ale obecně věříme v existenci OWF, takže teoreticky máme i PRG.

P, NP jsou worstcase třídy, tedy existuje problém a existují instance, které jsou těžké vyřešit. V krypto chceme problémy obtížné on-average a navíc chceme generovat i obtížné problémy rovnou s jejich řešením.

`http://blog.computationalcomplexity.org/2004/06/impagliazzos-five-worlds.html`

Jednosměrné funkce OWF

Definice: Řekneme, že $f\colon \{0,1\}^* \to \{0,1\}^*$ je jednosměrná funkce *OWF*, pokud:

1. f lze snadno vyhodnotit (evaluate) v polynomiálním čase
2. pro všechny PPT A existuje negligible ε, taková že

$$\forall n \in \mathbb{N}\colon \Pr[A(f(x), 1^n) \in f^{-1}(f(x))] \leq \varepsilon(n)$$

kde pravděpodobnost je přes uniformně náhodné $x \in \{0,1\}^n$ a mince A. Navíc adversary A nepotřebuje najít x, ale stačí mu libovolný předobraz $f(x)$ (jinak bychom museli konstantní nulu považovat za jednosměrnou funkci, což jistě nechceme).

Řekneme, že $f\colon \{0,1\}^n \to \{0,1\}^\ell$, kterou lze vyhodnotit v čase $\ll t$, je (t, ε)-jednosměrná funkce, pokud je bezpečná oproti všem adversary A běžícím v čase t.

factoring Násobení dvou stejně dlouhých čísel: $f(x, y)$ kde $\|x\| = \|y\| = n$ a $f(x, y) = xy$. Přesněji řečeno f bere náhodný řetězec z a rozdělí ho na dvě stejně dlouhé části (pokud je liché délky, zapomene poslední bit), které vynásobí.

Definice: Factoring předpoklad: pro všechny PPT A existuje negligible ε takové, že

$$\Pr[A(N) \in \{P, Q\}] \leq \varepsilon(n)$$

kde P, Q jsou náhodná prvočísla délky n a $N = PQ$.

Silvio Micali řekl: "Cryptographers seldom sleep well" protože neví, kdo najde polynomiální algoritmus, který rozboří jejich předpoklad.

Nejdelší zatím prolomená RSA challenge měla 768 bitů a trvalo to zhruba 2000 CPU let na 2.2 GHz single core.

Nejlepší (asymptoticky) algoritmus na factoring má čas $\exp(\mathcal{O}(n^{1/2} \log^{1/2} n))$, nejlepší heuristický $\exp(\mathcal{O}(n^{1/3} \log^{1/3} n)))$.

Fakt: pokud factoring předpoklad platí, pak je násobení OWF.

Pokud existuje kvantový počítač, pak existuje efektivní algoritmus na faktorizaci (Shor's algorithm).

subset sum $f(x_1, \ldots, x_n, S) = (x_1, \ldots, x_n, \sum_{i \in S} x_i \mod 2^n)$ kde $x_i \in \{0, 1\}^n$ a $S \subseteq 2^{\{1,\ldots,n\}}$, tedy jde o vyřešení soustavy rovnic. Víme, že tento problém je NP-úplný.

DES, AES Blokové šifry DES a AES (ta druhá se používá například ve WiFi routerech) jsou heuristické konstrukce blokových šifer, tedy to jsou kandidáti na jednosměrné funkce, ale neumíme dokázat jejich bezpečnost. Prolamovat umíme v podstatě jen pomocí bruteforce.

Je jednoduché rozmyslet, že funkce, kterou lze vyhodnotit v čase t_0 nemůže být:

1. $(\mathcal{O}(2^n t_0), 1)$-jednosměrná,
2. $(\mathcal{O}(n), \max\{2^{-n}, 2^{-\ell}\})$-jednosměrná.

Obecně se jedná o trade-off mezi 1. a 2.

Co vlastně OWF schovávají? Třeba subset-sum vrací n^2 bitů $\overline{x}$. Pro f jednosměrnou a $g(x, y) = (x, f(y))$ kde $|x| = |y|$ je funkce g také jednosměrná, ale vrací polovinu vstupu.

Příště ukážeme, že existence OWF implikuje existenci PRG, kde myšlenkou bude hardcore bit.