Pravděpodobnostní algoritmy

17. a 18. přednáška z kryptografie

Obsah

Pravděpodobnostní algoritmy

Generování náhodných bitů

Předpokládejme, že máme algoritmus, který generuje náhodný bit. (Jak to dělá, to ponecháme stranou.)
Máme tedy novou instrukci (na stejné úrovni, jako aritmetické instrukce "součet dvou bitů" a "součin dvou bitů")

$$\gamma \leftarrow RAND,$$

která přiřadí do proměnné γ náhodně nulu či jedničku tak, že

- $P[\gamma = 1] = P[\gamma = 0] = \frac{1}{2}$,
- výsledek instrukce *RAND* je nezávislý na jejích předchozích použitích.

Budeme předpokládat, že jedno zavolání instrukce *RAND* trvá konstantní čas $O(1)$.

Pravděpodobnostní algoritmy

Definice

- *Pravděpodobnostní algoritmy* jsou algoritmy, které používají instrukci *RAND*.
- *Deterministické algoritmy* instrukci *RAND* nepoužívají.

Na úrovni algoritmů budeme náhodné generování zapisovat takto:

- $y \xleftarrow{\not{e}} \{0,1\}$ vygeneruje náhodný bit v čase $O(1)$
- $y \xleftarrow{\not{e}} \{0,1\}^{\times l}$ vygeneruje řetězec náhodných bitů délky l v čase $O(l)$

Pravděpodobnostní algoritmy

Pro pravděpodobnostní algoritmus A a vstup x zavedeme náhodné veličiny:

- $LOOPS$ = počet opakování cyklu při daném běhu algoritmu
- $LOOPTIME$ = čas běhu jednoho cyklu
- $TIME$ = celkový čas běhu algoritmu
- $OUTPUT$ = hodnota výstupu při daném běhu algoritmu

Hodnoty těchto náhodných veličin budou záviset na výsledcích instrukce $RAND$ v daném běhu algoritmu A se vstupem x.

Bude nás zajímat očekávaný čas běhu algoritmu A se vstupem x, aneb střední hodnota náhodné veličiny $TIME$,
a pravděpodobnostní rozdělení výstupní veličiny $OUTPUT$.

Pravděpodobnostní prostor

Přesnější postup: Vytvoříme spočetný pravděpodobnostní prostor simulující chování pravděpodobnostního algoritmu A se vstupem x při různých výsledcích instrukcí *RAND*.

- $\Omega = \{\omega \in \{0,1\}^{\times l}, l \in \mathbb{N}; \omega = \text{přesná trasa běhu}\}$
  Přesná trasa běhu (exact execution path) je posloupnost 0 a 1, v níž každý člen odpovídá jedné instrukci algoritmu, přičemž poslední člen odpovídá instrukci *HALT*. Dále pokud je $i-$tá instrukce *RAND*, tak je $i-$tý člen použit jako výsledek instrukce *RAND*.
- $P(\omega) = \frac{1}{2^{|\omega|}}$, kde $|\omega| = l$ je délka posloupnosti ω.
- Lze dokázat, že $\sum_{\omega \in \Omega} 2^{-|\omega|} = \alpha \leq 1$. Říkáme, že algoritmus A zastaví na vstupu x s pravděpodobností α. Pokud $\alpha = 1$, tak $P : \Omega \mapsto < 0, 1 >$ je pravděpodobnostní funkce na Ω.

Náhodná veličina

Náhodné veličiny budou pak zobrazení definovaná na Ω:

- $TIME(\omega) = |\omega|$
- $OUTPUT(\omega) =$ výstup algoritmu A se vstupem x, jestliže posloupnost ω simuluje běh algoritmu (má na odpovídajících místech výsledky instrukcí $RAND$).

Potom bychom spočetli pravděpodobnost, že čas běhu je l, takto:

- $P[TIME = l] = P(\{\omega \in \Omega, TIME(\omega) = l\}) = \frac{s}{2^l}$, kde $s =$počet přesných tras běhu délky l.

Budeme s náhodnými veličinami zacházet na intuitivní rovině, aniž bychom zacházeli do detailů výpočtu v příslušném pravděpodobnostním prostoru.

Diskrétní rozdělení

Diskrétní rozdělení náhodné veličiny

Nechť náhodná veličina X nabývá spočetně mnoha hodnot z množiny $M = \{x_i, i \in I\}$, kde $I \subseteq \mathbb{N}$.
Pravděpodobnostní funkce P je každá nezáporná funkce z M do $\mathbb{R}$, která splňuje $\sum_{i \in I} P[X = x_i] = 1$.
Pravděpodobnostní funkce udává diskrétní rozdělení náhodné veličiny na množině M.

Střední hodnota náhodné veličiny

Střední hodnota náhodné veličiny X je definována jako $E(X) = \sum_{i \in I} x_i P[X = x_i]$ (vážený průměr).
Střední hodnota je lineární funkce: $E(aX + bY) = aE(X) + bE(Y)$ pro libovolné náhodné veličiny X, Y na množině M a $a, b \in R$.

Diskrétní rozdělení

- Náhodná veličina X má *rovnoměrné rozdělení* na množině $\{1, 2, \ldots, m\}$, pokud $P[X = i] = \frac{1}{m}$ pro každé $1 \leq i \leq m$.
  Střední hodnota $E(X) = \frac{m+1}{2}$.
- Náhodná veličina X má *alternativní rozdělení* s parametrem p na množině $\{0, 1\}$, pokud $P[X = 1] = p$, $P[X = 0] = 1 - p$.
  Střední hodnota $E(X) = p$.
- Náhodná veličina X má *geometrické rozdělení* s parametrem p na množině $\{1, 2, 3, \ldots\} = \mathbb{N}^+$, když pro každé $i \geq 1$ je $P[X = i] = (1 - p)^{i-1} p$.
  Střední hodnota $E(X) = \frac{1}{p}$.

  (Pokus má alternativní rozdělení s pravděpodobností úspěchu p. Geometrické rozdělení udává pravděpodobnost, že první úspěch nastane při $i-$tém opakování pokusu.)

Pravděpodobnostní algoritmy

Algoritmus - házení mincí, dokud nepadne panna

- repeat $y \overset{\not{c}}{\leftarrow} \{0, 1\}$
- until $y = 1$

Analýza algoritmu

Pravděpodobnost, že algoritmus zastaví po jednom cyklu je $\frac{1}{2}$.
Náhodná veličina $LOOPS$ má geometrické rozdělení s parametrem $p = \frac{1}{2}$, očekávaný počet cyklů tedy je $E(LOOPS) = 2$.

Pravděpodobnost, že počet cyklů bude aspoň k je rovna $P[LOOPS \geq k] = \frac{1}{2^{k-1}}$, $\lim_{k \to \infty} P[LOOPS \geq k] = 0$.
Algoritmus se zastaví s pravděpodobností 1, přestože počet kroků algoritmu není omezený.
Situace, že se algoritmus nezastaví, má pravděpodobnost 0.

Pravděpodobnostní algoritmy

Algoritmus GT (=Generate and Test)

Máme dva pravděpodobnostní algoritmy $A(x)$ a $B(x, y)$, kde B vrací *true* či *false*. Algoritmus $GT(x)$ je kombinuje takto:

- repeat $y \leftarrow A(x)$
- until $B(x, y)$
- output y

Analýza algoritmu GT

- Pokud A zastaví s pravděpodobností 1 na vstupu x a pro každý výstup y platí, že B zastaví s pravděpodobností 1 na vstupu (x, y), přitom pro některá y je pravděpodobnost, že $B(x, y)$ vrátí *true*, kladná, pak také GT zastaví s pravděpodobností 1 na vstupu x.

Pravděpodobnostní algoritmy

Analýza algoritmus GT

Buď $\mathcal{H}_1$ jev, že algoritmus zastaví po provedení prvního cyklu, a T množina všech možných výstupů algoritmu $A(x)$.

- Náhodná veličina $LOOPS$ má geometrické rozdělení s parametrem $p = P[\mathcal{H}_1]$.
- $E(TIME) = E(LOOPS)\,E(LOOPTIME) = \frac{1}{p}E(LOOPTIME)$
- pro každé $t \in T$ je $P[OUTPUT = t] = P[OUTPUT = t|\mathcal{H}_1]$

Generování náhodných čísel

Algoritmus RN (=Random Number)

Vstup: přirozené číslo $m \geq 1$

Výstup: náhodné přirozené číslo menší než m

- $l \leftarrow \lceil \log_2(m) \rceil$ (Aneb: $2^{l-1} < m \leq 2^l$)
- repeat
  - $y \overset{\mathcal{U}}{\leftarrow} \{0,1\}^{\times l}$ (Typ String)
  - $n \leftarrow \sum_{i=0}^{l-1} y_i 2^i$ (Typ Integer)
- until $n < m$
- output n

Generování náhodných čísel

Analýza algoritmu RN

- Algoritmus zastaví s pravděpodobností 1.
- $LOOPS$ má geometrické rozdělení s parametrem $p = \frac{m}{2^l} > \frac{1}{2}$, očekávaný počet cyklů tedy je $E(LOOPS) < 2$.
- Čas potřebný pro jeden cyklus je $O(l)$, očekávaný čas je $E(TIME) \in O(2l) = O(l)$, kde $l = \mathrm{len}(m)$.
- Výstup má rovnomětné rozdělení na množině $\{0, \ldots, m-1\}$, tedy $P[OUTPUT = n] = \frac{1}{m}$ pro každé $0 \leq n \leq m-1$.

Generování náhodných čísel

Algoritmus RN (=Random Number)

Vstup: přirozená čísla $1 \leq m_1 < m_2$

Výstup: náhodné přirozené číslo z množiny $\{m_1, \ldots, m_2\}$

- $l \leftarrow \lceil \log_2(m_2 + 1) \rceil$
- repeat
  - $y \overset{q}{\leftarrow} \{0,1\}^{\times l}$ (Typ String)
  - $n \leftarrow \sum_{i=0}^{l-1} y_i 2^i$ (Typ Integer)
- until $m_1 \leq n \leq m_2$
- output n

Očekávaný čas je $O(l)$ a výstup má rovnoměrné rozdělení na množině $\{m_1, \ldots, m_2\}$.

Generování náhodných čísel

Algoritmus GT (=Generate and Test) konkrétněji

Vstup: konečná množina T a její neprázdná podmnožina T'
Výstup: náhodný prvek z T'

- repeat $y \overset{\not{d}}{\leftarrow} T$
- until $y \in T'$
- output y

Předpokládáme, že umíme náhodně vygenerovat prvek z T v očekávaném čase $O(f)$ a že výstup tohoto algoritmu je rovnoměrně rozdělený na množině T.
Dále předpokládáme, že umíme efektivně testovat, zda $y \in T'$ v očekávaném čase $O(g)$, a že $T' \neq \emptyset$.
Přitom oba algoritmy zastaví s pravděpodobností 1.

Generování náhodných čísel

Analýza algoritmu GT

- Algoritmus GT zastaví s pravděpodobností 1.
- $LOOPS$ má geometrické rozdělení s parametrem $p = \frac{|T'|}{|T|} > 0$, očekávaný počet cyklů tedy je $E(LOOPS) = \frac{|T|}{|T'|}$.
- Očekávaný čas jednoho cyklu je $E(LOOPTIME) \in O(f + g)$, celkový očekávaný čas tedy je $E(TIME) \in O(\frac{|T|}{|T'|}(f + g))$.
- Výstup má rovnomětné rozdělení na množině T', tedy $P[OUTPUT = t] = \frac{1}{|T'|}$ pro každé $t \in T'$, $P[OUTPUT = t] = 0$ pro každé $t \in T \setminus T'$.

Generování náhodných prvočísel

Algoritmus RP (=Random Prime)

Vstup: přirozené číslo $m \geq 2$

Výstup: náhodné prvočíslo mezi 2 a m

(resp. náhodné $l-$bitové prvočíslo)

- repeat $n \overset{\mathcal{U}}{\leftarrow} \{2, \ldots m\}$

  (resp. $n \overset{\mathcal{U}}{\leftarrow} \{2^{l-1}, \ldots, 2^l - 1\}$)
- until $IsPrime(n)$
- output n

Algoritmus $IsPrime(n)$ na testování prvočíselnosti zatím berme jako "černou skříňku", která vrací pro každé $n \in \mathbb{N}$ *true* či *false*.

Hustota prvočísel

Pro časovou analýzu algoritmu RP potřebujeme odhadnout, kolik je prvočísel a jaká je jejich "hustota" mezi přirozenými čísly.

Eukleidova věta

Existuje nekonečně mnoho prvočísel.

Tvrzení

Pro každé $n \in \mathbb{N}$ lze najít n po sobě jdoucích složených čísel.
(Mezi po sobě jdoucími prvočísly jsou libovolně velké "díry".)

Hustota prvočísel

Označme $\pi(m)$ počet prvočísel mezi 1 až m, včetně m.

Čebyševova věta

Pro každé přirozené číslo $m \geq 2$ platí: $\pi(m) \in \Theta(\frac{m}{\ln(m)})$

Tvrzení

Pro každé přirozené číslo $m \geq 2$: $\pi(m) \geq \frac{\ln(2)}{2} \frac{m}{\ln(m)} \doteq 0,35 \frac{m}{\ln(m)}$

Hustota prvočísel

Důsledek

Protože $\ln(m) = \frac{\log_2(m)}{\log_2(e)}$, platí také $\pi(m) \in \Theta(\frac{m}{\mathrm{len}(m)})$,
aneb existují $c_1, c_2 > 0$ tak, že pro všechna $m \geq m_0$ platí:

$$c_1 \frac{1}{\mathrm{len}(m)} < \frac{\pi(m)}{m} < c_2 \frac{1}{\mathrm{len}(m)}$$

Poznámka

Prvočísel do $m = 1000$ je celkem 168.

Čebyševův odhad je $\frac{m}{\ln(m)} = \frac{1000}{\ln(1000)} \doteq 145$. Přitom $\mathrm{len}(1000) = 10$,
náš odhad $\frac{m}{\mathrm{len}(m)} = \frac{1000}{\mathrm{len}(1000)} \doteq 100$ je tedy o něco nepřesnější.

Hustota prvočísel

Bertrandův postulát

Pro každé přirozené číslo $m \geq 1$ je $\pi(2m) - \pi(m) > \frac{m}{3\ln(2m)}$.
Aneb prvočísel mezi m a $2m$ je $\Omega(\frac{m}{\ln(m)})$.

Důsledek

Existuje $c > 0$ tak, že pro všechna $m \geq m_0$ platí:

$$c\,\frac{1}{\mathrm{len}(m)} < \frac{\pi(2m) - \pi(m)}{m}$$

Generování náhodných prvočísel

Analýza algoritmu RP pro deterministický IsPrime(n)

- Budeme předpokládat, že algoritmus *IsPrime*(n) pracuje pro všechna $n \leq m$ v čase $O(\tau(l))$, kde $l = \mathrm{len}(m)$, a že $\tau(l) > l$. Pak čas potřebný pro jeden cyklus je $O(\tau(l))$.
- *LOOPS* má geometrické rozdělení s parametrem $p = \frac{\pi(m)}{m-1}$, resp. s parametrem $p = \frac{\pi(2^l)-\pi(2^{l-1})}{2^{l-1}}$ pro $l-$bitové prvočíslo. Díky větám Čebyševově a Bertrandově máme v obou případech odhad $p > \frac{c}{l}$ pro vhodnou konstantu $c \doteq \frac{1}{3}$.
- Očekávaný čas je $E(TIME) \in O(l\tau(l))$.
- Výstup je rovnoměrně rozdělen na množině všech prvočísel.

Generování náhodných prvočísel

Analýza algoritmu RP pro pravděpodobnostní IsPrime(n)

Nechť algoritmus *IsPrime*(n) je pravděpodobnostní algoritmus, který je zatížen jednostrannou chybou: pro n prvočíslo je odpověď *true* jistá, pro n složené číslo je odpověď *true* také možná a to s pravděpodobností nejvýše ϵ.

- Budeme předpokládat, že algoritmus *IsPrime*(n) pracuje pro všechna $n \leq m$ v očekávaném čase $O(\tilde{\tau}(l))$, kde $l = \mathrm{len}(m)$, a že $\tilde{\tau}(l) > l$. Pak očekávaný čas pro jeden cyklus je $O(\tilde{\tau}(l))$.
- *LOOPS* má geometrické rozdělení s parametrem $p > \frac{\pi(m)}{m-1}$, neboť i pro složené n může algoritmus skončit.
  Díky Čebyševově větě dostáváme $p > \frac{c}{l}$ pro vhodné $c \doteq \frac{1}{3}$.
- Očekávaný čas je
  $E(TIME) = E(LOOPTIME)E(LOOPS) \in O(l\tilde{\tau}(l))$.

Generování náhodných prvočísel

Analýza algoritmu RP pro pravděpodobnostní IsPrime(n)

- $P[OUTPUT = n] = \beta < \frac{1}{\pi(m)}$ je stejná pro každé prvočíslo $n \leq m$.
- $P[OUTPUT = n] > 0$ i pro n složené číslo.
- Odhadneme celkovou pravděpodobnost jevu, že výstup je složené číslo.
  $P[n \text{ složené} | IsPrime(n)] = \frac{P[IsPrime(n)|n \text{ složené}]\, P[n \text{ složené}]}{P[IsPrime(n)]} <$
  $< \frac{\epsilon}{\frac{\pi(m)}{m}} < \frac{\epsilon l}{c}$
  Pravděpodobnost, že výstup je složené číslo je tedy $O(\epsilon l)$.
  (Odhad je velmi hruhý, ale postačující pro odpověď na otázku, jak malé máme volit ϵ.)

Generování náhodných faktorizovaných čísel

Algoritmus RS (=Random Sequence)

Vstup: přirozené číslo $m \geq 2$

Výstup: nerostoucí posloupnost čísel mezi 1 a m

- $n_0 \leftarrow m$, $k \leftarrow 0$
- repeat
  - $k \leftarrow k + 1$
  - $n_k \overset{q}{\leftarrow} \{1, \ldots, n_{k-1}\}$ (totéž číslo může být vybráno znova)
- until $n_k = 1$
- output $(n_1, \ldots, n_k)$

Očekávaný čas je $O(l^2)$, kde $l = \mathrm{len}(m)$ (intervaly se budou zhruba půlit, očekáváme $\log_2(m)$ náhodných výběrů).

Generování náhodných faktorizovaných čísel

Algoritmus RFN (=Random Factored Number)

Vstup: přirozené číslo $m \geq 2$, (resp. počet bitů l)

Výstup: náhodné faktorizované číslo $n \leq m$, (resp. $l-$bitové n)

- repeat
  - vygeneruj nerostoucí posloupnost čísel do m pomocí algoritmu RS, získáme $(n_1, \ldots, n_k)$, v níž se mohou čísla opakovat
  - vyber z ní podposloupnost všech prvočísel pomocí $IsPrime(n_i)$, získáme $(p_1, \ldots, p_s)$, přitom zachováme všechny duplicity
  - $n \leftarrow \prod_{i=1}^{s} p_i$ (jakmile násobení překročí m, nebudeme dále násobit)
  - $x \overset{q}{\leftarrow} \{1, \ldots, m\}$ (aby n bylo náhodně dostatečně velké)
- until $x \leq n \leq m$ (resp. $2^l \leq n < 2^{l+1}$)
- output n, $(p_1, \ldots, p_s)$

Generování náhodných faktorizovaných čísel

Analýza algoritmu RFN

Pro deterministický *IsPrime*(n) pracující v čase $O(\tau(l))$:

- Očekávaný čas je $E(TIME) \in O(l^2\tau(l))$.
- Výstup je rovnoměrně rozdělen na množině $\{1, \ldots, m\}$.

Pro pravděpodobnostní *IsPrime*(n) pracující v očekávaném čase $O(\tilde{\tau}(l))$, který může vrátit *true* pro složené číslo s pravděpodobností nejvýše ϵ:

- Očekávaný čas je $E(TIME) \in O(l^2\tilde{\tau}(l))$.
- Všechny správně faktorizované výstupy jsou stejně pravděpodobné.
- Pravděpodobnost, že výstup je špatně faktorizovaný je $O(\epsilon l^2)$ pro dostatečně malé ϵ, zhruba $\epsilon l \leq \frac{1}{2}$.

Generování náhodného prvočísla p s faktorizací $p-1$

Algoritmus RPF

Vstup: přirozené číslo $m \geq 2$, (resp. počet bitů l)
Výstup: náhodné prvočíslo $p \leq m+1$ (resp. $l-$bitové p), spolu s faktorizací pro $p-1$

- repeat
  - vygeneruj náhodné faktorizované číslo do m (resp. $l-$bitové) pomocí algoritmu RFN; získáme n, $(p_1, \ldots, p_s)$
- until $IsPrime(n+1)$
- $p \leftarrow n+1$
- output p, $(p_1, \ldots, p_s)$ faktorizace pro $p-1$

Generování náhodného prvočísla p s faktorizací $p-1$

Analýza algoritmu RPF

Pro pravděpodobnostní *IsPrime*(n) pracující v očekávaném čase $O(\tilde{\tau}(l))$, který může vrátit *true* pro složené číslo s pravděpodobností nejvýše ϵ:

- Očekávaný čas je $O(l^3\tilde{\tau}(l))$.
- Každé prvočíslo p se správně faktorizovaným $p-1$ je vybráno se stejnou pravděpodobností.
- Pravděpodobnost, že výstup p není prvočíslo nebo $p-1$ není správně faktorizováno je $O(\epsilon l^2)$ pro dostatečně malé ϵ, zhruba $\epsilon l \leq \frac{1}{2}$.

Pravděpodobnostní algoritmy

Literatura

- Shoup: A Computational Introduction to Number Theory and Algebra. Kapitola 9.
- Základy teorie pravděpodobnosti najdete tamtéž v kapitole 8, odstavce 1-4 a 10.
- Věty o prvočíslech najdete tamtéž v kapitole 5.

  http://shoup.net/ntb/