Pokročilá kryptologie

Generátory náhodných čísel

prof. Ing. Róbert Lórencz, CSc., Ing. Josef Hlaváč, Ph.D.

České vysoké učení technické v Praze, Fakulta informačních technologií
Katedra informační bezpečnosti

Obsah přednášky

- Kryptoanalýza
- Náhodná čísla v kryptografii
  - Entropie
  - Pseudonáhodné generátory
  - Skutečně náhodné generátory
- Testování náhodnosti

Kryptoanalýza (1)

Kryptoanalýza je věda zabývající se luštěním matematických mechanismů ochrany dat,

- proto je nerozlučně spjata s kryptografií.
- Kryptograf navrhuje a inovuje matematické mechanismy kryptografie na základě potřeb tvůrců informačních systémů a aktuálních kryptoanalytických poznatků.
- Sjednocením kryptografie a kryptoanalýzy vzniká interdisciplinární obor – kryptologie.
- Matematika je hlavním nástrojem kryptoanalýzy.
- V klasické podobě se v kryptoanalýze pracuje s páry otevřený a šifrový text (v ojedinělých případech jen se šifrovým textem)

V současnoti se ale moderní kryptoanalýza neobejde bez teoretické informatiky, aplikované fyziky a jiných oborů zasahujících do informačních technologií.

Kryptoanalýza (2)

- Standardní přístup kryptoanalytika je víra v existenci bezpečného schématu, tj. pokud lze něco prolomit, tak musel někdo udělat chybu, protože chybovat je lidské...
- Klasický kryptoanalytik pracuje s páry otevřeného textu (OT) a šifrového textu (ŠT), případně jen se ŠT nebo jinými veřejně dostupnými informacemi (veřejný klíč).
- Kryptoanalytik hledá způsob jak získat utajovanou informaci, tj. OT nebo klíč ze ŠT nebo dvojic OT a ŠT nebo tajný klíč z veřejného klíče...
  - Hledá matematické zákonitosti mezi OT a ŠT (algebraická, lineární, diferenciální kryptoanalýza atd.).
  - Hledá způsob, kterým by byl schopen získat tajný klíč (faktorizační algoritmy u asymetrické kryptografie při znalosti veřejného klíče).
  - Hledá způsob, jak nalézt kolize u hashovacích funkcí.
  - Snaží se najít periodicity u pseudonáhodných generátorů ...
- Každé “hledání” musí byt provedeno v “rozumném čase”.

Kryptoanalýza (3)

Častý přístup současné kryptografie typu „black-box“

- Autonomní, snadno aplikovatelné moduly.
- Nízké povědomí až aktivní nezájem o vnitřní uspořádání.
- Zřetelný rozdíl mezi „zpravodajským“ a komerčním pojetím.
- Neznalost až vědomá ignorace elementárních principů.
- Chybí použitelný standard kvality.

Současné metody kryptoanalýzy

- Překvapivé útoky v neočekávaných místech systému – obvykle velmi efektivní a těžko odhalitelné postupy.
- Postranní kanály – podcenění fyzikálních projevů zařízení a jiných implementačních faktorů.
- Podcenění heuristické povahy kryptografie.
- Sociální techniky – podcenění lidského faktoru.

Kryptoanalýza (4)

Postranní kanály je interdisciplinární zkoumání *praktických* problémů kryptoanalýzy.

Teorie postranních kanálů – nová kategorie elegantních a účinných útočných metod.

Postranní kanál je každý nežádoucí způsob výměny informací mezi kryptografickým modulem a jeho okolím.

Typy postranních kanálů:

- Časový
- Napěťově - proudový
- Elektromagnetický
- Chybový
- Kleptografický
- ...

Kryptoanalýza (5)

- **Dříve** měl kryptoanalytik k dispozici zachycený ŠT (pár OT a ŠT), v lepším případě měl i popis použité metody.
- **Dnes** analytik komunikuje přímo s napadeným systémem - dává mu povolené příkazy, útok připomíná herní partii – výhrou analytika je prolomení systému.
- **Prokazatelná bezpečnost** je zatím iluzí. Problém je, že si myslíme, že systém je tak bezpečný, jak složitý je problém, o kterém si myslíme, že je neschůdný.
  - Místo **myslíme si** zde ovšem má být **umíme dokázat**.
- Měli bychom očekávat u každého algoritmu oslabení ze dne na den
  - Na druhé straně je realita jiná – aplikace nejsou technicky schopny přejít rychle na jiný algoritmus; některé to nedokážou vůbec; změna algoritmu není procesně podchycena (krizové scénáře, atp.).

Kryptoanalýza (6)

V současnosti kryptoanalýza je silná zbraň.

V moderních útocích se kombinují:

- Elementární matematické slabiny.
  - Původně složitý problém může mít překvapivě snadné řešení.
- Zranitelnosti chybné implementace - zejména postranní kanály.
  - Napadený modul pracuje sám proti sobě.
  - Některé úpravy, které vylepšují implementaci (nižší odběr, rychlejší výpočet atd.) vyzařují víc informací.
- Slabiny lidského faktoru – sociální inženýrství.
  - Zmatený uživatel sám spolupracuje s útočníkem.
  - Predikovatelnost reakcí skupiny uživatelů na definované vnější podněty.

Náhodná čísla v kryptografii – Motivace (1)

Mnoho kryptografických aplikací vyžaduje náhodná čísla

- Generování kryptografických klíčů
- Generování čísel *nonce*, *salt*, výplní (*padding*)
- Vernamova šifra (*one-time pad*)

Vyžadovaná „kvalita“ náhodnosti se u různých aplikací liší

- *Nonce* v některých protokolech stačí jedinečné
- Generování klíčů vyžaduje vyšší kvalitu
- Záruka nerozluštitelnosti Vernamovy šifry platí jen v případě, že klíč byl získán ze skutečně náhodného zdroje s vysokou entropií

Náhodná čísla v kryptografii – Motivace (2)

Základní princip:

„Kryptosystém je jen tak silný, jak silný je jeho nejslabší článek.“

Důsledek: Chybně navržený nebo chybně použitý generátor náhodných čísel může představovat fatální slabinu celého kryptosystému.

Náhodná čísla v kryptografii – Pojmy (1)

Náhodné číslo je číslo vygenerované procesem, který má nepředvídatelný výsledek a jehož průběh nelze přesně reprodukovat. Tomuto procesu říkáme *generátor náhodných čísel* (RNG – Random Number Generator).

U jednoho čísla nelze dost dobře diskutovat, zda je nebo není generováno generátorem náhodných čísel. Pracujeme tedy vždy s *posloupností náhodných čísel* neboli též *náhodnou posloupností*.

V počítači se čísla reprezentují pomocí bitů. Namísto náhodných čísel tedy často pracujeme s *náhodnými bity* resp. *generátory náhodných bitů* a *posloupnostmi (řetězci) náhodných bitů*.

Náhodná čísla v kryptografii – Pojmy (2)

Od náhodných posloupností očekáváme *dobré statistické vlastnosti*:

- Rovnoměrné rozdělení – všechny hodnoty jsou generovány se stejnou pravděpodobností
- Jednotlivé generované hodnoty jsou *nezávislé* – není mezi nimi žádná korelace

Důležitým pojmem je *entropie*.

Entropie (1)

Veličina *entropie* popisuje míru náhodnosti – jak obtížné je hodnotu (náhodné číslo, náhodnou posloupnost, řetězec náhodných bitů) uhodnout.

Entropie je mírou nejistoty či nepředvídatelnosti hodnoty a závisí na pravděpodobnostech možných výsledků procesu, který ji generuje. Entropie je dána vztahem

$$H(X) = -\sum_{i-1}^{n} p_i \log_2 p_i \tag{1}$$

kde X je generovaná hodnota (např. řetězec náhodných bitů dané délky) a $p_1, \ldots, p_n$ jsou pravděpodobnosti všech hodnot $X_1, \ldots, X_n$, které je daný generátor schopen vygenerovat.

Entropie (2)

Entropie se vždy vztahuje k útočníkovi a jeho schopnosti (či neschopnosti) předpovědět vygenerovanou hodnotu. Pokud útočník následující generovanou hodnotu s jistotou zná, entropie je nulová (a nulová je i bezpečnost aplikace, která takto generovanou náhodnou hodnotu využívá).

Lze též říci, že entropie vyjadřuje průměrný počet bitů nutných k zakódování hodnoty při použití optimálního kódování, nebo že entropie vyjadřuje obsažené množství informace vyjádřené v bitech.

Kdy je entropie generátoru náhodných bitů maximální?

Entropie generátoru je maximální, pokud se pro danou délku (počet bitů) generují všechny možné posloupnosti, každá z nich se stejnou pravděpodobností.

Pseudonáhodné generátory – Úvod

Počítače pracují deterministicky $\rightarrow$ jak generovat náhodná čísla?

Generátor pseudonáhodných čísel (PRNG): Algoritmus, jehož výstupem je posloupnost, která sice ve skutečnosti *není* náhodná, ale která se *zdá být* náhodná, pokud útočníkovi nejsou známy některé parametry generátoru.

- Algoritmické $\Rightarrow$ snadno realizovatelné
- Obvykle rychlé
- Zpravidla mají dobré statistické vlastnosti
- Ale: výstup je předvídatelný

Lineární kongruenční generátor (1)

Jeden z nejstarších a nejznámějších způsobů generování pseudonáhodných čísel je *lineární kongruenční generátor*:

$$X_{n+1} = (aX_n + c) \bmod m \tag{2}$$

kde X je posloupnost pseudonáhodných čísel, $m > 0$ je modul (často se používá mocnina dvou), a je násobitel, c je inkrement a X_0 je počáteční hodnota (*seed*).

Pseudonáhodná posloupnost X se opakuje nejvýše po m iteracích. Tohoto maxima dosáhneme, pokud jsou splněny všechny následující podmínky:

- Čísla c a m jsou nesoudělná
- $a - 1$ je dělitelné všemi prvočiniteli m
- Pokud $4|m$, pak také $4|a - 1$

Lineární kongruenční generátor (2)

Kvalita generátoru je velmi závislá na volbě parametrů m, a, c.
Nicméně z kryptologického hlediska jde o velmi problematický RNG:

- Pokud se generovaná čísla použijí jako souřadnice bodů v n-rozměrném prostoru, výsledné body budou ležet na nejvýše $m^{\frac{1}{n}}$ hyperrovinách.
- Když m je mocnina dvou, nízké bity X mají mnohem kratší periodu než celá posloupnost. Generátor například střídavě produkuje sudá a lichá čísla.
  (Pozn. Proto se z hodnot X_n zpravidla vybírají jen některé bity.)

Lineární kongruenční generátor *není kryptograficky bezpečný*.

Kryptograficky bezpečné PRNG (1)

Požadavky na *kryptograficky bezpečné* pseudonáhodné generátory:

- „*Next-bit test*“: Je-li známo prvních k bitů náhodné posloupnosti, neexistuje žádný algoritmus s polynomiální složitostí, který by dokázal předpovědět $(k+1)$. bit s pravděpodobností úspěchu vyšší než $1/2$.
- „*State compromise*“: I když je zjištěn vnitřní stav generátoru (ať už celý nebo zčásti), nelze zpětně zrekonstruovat dosavadní vygenerovanou náhodnou posloupnost. Navíc, pokud do generátoru za běhu vstupuje další entropie, nemělo by být možné ze znalosti vnitřního stavu předpovědět vnitřní stav v následujících iteracích.

Většina používaných PRNG tyto požadavky splňuje jen za určitých podmínek.

Kryptograficky bezpečné PRNG (2)

Příklady, jak realizovat kryptograficky bezpečné PRNG:

- Bezpečná bloková šifra v režimu čítače: Náhodně se zvolí klíč (*seed*) a počáteční hodnota čítače i. Zvoleným klíčem se postupně šifrují hodnoty i, $i + 1$, atd. Je zřejmé, že perioda u n-bitové blokové šifry je 2^n a nesmí dojít k vyzrazení klíče a počáteční hodnoty čítače.
- Kryptograficky bezpečná hešovací funkce aplikovaná na čítač: Náhodně se zvolí počáteční hodnota čítače i. Postupně se hashuje i, $i + 1$, atd. Opět nesmí dojít k prozrazení počáteční hodnoty čítače.
- Proudové šifry jsou v zásadě PRNG, s jehož výstupem se XORuje plaintext.
- Algoritmy založené na teorii čísel, u kterých byl proveden (alespoň nějaký) důkaz bezpečnosti.

Blum-Blum-Shub PRNG (1)

Příkladem PRNG, u kterého se má za to, že je kryptograficky bezpečný, je algoritmus Blum-Blum-Shub:

$$X_{n+1} = X_n^2 \bmod m \tag{3}$$

Modul $m = pq$ je součinem dvou velkých prvočísel p a q. Počáteční prvek (*seed*) je $X_0 > 1$. Mělo by platit, že $p, q \equiv 3 \pmod 4$, a $\gcd(\phi(p-1), \phi(q-1))$ by měl být malý.

Výstupem zpravidla není přímo hodnota X_n, ale její parita nebo několik nejméně významných bitů.

Blum-Blum-Shub PRNG (2)

Pseudonáhodný generátor Blum-Blum-Shub:

- Pomalý
- Poměrně silný důkaz bezpečnosti (spojuje ji s výpočetní náročností faktorizace celých čísel)
- Lze přímo spočítat i-tý prvek posloupnosti:

$$X_i = (X_0^{2^i \bmod (p-1)(q-1)}) \bmod m \tag{4}$$

Kryptograficky bezpečné PRNG (3)

Zmíněné PRNG vyžadují náhodný a tajný vstup, *seed*:

- Bez nějaké *skutečné* náhody se stejně neobejdeme
- Kvalita PRNG se odvíjí i od kvality generování hodnoty *seed*

Entropie výstupu PRNG: dána entropií, která vstupuje (*seed*), **algoritmus samotný nikdy nemůže entropii zvyšovat**.

Skutečně náhodné generátory (1)

Společná vlastnost kryptograficky bezpečných PRNG: Neobejdou se bez parametrů (zejm. *seed*), které je třeba zvolit náhodně. PRNG tedy samy o sobě v kryptografii nestačí, je třeba umět generovat skutečně náhodné hodnoty resp. bity.

Generátory skutečně náhodných čísel (TRNG – True Random Number Generator) využívají *zdroj entropie*, kterým je zpravidla nějaký fyzikální jev nebo vnější vliv. Například:

- Radioaktivní rozpad (projekt HotBits)
- Atmosférický šum (viz projekt random.org)
- Teplotní šum, např. na analogových součástkách
- Chování uživatele (pohyb myší, prodlevy při psaní na klávesnici)
- ...

Skutečně náhodné generátory (1)

Vlastnosti generátorů skutečně náhodných čísel:

- Výstup není předvídatelný, i když známe všechny parametry
- Výstup má zpravidla horší statistické vlastnosti $\rightarrow$ je nutné následné zpracování
- Implementace je složitější, často vyžaduje dodatečný a dedikovaný hardware
- Zdroj entropie je třeba průběžně testovat, časem se mohou zhoršit jeho vlastnosti

Skutečně náhodné generátory (2)

Následné zpracování (*post-processing*) má za cíl vylepšit statistické vlastnosti TRNG, zejména:

- Odstranění nevyváženosti jedniček a nul (*bias*) a zajištění rovnoměrného rozdělení
- Extrakce entropie – zvýšení entropie výstupních bitů za cenu snížení rychlosti jejich generování (*bitrate*)

Skutečně náhodné generátory (3)

John von Neumannův dekorelátor je schopen eliminovat nevyváženost a snížit korelovanost výstupu. Bity se odebírají po dvou. Výstup je následující:

Vstup	**Výstup**
00, 11	– (vstup se zahodí)
01	0
10	1

Skutečně náhodné generátory (4)

Další možnosti, jak zlepšit statistické vlastnosti výstupu z TRNG:

- Výstup TRNG se XOR-uje s výstupem kryptograficky silného PRNG
- Sloučení (XOR) výstupů dvou nebo více různých TRNG („*software whitening*“)
- Hešování výstupu TRNG kryptograficky kvalitní hešovací funkcí

Testování náhodných generátorů (1)

K ověření vlastností náhodných generátorů se používají statistické testy. Testy ověřují, zda generovaná posloupnost splňuje některé vlastnosti náhodné posloupnosti.

Pozor: Statistickými testy lze ukázat, že daný generátor nejspíše NENÍ kvalitní, ale nelze prokázat, že JE kvalitní. Pokud generátor „projde“ všemi testy, je pořád možné, že obsahuje slabinu, kterou testy (vzhledem k tomu, jak jsou postaveny) neodhalily.

Statistické testy jsou založeny na *testování statistických hypotéz* na určité *hladině významnosti*. Nulovou hypotézou je, že testovaná posloupnost je náhodná. Testy vrací *p-hodnotu*, která vyjadřuje sílu důkazů proti nulové hypotéze. Pokud tato *p-hodnota* překročí určitou mez, považujeme nulovou hypotézu za neplatnou.

Testování náhodných generátorů (2)

Příklady testů náhodnosti:

- Frekvenční test – testuje, zda testovaná posloupnost bitů obsahuje přibližně stejný počet nul jako jedniček. Testuje se jednak celá posloupnost, jednak dílčí podposloupnosti.
- „Runs“ test – testuje, zda počet a délka řetězců po sobě jdoucích stejných bitů (samých 1 nebo samých 0) v rámci testované posloupnosti bitů odpovídá náhodné posloupnosti. Opět se testuje celá posloupnost i dílčí podposloupnosti.
- Test hodností matic – zaměřuje se na hodnosti disjunktních podmatic, cílem je odhalit lineární závislost podposloupností pevné délky.
- Spektrální test – diskrétní Fourierova transformace, snaží se odhalit periodicitu

Testování náhodných generátorů (3)

- Maurerův univezální statistický test – testuje, zda lze posloupnost výrazněji zkomprimovat bez ztráty informací. Výrazně komprimovatelná sekvence neobsahuje dostatek entropie.
- „Monkey“ test – Generátor se použije ke generování „písmen“ nějaké „abecedy“ a testuje se, jak často se vyskytují předem určená „slova“ složená z těchto „písmen“. Název testu pochází od příslovečné opice u psacího stroje.
- Testy založené na narozeninovém paradoxu. Např. „Birthday Spacing“ test – zvolí se náhodné body na velkém intervalu, mezery mezi body by měly asymptoticky mít Poissonovo rozdělení.

Testování náhodných generátorů (4)

Uvedené testy bývají součástí obecně známých a používaných sad („baterií“) testů.

Známé sady testů:

- Diehard (George Marsaglia): 12 různých testů, poměrně silných
- Dieharder (Robert G. Brown): re-implementace testů Diehard podle jejich popisu, přidána řada dalších testů
- NIST (National Institute of Standards and Technology): 16 testů

Tyto sady byly nicméně vyvinuty převážně pro testování PRNG. Při testování TRNG je třeba důkladně analyzovat zdroj entropie a navrhnout a provést cílené testy, které by odhalily případné slabiny specifické pro tento zdroj entropie.

Statistický test (1)

Testování hypotéz

- Vstup - pozorované hodnoty *náhodné veličiny* (dat) a *statistický test* odpovídající testované *nulové hypotéze* umožňující ověřit její platnost.
- Statistický test - testovací statistika (test statistic)
  - transformace pozorovaných hodnot (náhodného výběru) pocházejících z určitého rozdělení pravděpodobnosti
  - náhodná veličina s nějakým rozdělením pravděpodobnosti
  - *null distribution*: rozdělení pravděpodobnosti testovací statistiky za platnosti nulové hypotézy H_0

Testování

- Na základě dat vypočítáme hodnotu testovací statistiky, kterou srovnáme s kvantilem označovaným jako tzv. kritická hodnota, jejího rozdělení pravděpodobnosti odpovídajícím zvolené hladině významnosti testu a

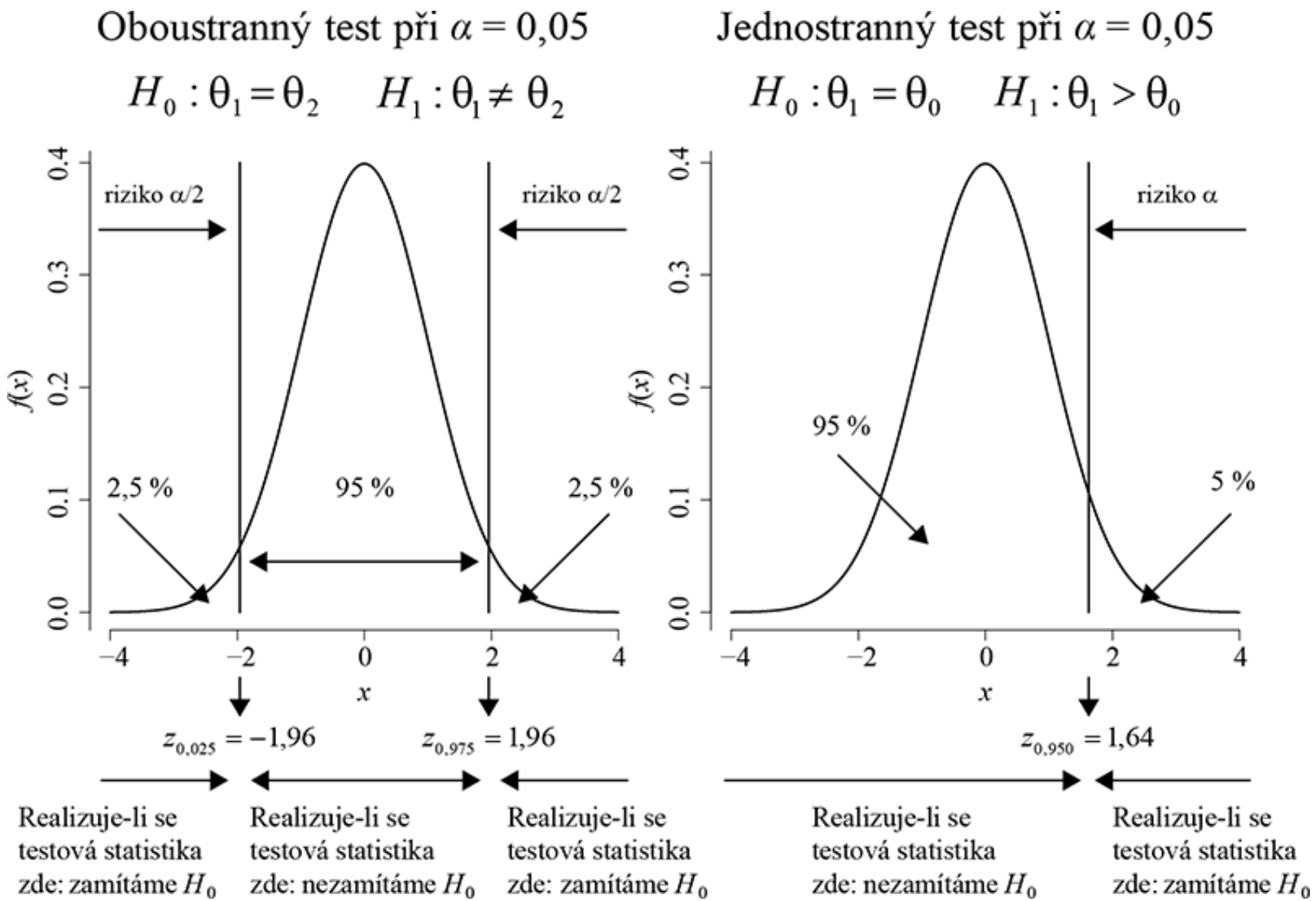
Statistický test (2)

- Pohybuje-li se hodnota realizace testovací statistiky v rozmezí běžných hodnot daných rozdělením pravděpodobnosti testovací statistiky za platnosti nulové hypotézy, H_0 (hodnota realizace nepřekračuje kritickou hodnotu), pak nulovou hypotézu nezmítáme.
- Představuje-li hodnota realizace testovací statistiky extrémnější (méně pravděpodobnou) hodnotu v rámci rozdělení pravděpodobnosti odpovídajícího H_0, než je kritická hodnota (kvantil rozdělení) odpovídající zvolenému riziku a, H_0 zamítáme.
- Hodně pravděpodobné nebo běžné hodnoty realizace testové statistiky v rámci rozděleni pravděpodobnosti testovací statistiky za platnosti H_0 potvrzují platnost statistické hypotézy
- Málo pravděpodobné až extrémní hodnoty realizace testovací statistiky do tohoto rozděleni nepatří $\Rightarrow$ potvrzují neplatnost H_0

Statistický test (3)

- U zvolené alternativní hypotézy riziko špatného rozhodnutí, buď rovnoměrně rozdělujeme na obě extrémní varianty výsledku (extrémně nízké i vysoké hodnoty testové statistiky) $\Rightarrow$ oboustranný test,
- nebo uvažujeme pouze jednu extrémní variantu výsledku (buď extrémně nízké, nebo extrémně vysoké hodnoty testové statistiky) $\Rightarrow$ jednostranný test.
- Obrázek ukazuje kritické hodnoty pro testovací statistiku se standardním normálním rozdělením, hladinou významnosti $a = 0,05$ a oboustrannou i jednostrannou alternativou. Pro oboustrannou alternativu jsou kritickými hodnotami kvantily $z_{a/2}$ a $z_{1-a/2}$, tedy kvantily $z_{0,025}$ a $z_{0,975}$ (čísla -1,96 a 1,96), standardního normálního rozděleni $N(0,1)$, zatímco pro jednostrannou alternativu je kritickou hodnotou kvantil z_{1-a}, tedy kvantil $z_{0,95}$ (číslo 1,64).

Statistický test (4)



Statistický test (5)

Realizaci testovací statistiky v oblasti málo pravděpodobných hodnot rozděleni pravděpodobnosti za platnosti H_0 znamená, že nastala jedna ze dvou situací:

1. H_0 platí a my jsme pozorovali málo pravděpodobný jev
2. H_0 neplatí

Pozorování málo pravděpodobných jevu máme ošetřeno rizikem a (pravděpodobnosti chyby I. druhu), jinými slovy málo pravděpodobné jevy jsou součásti našeho rizika, proto se v takovém případě kloníme k druhé možnosti a zamítáme H_0.
Zamítáme-li H_0, je vždy nutné tuto informaci doplnit právě hodnotou a, tedy informací, na jaké hladině významnosti jsme test prováděli.

Zdroj: http://portal.matematickabiologie.cz

Statistický test (6)

P-hodnota a její interpretace

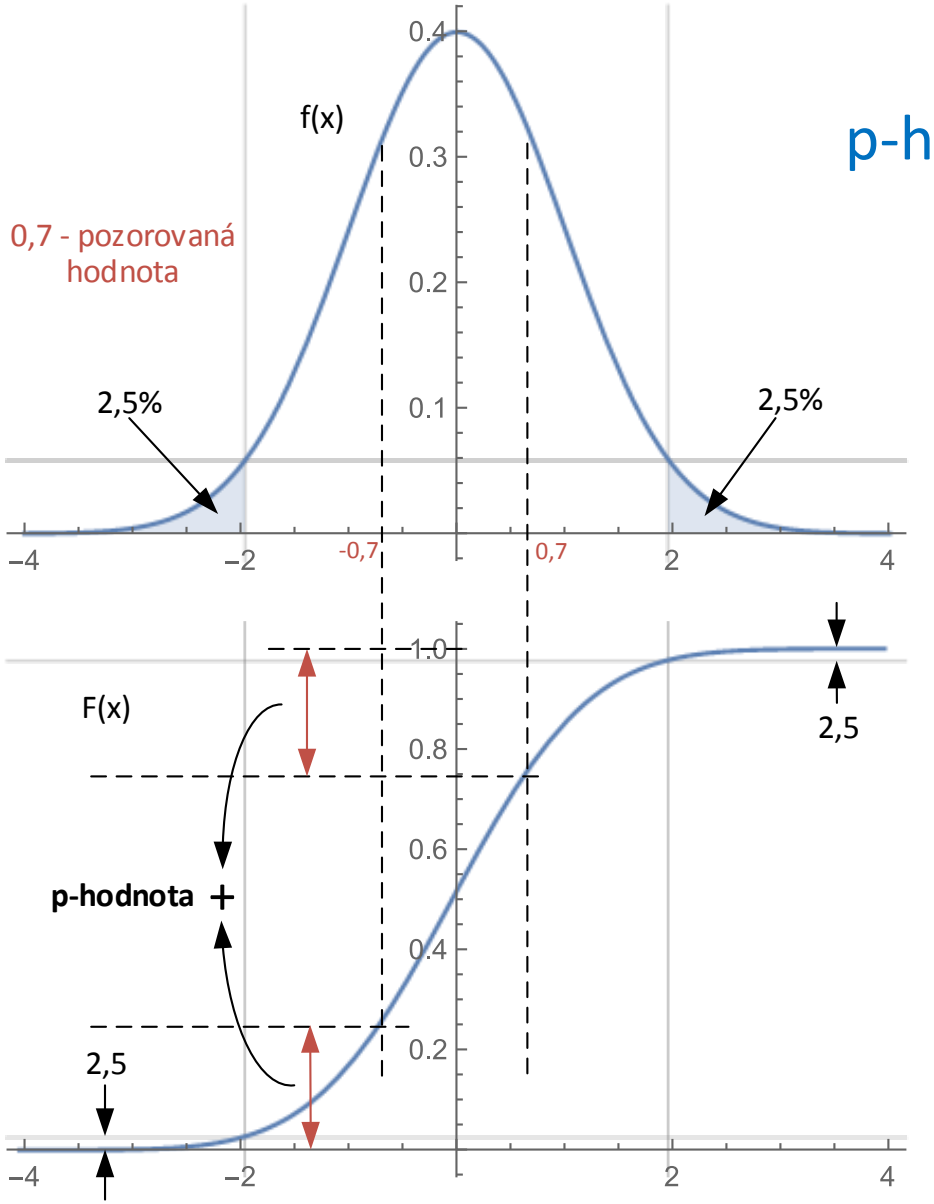
- Místo porovnání hodnoty testovacího kritéria s kritickými hodnotami lze pro rozhodování o platnosti či neplatnosti H_0 použít i *p*-hodnotu (*p*-value).
- *P*-hodnota je pravděpodobnost za platnosti H_0, s níž bychom, vzhledem k jednostrannosti/oboustrannosti testu získali stejnou nebo ještě méně pravděpodobnou hodnotu testovací statistiky.
- Formálně lze *p*-hodnotu definovat i jako nejmenší hladinu významnosti testu, při níž na daných datech ještě zamítneme H_0.
- Čím nižší *p*-hodnota testu je, tím menší nám tento test indikuje pravděpodobnost, že platí H_0. Vyjde-li nám při vyhodnocení statistického testu *p*-hodnota „blízká nule“ (standardně jsou opět přijímány hranice: 5 % a 1 %), znamená to, že naše H_0 má velmi malou oporu v pozorovaných datech a můžeme ji zamítnout.

Statistický test (7)

- Rozhodování o platnosti či neplatnosti H_0 tedy probíhá tak, že výslednou p-hodnotu testu srovnáme se zvolenou hladinou významnosti a s tím, že H_0 je zamítána ve chvíli, kdy p-hodnota testu klesne pod tuto hladinu.
- Tedy, ve chvíli, kdy riziko falešně pozitivního výsledku v souvislosti se zamítnutím H_0 klesne pod vybranou hladinu (napr. 5% nebo 1%), pak ji zamítáme.
- Je-li tedy např. p-hodnota menší než 0,05, H_0 zamítáme a hovoříme o statisticky významném výsledku na hladině významnosti $a = 0{,}05$. Rozhodujeme-li o platnosti H_0 pomocí p-hodnoty, lze p-hodnotu chápat jako číselný indikátor platnosti nebo neplatnosti H_0 vyjádřený na pravděpodobnostní škále.
- Jako každý indikátor, muže i p-hodnota indikovat špatný výsledek (může hrozit jak chyba I. druhu/II. druhu).

Zdroj: http://portal.matematickabiologie.cz

Statistický test 8



p-hodnota (p-value)

p-hodnota vyjadřuje pravděpodobnost obdržení pozorované hodnoty -0,7 nebo extrémnější, za předpokladu platnosti H_0

Statistický test 9

Tabulka chyb při vyhodnocování

TRUE SITUATION	Accept H_0	Accept H_a (reject H_0)
Data is random (H_0 is true)	No error	Type I Error
Data is not random (H_a is true)	Type II error	No error

Příklad útoku

Příklad skutečného kryptoanalytického útoku s využitím slabin náhodného generátoru: bezkontaktní RFID karty s čipem Mifare Classic

- Použitý šifrovací algoritmus (Crypto-1) nebyl veřejně znám („*security by obscurity*“ – nikdy nefunguje). Ukázalo se, že je poměrně slabý.
- Autentizační protokol je typu *challenge-response*. Využívá „náhodně“ generovanou hodnotu *nonce*.
- Náhodný generátor je realizován prostým čítáním hodinových pulsů od přiložení karty ke čtečce (=zapnutí napájení).
- Když čtečka vyšle autentizační příkaz vždy ve stejnou dobu od zapnutí napájení karty, *hodnota nonce bude vždy stejná* (!) – a výrazně se zmenší stavový prostor, který je třeba hrubou silou prohledat k nalezení klíče.