Náhodná čísla a instrukce RdRand

RNDr. Jiří Hladký
hladky.jiri@gmail.com
13. 3. 2014

Obsah

- True Random Generators
- Pseudo-Random Numbers
- Kryptograficky bezpečné PRNG
- AES
- RdRand

Použití náhodných čísel

- Kryptografie, šifrování
- Počítačové simulace
- Statistické metody (Randomization)
- Hry

Fyzikální metody

- “True” random numbers
- Založeny na fyzikálních procesech
  - Radioaktivní rozpad
  - Kvantové jevy
  - Termální šum
  - Kosmické záření
  - Elektromagnetický šum v atmosféře

Fyzikální metody

- Nejsou reprodukovatelné
- Problém s biasem
- Stárnutí
- Problém s rychlostí (poměr cena/výkon)
- Potřeba testovat výstup
- Výstup je často upravován (whitening)

Online služby

- HotBits – radioaktivní rozpad
- Random.org – atmosferický šum
- randomnumbers.info – kvantová optika
- random.irb.hr – kvantová optika

Pseudorandom numbers

- Algoritmické metody
- Reprodukovatelnost
- Snadná dostupnost

Lineární generátory

- LCG $X_n \equiv (a\,X_n + c) \bmod m$
- Combined LCG $X_n \equiv \left(\sum_{j=1}^{k} (-1)^{j-1}\, Y_{i,j} \right) \bmod \left((m_1 - 1) \right)$
- Mersenne Twister
- Velmi dobře prozkoumané
- Rychlé
- Oblíbené pro počítačové simulace

Nelineární generátory

- Inversive congr. gen $X_{n+1} \equiv (a\, X_n^{-1} + b) \bmod m$
- Explicit inversive congruential generator $X_n \equiv \left(a\,(n_0 + n) + b\right)^{-1} (\bmod\, m)$
- RANLUX
- AES jako RNG
- Nemají mřížkovou strukturu
- Obecně pomalé (kromě AES)

Kryptograficky bezpečné PRNG

- Vyhovují next-bit testu
- Pokud byl jejich vnitřní stav kompromitován, nedá se odvodit jejich předchozí výstup

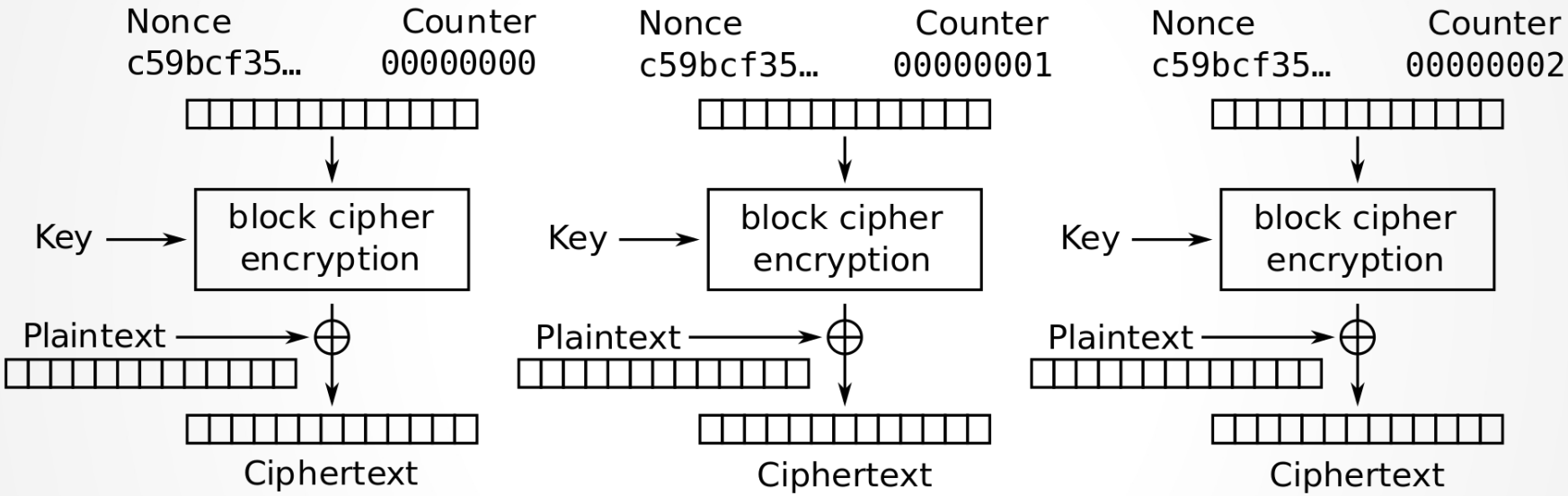
Kryptograficky bezpečné PRNG

- Bloková šifra (AES) v CTR nebo OFB módu
- Kryptograficky bezpečná hashovací funkce (SHA-1) v CTR módu
- Blum Blum Shub $X_{n+1} \equiv X_n^2 \bmod M \, , M = pq$
- Fortuna

Advanced Encryption Standard

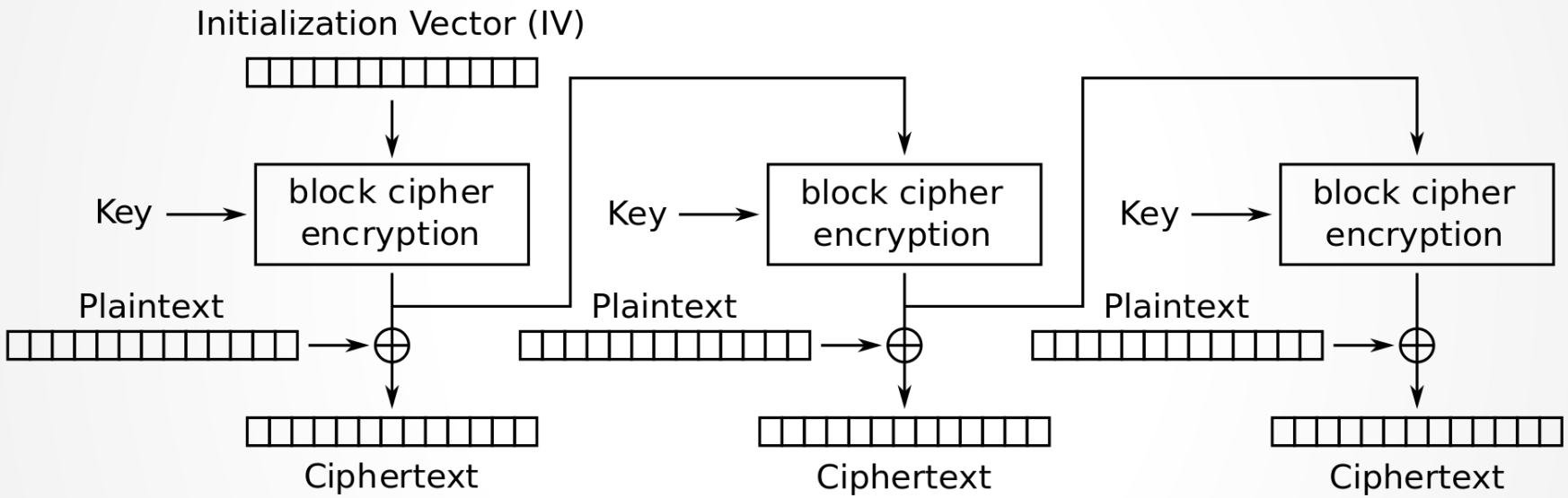
- NIST standard 2001
- Symetrická bloková šifra
- Velikost bloku: 128 bitů
- Délka klíče: 128, 192, 256 bitů
- Počet cyklů: 10, 12, 14
  - Nelineární operace (S-box, substituční šifra)
  - Lineární mixování
  - XOR se sub-klíčem

Counter Mode



Counter (CTR) mode encryption

Output feedback



Output Feedback (OFB) mode encryption

Příkazy

- `openssl enc -e -aes-128-ctr -K 0 -iv 0 -p -in /dev/zero -out >(pv >/dev/null)`
- `openssl enc -e -aes-128-ofb -K 0 -iv 0 -p -in /dev/zero -out >(pv >/dev/null)`
- Rychlost 1.5 GiB/s s HW AES-NI instrukcí

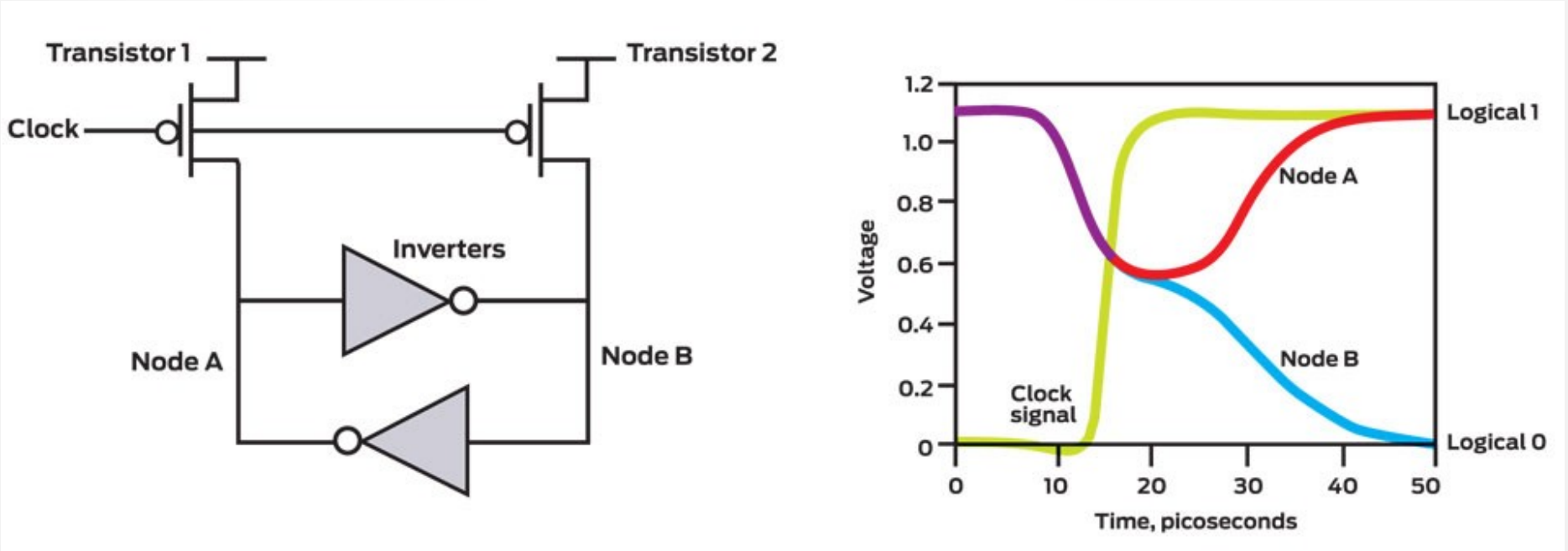
Instrukce RdRand

- HW RNG široce dostupný
- 2011 Intel Ivy Bridge
- Rychlost 800MiB/s
- Jeden na čipu
- Kaskádová konstrukce

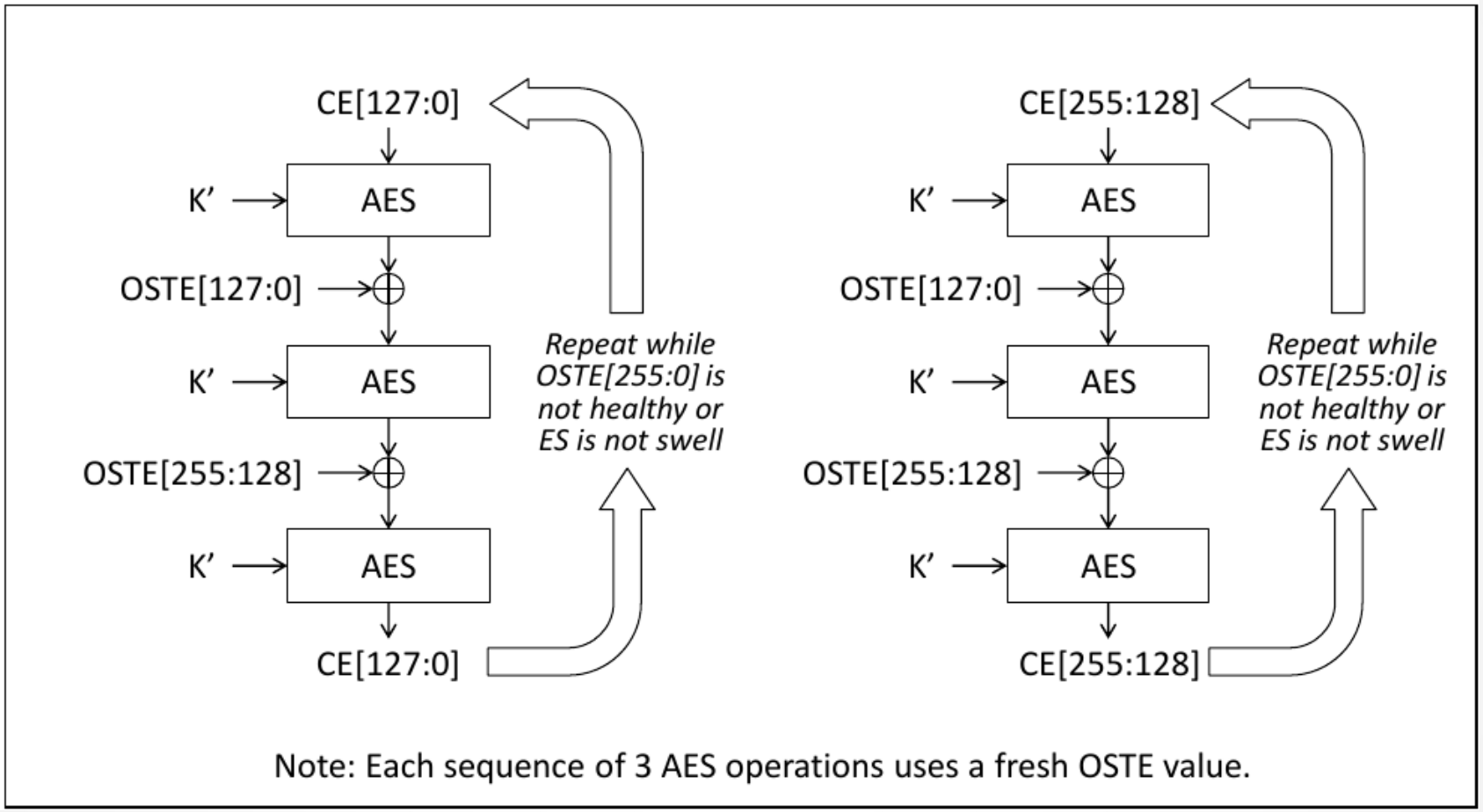
Konstrukce RdRand

- HW zdroj entropie
- Online test, případné korekce zdroje entropie (bias)
- Extrakce entropie AES-CBC-MAC
- Kryptograficky bezpečný RNG založený na AES CTR

Zdroj entropie



Extrakce entropie



AES CTR DRBG: Reseed

Let K denote the key for the DRBG,V the 128-bit counter, and C the number of outputs since the last reseeding. All three are initialized to zero at reset. Then the DRBG reseeds as follows.

1. V[15:0] = (V[15:0] + 1) mod 65536
2. Temp = AES (K,V)
3. V[15:0] = (V[15:0] + 1) mod 65536
4. V = CE[255:128] XOR AES(K,V)
5. K = CE[127:0] XOR Temp
6. C = 0.

AES CTR DRG: Generate

```
V[15:0] = (V[15:0] + 1) modulo 65536
C = C+ 1
Output = AES(K,V)
If (update needed)
{
        V[15:0] = (V[15:0] + 1) modulo 65536
        Temp = AES(K,V)
        V[15:0] = (V[15:0] + 1) modulo 65536
        V = AES(K,V)
        K = Temp
}
```

- Výstup: maximálně 512 128-bitových bloků

Kontroverze: NSA

- Working with **hardware** and software vendors to weaken encryption and random number generators.
- We already know NSA/GCHQ have been collaborating with hardware manufacturers to 'enable' decryption on several major VPN encryption chips.
- Řešení: RdRand jako vstup pro CSPRNG (Fortuna nebo AES-CTR)

Links

- http://www.cryptography.com/public/pdf/Intel_TRNG_Report_20120312.pdf
- https://github.com/BroukPytlik/RdRand
- http://www.iro.umontreal.ca/~simardr/testu01/tu01.html
- http://pracrand.sourceforge.net/
- http://www.gnu.org/software/gsl/manual/html_node/Random-Number-Generator-Performance.html#Random-Number-Generator-Performance
- https://github.com/waitman/libfortuna

Thank you!