RNG with compromise recovery

Homework III.

PA193 – Secure coding

Marek Sýs
Faculty of Informatics, Masaryk University, Brno, CZ

CROCS
Centre for Research on Cryptography and Security

Task

1. Design and implement your own **secure** RNG
2. RNG provides method **`generateData(byte[] buffer, int length);`** which will fill `buffer` with required amount of pseudorandom data (`length` parameter)
3. RNG should be capable to recover from compromise of its internal state by an attacker. After recovery, attacker should not be able to predict pseudorandom data produced by RNG
4. RNG should recover as fast as possible
5. Test output of your RNG with NIST STS or Dieharder battery

What to submit

- Upload your solution to IS homework vault
  - Three files
  - your program (*.c, *.cpp, *.java, *.py…)
  - Result.txt – result of randomness testing
  - Text – description of your program, interpretation of results and RNG characteristics (recovery speed, security)
- Discuss properties of your recovery mechanism
  - speed, security
- Deadline: 27.10.2016 23:59 (full number of 7 points)
  - Every additional 24h started means 2 points penalization