# RNG with compromise recovery

# Homework IV.

**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 you rown **secure** RNG.
2. RNG provides method **generateData(byte[] buffer, int length);** which will fill buffer with required amount of pseudorandom data (length paramater)
3. RNG should be capable to recover from compromise of its internal state by an attacker. After recovery, should not be able to predict pseudorandom produced by RNG.
4. RNG should recover as fast as possible.
5. Test output of your RNG with NIST STS, Dieharder or TestU01 battery.

# What to submit

- Upload your solution to IS homework vault
  - Three files
  - Your program (*.c, *.cpp, *.java, *.py,…)
  - Results.txt – results 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: 02.11.2017 23:59 (full number of 5 points)
  - Every additional 24h started means 1.5 points penalization