KU LEUVEN

Canary Numbers:

Design for Light-weight Online Testability of True Random Number Generators

Vladimir Rožić, Bohan Yang, Nele Mentens and Ingrid Verbauwhede

HECTOR

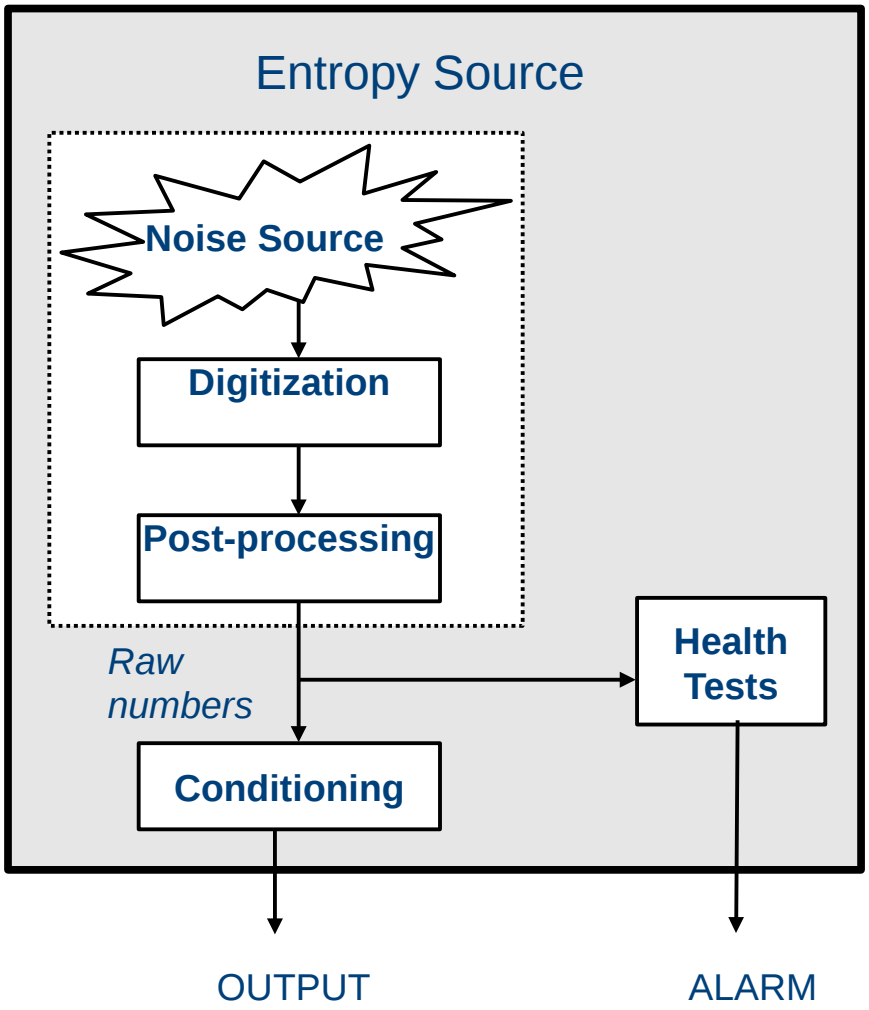
Acknowledgment

This work is supported in part by the **European Commission** through the Horizon 2020 research and innovation program under grant agreement No 644052 **HECTOR**

HECTOR

KU LEUVEN

Generic TRNG Architecture



-False alarm rate vs. usefulness

-Better performance for longer sequences

-High latency

The role of the canary

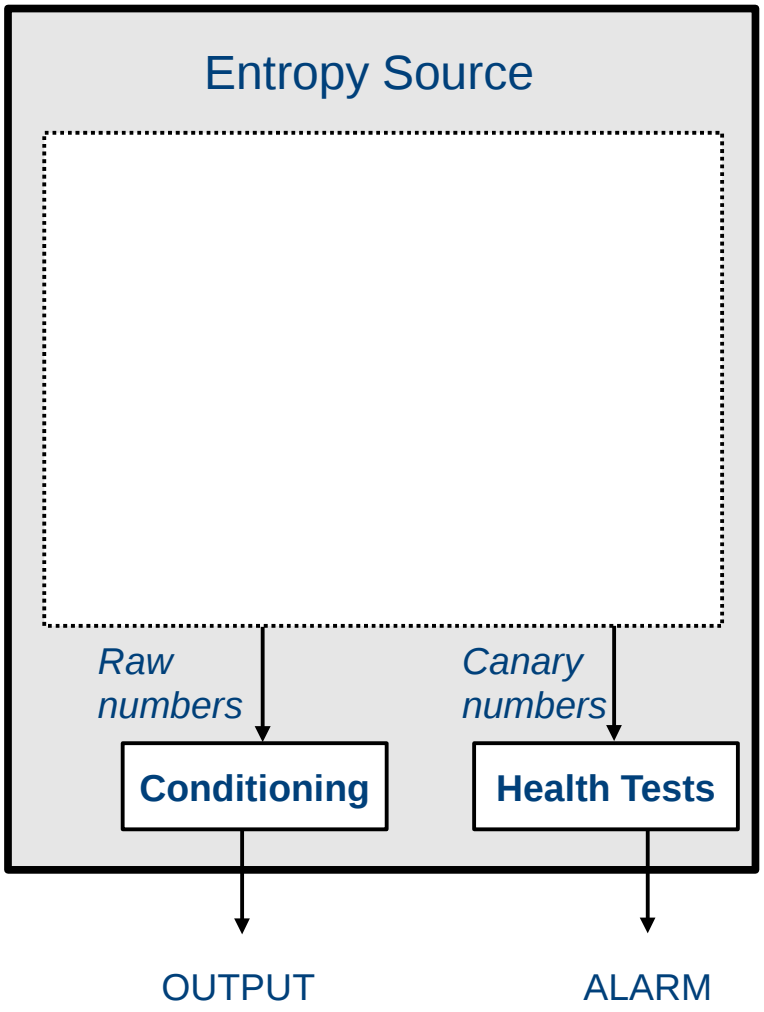
-Early-warning threat detection

-Canaries in security:

-Software: Canary values, a countermeasure against the buffer overflow attack.

-Hardware: Canary logic, redundant logic paths with high propagation delay

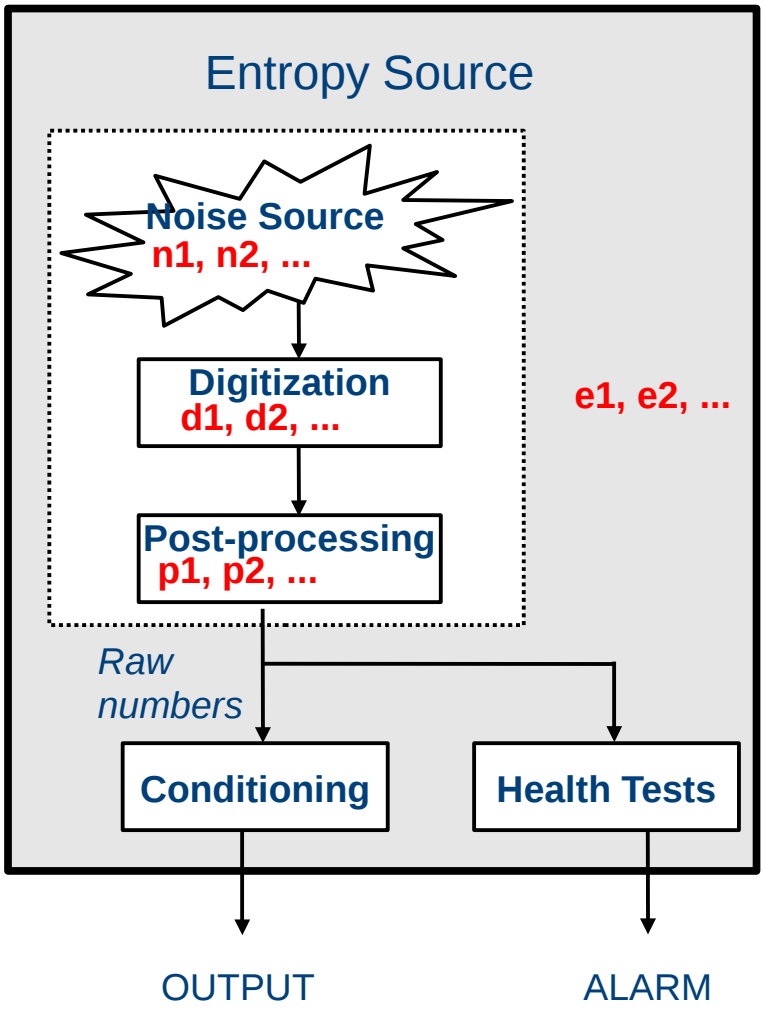
Canaries in TRNGs



-GOALS:

- Higher sensitivity to attacks
- Early attack detection
- Statistical testing on the canary numbers
- Low false positive error rate
- High usefulness
- Low latency
- Low area

TRNG parameters



-Design parameters

- Noise Source (n_1, n_2,...)
- Digitization (d_1, d_2, …)
- Post-processing (p_1, p_2, ...)

-Environment parameters (e_1, e_2, …)

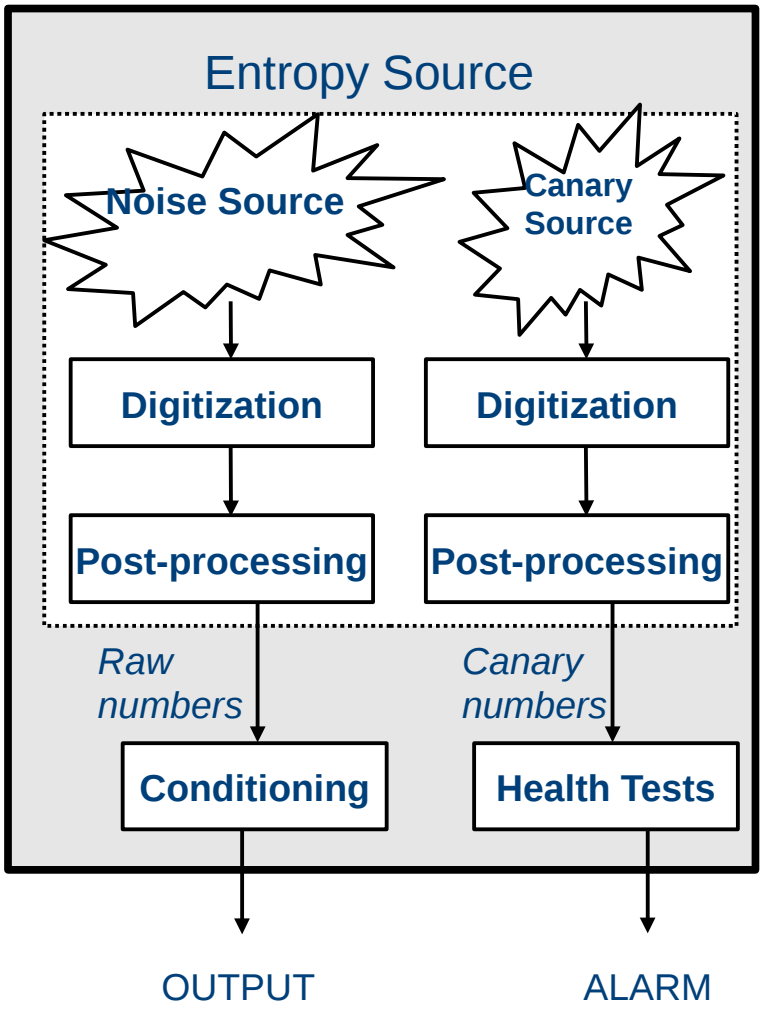
- Critical parameter e_c

Entropy and Testability

$$\left.\frac{\partial H_{raw}}{\partial e_c}\right|_{e_c = e_{c,OP}} \approx 0$$

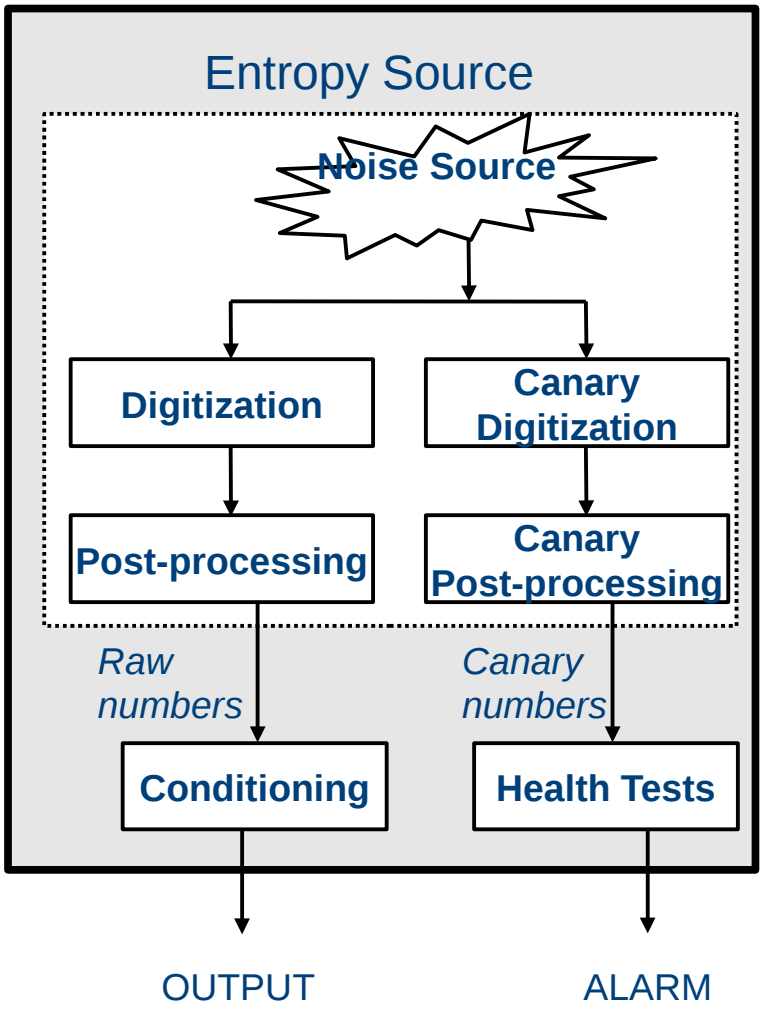
$$testability = \left.\frac{\partial f}{\partial e_c}\right|_{e_c = e_{c,OP}}$$

Replica-based architecture



- Weaker replica of the noise source
- Design space (n_1, n_2, ...)
- Detects global changes in environment
- Not a stand-alone countermeasure

Canary-extraction based architecture



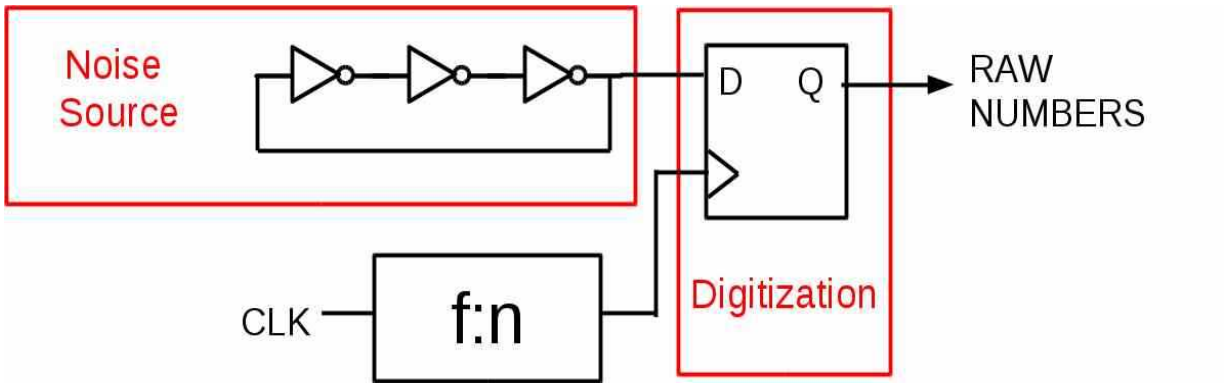
-Weaker processing of the noise

-Design space (d_1, d_2…p_1, p_2,...)

-Testing the noise source

KU LEUVEN

Case Study 1: Elementary TRNG



Stochastic model

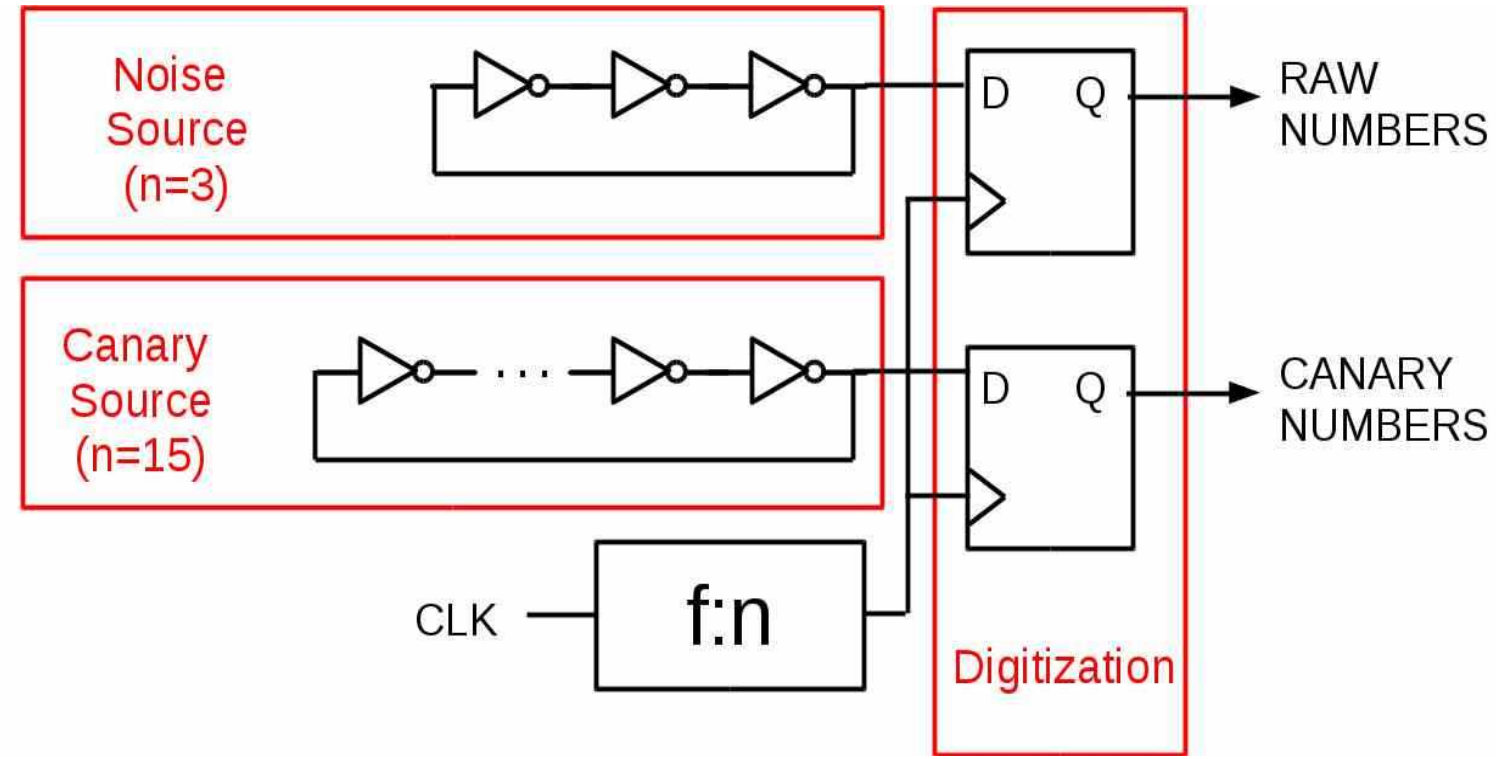
[2] M. Baudet et. al., On the Security of Oscillator-based Random Number Generators. Journal of Cryptology 24(2), 2011.

Critical parameter: jitter accumulation rate

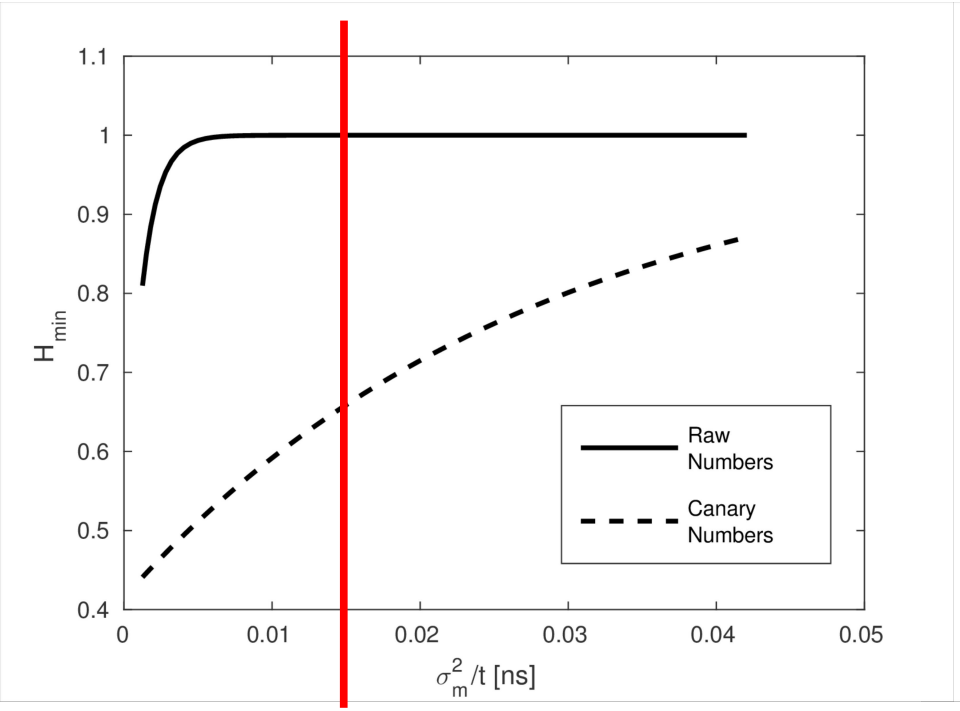
Replica-based architecture

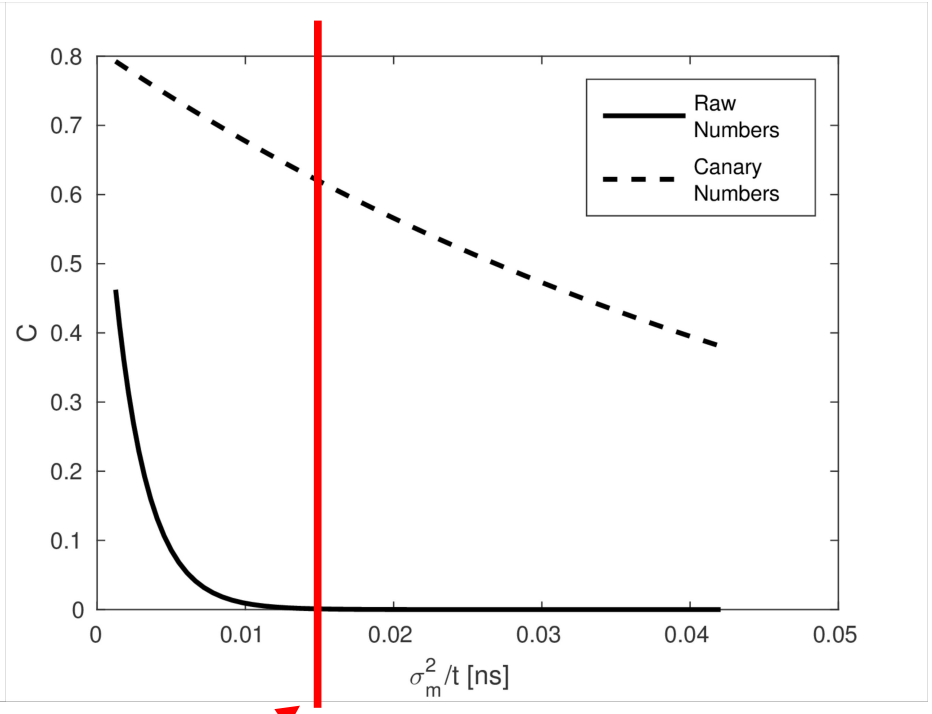
-RO length

Case Study 1: Elementary TRNG



Case Study 1: Elementary TRNG





Operating point

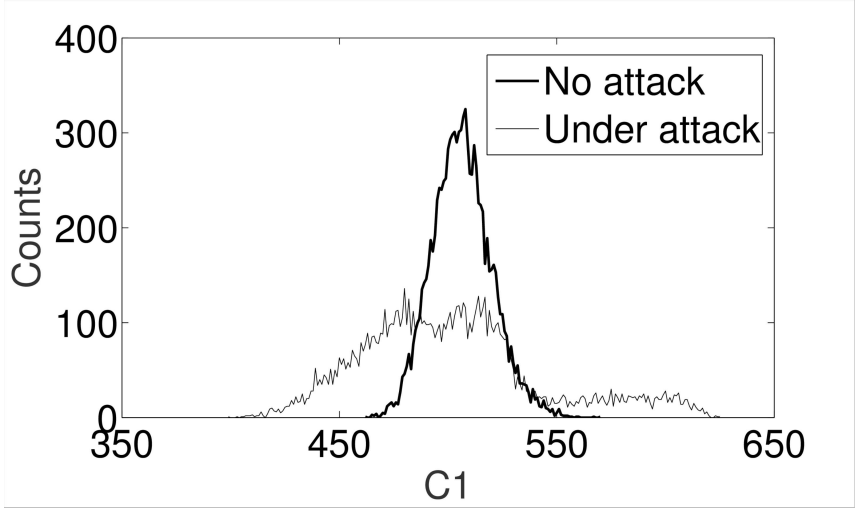
KU LEUVEN

Case Study 1: Elementary TRNG

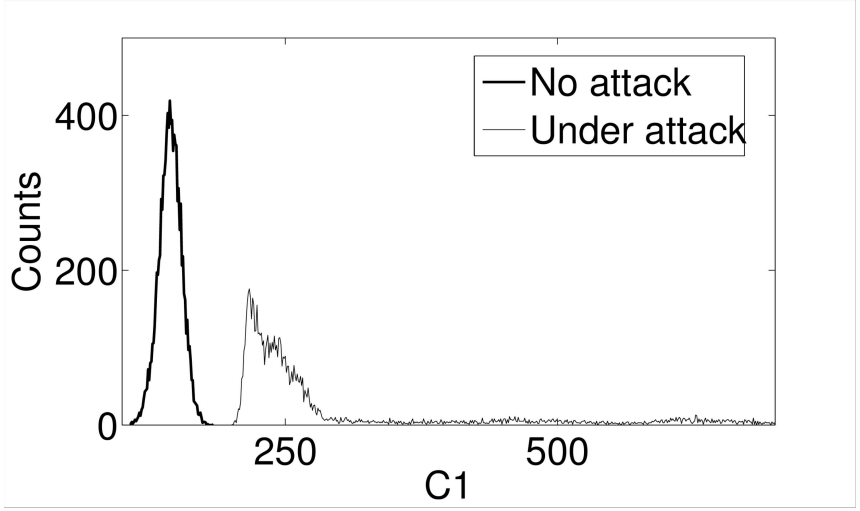
- EXPERIMENT:
  - Collect 10000 sequences of 1024b
  - Compute auto-correlation coefficients
  - Attack: FPGA cooled down using freezing spray
  - Compare Distributions

Case Study 1: Elementary TRNG

RAW NUMBERS

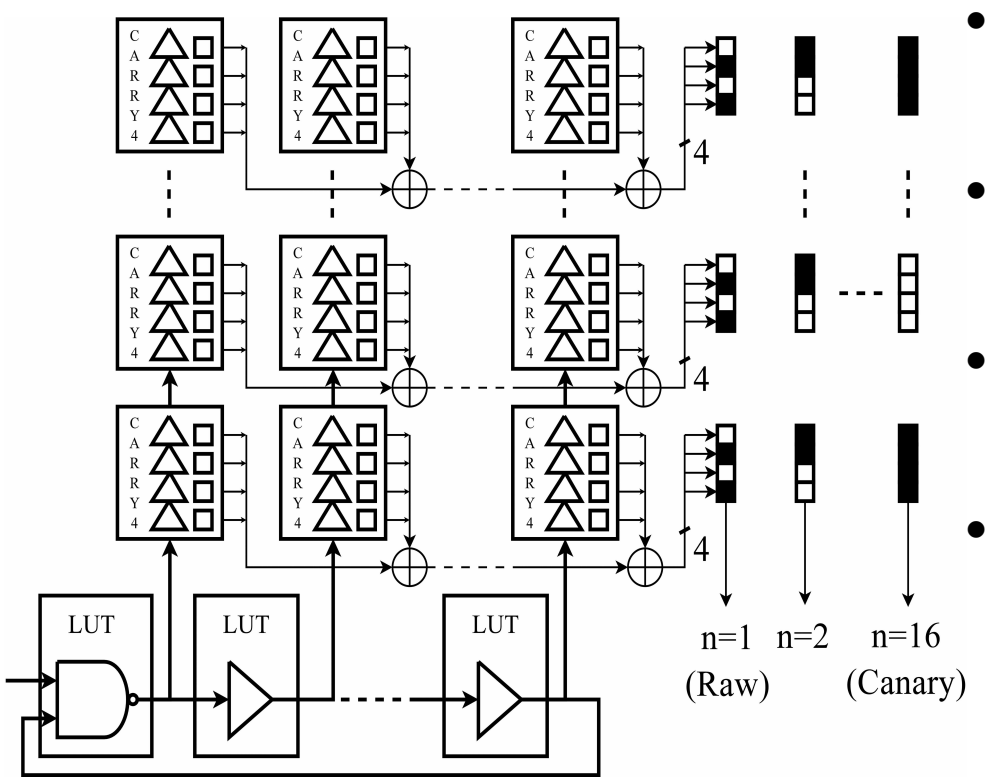


CANARY NUMBERS



KU LEUVEN

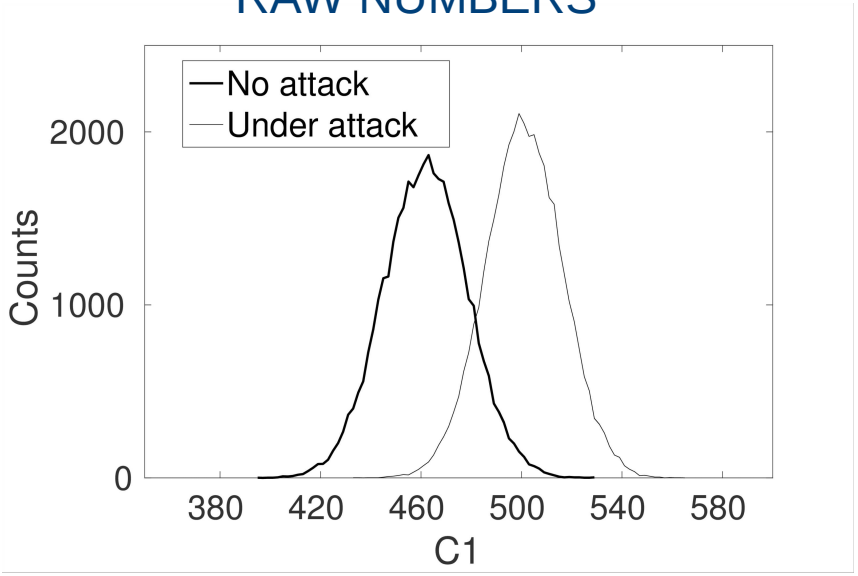
Case Study 2: Delay-chain TRNG



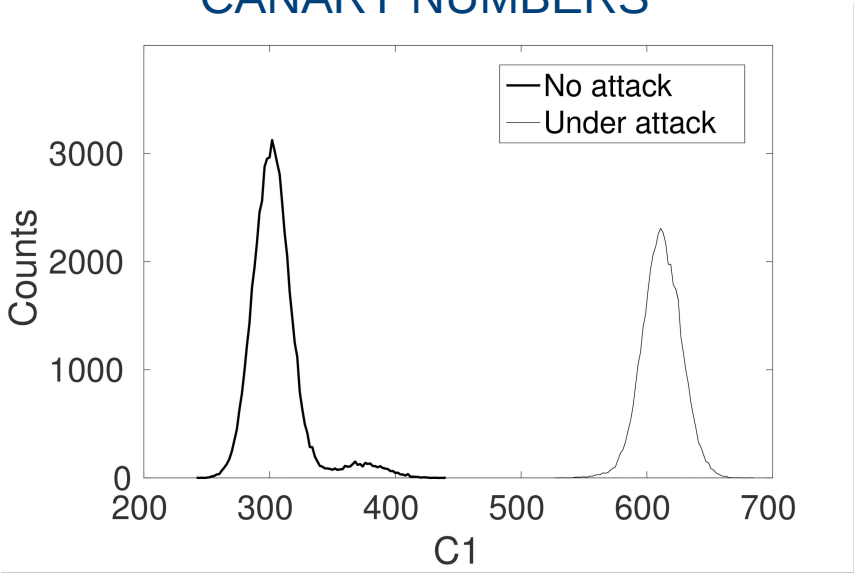
- Noise Source: Ring-oscillator
- Digitization: Tapped delay lines
- Post-processing: Priority encoder
- Canary extraction: Time-to-Digital Conversion with lower precision

Case Study 2: Delay-chain TRNG

RAW NUMBERS



CANARY NUMBERS

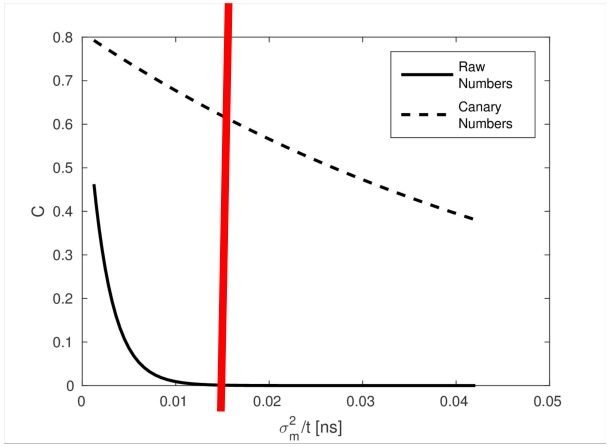


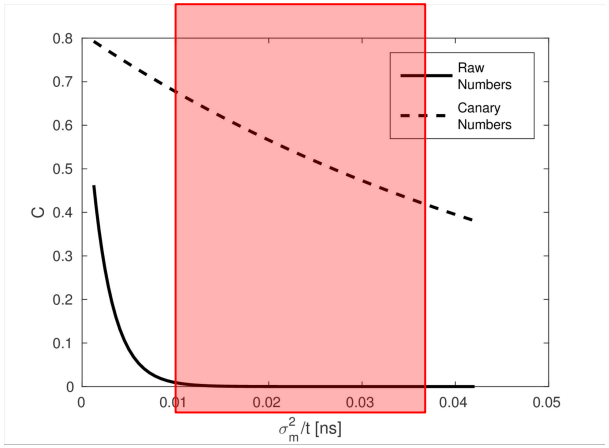
KU LEUVEN

Conclusions

- A promising testing strategy for some TRNGs
- Improved distinguish-ability for Elementary TRNG and Delay-chain TRNG
- 1024 bits per sequence is probably not enough

Future work





- Challenges:
  - From operating point to operating range
  - Exploring other TRNG designs

Questions?

KU LEUVEN