Circuits for True Random Number Generation with On-Line Quality Monitoring

Arnaud Tisserand

Circuits for True Random Number Generation with On-Line Quality Monitoring

Arnaud Tisserand

CNRS, IRISA laboratory, CAIRN research team

Claude Shannon Institute Workshop on Coding & Cryptography
May 23–24, 2011, UCC

Summary

- Motivations and context
- Randomness quality evaluation
- True random number generators (TRNGs)
- OCHRE circuits
- Conclusion, future prospects, references

The "Random Group" at CAIRN–IRISA

Researchers:

- Prof. Olivier Sentieys (ENSSAT–Univ. Rennes–INRIA)
- Dr. Arnaud Tisserand (CNRS)

PhD student:

- Dr. Renaud Santoro (2006–2009)

Engineers:

- Thomas Anger (2009–2010)
- Arnaud Carer (CAIRN)
- Philippe Quémerais (ENSSAT–Univ. Rennes)

Master student:

- Mohamed Habibi (2011)

Some Applications of Random Numbers

- **Lotteries, games and gambling**
- **Cryptography and security**:
  key generation, initialization vectors, padding, nonces, stream ciphers, masking, blinding, randomization, ...
- **Probabilistic / randomized algorithms**:
  Monte Carlo simulations, Las Vegas algorithms, ...
- **VLSI testing**:
  random patterns, random schedules, ...
- **Digital communications**:
  channel estimation/modeling, simulation, ...

Random Numbers: "Definition"

Simple definition (B. Schneier):

The output:

- looks random
- is unpredictable
- cannot be reproduced (for TRNGs)

Standard definition (e.g. see D. Knuth [4, chap. 3]):

The sequence of numbers $(x_1, x_2, x_3, \ldots, x_{n-1}, x_n)$, with $\forall i, x_i \in \mathcal{S}$, is *random* when the n numbers are:

- statistically independent
- uniformly distributed (*equally probable*)
- unpredictable

Chapter 3, *Random Numbers*, from [4] D. E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, 3rd edition, 1997

Randomness Quality Evaluation is Required

Extract from [8] (1988):

"*Many generators have been written, most of them have demonstrably non-random characteristics, and some are embarrassingly bad.*"

Randomness quality evaluation required at :

- design time (ensures minimal randomness for ideal environment)
- run time (check environment modifications, attacks)

Randomness quality evaluation methods:

- mathematical and physical models
- statistical tests

AND

[8] S. K. Park and K. W. Miller. Random number generators: Good ones are hard to find. *Communications of the ACM*, 31(10):1192–1201, October 1988

Statistical Tests for Randomness Evaluation

- FIPS 140-1 (1994): Security Requirements for Cryptographic Modules, *4 basic tests* (removed in 140-2 version, 2001/2002)
- AIS 31 (2001+): Functionality Classes and Evaluation Methodology for Physical Random Number Generators. *Specific tests for TRNGs*
- NIST 800-22 (2008+): A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. *More complete test suite*
- DIEHARD (1995): by G. Marsaglia, `http://www.stat.fsu.edu/pub/diehard/`
- DIEHARDER (2003+): *very complete test suite*, maintained by R. Brown, `http://www.phy.duke.edu/~rgb/General/dieharder.php`
- TestU01: C library from P. L'Ecuyer and R. Simard [6] (2007+)
- Universal test from U. Maurer [7] (1992)
- ...

FIPS 140-1 Statistical Tests

Input: sequence S of 20000 bits (from the RNG)
Output: PASS or FAIL

- Monobit: check that $9654 < \#1(S) < 10346$
- Poker: split S into 5000 4-bit blocs, $\#B_i(S)$ number of blocs equal to i ($(0000)_2$, $(0001)_2$, ..., $(1111)_2$), with $0 \leq i \leq 15$, check that

$$\chi^2_{15}: \quad 1.03 < \left(\frac{16}{5000} \times \sum_{i=0}^{15} \#B_i(S)^2 - 5000 \right) < 57.4$$

- Run: check that $\#run(k,S) \in I_k$ for $1 \leq k \leq 6$ and

k	1	2	3	4	5	6+
I_k	[2267, 2733]	[1079, 1421]	[502, 748]	[223, 402]	[90, 223]	[90, 223]

- Long run: check that $\forall k \geq 34, \#run(k,S) = 0$

Notation: $run(k,S)$ = string of k *consecutive* 0s/1s (i.e. $1\underbrace{0\ldots0}_{k}1$ or $0\underbrace{1\ldots1}_{k}0$)

Statistical Tests Examples and Limitations

FIPS 140-1 tests

20kb sequences

Results format →

X = P or F

monobit	poker	long run	run(0,1)	run(1,1)	run(0,2)	run(1,2)	run(0,3)	run(1,3)	run(0,4)	run(1,4)	run(0,5)	run(1,5)	run(0,6+)	run(1,6+)
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

generator	x_0 (*)	FIPS 140-1 test results
Mersenne twister	$\approx \forall x_0$	PPPPPPPPPPPPPPP
$x_n = 1583458089 x_{n-1} \bmod 2^{31} - 1$	many x_0	PPPPPPPPPPPPPPP
	19, 23233, ...	FPPPPPPPPPPPPPP
	137, 1117, ...	FFPPPPPPPPPPPPP
$x_n = 331 x_{n-1} \bmod 1021$	1, ...	FFPFFPPPPFPFPPF
$x_n = x_{n-1} + 1$	1, 2, ...	FFPFPPPPPPPPPFP
	999, ...	PFPPPPPPPPPPPPP

(*) some bad values for x_0

Types of Random Number Generators

Pseudo random number generator (PRNG):

- deterministic algorithms
- very high throughput and good statistical properties
- various algorithms → quality/throughput/cost tradeoffs

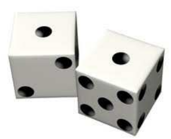
True random number generator (TRNG): THIS WORK

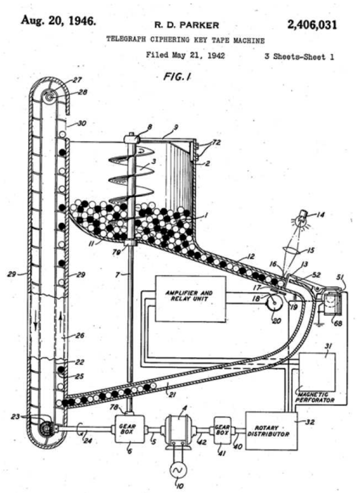
- non-deterministic algorithms (physical random source)
- limited throughput
- quality = *func*(environment parameters, ...) → attacks

Hybrid random number generator (HRNG):

- HRNG = TRNG + PRNG
- very high speed and very good quality
- selection needs more research

Historical Hardware TRNGs





ATT Patent 1946, source: P. Kohlbrenner

TRNGs Selection

Physical noise source:

- quantum physics
- radioactive decay
- atmospheric noise
- thermal/Johnson noise
- jitter in ring oscillator sampling
- meta-stability
- noises in circuits: 1/f, shot, popcorn, crosstalk, ...
- ...

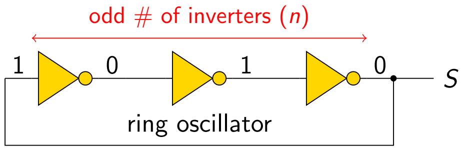
Characteristics:

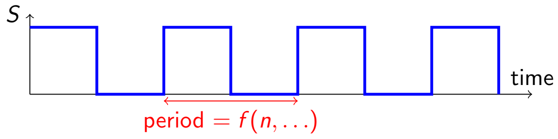
- throughput (? Mb/s)
- randomness quality (bias, entropy/bit, stability, effects of environment variations, ...)
- security → fully integrated in the chip
- cost (silicon area, power consumption)

Free Running Ring Oscillator

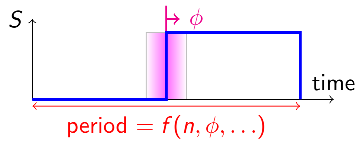
inverter:

in	out
0	1
1	0

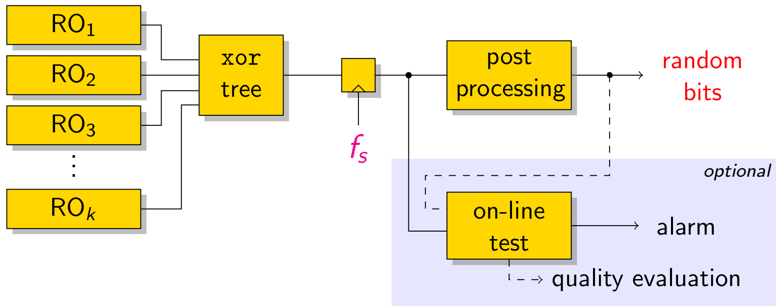




ϕ random jitter
(*timing/phase instability*)



Example of Ring Oscillator (RO) Based TRNG



Description:

- k free running ring oscillators
- f_s is the sampling frequency
- post processing: enhance statistical parameters
- on-line quality test (environment variations, attacks, ...)

Post Processing

Purpose: enhance statistical parameters of the output sequence

- reduce bias $Pr(x = 1) = 0.5 + \epsilon$ (AIS 31: $\epsilon < 0.0173$)
- increase entropy per bit (the real randomness)

Typical post processing methods:

- Von Neumann correction

input bits	(0,0)	(0,1)	(1,0)	(1,1)
output bit	*none*	1	0	*none*

- Linear feedback shift register (LFSR)
- Hash function (e.g. SHA)
- Ciphering (e.g. AES)
- Resilient function (e.g. error code computations)
- ...

Trade-off: entropy per bit, data rate, cost, quality

RO Based TRNG Example

[14] B. Sunar, W. J. Martin, and D. R. Stinson. A provably secure true random number generator with built-in tolerance to active attacks. *IEEE Transactions on Computers*, 56(1):109–119, January 2007

Description:

- $k = 114$ RO of 13 inverters
- resilient function: BCH(256, 13, 113) code
- mathematical model (but not realistic assumptions)
- data rate 2.5 Mb/s on FPGA

Problems:

- very complex calibration (external measurement of the jitter!!!)
- too many transitions in the `xor` tree
- setup/hold violations in the flip-flop
- ...

Other TRNG References

[5] P. Kohlbrenner and K. Gaj. An embedded true random number generator for FPGAs. In *Proc. Field Programmable Gate Arrays (FPGA)*, pages 71–78. ACM Press, February 2004

[3] V. Fischer, M. Drutarovsky, M. Simka, and N. Bochard. High performance true random number generator in altera stratix FPLDs. In *Proc. Field Programmable Logic and Applications (FPL)*, volume 3203 of *LNCS*, pages 555–564. Springer, August 2004

[13] D. Schellekens, B. Preneel, and I. Verbauwhede. FPGA vendor agnostic true random number generator. In *Proc. Field Programmable Logic and Applications (FPL)*, LNCS, pages 1–6. Springer, August 2006

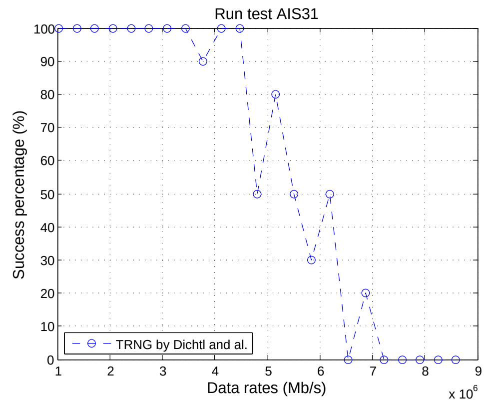
[2] M. Dichtl and J. D. Golic. High-speed true random number generation with logic gates only. In *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, volume 4727 of *LNCS*, pages 45–62. Springer, September 2007

[15] I. Vasyltsov, E. Hambardzumyan, Y.-S. Kim, and B. Karpinskyy. Fast digital TRNG based on metastable ring oscillator. In *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, volume 5154 of *LNCS*, pages 164–180. Springer, September 2008

[1] J.-L. Danger, S. Guilley, and P. Hoogvorst. High speed true random number generator based on open loop structures in FPGAs. *Microelectronics Journal*, 40(11):1650–1656, November 2009

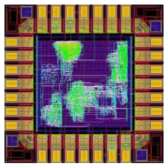
Example of Measurements on FPGAs

TRNG from [2] (Altera Stratix II):



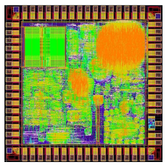
OCHRE Circuits (On-Chip Randomness Extraction)

OCHRE V1, 2009Q2 — 100 % OK



- 1 mm^2 9.04 mW (CMOS 130 nm 1.2 V STMicro)
- TRNG from [13] (110 RO with 3 inv.) → 195 MHz
- PRNG (cellular automaton) → 819 MHz
- FIPS 140-1 embedded statistical tests

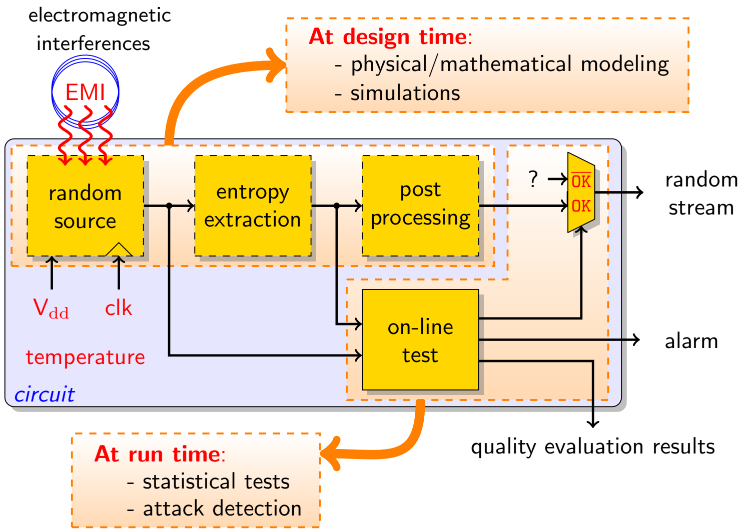
OCHRE V2, 2010Q4 — *under test*



- 4 mm^2 (CMOS 130 nm 1.2 V STMicro)
- TRNGs [14]/[15]/[2]/[16] → 235/628/746/363 MHz
- FIPS 140-1 and AIS 31 tests → 80 MHz
- AES → 241 MHz

We also have FPGA implementations (Xilinx, Altera and Actel).

TRNG Design and Use



Conclusion & Future Prospects

- TRNGs are important elements of security systems
- Randomness quality evaluation is very complex
  → accurate modeling, simulations, measurements, ...
- Embedded security systems → on-line quality evaluation
- We have implementations for both ASIC and FPGA targets
- Currently: very intensive measurements
  Environment parameters: V_{dd}, temperature, electromagnetic radiations, clock variations, ...

Future research topics:

- Hybrid generators = TRNG + PRNG
- Design space exploration at system level

References I

NIST references on RNG:
`http://csrc.nist.gov/groups/ST/toolkit/rng/index.html`

[1] J.-L. Danger, S. Guilley, and P. Hoogvorst.
High speed true random number generator based on open loop structures in FPGAs.
Microelectronics Journal, 40(11):1650–1656, November 2009.

[2] M. Dichtl and J. D. Golic.
High-speed true random number generation with logic gates only.
In *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, volume 4727 of *LNCS*, pages 45–62. Springer, September 2007.

[3] V. Fischer, M. Drutarovsky, M. Simka, and N. Bochard.
High performance true random number generator in altera stratix FPLDs.
In *Proc. Field Programmable Logic and Applications (FPL)*, volume 3203 of *LNCS*, pages 555–564. Springer, August 2004.

References II

[4] D. E. Knuth.
Seminumerical Algorithms, volume 2 of *The Art of Computer Programming*.
Addison-Wesley, 3rd edition, 1997.

[5] P. Kohlbrenner and K. Gaj.
An embedded true random number generator for FPGAs.
In *Proc. Field Programmable Gate Arrays (FPGA)*, pages 71–78. ACM Press, February 2004.

[6] P. L'Ecuyer and R. Simard.
TestU01: A C library for empirical testing of random number generators.
ACM Transactions on Mathematical Software, 33(4):22:1–40, August 2007.

[7] U. M. Maurer.
A universal statistical test for random bit generators.
Journal of Cryptology, 5(2):89–105, January 1992.

References III

[8] S. K. Park and K. W. Miller.
Random number generators: Good ones are hard to find.
Communications of the ACM, 31(10):1192–1201, October 1988.

[9] R. Santoro, S. Roy, and O. Sentieys.
Search for optimal five-neighbor FPGA-based cellular automata random number generators.
In *Proc. Int. Symp. Signals, Systems and Electronics (ISSSE)*, pages 343–346, Montreal, Canada, July 2007.

[10] R. Santoro, O. Sentieys, and S. Roy.
On-line monitoring of random number generators for embedded security.
In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 3050–3053, Taipei, Taiwan, May 2009.

[11] R. Santoro, O. Sentieys, and S. Roy.
On-the-fly evaluation of FPGA-based true random number generator.
In *Proc. Int. Symp. on VLSI (ISVLSI)*, pages 55–60. IEEE Computer Society, May 2009.

References IV

[12] R. Santoro, A. Tisserand, O. Sentieys, and S. Roy.
Arithmetic operators for on-the-fly evaluation of TRNGs.
In *Proc. Advanced Signal Processing Algorithms, Architectures and Implementations XVIII*, volume 7444, pages 1–12, San Diego, California, U.S.A., August 2009. SPIE.

[13] D. Schellekens, B. Preneel, and I. Verbauwhede.
FPGA vendor agnostic true random number generator.
In *Proc. Field Programmable Logic and Applications (FPL)*, LNCS, pages 1–6. Springer, August 2006.

[14] B. Sunar, W. J. Martin, and D. R. Stinson.
A provably secure true random number generator with built-in tolerance to active attacks.
IEEE Transactions on Computers, 56(1):109–119, January 2007.

[15] I. Vasyltsov, E. Hambardzumyan, Y.-S. Kim, and B. Karpinskyy.
Fast digital TRNG based on metastable ring oscillator.
In *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, volume 5154 of *LNCS*, pages 164–180. Springer, September 2008.

References V

[16] S.-K. Yoo, D. Karakoyunlu, B. Birand, and B. Sunar.
Improving the robustness of ring oscillator TRNGs.
ACM Transactions on Reconfigurable Technology and Systems, 3(2):9:1–30, May 2010.

The end, some questions ?

Contact:

- `mailto:arnaud.tisserand@irisa.fr`
- `http://www.irisa.fr/prive/Arnaud.Tisserand/`
- CAIRN Group `http://www.irisa.fr/cairn/`
- IRISA Laboratory, CNRS–INRIA–Univ. Rennes 1
  6 rue Kérampont, BP 80518, F-22305 Lannion cedex, France

Thank you