Bundesamt
für Sicherheit in der
Informationstechnik

Security Evaluation of Physical RNGs

Werner Schindler,
Workshop on Randomness and Arithmetics for Cryptography on Hardware
Roscoff, April 16, 2019

Overview

- Introduction and motivation
- Classification of random number generators (RNGs) and generic security requirements
- Evaluation criteria for physical RNGs
  - stochastic model
  - online test, total failure test, start-up test
- AIS 31 (and AIS 20)
  - history and main features
  - experiences and impact
- Conclusion

Introduction and motivation

Random numbers are needed almost everywhere …

- symmetric keys, IVs for block ciphers, session keys,
- challenges, nonces
- signature keys (RSA, ECDSA)
- ephemeral keys (ECDSA, DSA)
- protocols
- blinding and masking values (→ protection against side-channel attacks)
- zero-knowledge proofs
- ...

Well-known flaws (I)

Example 1:

- RNG in a Debian Linux Distribution (OpenSSL, 2008) [10]
- The RNG could only output 2^{15} different random numbers.
- Reason: Accidentally, a line of code had been commented out.

Well-known flaws (II)

Example 2:

- Flaw in Taiwan’s smart ID cards
- Bernstein et al. (2013) [BCC+13] were able to factorise 184 out of about two million 1024-bit RSA moduli from a public certificate database.
- <u>One reason</u> (amongst others): The prime
  0xc00000000000000000000000000000000000000000000000000000000000000000000000000
  000000000000000000000000000000000000000000000000002f9
  (= next prime after $2^{511} + 2^{510}$) occurred 46 times!
- Presumably, the two most significant bits were set ‘11’ to ensure 1024 bit moduli,
  and in all 46 cases 510 zeroes were generated in a row!
- Of course, it is actually impossible that this happens by chance.

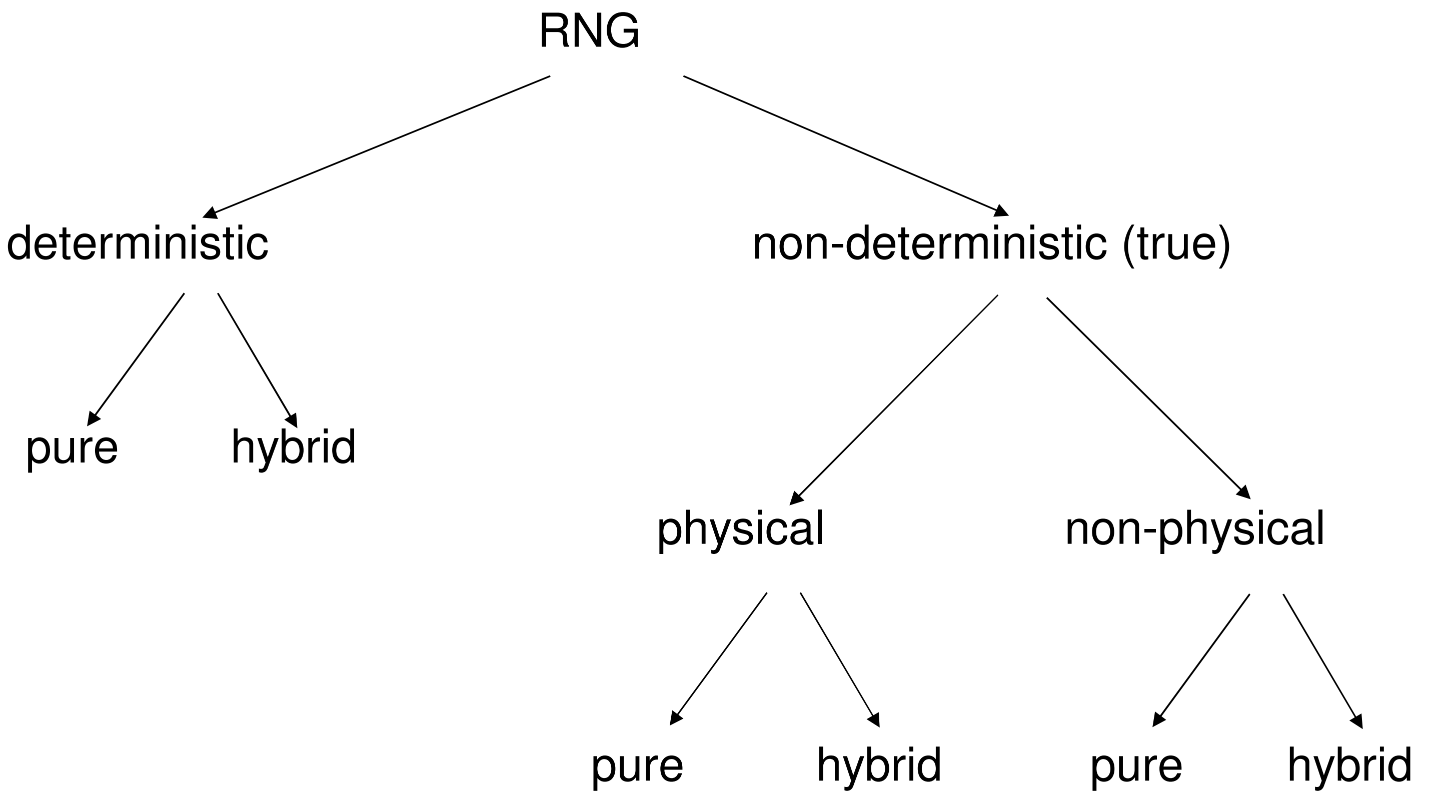
‚Natural‘ security requirements

- Which properties should random numbers have?
  - The random numbers should assume all admissible values with equal probability.
  - The assumed values should be independent from predecessors and successors.

- These requirements characterise an ideal random number generator (a mathematical construct!)
- An ideal RNG does not exist in the real world, and if one existed it would not be possible to verify the idealness.

The development of secure RNGs and their trustworthy evaluation are not trivial tasks.

Classification of real-world RNGs



RNG

deterministic

non-deterministic (true)

pure

hybrid

physical

non-physical

pure

hybrid

pure

hybrid

[8], Fig. 2.1

Remark

- deterministic RNGs (DRNGs) a.k.a pseudorandom number generators
- The output of pure DRNGs is completely determined by the seed value.
- Hybrid RNGs show design features from both deterministic RNGs and true RNGs.
- The core of a physical RNG (PTRNG) is the noise source (dedicated hardware).
- Non-physical true RNGs (NPTRNGs) exploit user's interaction and / or system data.

  Typically, NPTRNGs are implemented on PCs or servers.

  Example: /Linux /dev/random and /dev/urandom

Security requirements (I)

- R1: good statistical properties
- R2: backward secrecy and forward secrecy

  (The knowledge of sub-sequences of random numbers *shall not allow to practically compute* predecessors or successors or *to guess them with non-negligibly larger probability* than without knowledge of these sub-sequences.)

DRNGs: Verification of the security requirements

- R1: by statistical tests
- R2: A DRNG is usually composed of well-known cryptographic primitives. The security proof usually traces back to the properties of the primitives.

  Example (possible conclusions in security proofs):

  - Breaking the forward secrecy is as at least as hard as mounting a chosen-plaintext attack on the AES.
  - Breaking the backward secrecy is at least as hard as finding a pre-image of the SHA-256.
  - …

  Security proofs for DRNGs usually exploit well-known and established cryptographic results.

DRNGs: Additional security requirements (I)

- R3: enhanced backward secrecy

  (It shall not be *practically feasible* to compute preceding random numbers from the internal state or to guess them with non-negligibly larger probability than without knowledge of the internal state.)
- The enhanced backward secrecy protects previous random numbers even if the internal state of the DRNG has been compromised.

DRNGs: Additional security requirements (II)

- For particular applications a further security requirement may be desirable, too:
- R4: *enhanced forward secrecy*

  (It shall not be practically feasible to compute future random numbers from the internal state or to guess them with non-negligibly larger probability than without knowledge of the internal state.)

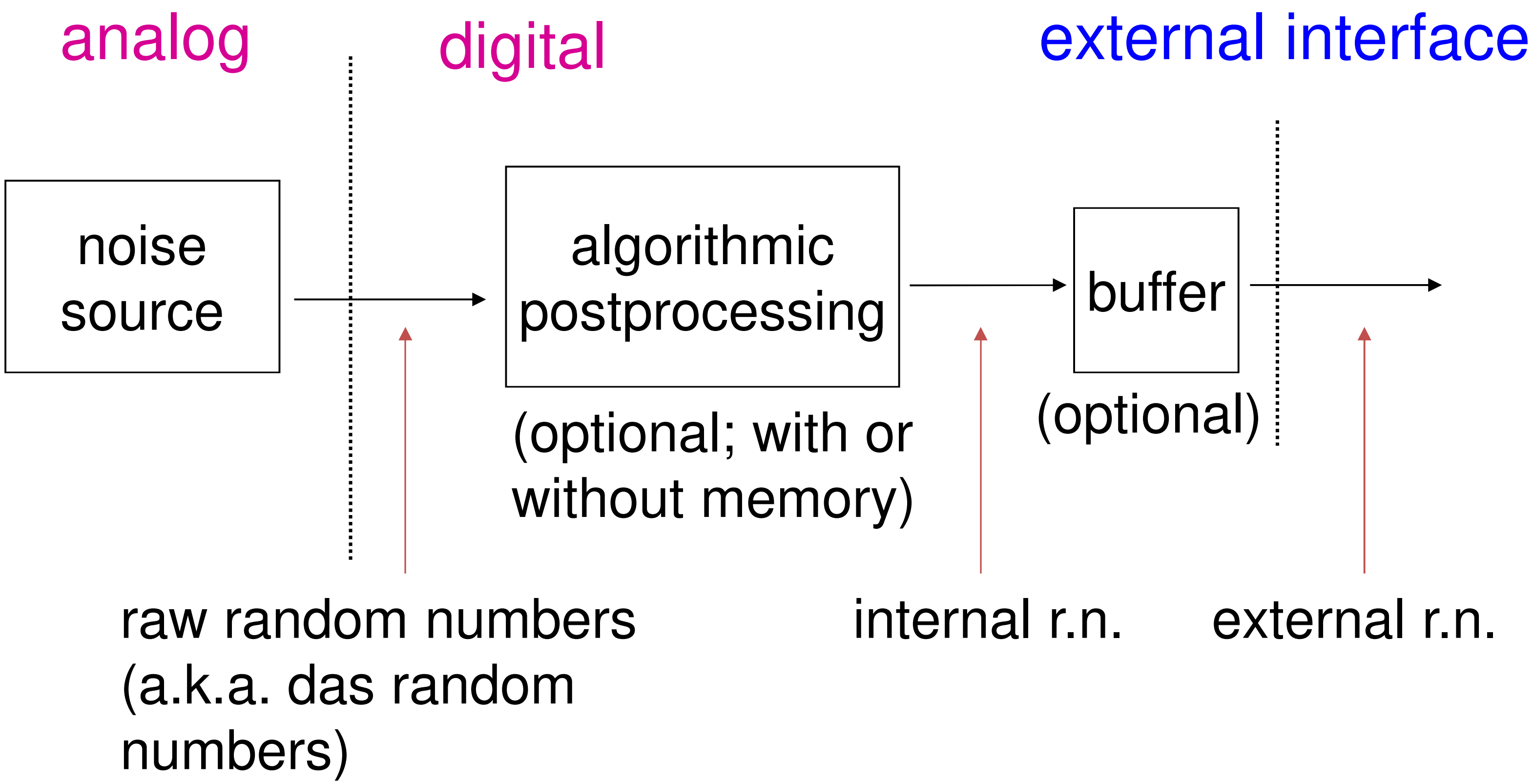
Note:

- Pure DRNGs cannot fulfil R4.
- Hybrid DRNGs *may* fulfil R4 after fresh entropy has been added.
- The security requirements R3 and R4 are DRNG-specific.
- For true RNGs R3 and R4 are usually ‘automatically’ guaranteed by R2.

Evaluation criteria for physical RNGs (PTRNGs)

Bundesamt
für Sicherheit in der
Informationstechnik

Physical RNG (schematic design)



analog

digital

external interface

noise source

algorithmic postprocessing

(optional; with or without memory)

buffer

(optional)

raw random numbers (a.k.a. das random numbers)

internal r.n.

external r.n.

[8], Fig. 2.4

Noise source

A noise source is a special type of entropy source that consists of dedicated hardware.

The noise source uses / exploits, for instance,

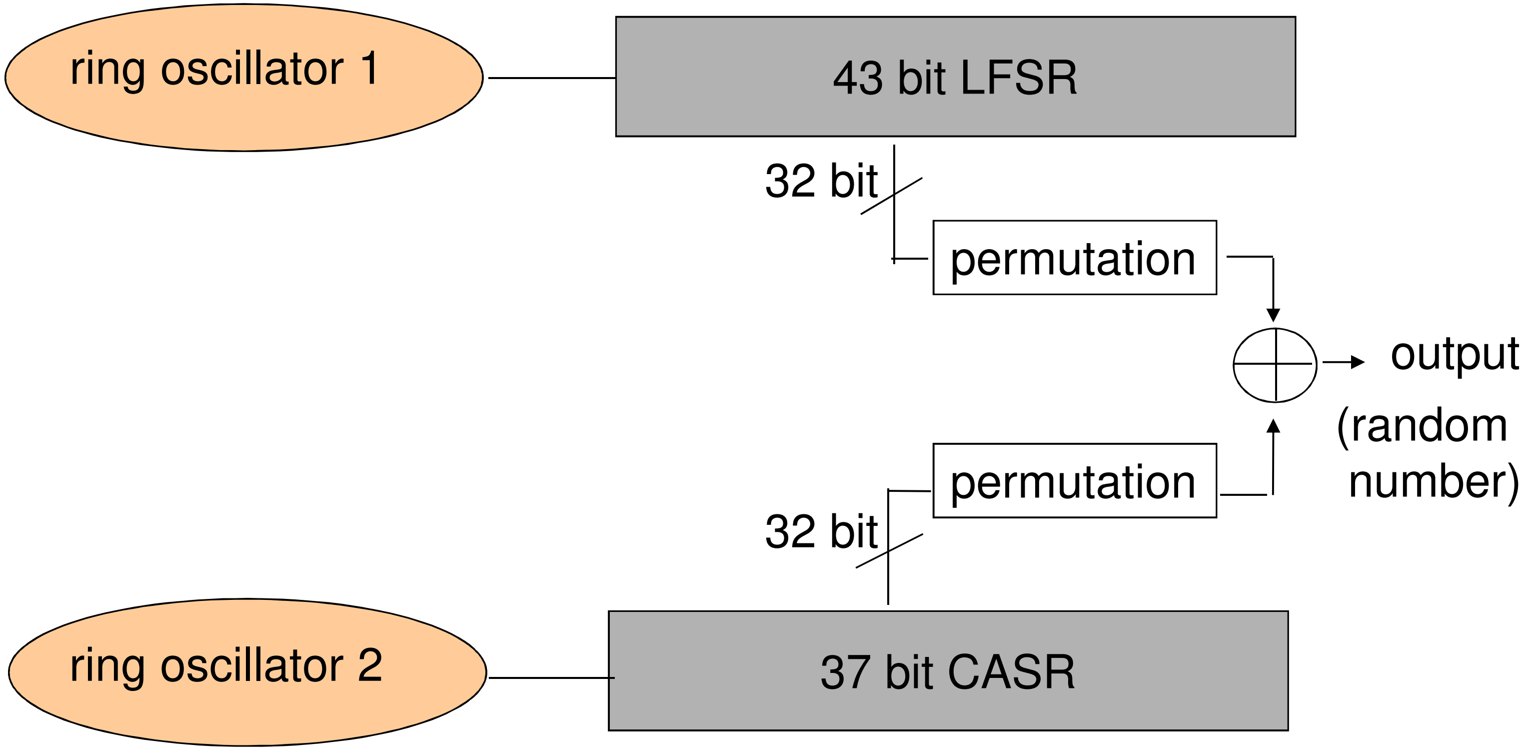
- noisy diodes
- free-running oscillators
- ring oscillators
- radioactive decay
- quantum photon effects
- …

Evaluation of the security requirements

- R1: The statistical properties of RNGs are checked by statistical tests.
- This is the easy part of the evaluation.
- Aren’t good statistical properties sufficient for true RNGs?
- The answer is no!

Bundesamt
für Sicherheit in der
Informationstechnik

Example 3: A PTRNG design presented at CHES 2002 [11]



CASR = Cellular Automaton Shift Register (GF(2)-linear)

[9], Fig. 3.2

Example 3 (II)

- The intermediate time between two outputs of random numbers should exceed a minimum number of LFSR and CASR cycles.
- The developers modified the design until a set of statistical tests had been passed [11].
- Dichtl (CHES 2003) [2] presented an attack (for the specified minimum intermediate time between consecutive random numbers *under the assumption that all design details would be known*).
- Good statistical properties are not enough!
- Schindler (Cryptography & Coding 2003) [7] derived lower and upper bounds for the entropy per bit (depending on the jitter of the ring oscillators).

Entropy (I)

Definition: Let X denote a random variable that assumes values in a finite set $S = \{s_1, \ldots, s_t\}$. The (Shannon) entropy of X is given by

$$\boxed{H(X) = \sum_{j=1}^{t} \mathrm{Prob}(X = s_j) \cdot \log_2 (\mathrm{Prob}(X=s_j))}$$

- $0 \leq H(X) \leq \log_2 | S |$
    - special case ($| S | = 2$): $0 \leq H(X) \leq 1$
- min entropy: $H_{min}(X) = \min \{- \log_2(\mathrm{Prob}(X=s_j)) \mid j = 1, \ldots, t\}$

Entropy (II)

- Entropy cannot be measured like temperature, voltage etc.
- Universal entropy estimators do not exist.
- Entropy is a property of random variables, not of random numbers.
- Model: In the following we assume that the random numbers are realisations (i.e. values that are taken on) by random variables X_1,X_2, ….
- Aim of a security evaluation: Verify a lower entropy bound per internal random bit.
- Attention! If one considers only the bias one gets an upper entropy bound because dependencies reduce the amount of entropy.

Stochastic model (I)

- Ideally, a stochastic model specifies a family of probability distributions that contains the true distribution of the internal random numbers.
- Alternatively, the stochastic model may specify a family of distributions that contain the distribution
  - of the raw random numbers or
  - of ‚auxiliary' random variables

  if this allows to estimate the (average) increase of entropy per internal random number.
- The specified family of probability distributions depends on one or on several parameters.

Example 4: Coin tossing (I)

- **PTRNG:** A single coin is tossed repeatedly.

  "Head" (H) is interpreted as 1, "tail" (T) as 0.
- **Stochastic model:**

  The observed sequence of random numbers (here: heads and tails) are interpreted as values that are assumed by random variables $X_1, X_2, \ldots$.
- The random variables $X_1, X_2, \ldots$ *are assumed to be independent and identically distributed.* (Justification: Coins have no memory.)
- $p := \mathrm{Prob}(X_j = H) \in [0,1]$ with unknown parameter p

Example 4: Coin tossing (II)

Entropy estimation (based on the stochastic model)

- Observe a sample $x_1, x_2, \ldots, x_N$

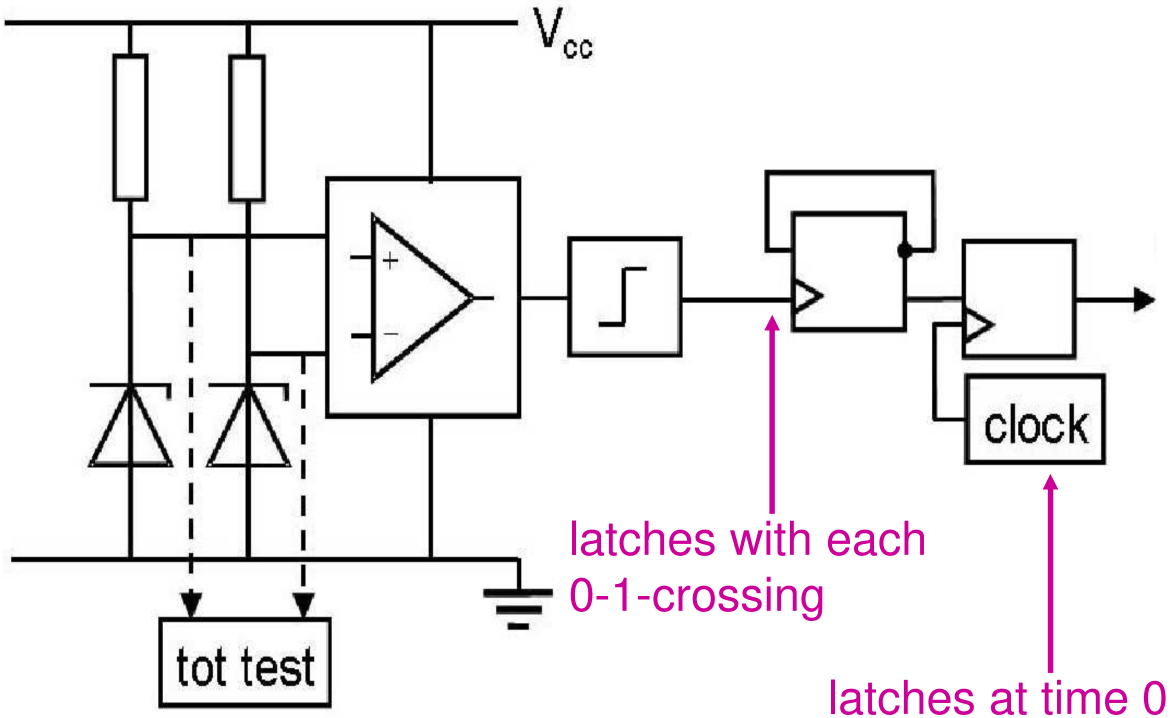
  Set $\tilde{p} := \#\{j \leq N \mid x_j = H\} / N$
- To obtain an estimate for the entropy $H(X_1)$ substitũte p into the entropy formula:

  $\tilde{H}(X_1) = -\,(\,\tilde{p} * \log_2(\tilde{p}) + (1-\tilde{p}) * \log_2(1-\tilde{p}))$

Stochastic model (II)

- For the coin tossing example the stochastic model depends on one parameter (on p).
- Its justification is much easier than for a *physical model*, which would consider the mass distribution etc.
- The ‘price’ is that the parameter p has to be estimated (easy task!)
- Moreover, for different coins the parameter p may vary to some degree. The stochastic model covers all coins.
- The parameter(s) are estimated first, and then an entropy estimate is computed (as in Example 4).
- For physical RNGs the justification of the stochastic model is usually more difficult and requires more sophisticated arguments. Ideally, it should be confirmed by experiments.

Example 5: Killmann & Schindler (CHES 2008) [4]



internal random numbers

$y_1, y_2, \ldots$

latches with each 0-1-crossing

latches at time 0,s,2s,…

c.f. [4], Fig. 1

Example 5: Stochastic model

- t_n: time between the (n-1)th and the nth upcrossing
- raw random numbers [only virtually]
  r_n := number of 0-1-crossings of the input voltage of the Schmitt trigger in time period ((n-1)s,ns]
- internal random numbers
  $y_n \equiv y_{n-1} + r_n \equiv y_0 + r_1 + \ldots + r_n \pmod 2$
- $T_1, T_2, \ldots$ is stationary (mild assumption) →

  $R_1, R_2, R_3 \ldots$ is stationary [raw random numbers] →

  $Y_1, Y_2, Y_3 \ldots$ is stationary [internal random numbers] →

  $W_1, W_2, W_3 \ldots$ is stationary [auxiliary random numbers]

→ 2-parameter family of distributions (depending on the expectation and the generalised variance)

Example 6: Haddad, Fischer, Bernard, Nicolai (CHES 2015) [3]

- Source of randomness: transient effect ring oscillator (TERO)
- Thorough analysis of the electric processes in the TERO structure

  → stochastic model of the TERO

  → stochastic model of the complete RNG
- Implementation of the RNG design (28 nm CMOS ASIC)

- There are several papers on the evaluation of PTRNGs on the basis of stochastic models in the literature.

PTRNG in operation: Security measures

test	**aim**
total failure test	shall detect a total breakdown of the noise source (almost) immediately; r.n.’s, which have been generated after that instant, shall not be output
start-up test	shall ensure the functionality of the physical RNG when it is started
online test	shall detect non-tolerable weaknesses of the random numbers sufficiently soon

Security evaluation (summary)

A trustworthy security evaluation should verify

- the suitability of the RNG design
- a lower entropy bound
- the effectiveness of the online test and the tot test
- the appropriateness of the specified consequences of a noise alarm

Note:

- The online test should be tailored to the stochastic model.
- The total failure test should be based on a sound failure analysis.

AIS 31 (and AIS 20)

‚History‘

- Until the end of the nineties the design of PTRNGs often seemed to follow the motto ‚security by obscurity‘.
- There was a lack of appropriate evaluation criteria.

Common Criteria

- provide evaluation criteria for IT products, which
  - shall permit the comparability between
  - independent security evaluations.
- A product or system that has successfully been evaluated is awarded with an internationally recognized (to particular assurance levels) IT security certificate.
- The Common Criteria and the corresponding evaluation manuals *do not* specify evaluation criteria for random number generators.

AIS 20 and AIS 31 (I)

- In the German evaluation and certification scheme the evaluation guidance documents

AIS 20: Functionality Classes and Evaluation Methodology for Deterministic Random Number Generators

AIS 31: Functionality Classes and Evaluation Methodology for Physical Random Number Generators

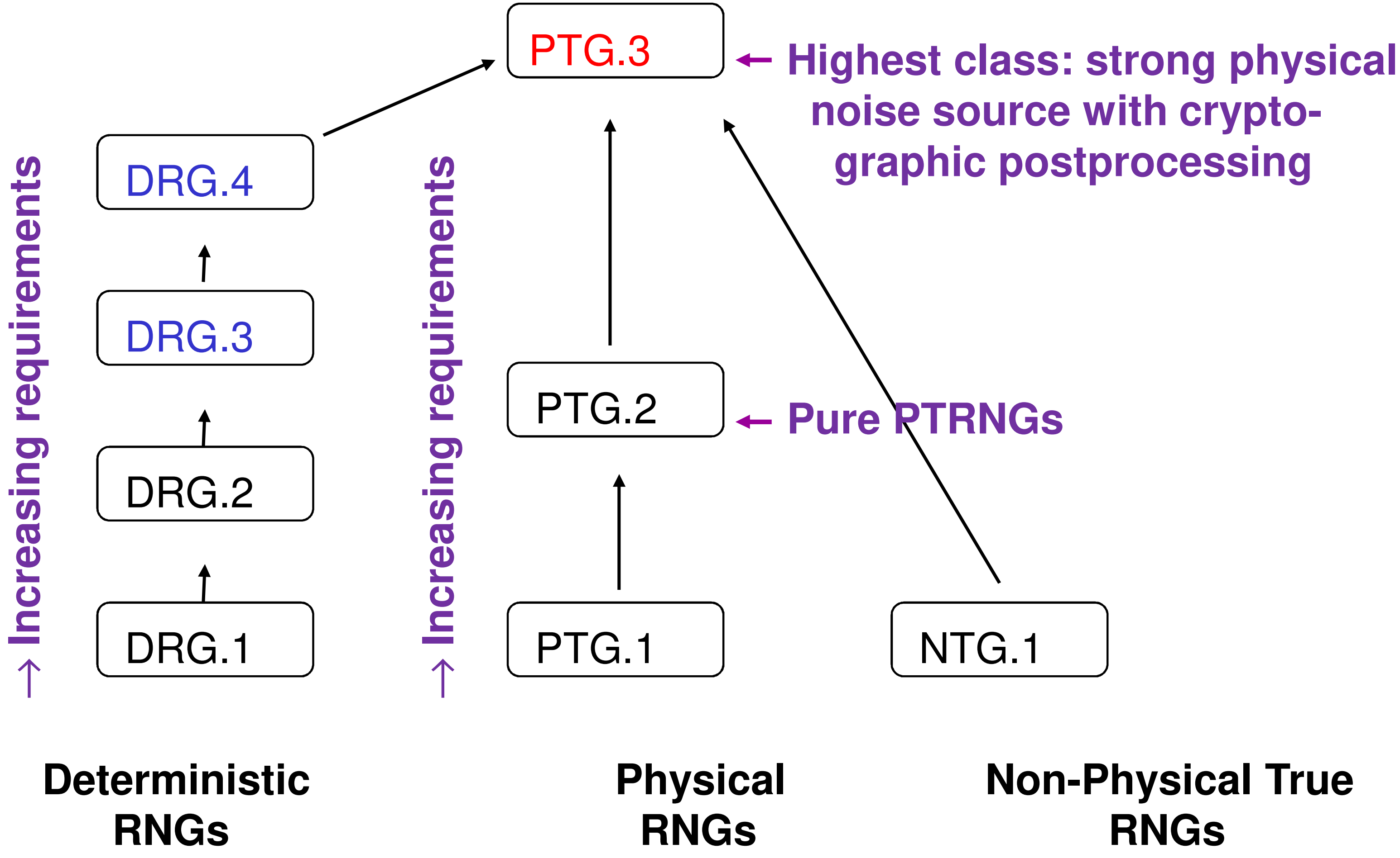
have been effective since 1999 and 2001, respectively.

- The mathematical-technical reference [5] has been updated in 2011 (in English → BSI website).

AIS 20 and AIS 31 (II)

- The AIS 20 and AIS 31 are technically neutral (no approved designs). Instead, several functionality classes are defined.
- The applicant has to give evidence that the RNG meets the requirements.
- PTRNGs are usually integrated in chips / smart cards.
- The AIS 20 and AIS 31 have been well-tried in practice.
- Companies from several countries have received certificates, which confirm that their PTRNGs are conformant to particular functionality classes (AIS 20 and AIS 31).

AIS 20 / AIS 31: Functionality classes



PTG.3

← **Highest class: strong physical noise source with crypto-graphic postprocessing**

DRG.4

DRG.3

DRG.2

DRG.1

PTG.2

← **Pure PTRNGs**

PTG.1

NTG.1

↑ Increasing requirements

↑ Increasing requirements

Deterministic RNGs

Physical RNGs

Non-Physical True RNGs

Functionality classes DRG.2 & DRG.3

DRG.2:

- high seed entropy
- good statistical properties
- backward secrecy and forward secrecy

DRG.3:

- DRG.2-conformance
- + enhanced backward secrecy

Note: The functionality class DRG.3 ensures the privacy of old random numbers even if the internal state has been compromised. This demands a one-way state transition function, which may be costly (e.g. for smart cards).

Functionality class PTG.2

- The internal random numbers may have a small entropy defect (due to bias, dependencies).
- effective online tests, total failure tests, start-up tests

Note:

- PTG.2- conformant RNGs are suitable to seed DRNGs.
- The entropy defect is of little importance for the generation of symmetric keys, challenges etc.
- In general, the BSI prefers DRG.4 or (even better) PTG.3-conformant RNGs.

Functionality classes DRG.4 [hybrid DRNGs]

- DRG.3-conformant RNG
- A strong PTRNG (typically, a PTG.2-conformant) reseeds / updates the internal state (upon request, event-driven, after k random numbers).

Functionality class PTG.3

- Internal random numbers from a PTG.2-compliant RNG are post-processed by a DRG.3-compliant RNG with memory.
- The post-processing algorithm must not ‚extend‘ the input data.

Note:

- PTG.3 is the highest class as it combines a strong physical noise source with a cryptographic post-processing algorithm.
- Moreover, PTG.3-conformant RNGs should provide better protection against implementation attacks (side-channel attacks, fault attacks) than other functionality classes.

Functionality class NTG.1

- good statistical properties
- high entropy per bit

Note:

- Problem / disadvantage: The platform is not under the control of the RNG designer. For very sensitive applications the BSI recommends RNGs from the functionality classes PTG.3 and DRG.4.

Note:

- An analysis of /dev/random (and /dev/urandom) for the Linux kernels from the last years can be found on the BSI website [6].

AIS 31: Influence

- Over the years the AIS 31 has certainly influenced the design of physical RNGs.
- The AIS 31 has had significant influence on scientific research. Many papers and PhD theses considered physical RNG and their conformance to the AIS 31 by analysing stochastic models.
- In particular, Viktor Fischer (University of Saint Etienne) and his research group has performed a lot of research work in this field.
- The AIS 31 is also applied in the French certification scheme.

  Certificates, which confirm the PTG.2-conformance, have mutually been recognized between the BSI and ANSSI since 2015.

Other national and international standards (small selection) (I)

- NIST SP 800-22
  - discusses 15 statistical tests and testing strategies
  - describes 9 DRNGs
  - focuses on statistical testing, cannot serve as a substitute for a solid security analysis (pointed out there)

Other national and international standards (small selection) (II)

- NIST SP 800-90 B (entropy sources):
  - Compared to its draft the standard has noticeably moved towards the AIS 31.
  - The applicant has to justify his entropy claim (*may* be done by a stochastic model).
  - specifies several predictors and statistical tests
  - further standards: NIST SP 800-90 A (DRNGs) and NIST SP 800-90 C (compositions of true and deterministic RNGs)

Other national and international standards (small selection) (III)

- ISO 20543 (upcoming, standard should appear soon):

  ‘Standard Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC15408’
- status: DIS
  - treats PTNGs, NPTRNGs, DRNGs
  - PTRNGs: a stochastic model is mandatory
  - health tests and total failure tests
  - distinguishes between PTRNGs and NPTRNGs

Conclusion

- RNGs are important components of cryptographic implementations.
- Design and evaluation of RNGs are not easy tasks.
- The AIS 31 has introduced a new evaluation methodology for physical RNGs (‘white-box analysis’ based on a stochastic model).
- The AIS 31 has had influence on scientific research, on the design of physical RNGs and on other RNG standards.

List of references (I)

[1] D. Bernstein, Y.-A. Chang, C.-M. Cheng, L.-P. Chou, N. Heninger, T. Lange, N. van Someren: Factoring RSA Keys from Certified Smartcards: Coppersmith in the Wild. In: K. Sako, P. Sarkar (eds.): Asiacrypt 2013, Part II, Springer, LNCS, Vol. 8270, Berlin 2013, 341-360.

[2] M. Dichtl: How to Predict the Output of a Hardware Random Number Generator. In: C.D. Walter, C.K. Koc, C. Paar (eds.): Cryptography and Embedded Systems - CHES 2003, Springer, LNCS, Vol. 2779, Berlin 2003, 181-188.

[3] P. Haddad, V. Fischer, F. Bernard, J. Nicolai: A Physical Approach for Stochastic Modeling of TERO-Based TRNG. In: T. Güneysu, H. Handschuh (eds.): Cryptographic Hardware and Embedded Systems – CHES 2015, Springer, LNCS, Vol. 9293, Berlin 2015, 357-372.

[4] W. Killmann, W. Schindler: Security Analysis of a Particular Design for a Physical Random Number Generator. In: E. Oswald, P. Rohatgi (eds.): Cryptographic Hardware and Embedded Systems - CHES 2008, Springer, LNCS, Vol. 5154, Berlin 2008, 146-163.

[5] W. Killmann, W. Schindler: A Proposal for: Functionality Classes for Random Number Generators. Version 2.0 (18.09.2011), mathematical-technical reference of the AIS 20 and AIS 31 (can be downloaded from the BSI website).

List of references (II)

[6] S. Müller: Documentation and Analysis of the Linux Random Number Generator. Report, produced by atsec information security GmbH , by order of Bundesamt für Sicherheit in der Informationstechnik (BSI), last updated: January 2019 (can be downloaded from the BSI website).

[7] W. Schindler: A Stochastical Model and Its Analysis for a Physical Random Number Generator Presented at CHES 2002. In: K.G. Paterson (ed.): Cryptography and Coding - IMA 2003, Springer, LNCS, Vol. 2898, Berlin 2003, 276-289.

[8] W. Schindler: Random Number Generators for Cryptographic Applications. In: C.K. Koc (ed.): Cryptographic Engineering, Springer, Berlin 2009, 5 - 23.

[9] W. Schindler: Evaluation Criteria for Physical Random Number Generators. In: C.K. Koc (ed.): Cryptographic Engineering, Springer, Berlin 2009, 25 - 54.

[10] B. Schneier: Random Number Bug in Debian Linux (May 19, 2008) http://www.schneier.com/blog/archives/2008/05/random_number_b.html

[11] T. Tkacik: A Hardware Random Number Generator. In: B.S. Kaliski Jr., C.K. Koç, C. Paar (eds.): Cryptographic Hardware and Embedded Systems - CHES 2002, Springer, LNCS, Vol. 2523, Berlin 2003, 450-453.

Thank you for your kind attention!

Kontakt

Werner Schindler
Referatsleiter (head of section)
Prüfung von Kryptoverfahren

Werner.Schindler@bsi.bund.de
Tel. +49 (0) 228 9582 5652
Fax +49 (0) 228 10 9582 5652
Bundesamt für Sicherheit in der Informationstechnik (BSI)
Godesberger Allee 185-189
53175 Bonn
www.bsi.bund.de
www.bsi-fuer-buerger.de