Bundesamt
für Sicherheit in der
Informationstechnik

Tailored RNGs for Low-Cost Devices (General Considerations)

Werner Schindler

Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn, Germany

La Grande Motte, June 23, 2016

Outline

- Introduction and motivation
- RNGs for low-cost devices (impact on the evaluation and the design)
  - Physical RNGs
  - Deterministic RNGs
- Conclusions

RNGs in real world devices

- High-end smart cards and general purpose hardware (PC, server etc.) normally use RNGs, which allow the broadest possible range of applications.
- Usually,
  - smart cards use physical RNGs (**PTRNGs**) or deterministic RNG (**DRNGs**)
  - PCs, server etc. use non-physical non-deterministic RNGs (**NTRNGs**)
    Example: /dev/random (Linux)

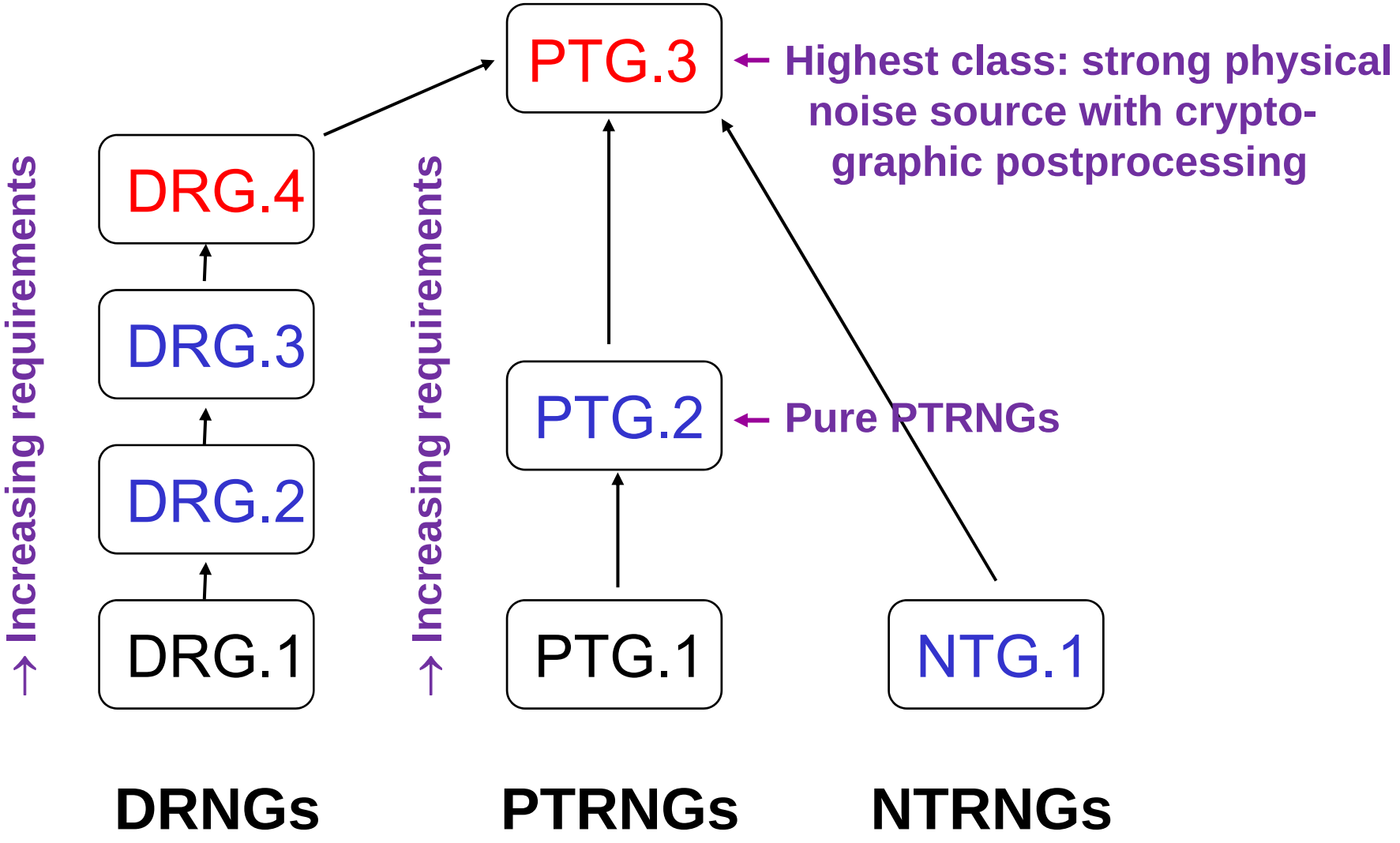
AIS 20 and AIS 31

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, resp. since 2001 (mathematical-technical reference updated in 2011)

Functionality classes



General purpose RNGs

- These functionality classes were designed for general purpose RNGs.
- The AIS 20 and AIS 31 explain how RNGs shall be evaluated. All problems solved?

RNGs for low-cost devices

- Principally, allowing a broad range of possible applications is a positive feature.
- However, general purpose RNGs may require substantial resources (memory, energy, size).
- Low-cost devices apply resource-saving lightweight (or even ultra-lightweight) cryptographic algorithms, which are tailored to the application(s).

RNGs for low-cost devices (II)

- It seems to be natural to tailor RNGs on low-cost devices to the needs of the applications, following the principles of lightweight cryptography.
- What does this mean for the design and the security evaluation of RNGs?
- The lightweight implementations are very different in terms of resources and security requirements.

In the following we will not provide ready-to-use solutions but basic considerations, which might serve as a basis for discussions / for appropriate designs.

Security evaluation of PTRNGs

- Primary Goal: Estimate the entropy per random bit
- Entropy is a property of random variables but not of random numbers. General entropy estimators do not exist.
- Main task: Develop, verify and analyse a stochastic model ($\rightarrow$ entropy estimate)

Security evaluation of PTRNGs (II)

A trustworthy security evaluation should verify the suitability of

- the RNG design
- the online test, the tot test and the startup test.

The online test should be tailored to the stochastic model.

The tot test (total failure test) shall consider all realistic scenarios of total failures.

PTRNGs in low-cost devices

- The stochastic model (development, verification, analysis) is the most complex part of a security evaluation. Can the tasks be simplified / be reduced for low-cost devices?
- **The answer is: no!**
- Reason: The keys of lightweight algorithms are shorter than for 'normal' cryptographic algorithms but the entropy per key bit should not be smaller.

The lower bound for the entropy per random bit cannot be scaled down in a natural way.

PTRNGs in low-cost devices (II)

- Can the tot test, the self test or the online test be dropped?
  - Principally yes, but only at the risk of the unnoticed use of weak random numbers!
  - It depends on the application and on the assumed attack potential whether this is a reasonable option.
- Secure alternative: At the cost of the output rate more resource-efficient PTRNG designs might be utilized if the application permits.

Security evaluations of DRNGs

- The state transition function and the output function are usually composed of cryptographic primitives.
- A security evaluation of a DRNG shall verify
  - that the seed entropy is sufficiently large
  - that the random numbers have appropriate statistical properties.
  - which of the following security properties are fulfilled
    - forward secrecy
    - backward secrecy
    - enhanced backward secrecy

Security evaluations of DRNGs (II)

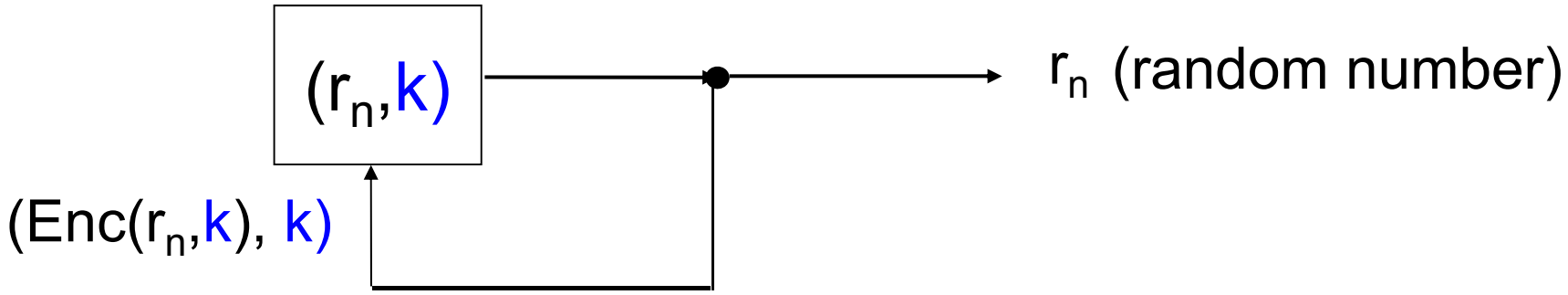
- The security properties of the DRNG
  - ‘forward secrecy’,
  - ‘backward secrecy’
  - and ‘enhanced backward secrecy’

  are traced back to the security properties of the cryptographic primitives.

Example 1

Enc: block cipher (AES, Triple-DES etc.)

k: key (to be kept secret)



internal state: $s_n = (r_n,k)$

$$s_{n+1} = (\mathrm{Enc}\,(r_n, k), k) =: (r_{n+1},k)$$

Example 1 (II)

Assumption: The attacker knows $r_i, r_{i+1}, \ldots, r_{i+j}$

- Task: Find $r_{i+j+1} = \mathrm{Enc}(r_{i+j}, k)$:

  Note that $r_{i+1} = \mathrm{Enc}(r_i, k)$, …, $r_{i+j} = \mathrm{Enc}(r_{i+j-1}, k)$

  → specific chosen-plaintext attack on $\mathrm{Enc}(\cdot, k)$

  (for AES, for instance) → forward secrecy

- Task: Find $r_{i-1} = \mathrm{Dec}(r_i, k) = \mathrm{Enc}^{-1}(r_i, k)$:

  Note that $r_{i+j-1} = \mathrm{Dec}(r_{i+j}, k)$, …, $r_i = \mathrm{Dec}(r_{i+1}, k)$

  → specific chosen-plaintext attack on $\mathrm{Dec}(\cdot, k)$

  (for AES, for instance) → backward secrecy

This security proof is typical for DRNGs.

DRNGs for low-cost devices

- Natural approach:
  Take a general purpose DRNG and replace the cryptographic primitives by corresponding lightweight primitives (e.g., AES by Present).
- This may reduce the security level of the DRNG.
  However, this should not be critical if the security level of the DRNG is still $\geq$ the security level of the lightweight cipher, which uses the random numbers.

DRNGs for low-cost devices (II)

- This approach definitely saves resources without affecting the security level of the lightweight cipher.
- However, the ‘reduced’ design may still be (too) costly. Are further savings possible?

Enhanced backward secrecy

- Enhanced backward secrecy guarantees the secrecy of prior random numbers even if the internal state has been compromised (e.g. by a hardware attack).
- One-way state transitions functions ensure enhanced backward secrecy.
- The implementation of a one-way function may be too costly for a low-cost device.

Which security properties are relevant ?

- The designer should analyse carefully, which of these security properties (forward secrecy, backward secrecy, enhanced backward secrecy) are actually needed by the application (← threat model).
- Moreover, the designer has to determine a suitable threshold for the seed entropy.
- On the basis of this analysis the designer may try to develop a resource-saving DRNG design, which ensures the needed security properties.

Seeding the DRNG

- If the low-cost device has enough permanent memory the DRNG may be seeded within a personalisation process.
- If not, the DRNG has to be seeded after power-up or before usage. Then the device needs a PTRNG.
  - The PTRNG need not be PTG.2-conformant but the seeding process (= special purpose application) shall guarantee enough seed entropy.

Implementation attacks

- Whether implementation attacks on the RNG (side-channel attacks, fault attacks etc.) need to be considered depends on the application and on the threat model.

Conclusion

- Low-cost devices apply lightweight cryptographic algorithms.
- It is a natural question whether the lightweight principle can be transferred to RNGs, too.
- The evaluation methodology for RNGs is not easier for low-cost devices.
- For PTRNGs the entropy requirements on the random numbers can not be relaxed but resources might be saved at the cost of performance.
- DRNGs may be scaled down in a natural way.

Bundesamt
für Sicherheit in der
Informationstechnik

Contact

Bundesamt für Sicherheit in der Informationstechnik (BSI)

Werner Schindler
Godesberger Allee 185-189
53175 Bonn, Germany

Tel: +49 (0)228-9582-5652
Fax: +49 (0)228-10-9582-5652

Werner.Schindler@bsi.bund.de
www.bsi.bund.de
www.bsi-fuer-buerger.de