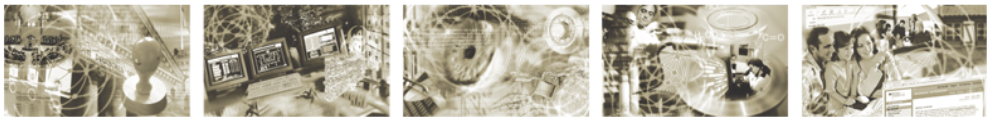
Bundesamt
für Sicherheit in der
Informationstechnik



Evaluation of random number generators

Version 0.10

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582-111
E-Mail: zertifizierung@bsi.bund.de
Internet: https://www.bsi.bund.de

Table of content

Tables

1 Introduction

Random Number Generators (RNGs) are incorporated into many products and play an important role in numerous cryptographic applications. However, the Information Technology Security Evaluation Criteria [ITSEC] and the Common Criteria ([CCV31_1], [CCV31_2], [CCV31_3]) do not specify any uniform evaluation criteria for RNG, nor do their corresponding evaluation methodologies (Information Technology Security Evaluation Manual [ITSEM]) and Common Evaluation Methodology [CEM]) specify such criteria.

As an increasing number of Common Criteria evaluation procedures in the German scheme include the assessment of security functionality relying on the secure generation of random numbers, the German certification body decided to update the methodology for the evaluation of random number generators.

In contrast to the former two documents [AIS20An] and [AIS31An], the new methodology is split into several documents. This document is the master document referencing the current versions of all documents representing the updated methodology in chapter 4 and in the bibliography. Furthermore, it contains general information and regulations that apply to all types of RNGs.

The terms “shall”, “should” and “may” if printed in bold and italic, are used as in the CC and the [CEM]. The word “shall” is used where the respective statement results directly from corresponding requirements in the CC or the [CEM]. The word “should” expresses a recommendation to the developer and the evaluator, as appropriate.

2 Specification of the generation of random numbers in STs and PPs

This chapter describes what has to be taken into account for the specification of random number generators in Security Targets and Protection Profiles, as well as the evaluation of those.

If the ST does not describe any SFR for random number generation, the evaluator shall identify the internal usage of the RNG by other security mechanisms and the resulting functional requirements for the RNG necessary to ensure the security of these mechanisms. If the RNG fails to meet these functional requirements, then the security mechanism using the random number will fail and as a result, the SFR implemented by this security mechanism will be violated.

2.1 Describing random number generation in STs and PPs

To harmonize the specification of random number generators within STs and PPs that are certified in the German scheme, an author of such a document *should* take the following information into account.

The document [KS2011] defines in chapter 3 the family FCS_RNG - Generation of random numbers. The German certification scheme requires that this SFR is used in combination with the predefined RNG-classes mentioned in chapter 4 of [KS2011] to express the security functionality of generating random numbers within an ST.

If random number generation is provided as a security service to the user, the ST/PP *should* describe the SFR for this security service by means of the extended component FCS_RNG.1 and one of the pre-defined RNG-classes. There might be rare cases where this approach is not applicable but this is a decision on a case-by-case basis for every certification process. In this case the certification body *must/shall* be involved.

If the random number generation is used for internal security mechanisms, the ST *may* describe the SFR for random number generation by means of FCS_RNG as well. If chosen, this SFR *should* describe the quality of the random numbers, the intended purpose or the internal use of the random numbers and the evaluator will use this information as criteria for the vulnerability analysis.

If the author of a ST or PP does not include FCS_RNG.1 into the document because random number generation is used for internal purposes only, the RNG will nevertheless be assessed as if it was described as a security functionality in the ST. The evaluation should assure that security mechanisms relying on "good" random numbers will work as expected and no security objective for the TOE is violated. To avoid that a product fails the evaluation or the evaluation takes more time than expected because the RNG has to be evaluated although it was not mentioned in the ST or PP, it is strongly recommended to express the generation of random numbers within the ST or PP.

2.2 Evaluation of PPs and STs with respect to random number generators

The following sections describe aspects specifically related to the evaluation activities for the classes APE and ASE if the generation of random numbers is expressed as a security functionality within the Security Target.

2.2.1 TOE definition (APE_INT, ASE_INT)

The PP / ST introduction describes the TOE in a narrative way on different levels of abstraction: the TOE reference, the TOE overview and — in an ST only — the TOE description. (cf. [CCV31_1], A.4, B.4). The relevant evaluator work units state:

APE_INT.1-3, ASE_INT.1-5 The evaluator shall examine the TOE overview to determine that it describes the usage and major security features of the TOE.

The PP / ST of a TOE containing an RNG *may* claim that the RNG is used for:

1. Providing random number generation as a security service to the user (e.g., if the TOE is a security integrated circuit (SIC), it provides random numbers for the operating system running on the SIC as platform),
2. internal purposes only (e.g., generation of cryptographic keys), or
3. provision of random number generation as a security service to the user and for internal purposes.

In cases 1. and 3., the random number generation should be described in the PP / ST introduction. In case 2., the random number generation may or may not be mentioned in the PP / ST introduction.

2.2.2 Security problem definition (APE_SPD, ASE_SPD)

The evaluator *shall* examine the security problem definition (SPD) of the PP / ST to find any threat, organizational security policy, or assumption relevant for the RNG according to the following work units:

{APE, ASE}_SPD.1-2 The evaluator *shall* examine the security problem definition to determine that all threats are described in terms of a threat agent, an asset, and an adverse action.

{APE, ASE}_SPD.1-3 The evaluator *shall* examine that the security problem definition describes the OSPs.

{APE, ASE}_SPD.1-4 The evaluator *shall* examine the security problem definition to determine that it describes the assumptions about the operational environment of the TOE.

The PP or ST *may* describe threats relevant for the RNG as follows:

The threat *may* address a direct attack against the random number generation service provided by the TOE to the user (e.g., the TOE is a smart card integrated circuit providing random numbers to the operating system for the generation of cryptographic keys).

Example 1: “T.RNG: An attacker might guess the random numbers generated by the TOE and those that were provided to the user due to insufficient entropy of the random numbers.”

Note that insufficient entropy might be caused, e.g., by an error in the RNG design, manipulation of the entropy source of the TRNG, insufficient or compromised seeding of the DRNG, compromised or manipulated internal state of the DRNG, or failure. For the threats described in the ST, only attacks affecting the RNG operation are relevant; design errors shall be detected in the evaluation process.

The threat *may* address an attack against TOE services (e.g., generation of the cryptographic key for digital signature creation) provided to the user, which are based on or make use of a vulnerability of the random number generator implemented by the TOE (i.e., the cryptographic key might be guessed).

Example 2: "T.SCD_Gen: An attacker might reconstruct a cryptographic key by affecting the key generation of the TOE."

An organizational security policy (OSP) in a PP or ST *may* impose security rules for RNGs aiming at high secrecy and quality of a security service provided by the TOE as follows:

Example 3: "OSP.PTRNG: The TOE provides random numbers generated by a physical random number generator as a security service for cryptographic purposes."

Security objectives like this might be appropriate for security integrated circuits providing random number generation for the operating system running on the chip.

Example 4: "OSP.KeyGen_PTRNG: The generation of cryptographic keys shall use physical random number generators."

Security objectives like this might be appropriate for cryptographic modules (e.g. cryptoboxes, smart cards) generating highly-secure cryptographic keys.

Physical random number generators should provide large entropy and should be independent from any other processes. Therefore, they might be appropriate for the generation of long term and very important keys (e.g., master encryption keys or root public keys of a public key infrastructure). Note that if the cryptographic keys are generated by a DRNG, the entropy contained in the keys is limited by the entropy of the internal state, which in the case of binary vectors is upper bounded by the length of the internal state.

The PP or ST *shall* describe any assumptions on the operational environment defined for the operation of the TOE and especially for the RNG, if applicable. The PP or ST *may* include assumptions like these:

- Environmental conditions for the operation of the internal entropy source of the PTRNG (e.g., power consumption, temperature),
- seeding procedure of the DRNG,
- external entropy sources and their properties assumed for the NPTRNG.

Example 5: [GuPR06] describes a use case where a Linux operating system was installed on an embedded router without any hard disk drives, keyboard, mouse, or other entropy sources for /dev/urandom. The router fails to generate cryptographic keys with sufficient entropy, thus an attacker might guess these keys. A clearly-stated assumption about the IT environment of Linux and a corresponding description of the IT environment in the guidance documentation should prevent such problems.

2.2.3 Security objectives (APE_OBJ, ASE_OBJ)

The PP or ST *shall* describe RNG-specific security objectives if random number generation is provided as a security service to the user. They *should* describe the intended purpose of the RNG security service.

Example 6: The security objective of the PP or ST may describe the security service for random number generation that provides random numbers of 128-bit length like this:

”OT.PTRNG: The TOE shall provide a security service for random numbers generated by means of a physical random number generator containing at least fresh 100-bit min-entropy in each 128-bit output.”

”OT.DRNG: The TOE shall provide a security service for random numbers containing at least 100-bit min-entropy in the output between seeding.”

The 100-bit min-entropy in the output prevents guessing of this output. The difference between these two objectives is that the statement about guessing is valid for each output number in the case of OT.PTRNG and for each instantiation of the DRNG (after seeding) in the case of OT.DRNG. Specific security features of the DRNG might ensure that this potential vulnerability is not exploitable (e.g., forward secrecy, enhanced backward secrecy).

The PP or ST *may* describe RNG-specific security objectives if random number generation is internally used for the generation of secrets for cryptographic or authentication purposes, e.g., the PP or ST will describe an RNG-specific security objective to enforce an OSP.

Example 7: The security problem definition of a PP or ST contains the organizational security policy OSP.KeyGen_PTRNG, as given in Example 4. The security objective of the PP or ST may cover the OSP like this:

“OT.KeyGen_PTRNG: The TOE shall generate secure cryptographic keys with maximum entropy using an internal physical random number generator.”

Note that the internal physical random number generator provides fresh entropy for the generation of cryptographic keys, but the entropy of the keys depends on the key generation algorithm and its implementation.

The security objective for the operational environment *shall* cover any assumptions about the RNG. Depending on the RNG type, the PP or ST *may* describe different security objectives for the operational environment to ensure secure operation of the RNG. The evaluator is reminded that the guidance documentation shall describe the security measures to be implemented by the user to fulfil these security objectives for the environment.

Example 8: The TOE is a smart card consisting of hardware and software and implements a PTRNG. The PTRNG may require specific operational conditions, e.g., upper and lower bounds for the operating temperature, in order to meet the security objective (for example, OT.KeyGen_PTRNG above). The security objectives for the operational environment might address this dependency as follows:

“OE.OperCond: The operational environment shall ensure the following environmental conditions for the TOE: …”

Security objectives for the operational environment may indicate a critical dependency on environmental conditions. It is up to the consumer to decide whether this dependency is appropriate for the intended usage (e.g., in a trust centre) or not appropriate (e.g. in an uncontrolled environment). In the case of state-of-the-art cryptographic modules and even smart cards, TOE environmental controls (e.g., sensors) ensure a secure state of the TOE if these environmental conditions are not met. If the PP or ST does not contain a security objective for the TOE to ensure the availability of the key generation service, the secure state will not violate the OT.KeyGen_PTRNG security objective. The PP or ST *should* describe the operational conditions under which the operation of the TOE is normally ensured, but this may not be a security objective for the operational environment.

Example 9: For a device like a PDA with a DRNG, the operational environment may describe a seeding procedure like this:

> "OE.DRNG: The personalization of the TOE shall ensure seeding of the DRNG with a confidential seed sequence of at least 40-bit min-entropy."

The initial entropy of the seed allows generation of secure passwords (cf. FIA_SOS.2 in combination with FIA_AFL.1), but might not be sufficient for generation of cryptographic authentication keys used in a challenge-response protocol if high resistance against attacks (cf. AVA_VAN.5) is claimed in the PP / ST. The reason is that passwords and cryptographic keys are subject to different attack scenarios .

Example 10: For an operating system implementing an NPTRNG, the operational environment may describe the necessary uses of input from keyboard, mouse and hard drive as entropy source:

> "OE.NPTRNG: The platform shall use a hard drive to store the TOE and user data. The human user shall use a keyboard and mouse for interaction with the TOE during start-up."

Note that solid state drives and automatic operation without human interaction might not provide the necessary entropy input.

2.2.4 Extended component definition (APE_ECD, ASE_ECD)

Most work units of the families APE_ECD and ASE_ECD do not have RNG-specific aspects, but APE_ECD.1-13 and ASE_ECD.1-13 do include RNG-specific aspects:

{APE, ASE}_ECD.1-13/: The evaluator *shall* examine the extended components definition to determine that each extended component may not be clearly expressed using existing components.

The CC already provide some hints to express the generation of random numbers using FIA_SOS.2 or FCS_CKM.1. According to annex G.3 of the CC, part 2, FIA_SOS is mostly used to generate secrets for authentication, and FCS_CKM.1 does not allow to describe the nature of the secrets used for key generation. Furthermore, the existing SFRs in the CC, part 2, do not allow to describe the characteristics of DRNGs and PTGs, specific cryptographic methods like padding, etc. In contrast, FCS_RNG.1 in combination with the pre-defined RNG classes allows to address the generation of random numbers in general and is better suited to express all relevant information within the Security Target.

The PP or ST writers *shall* use the component FCS_RNG.1 and the pre-defined RNG-classes defined in [KS2011] to describe the specific requirements for the RNG implemented by the TSF. This component does not need any extended assurance component.

2.2.5 Security requirements (APE_REQ, ASE_REQ)

Most work units of the family APE_REQ and ASE_REQ do not have RNG-specific aspects, but {APE, ASE}_REQ.2-3 and {APE, ASE}_REQ.2-13 do include RNG-specific aspects.

A PP *may* use the component FCS_RNG.1 and the pre-defined class DRG.{1,2,3,4} without further operation in the element FCS_RNG.1.1. An ST *shall* perform all open operations and may add security capabilities. The evaluator is reminded that the work unit APE_REQ.2-11 requires careful examination of all operations left open in the SFR FCS_RNG.1.

The evaluator determines that RNG-specific terms used in the operation of the SFR are defined.

{APE, ASE}_REQ.2-3 The evaluator *shall* examine the {PP, ST} to determine that all subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs are defined.

The definition may refer to further explanation or scheme documents (e.g. [KS2011]).

Example 11: The PP writer performs the operation of the element FCS_RNG.1.1 with a reference to or a copy of the capability list of the predefined class as follows:

"FCS_RNG.1.1 The TSF shall provide a *deterministic*[1] random number generator *as defined in [NIST800-90]*[2] that implements: *capability list of class DRG.2 as defined in [KS2011]*[3]. "

As shown in the example above, referring to standards like [NIST800-90] shall be incorporated into the SFR by means of an appropriate refinement in FCS_RNG.1.1.

If a PP / ST writer specifies a PTG, the security capability "(PTG.2.7) The average entropy per internal random bit exceeds 0.997" uses entropy to describe the quality of random numbers. The PP / ST writer performing this operation identifies

1. the type of entropy (e.g., Min-entropy or Shannon entropy) and
2. the quantity of the entropy (e.g., 7.9-bit per octet). An operation like "7.9" or "7.9-bit" is incomplete and incoherent.

In addition, the evaluator determines that the statement of security requirements shall be internally consistent.

{APE, ASE}_REQ.2-13 The evaluator *shall* examine the statement of security requirements to determine that it is internally consistent.

Some possible conflicts related to RNGs are:

- The PP / ST refers to different types of RNGs, but it is not clearly stated which one is used for a specific purpose.
- The PP / ST describes the use of DRNGs without any assumption about their initialization process or the external RNG seeding the DRNG.
- The PP / ST describes the use of NPTRNGs without any assumption about their operational environment, including the external entropy source of the NPTRNG.

2.2.6 TOE summary specification (ASE_TSS)

The TOE summary specification (TSS) enables evaluators and potential consumers to gain a general understanding of how the TOE is implemented. This is evaluated according to several work units like:

ASE_TSS.1-1 The evaluator *shall* examine the TOE summary specification to determine that it describes how the TOE meets each SFR.

1 [selection: *physical*, *non-physical true*, *deterministic*, *hybrid physical*, *hybrid deterministic*]

2 [refinement: *as defined in [NIST800-90]]*

3 [assignment: *list of security capabilities*]

If the TOE provides random number generation as a service for the user, the TSS *shall* describe how the TOE implements the RNG to meet the SFR addressing this service. If the TOE generates random numbers for internal use (e.g., for generating cryptographic keys) described by an SFR in the ST, the TSS *may* describe how the TOE implements the RNG to meet the respective SFR.

The TSS should inform potential consumers about:

- the type of the RNG: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic, and
- the most important security capabilities of the RNG implemented by the TOE.

The [CEM], paragraph 500, reminds the evaluator “that the objective of each description is to provide potential consumers of the TOE with a high-level view of how the developer intends to satisfy each SFR and that the descriptions therefore should not be overly detailed”. The type and security capabilities of the RNG might be of interest for consumers to understand, e.g., how key generation policies are met. Other security properties might be important for the evaluation process only, e.g., the quality of the generated random numbers if they are used internally for cryptographic key generation only.

3 Pre-defined RNG classes

[KS2011] defines the DRNG classes DRG.1, DRG.2 and DRG.3 by means of the SFR FCS_RNG.

In a ST it is recommended to rely on approved standards as listed below whereas not implementing an approved method might result in a potential vulnerability:

- [ISO18031] describes requirements for PTRNG, DRNG and NPTRNG.
- [RFC4086] specifies best practices for RNGs.
- [NIST800-90] describes NIST approved DRNGs.
- [RNGVS] describes a test methodology for the DRNGs defined in [NIST800-90].
- [FI186]: This standard's Annex 3 describes a method to produce pseudo-random numbers of 160 bits[4].

3.1 Intended use for pre-defined RNG-classes

The following table lists the allowed RNGs together with their intended operation. The RNGs are defined in [KS2011], chapter 4:

RNG	***Intended operation***
DRG.{1, 2, 3, 4}, PTG.{2,3}, NTG.1	- generation of challenges in cryptographic protocols or initialization, - generation of initialization vectors for block ciphers in special operational modes, - generation of random numbers for zero-knowledge proofs, provided that previous random numbers need not be protected,
DRG.{2, 3, 4}, PTG.{2,3}, NTG.1	- generation of symmetric or asymmetric cryptographic keys, - generation of (pseudo-)random padding bits, - generation of random numbers for the derivation of cryptographic keys (e.g. ECDH or DH), - generation of unpredictable but publicly known initialization vectors / nonces.
DRG.{2, 3, 4}, PTG.3, NTG.1	- Strong recommendation for creation of ephemeral keys for digital signature algorithms like ECDSA or DSA.
DRG.{3, 4}, PTG.3, NTG.1	- Strong recommendation for creation of random numbers in environments where the internal state of a RNG is not physically protected.

Table 1: Intended operations for predefined RNG classes

4 Note the draft of FIPS 186-3 refers to [NIST800-90] for generation random numbers.

In related real-world applications there might be special requirements, and progress in cryptanalysis in the next years might make it necessary or at least advisable to select an RNG from a higher class. For these reasons the choice of an appropriate RNG-class is a decision on a case-by-case basis for every certification process.

4 Methodology documents

Currently, the evaluation methodology for the evaluation of random number generators in the German scheme consists of the following documents:

- [KS2011] provides the theoretical background, a reference list, abbreviations, the terminology, the definition for different types of RNGs, much information regarding the different RNGs and examples in one document. A developer applying for certification of a product that uses the generation of random numbers as a security functionality is strongly recommended to read this document. The same applies to evaluators who are going to work in the field of cryptographic assessment.
- The evaluation methodology provides a template for the evaluation of DRNGs [DRGEV] and of PTRNGs [PTGEV]. This template contains work units expressing the requirements for the evaluation of the developer evidence, tests and penetration tests. Similar to the [CEM] each work unit contains further hints that must be taken into account by the evaluator. The format of the template is derived from the templates for ETR parts published along with the AIS 14. The BSI plans to publish corresponding templates for the evaluation of NPTRNGs as well.
- Analogously to the templates for the evaluation report the methodology currently contains a document with information about the required evaluation evidence for DRNGs [DRGDEV] and PTRNGs [PTGDEV]. This information is intended for developers and evaluators likewise. From the document developers can deduce what they have to provide for the evaluation of their DRNG or PTRNG, and evaluators gain an understanding about the evaluation evidence their work is based on. The BSI plans to publish corresponding documents for the description of the required evaluation evidence for NPTRNGs as well.
- Compared to the former approach the methodology was changed considerably. It is the intention of the BSI to avoid gaps in the methodology due to errors or unwitting omissions that were newly introduced by reworking the documents. Therefore, the basic documents [AIS20An] and [AIS31An] referring to the evaluation of deterministic and physical true random number generators of the former approach are not superseded by the new documents. They are still part of the methodology and should be taken in account if the newly created documents do not provide answers to certain questions.

The BSI will continuously update the documents to take new information and findings into account.

Appendix

Literature

AIS20An	W. Schindler: AIS 20: Functionality classes and evaluation methodology for deterministic random number generators, Version 2.0, English Translation, 02.12.1999
AIS31An	W. Killmann, W. Schindler: A Proposal for: Functionality Classes and Evaluation Methodology for True (Physical) Random Number Generators, Version 3.1, English translation, 25.09.2001
CCV31_1	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 4, CCMB-2012-09-001, September 2012
CCV31_2	Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements, Version 3.1, Revision 4, CCMB-2012-09-002, September 2012
CCV31_3	Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements, Version 3.1, Revision 4, CCMB-2012-09-003, September 2012
CEM	Common Methodology for Information Technology Security Evaluation (CEM): Evaluation Methodology, Version 3.1, Revision 4, CCMB-2012-09-004, September 2012
DRGDEV	Developer evidence for the evaluation of a deterministic random number generator, Version 0.9, February 28, 2013
DRGEV	Evaluation Report as part of the Evaluation Technical Report, Part B, ETR-Part, Deterministic Random Number Generator, Version 0.10, February 28, 2013
FI186	NIST: FIPS PUB 186-2, Specifications for the Digital SignatureStandard (DSS), with Change Notice 1, October 2001
GuPR06	Z. Guttermann, B. Pinkas, T. Reinman: Analysis of the Linux Random NumberGenerator, The Hebrew University of Jerusalem, March 6, 2006
ISO18031	ISO/IEC 18031: Random Bit Generation, November 2005
ITSEC	Information Technology Security Evaluation Criteria (ITSEC), Provisional Harmonised Criteria, Version 1.2, June 1991
ITSEM	Information Technology Security Evaluation Manual (ITSEM), Provisional Harmonised Methodology, Version 1.0, September 1993
KS2011	W. Killmann, W. Schindler, „A proposal for: Functionality classes for random number generators“, Version 2.0, September 18, 2011
NIST800-90	Elaine Barker, John Kelsey: NIST Special Publication 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised), March 2007

PTGDEV	Developer evidence for the evaluation of a physical true random number generator, Version 0.8, February 28, 2013
PTGEV	Evaluation Report as part of the Evaluation Technical Report, Part B, ETR-Part, True Physical and Hybrid Random Number Generator, Version 0.7, February 28, 2013
RFC4086	D. Eastlake, S. Crocker, J. Schiller: RFC 4086 Randomness Requirements for Security, June 2005
RNGVS	National Institute of Standards and Technology, Information Technology Laboratory,Computer Security Division: The Random Number Generator Validation System(RNGVS), January 31, 2005