A Closer Look at Security in Random Number Generators Design

Viktor Fischer

A Closer Look at Security in Random Number Generators Design

Viktor Fischer

Laboratoire Hubert Curien UMR 5516 CNRS
Jean Monnet University, member of University of Lyon
Rue du Prof. Benoit Lauras, 18, 42000, Saint-Etienne, France
http://laboratoirehubertcurien.fr/spip.php?rubrique29

Abstract. The issue of random number generation is crucial for the implementation of cryptographic systems. Random numbers are often used in key generation processes, authentication protocols, zeroknowledge protocols, padding, in many digital signature and encryption schemes, and even in some side channel attack countermeasures. For these applications, security depends to a great extent on the quality of the source of randomness and on the way this source is exploited. The quality of the generated numbers is checked by statistical tests. In addition to the good statistical properties of the obtained numbers, the output of the generator used in cryptography must be unpredictable. Besides quality and unpredictability requirements, the generator must be robust against aging effects and intentional or unintentional environmental variations, such as temperature, power supply, electromagnetic emanations, etc. In this paper, we discuss practical aspects of a true random number generator design. Special attention is given to the analysis of security requirements and on the way how this requirements can be met in practice.

Keywords: Random number generation, cryptographic hardware, data security, statistical tests, digital design

1 Introduction

Random number generators (RNGs) are one of basic cryptographic primitives used to design cryptographic protocols. Their applications include - but are not limited to - the generation of cryptographic keys, initialization vectors, challenges, nonces and padding values, and the implementation of countermeasures against side channel attacks. RNGs aimed at cryptographic applications must fulfill basic security requirements. First of all, their output values must have good statistical properties and be unpredictable. In modern designs, some additional features are required: the generator must be inherently secure, robust and resistant to attacks and/or tested on line using generator specific tests.

The security of cryptographic systems is mainly linked to the protection of confidential keys. In high end information security systems, when used in an uncontrolled environment, cryptographic keys should never be generated outside

the system and they should never leave the system in clear. For the same reason, if the security system is implemented in a single chip (cryptographic system-on-chip), the keys should be generated inside the same chip. Implementation of random number generators in logic devices (including configurable logic devices) is of paramount importance.

There are three basic challenges in modern embedded TRNG design: (i) finding a good quality source of randomness (available in the digital technology); (ii) finding an efficient and robust principle of randomness extraction; (iii) guaranteeing the security (e.g. by a robust design or by an efficient online testing).

Historically, three basic RNG classes are used in cryptography: deterministic, nondeterministic (physical) and hybrid random number generators.

Deterministic (pseudo-) random number generators (DRNGs) are mostly fast and have good statistical properties. They are usually used as key generators in stream ciphers. Due to the existence of some underlying algorithms, DRNGs are easy to implement in logic devices. However, if the algorithm is known, the generator output is predictable. Even when the algorithm is not known but some of the generator output sequences have been recorded, its behavior during the recorded sequence can be used in future attacks.

Physical (true-) random number generators (TRNGs) use physical processes to generate random numbers. If the underlying physical process cannot be controlled, the generator output is unpredictable and/or uncontrollable. The final speed of TRNGs is limited by the spectrum of the underlying physical phenomenon and by the principle used to extract entropy from it (e.g. sampling frequency linked with the noise spectrum). The statistical characteristics of TRNGs are closely related to the quality of the entropy source, but also to the randomness extraction method. Because physical processes are subject to fluctuations, the statistical characteristics of TRNGs are usually worse than those of DRNGs.

Hybrid random number generators (HRNGs) represent a combination of a (fast and good quality) deterministic RNG seeded repeatedly by a (slow but unpredictable) physical RNG. The designer has to find a satisfactory compromise between the speed of the generator and its predictability (by adjusting the time interval between seeds and the size of a seed).

TRNGs are the only cryptographic primitives that have not been subject to standardization up to now. However, before using the generator in practice, the principle and its implementation inside a cryptographic module has to be validated by an accredited institution as part of a security evaluation process. Generators that do not have a security certificate are considered to be *insecure* in terms of their use in cryptographic applications. Many TRNG designs exist, but only few of them deal with security. In this paper, we will focus on security aspects in TRNG design.

The paper is organized as follows. In Sec. 2, we present briefly basic approaches in TRNG design. In Sec. 3, we present and discuss basic TRNG design evaluation criteria and in Sec. 4 we analyze in detail TRNG security requirements. In Sec. 5, we sum up basic requirements for future secure TRNG designs. We conclude the paper in Sec. 6.

2 TRNG Design

The TRNG design styles evolved significantly in past few years. In the classical approach (see Fig. 1a), the designers usually proposed some (new) principle reflecting required design constraints such as area, throughput and/or power consumption. In the development phase, they obviously used FIPS 140-1 [9] or FIPS 140-2 statistical tests for verifying the quality of the generated bitstream, because these simple tests need short data files and they give a good quality estimation. In order to validate the final quality of the generated bitstream, the designer tested the generated data using standard statistical test suites like NIST SP 800-22 [20] or DIEHARD [19].

Even though statistical tests are required to evaluate the quality of the generated sequence, they cannot distinguish between pseudo random data generated by a deterministic generator and truly random data from a physical TRNG. This was one of the reasons, why German BSI (Bundesamt für Sicherheit in der Informationstechnik) proposed in 2001 a new methodology aimed at evaluation of physical random number generators. The AIS 31 methodology [15] defined several RND classes and their security requirements. It was updated in 2011 and new RNG classes were defined [16].

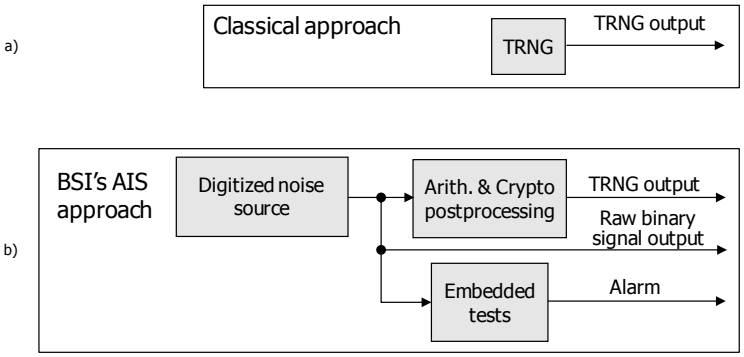

Fig. 1. Classical (a) and German BSI's (b) approach in TRNG design

According to the TRNG evaluation methodology proposed by BSI (see Fig. 1b), the generator should use an uncontrollable physical process as a source of randomness. Since physical phenomena used in TRNGs are mostly analog processes, some method enabling data conversion from analog to digital domain (as a part of randomness extraction procedure) is usually necessary.

The obtained unprocessed raw binary signal (so-called digital noise) can have low entropy and/or bad statistical properties (e.g. it can be biased). In this case, some post-processing algorithms can be used to enhance the statistical parameters of the output bitstream. While the algorithmic post-processing is optional, the following cryptographic post-processing can be strictly required according to the targeted security level. The cryptographic post-processing plays

very important security role if the source of randomness fails: (i) it can serve temporarily as a DRNG; (ii) according to the application security level, it should guarantee TRNG unpredictability in forward, backward or both directions.

Since the cryptographic algorithm implemented in the post-processing block behaves as a DRNG when a true randomness fails, the latest AIS methodology [16] merges evaluation of true random number generators and pseudorandom number generators into a common evaluation procedure and introduces new RNG subclasses (see Tab. 1): Physical TRNG (PTG.1 and PTG.2), Hybrid physical TRNG (PTG.3), Deterministic RNG (DRG.1, DRG.2 and DRG.3), Hybrid deterministic RNG (DRG.4) and Non-physical TRNG (NTG).

Table 1. New AIS RNG classes

RNG Class	AIS20/AIS31 Class	Comments
PTG.1	AIS31, P1	Physical TRNG with an internal total failure test of the entropy source and tests of non-tolerable statistical defects of the TRNG output
PTG.2	AIS31, P2	PTG.1 + a stochastic model of the entropy source and statistical tests of the raw binary signal
PTG.3	No counterpart	PTG.2 + cryptographic post-processing (hybrid PTRNG)
DRG.1	AIS20, K2, partly K3	DRNG with forward secrecy
DRG.2	AIS20, K3	DRG.1 + backward secrecy
DRG.3	AIS20, K4	DRG.2 + enhanced backward secrecy
DRG.4	No counterpart	DRG.3 + enhanced forward secrecy (hybrid DRNG)
NTG.1	No counterpart	Non-physical TRNG with entropy estimation

TRNG output post-processing can sometimes mask a serious faults, which standard statistical tests may fail to detect. Therefore, the unprocessed digital noise must be tested in classes with higher security requirements (PTG.2 and PTG.3). The dedicated tests should suit the generator's principle with particular reference to its potential weaknesses and should be executed on the fly.

3 TRNG Design Evaluation Criteria

True random number generators use different sources of randomness and numerous principles to extract it. TRNG designs (not TRNG implementations !) can be evaluated using three classes of criteria [1]: (i) characteristics related to the TRNG principle; (ii) design related characteristics; and (iii) security related characteristics.

3.1 Criteria Related to the TRNG Principle

This set of parameters determines the main characteristics of the generator. It includes parameters like source of randomness, method of randomness extraction, post-processing algorithms, output bitrate and its stability.

Source of Randomness
Logic devices are designed for the implementation of deterministic logic systems. Each unpredictable behavior in such a system (caused by a metastability, clock jitter, radiation errors, etc.) can have catastrophic consequences for the behavior of the overall system. For this reason, vendors of logic devices tend to minimize these causes. As a consequence, the TRNG design should always be critically examined in order to keep up with the evolution of the underlying technology.

Most logic devices do not contain analog blocks, so the sources of randomness are related to the operation of logic gates. Analog physical phenomena (like thermal, shot and flicker noise) are transformed to time domain instability of logic signals [13]. This can be seen as a variation in the delay of logic gates, analog behavior of logic gates between two logic levels (e.g. metastability) [18], [14] or randomness in two concurrent writings to RAM memory blocks [12], [11].

The instability of gate delays causes signal propagation variations over time. These variations can be seen as a clock period instability (the jitter) in clock generators containing delay elements assembled in a closed loop (ring oscillators). The variation in propagation time is also used in generators with delay elements in an open chain assembly [7].

Some generators use the tracking jitter introduced by phase locked loops (PLLs) available in digital technology [10].

Method of Randomness Extraction
In general, random numbers can be obtained in two ways: sampling random signals at regular time intervals or sampling regular signals at random time intervals. In synchronous systems, the first method is preferable in order to guarantee a constant bit rate on the output. In logic devices, randomness is often extracted by *sampling a jittery (clock) signal using synchronous or asynchronous flip-flops (latches)* and a reference (clock) signal.

The choice between synchronous and asynchronous flip-flops does not seem to be important in ASICs, but it is very important in FPGAs. This is because synchronous flip-flops are hardwired in logic cells as optimized blocks and their metastable behavior is consequently minimized. On the other hand, latches can usually only be implemented in Look up tables (LUTs) and are therefore subject to metastable behavior to a greater extent [7].

Another ways of extracting randomness are: (i) counting number of random events [28] or (ii) counting number of reference clock periods in a randomly changing time interval [26].

The randomness extraction method is usually linked to the basic principle of the generator and to the source of randomness. The randomness extraction procedure and post-processing are sometimes merged into the same block and cannot be separated [24]. In that case, the entropy of the randomness source is masked by post-processing and cannot be evaluated or tested correctly.

Arithmetic Post-processing of the Raw Binary Signal
The entropy source may have some weaknesses that lead to the generation of

non-random numbers (e.g. long sequences of zeros or ones). In this case, post-processing may be necessary to improve the statistical properties of random numbers, for example to increase entropy per bit, reduce bias and/or correlation.

The quality of the digital noise signal (the signal obtained at the randomness extraction block output) can deteriorate for several reasons: (i) the entropy of the source is not high enough (this can be the case if metastability is used as a source of randomness); (ii) the entropy, which is high in the original signal, is not efficiently extracted; (iii) the extracted samples are correlated. The entropy per bit at the output of the generator is mostly increased at the cost of reduction and/or variation in the bit rate. Most of arithmetic post-processing methods use some data compression technique in order to increase entropy per bit at generator's output.

Cryptographic Post-processing
This kind of the post-processing uses both diffusion and confusion properties of cryptographic functions. The perfect statistical characteristics of most of the encryption algorithms can be used to mask generator imperfections. One of advantages of this approach is that the encryption key can be used as a cryptographic variable to dynamically modify the behavior of the generator. Although this kind of post-processing block (the cipher) is rather complex and expensive, the TRNG can reuse (share) the cipher that is used for data encryption.

One of the most expensive (in time and area) but also one of the most secure methods is cryptographic post-processing based on hash functions. It uses diffusion and one-wayness (as opposed to encryption of the raw binary sisgnal) properties of hash functions to ensure the unpredictability of bits generated by the TRNG if a total breakdown of the noise source occurs. In this case, due to the non-linearity property of hash functions, the TRNG will behave like a cryptographically secure DRNG.

Output Bit Rate and its Stability
The speed is a secondary parameter (after security) in many cryptographic applications. Output bit rates from hundred kilobits per second up to 1 megabit per second are usually sufficient. However, there are some speed critical data security applications for which high speed generators are required. For example, Quantum cryptography requires a high bit rate (up to 100 megabits per second) because of the very low efficiency of key data transmission over the low-power optical channel.

High speed telecommunication servers can be given as a second example. They need to generate session keys regularly and at a high frequency (tens of megabits per second). For example a 10-Gbit Ethernet hub/server would need at least 20 Mbits/s random bits to generate one 128-bit session key for each 64kB data block in order to be able to face side channel attacks (giving 4k enciphered blocks per key).

Another aspect of the output bit rate that has to be considered is its regularity. Some generators give random numbers periodically, others generate output

in irregular time intervals. In the second case, a FIFO is required to accumulate the generated numbers. Another solution is to estimate the smallest bit rate available at the output and to sample the output at this rate. The disadvantage of the first solution is that, depending on the mean output bit rate and the need for random numbers, the FIFOs sometimes need to be very big. The disadvantage of the second solution is that if the estimated bit rate is incorrect, the random numbers may not be always available at the output.

3.2 Criteria Related to the TRNG design

Resource Usage
To evaluate practical usefulness of various TRNG principles, it is important to analyze the kind and number of resources needed for generator hardware implementation. Of course, the FPGA technology is more restrictive than its ASIC counterpart. In FPGAs, designers can use: LUT based or multiplexer based logic cells, embedded memory blocks, clock blocks featuring PLLs and DLLs, embedded RC oscillators, hardwired multipliers, programmable interconnections, etc.

FPGAs have many logic cells, so the use of logic cells (the logic area) is usually not a problem. However, the topology and electrical parameters of programmable interconnections are strongly technology dependent. Many TRNG designs require designer's manual intervention during placement and routing (P/R). Some designs can be easily implemented in one FPGA family, but could be difficult or impossible to implement in others. The choice and the number of embedded hardwired blocs is usually much more limited (PLLs, RC oscillators, multipliers, memory blocks) and varies with the vendor and the technology. The use of hardwired blocks can thus be a limiting factor for reusability of the TRNG principle.

Power Consumption
The power consumption of the generator is linked to its randomness source (e.g. the oscillator), to the clock frequency used and to the post-processing algorithm agility. In power critical applications, the generator can be stopped when not in use. However, the possibility to stop the bit stream generation can be used to attack the generator.

Technological Requirements
Compared to the implementation of TRNGs in ASICs, their implementation in FPGAs is much more restricted. Many TRNGs implemented in ASICs use analog components to generate randomness (e.g. chaos based TRNGs using analog to digital converters, free running oscillator based generators using thermal noise from diodes and resistors, etc.) and to process randomness (e.g. operational amplifiers, comparators, etc.).

Most of these functional blocks are usually not available in digital technology and especially in FPGAs, although some of them may be available in selected families, e.g. RC oscillators in Microsemi (Actel) Fusion FPGA, analog PLLs in most Altera and Actel families but not in old Xilinx families. From the point

of view of their feasibility in FPGAs, some generators are not feasible or are difficult to implement in FPGAs, some are feasible in selected FPGAs and the most general principles are feasible in all FPGAs.

Design Automation Possibilities
The fact that the generator uses resources that are available in given technology does not automatically mean that it can be implemented in this kind of technology. The range of tolerance of some technology parameters can be such that it prevents reliable implementation of the generator. This is especially true in FPGA technology.

The parameter that limits generator implementation in FPGAs is the availability of routing resources and their characteristics. Some generators require perfectly balanced routing. This necessitates perfect control of the module placement (e.g. symmetrical placement of two modules in relation to another module) and routing. While most FPGA design tools allow precise control of placement, the routing process is difficult or impossible to control (e.g. in the Microsemi family). Even when routing can be partially or totally controlled (e.g. Altera and Xilinx families), the delays in the configurable routing net vary so much from device to device that it is impossible to balance module interconnections in a general manner and the design will be device dependent, i.e. it has to be balanced manually for each device. Such manual intervention is not acceptable from the point of view of the practical implementation of the generator.

The best generators (very rare) can be mapped automatically (without manual intervention) in all FPGA families. From a practical point of view, implementation of the generator that requires manual P/R for each family and/or type of device, remains acceptable. However, generators that require manual optimization for each device are not tolerable in industrial applications.

3.3 Criteria Related to the TRNG security

Robustness, Resistance against Attacks
Besides defining the compression ratio, the entropy bound given by the statistical model can be used for security evaluation of the generator. Namely, it can help in estimating the robustness of the generator against intentional or unintentional environmental variations. Concerning attacks and resistance against them, there are three possibilities: (i) proof exists that the generator cannot malfunction as the result of any attack or of a changing environment (proof of security), (ii) neither security proof nor attack exists, (iii) some attack on a particular generator has been reported.

Existence of a Statistical Model and its Reliability
The randomness of generated bitstream can be characterized by the entropy increase per bit at the generator output. Unfortunately, entropy is a property of random variables and not of observed realizations (random numbers). In order to quantify entropy, the distribution of the random variables must be analyzed, e.g. by the use of a stochastic model.

Stochastic models are different from physical models. Figure 2 depicts the mechanical principle of the metastability (that is useful for understanding metastability in electronics). In this case, the physical model of the metastability would describe the form of the hill and the stochastic model would describe probability distribution of the ball final position according to the form and the width of the hill. In general, stochastic models are easier to construct.

The stochastic model must describe only random process that is indeed used as a source of randomness. The metastability in Fig. 2 is related to the ability of the ball to stay at the top of the hill during random time interval. It is clear, that it is very difficult (but not completely impossible) to place and to maintain the ball on the top. However, it is completely impossible to place it periodically exactly at the top in small time periods (in order to increase the bitrate) as is supposed to be done in papers presumably using metastability, e.g. in [18].

The stochastic model serves for estimating the lower entropy bound. This value should be used in the design of the arithmetic post-processing block: the lower entropy bound determines the compression ratio necessary for increasing the entropy per output bit to a value close to 1. It can also be used for testing the entropy of the generated random bits in real time (online tests).

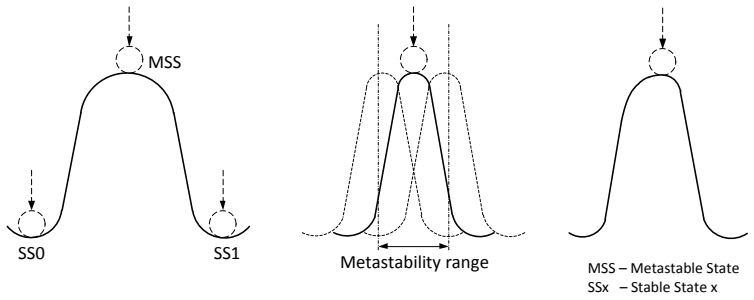

Fig. 2. Mechanical (physical) model of metastability

Inner Testability

Inner testability means that the generator structure enables evaluation of the entropy of the raw binary signal (if it is available) [6]. Indeed, in some designs, randomness extraction and post-processing are merged into the same process and the unprocessed random signal (the raw binary signal) is not available. Even if this signal is available, it is sometimes composed of a pseudo random pattern combined with a truly random bit stream [4].

The pseudo random pattern makes statistical evaluation of the raw signal more difficult. For this reason, we propose a new testability level: an *absolute inner testability*. The raw binary signal of the generator featuring absolute inner testability does not include a pseudo random pattern and contains only a true random bit stream. If (for some reason) the source of randomness fails, the raw signal of the generator will be zero. This fact can be used to detect very quickly and easily the generator's total failure.

3.4 TRNG Design Evaluation – Conclusions

The TRNG characteristics discussed in Sec. 3 are not all equally important. Security parameters like robustness, availability of a stochastic model, testability, etc. always take priority in a data security system. Their weight in TRNG evaluation is much higher than that of other parameters like power consumption, bit rate, etc. For this reason, we will analyze these criteria in more details and give some practical recommendations in the next section.

4 Main Security Issues in Published TRNG Designs

The output of a good TRNG should be indistinguishable from the output of an ideal TRNG, independently of operating conditions and time. The quality of the generator output bit stream and its security parameters including robustness against aging, environmental changes, attacks, existence of selftest and online tests are very important in the TRNG design.

4.1 Sensitivity of the TRNG to Variations of Operating Conditions

The quality of the generator output is tightly linked with the quality of the source of randomness and to the randomness extraction method used. The physical characteristics of the source of randomness (e.g. frequency spectrum) and the randomness extraction method determine the principal parameters of the generated bit stream: the bias of the output bit stream, correlation between subsequent bits, visible patterns, etc. While some of these faults can be corrected by efficient post-processing, it is better if the generator inherently produces a good quality raw bit stream.

It is of extreme importance that the generator is dimensioned to the *minimum* amount of random physical quantities (noise, jitter, etc.) that cannot be further reduced. The thermal noise can be considered as such a source of entropy. However, the total noise in digital devices is mostly a composition of random noises (such as thermal noise, shot noise, flicker noise, etc.) coming from global and independent local sources, but also of data dependent deterministic noises that can be very often manipulable.

If the extractor samples the source of randomness too fast, adjacent bits could be correlated. For this reason, it is good practice to check the generated bit stream for a short term auto correlation. It is also possible that the digital noise exhibits some other short term dependencies, which need to be detected by some generator specific tests.

The behavior of the generator is often influenced by external and/or internal electrical interferences. The most obvious effect of this will be discrete frequencies from the power supply and from various internal signals appearing in the noise spectrum.

The spectrum of the generated noise signal can be significantly influenced by a low frequency 1/f noise caused by semiconductors. Furthermore, the high

frequencies from the noise spectrum may be unintentionally filtered out by some internal capacities. Presumably white Gaussian noise will thus have a limited spectrum that will not be uniform.

Some generators can feature so called bad spots. Bad spots are short time periods, during which the generator ceases to work, due to some electrical interference or to extreme excursions of the generator's overloaded circuitry.

Another dangerous feature of the generator can be a back door, which refers to the deviations from uniform randomness deliberately introduced by the manufacturer. For example, let us suppose that instead of using some physical process, the generator would generate a high quality pseudo random sequence with a 40-bit seed. It would be impossible to detect this behavior by applying standard statistical tests on the output bit stream, but it would be computationally feasible for someone who knows the back door to guess successive keys.

When implementing TRNG as a part of a cryptographic system on chip, designers must take into account that the circuitry surrounding the generator will influence the generator's behavior by the data dependent noise present in power lines and by cross-talks. This impact is not so dangerous if two conditions are fulfilled: (i) the low entropy bound estimation of the generator does not include the digital noise from the system on chip activities; (ii) embedded online tests verify continuously that the effective entropy is not below this bound.

Very few papers evaluate the impact of the environment on the source of randomness and on the operation of the TRNG. The generator uses *all* the sources contributing to the selected phenomena. For example the clock jitter is determined by the local noise sources, but also by global sources from power supply, electromagnetic emanations etc. If the low entropy bound was estimated for the sum of noise sources, it will be sufficient for the attacker to put the generator to ideal conditions (low noise battery power supply, metallic shielding) in order reduce the entropy under the estimated low bound.

The generator's design must be evaluated in changing environmental conditions (temperature, electromagnetic emanations, etc.). It must be tested and embedded test validated for edge values (only one parameter is set to its maximal value) and corner values (more or all parameters are set to their critical value) of environmental parameters.

Recently, we have developed a set of evaluation boards (modules) aimed at fair TRNG benchmarking [5]. Five modules using five different FPGA families are available: Altera Cyclone III, Altera Arria II, Xilinx Spartan 3, Xilinx Virtex 5 and Microsemi Fusion. All the modules have the same architecture featuring selected FPGA device, linear power supply, two LDVS outpus for external jitter measurement and optionally 32 Mbits of external RAM for fast data acquisition. The modules are plugged to a motherboard containing linear power supplies (the card can be powered by battery, too), USB interface control device from Cypress. The modules are accessible remotely on demand and can be used for a fair evaluation of TRNG designs in the same working conditions. The new generation will be placed in an electromagnetic shielding and will communicate with PC via optical fibers.

4.2 Security Threats Related to Statistical Models and Entropy Estimators

Very few recent designs deal with stochastic models [23], [2], [3], [22], [8], [28]. The most comprehensive model of a two-oscillator based TRNG is presented in [2]. It characterizes randomness in the frequency domain. However, underlying physical hypotheses (clock jitter as a one-dimensional Brownian motion) must be still thoroughly evaluated.

A stochastic approach (an urn model) based on a known jitter size is presented by Sunar et al. in [23]. Unfortunately, it is based on several unrealistic assumptions criticized by Dichtl in [8]. Some of these assumptions, such as jitter overestimation (due to jitter measurement outside the device using standard input/output circuitry) can be corrected by using differential oscilloscope probes in combination with LVDS device outputs [25]. Unrealistic requirements given on the XOR gate were later resolved by Wold and Tan in [30].

However, the most security critical assumption of Sunar et al. turned out to be the mutual independence of rings (basic assumption for the validity of the model). It was shown in [4] that the rings are not independent and that up to 25% of them can be mutually locked. This phenomenon would reduce significantly the validity of the Sunar et al.'s model and consequently the entropy estimation and the security of the generator.

It is worth mentioning that Wold and Tan made another security critical attempt: since (by changing the original TRNG structure) the raw binary signal at the XOR gate output passed statistical tests more easily, they deduced that the entropy is sufficient enough (without measuring the jitter) and consequently they reduced considerably the number of rings (from 114 to 25). From the security point of view, this attempt is not acceptable, since it caused significant entropy reduction (according to the model, only few urns were filled).

The models presented in [3] are restricted to TRNGs based on coherent sampling [17], [26], [10]. However, these models have only limited practical value, because the first TRNGs in [17] and [26] have some technological limits (difficulty to set up precisely the generated clock signals periods) and the PLL-based TRNG from [10] uses the jitter with a complex profile (some deterministic jitter coming from the PLL depends on the characteristics of the PLL control loop).

4.3 Embedded TRNG Testing and Related Security Issues

In contrast to standard methods that tests only the TRNG output, the AIS methodology requires to test (for higher security levels) also the raw binary signal (see Fig. 1b). This new approach is motivated by the fact that the post-processing can mask serious defects of the generator. If a stochastic model of the physical randomness source is available, it can be used in combination with the raw signal to estimate the entropy and the bias depending on random input variables and depending on the generator principle.

The raw binary signal is also used in *Online tests*. Online tests should be applied to the digital noise signal while the generator is running. They provide

ways to stop the TRNG (at least temporarily) when a conspicuous statistical feature is detected. A special kind of online tests required by the AIS methodology is a "total failure test" or *Tot test* that should be able to immediately detect total failure of the generator.

Evaluating TRNGs is a difficult task. Clearly, it should not be limited to testing the TRNG output. Following the AIS methodology, the designer should also provide a stochastic model based on the noise source and the extraction process and propose statistical and online tests suited to the generator's principle. The AIS methodology does not favor or exclude any reasonable TRNG design. The applicant can also substitute alternative evaluation criteria, however these must be clearly justified.

Surprisingly, no design was evaluated in the literature following the AIS recommendations for high level security (separate testing of the raw binary signal and internal random numbers required for PTG.3 and PTG.4) up to now. Some papers just apply AIS tests T0 to T4 at the generator output. It is also worth pointing out that no paper proposed up to now a design-specific online test, not even a design-specific total failure test. Surprisingly, most of recent designs are still evaluated by their authors following the classical approach from Fig. 1a.

In our approach, we propose a new extension of security in TRNG design, which is depicted in Fig. 3. This approach simplifies significantly security evaluation, construction of the generator's stochastic model and last but not least, realization of simple and fast embedded test, while being entirely compatible with the AIS methodology.

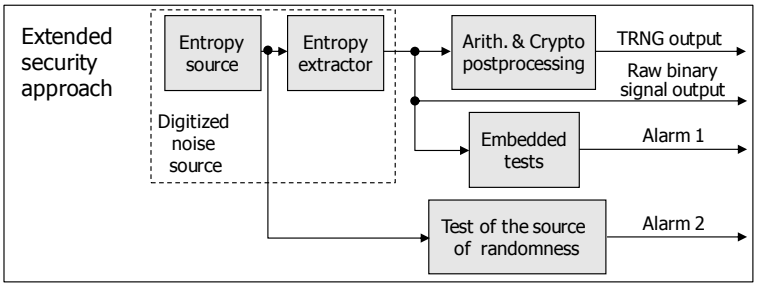

Fig. 3. New security approach in TRNG design based on embedded randomness testing

We propose to measure the source of entropy (e.g. the jitter) before the entropy extraction. This way, the randomness quantification is easier and more precise. Since the entropy extraction is an algorithmic process, it can be easily included in the stochastic (mathematical) model. However, two conditions must be fulfilled in our approach: (i) the method must to quantify exactly the same physical process that is used as a source of randomness; (ii) the entropy extraction algorithm must be included in the stochastic model very precisely. We have analyzed many recent TRNG principles. Unfortunately, only few of them

are directly (without modification) applicable. For example, we can cite those published in [17], [26], [10] and [27].

Some papers deal with implementation of embedded tests FIPS, NIST, etc. inside the device [21], [29]. Unfortunately, their authors do not consider the impact of the tests on the TRNG itself. The tests generate temporarily the digital noise (which let them pass more easily) and during the normal operation the effective noise (and consequently also entropy) can be significantly smaller.

5 Recommendations for Future Secure TRNG Designs

According to the previous analysis of TRNG designs and of security requirements in modern cryptographic systems, we propose designers to follow the next list of recommendations:

- Designers should clearly define the targeted security level. For example, in the context of the AIS procedure, they should specify the RND class.
- If higher security classes are targeted, the generator must be inner testable.
- A fast total failure test adapted to the TRNG principle must be proposed, implemented and executed continuously.
- If some online tests are *embedded* in the device, the designer should verify that the tests do not have any impact on the generated random numbers themselves, otherwise the tests must be executed continuously.
- If the generator makes part of a system on chip, the designer should verify that system working does not have a negative impact on the generator (i.e. that generation of random numbers cannot be manipulated by varying the system activity).
- The highest security can be obtained if the source of randomness (e.g. the jitter) is is measured online inside the device according to Fig. 3. In this case, the designer must pay particular attention to the fact that he measures *exactly* the same kind of physical parameter, which is used as a source of randomness. The same parameter must be used to build a stochastic model and to verify in real time the low entropy bound.
- The generator must be tested and embedded test validated in edge and corner values of environmental parameters.

6 Conclusion

In this paper, we have presented basic approaches to designing modern TRNGs. We have presented and discussed basic TRNG design evaluation criteria, such as sources of randomness and randomness extraction method applied, arithmetic and cryptographic post-processing method utilized, output bitrate and its stability, resource usage, power consumption, technological and design automation requirements, etc.

We have explained that security parameters like robustness, availability of a stochastic model, testability, etc. always take priority in a data security system.

We have also proposed a new level of testability criteria: the absolute inner testability. Furthermore, the new TRNG design approach testing the source of entropy before entropy extraction presented in this paper contributes to security enhancement of future TRNG design. We have also proposed a solution, which can serve for a fair TRNG benchmarking. In the last section, we have summed up several recommendations aimed at securing TRNG designs in general.

References

1. Badrignans, B., Danger, J.L., Fischer, V., Gogniat, G., Torres, L.: Security Trends for FPGAs, chap. 5, pp. 101–135. Springer, first edn. (2011)
2. Baudet, M., Lubicz, D., Micolod, J., Tassiaux, A.: On the security of oscillator-based random number generators. Journal of Cryptology 24, 1–28 (2010)
3. Bernard, F., Fischer, V., Valtchanov, B.: Mathematical Model of Physical RNGs Based on Coherent Sampling. Tatra Mt. Math. Publ. 45, 1–14 (2010)
4. Bochard, N., Bernard, F., Fischer, V., Valtchanov, B.: True-Randomness and Pseudorandomness in Ring Oscillator-Based True RandomNumber Generators. International Journal ofReconfigurable Computing, Article ID 879281 p. 13 (2010)
5. Bochard, N., Fischer, V.: A set of evaluation boards aimed at TRNG design evaluation and testing. Tech. rep., Laboratoire Hubert Curien, Saint-Etienne, France (March 2012), `http://www.cryptarchi.org`
6. Bucci, M., Luzzi, R.: Design of Testable Random Bit Generators. In: Cryptographic Hardware and Embedded Systems - CHES 2005. LNCS, vol. 3659, pp. 147–156. Edinburgh, UK, Springer Verlag (2005)
7. Danger, J.L., Guilley, S., Hoogvorst, P.: High Speed True Random Number Generator based on Open Loop Structures in FPGAs. Elsevier Microelectronics Journal 40(11), 1650–1656 (Nov 2009)
8. Dichtl, M., Golic, J.: High-Speed True Random Number Generation with Logic Gates Only. In: Cryptographic Hardware and Embedded Systems - CHES 2007. LNCS, vol. 4727, pp. 45–61. Vienna, Austria, Springer Verlag (2007)
9. FIPS, P.: 140-1: Security Requirements for Cryptographic Modules. National Institute of Standards and Technology 11 (1994)
10. Fischer, V., Drutarovsky, M.: True Random Number Generator Embedded in Reconfigurable Hardware. In: Cryptographic Hardware and Embedded Systems - CHES 2002. LNCS, vol. 2523, pp. 415–430. Redwood Shores, CA, USA, Springer Verlag (2002)
11. Güneysu, T.: True Random Number Generation in Block Memories of Reconfigurable Devices. In: Proc. Int. Conf. on Field-Programmable Technology – FPT 2010. pp. 200–207. IEEE (2010)
12. Gyorfi, T., Cret, O., Suciu, A.: High Performance True Random Number Generator Based on FPGA Block RAMs. In: Proc. Int. Symposium on Parallel and Distributed Processing. pp. 1–8. IEEE (2009)
13. Hajimiri, A., Lee, T.: A general theory of phase noise in electrical oscillators. Solid-State Circuits, IEEE Journal of 33(2), 179–194 (1998)
14. Holleman, J., Otis, B., Bridges, S., Mitros, A., Diorio, C.: A 2.92 muW Hardware Random Number Generator. IEEE Proceedings of ESSCIRC (2006)
15. Killmann, W., Schindler, W.: AIS 31: Functionality classes and evaluation methodology for true (physical) random number generators, version 3.1. Bundesamt fur Sicherheit in der Informationstechnik (BSI), Bonn (2001), `http://www.bsi.bund.de/zertifiz/zert/interpr/ais31e.pdf`

16. Killmann, W., Schindler, W.: A proposal for: Functionality classes for random number generators, version 2.0. Tech. rep., Bundesamt fur Sicherheit in der Informationstechnik (BSI), Bonn (September 2011), https://www.bsi.bund.de/EN/Home/home_node.html
17. Kohlbrenner, P., Gaj, K.: An Embedded True Random Number Generator for FPGAs. Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays pp. 71–78 (2004)
18. Majzoobi, M., Koushanfar, F., Devadas, S.: FPGA-Based True Random Number Generation Using Circuit Metastability with Adaptive Feedback Control. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2011. LNCS, vol. 6917, pp. 17–32. Nara, Japan, Springer Verlag (2011)
19. Marsaglia, G.: DIEHARD: Battery of Tests of Randomness. Online. Available at: http://stat.fsu.edu/pub/diehard/ (1996), http://stat.fsu.edu/pub/diehard/
20. Rukhin, A., Soto, J., Nechvatal, J., Smid, J., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S.: A statistical test suite for random and pseudorandom number generators for cryptographic applications, nist special publication 800-22. Online. Available at: http://csrc.nist.gov/ (2001), http://csrc.ncsl.nist.gov/publications/nistbul/html-archive/dec-00.html
21. Santoro, R., Sentieys, O., Roy, S.: On-line monitoring of random number generators for embedded security. Circuits and Systems, 2009. ISCAS 2009. Proceedings. IEEE International Symposium on (2009)
22. Simka, M., Drutarovsky, M., Fischer, V., Fayolle, J.: Model of a True Random Number Generator Aimed at Cryptographic Applications. Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on p. 4 (2006)
23. Sunar, B., Martin, W., Stinson, D.: A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks. IEEE TRANSACTIONS ON COMPUTERS pp. 109–119 (2007)
24. Tkacik, T.: A Hardware Random Number Generator. In: Cryptographic Hardware and Embedded Systems - CHES 2002. LNCS, vol. 2523, pp. 450–453. Redwood Shores, CA, USA, Springer Verlag (2003)
25. Valtchanov, B., Aubert, A., Bernard, F., Fischer, V.: Characterization of randomness sources in ring oscillator-based true random number generators in FPGAs. Design and Diagnostics of Electronic Circuits and Systems, 2008. DDECS 2010. 13th IEEE Workshop on pp. 1–6 (2010)
26. Valtchanov, B., Fischer, V., Aubert, A.: Enhanced TRNG Based on the Coherent Sampling. 2009 International Conference on Signals, Circuits and Systems (2009)
27. Varchola, M., Drutarovsky, M.: Embedded Platform for Automatic Testing and Optimizing of FPGA Based Cryptographic True Random Number Generators. RADIOENGINEERING 18(4), 631–638 (2009)
28. Varchola, M., Drutarovsky, M.: New High Entropy Element for FPGA Based True Random Number Generators. In: Mangard, S., Standaert, F.X. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2010. LNCS, vol. 6225, pp. 351–365. Springer (2010)
29. Veljkovic, F., Rozic, V., Verbauwhede, I.: Low-Cost Implementations of On-the-Fly Tests for Random Number Generators. In: Design, Automation, and Test in Europe – DATE 2012. EDAA (2012)
30. Wold, K., Tan, C.H.: Analysis and Enhancement of Random Number Generator in FPGA Based on Oscillator Rings. 2008 International Conference on Reconfigurable Computing and FPGAs pp. 385 – 390 (2008)