Attacking Hardware Random Number Generators in a Multi-Tenant Scenario

Yrjo Koyen, Adriaan Peetermans, Vladimir Rožić and Ingrid Verbauwhede
imec-COSIC, KU Leuven, Leuven, Belgium
firstname.lastname@esat.kuleuven.be

Abstract—**True random number generators are important building blocks for cryptographic systems and can be the target of adversaries that want to break cryptographic protocols by reducing the unpredictability of the used random numbers. This paper examines the viability of three different types of potential attacks on these generators when they are implemented on field programmable gate arrays, namely the voltage manipulation attack, the ring-oscillator locking attack and the replica observation attack. The proposed attacks only make use of the available programmable logic of the device and as such do not require physical access to it. They can technically be mounted remotely in a multi-tenant scenario by adversaries that only have bitstream write access to a part of the programmable logic. The attacks try to exploit interactions that can exist between an attack circuit and the targeted circuit because they reside on the same chip. The paper presents two case studies: an elementary ring oscillator design and a transition effect ring oscillator design. For the first case study, all three scenarios were tested and for the second case study, only the voltage manipulation attack scenario is examined. Our results show that this voltage manipulation attack is the most effective of the three proposed attacks.**

I. Introduction

Field-programmable gate arrays (FPGAs) are a popular platform for implementations of cryptographic systems due to their unique position on the performance-flexibility tradeoff. Cryptographic applications often rely heavily on the use of random numbers, for example as nonces, challenges or masks. In previous work, a variety of FPGA compliant true random number generator (TRNG) designs were presented, e. g. a simple ring-oscillator design by Baudet et al. [1], a metastability based TRNG by Majzoobi et al. [2], a multiple-ring oscillator based TRNG by Sunar et. al. [3] and a transition-effect ring-oscillator TRNG by Varchola et al. [4]. An overview and comparison of some designs is presented by Petura et al. in [5]. Adversarial attacks often target the weakest link in the system, which due to its high sensitivity to operating conditions, includes the on-chip TRNG circuit.

Various approaches have been successfully used to attack TRNGs in the past: in Cao et al. [6], the effects of variations in power supply voltage and ambient temperature on entropy source were observed. It was shown by Markettos and Moore in [7] that by injecting voltage oscillations in the power supply pin of a smart card, the produced entropy could be greatly reduced. Bayon et al. showed in [8] that this attack could also be done by only using contactless electromagnetic irradiation of the chip. The common difficulty of mounting any of these attacks is that they require close physical access to the target device.

With the emergence of FPGAs in the cloud, new scenarios become possible in which designs from multiple users can coexist within the same FPGA fabric and share its resources [9]. Such multi-tenant scenarios can introduce a new type of threat for security-sensitive systems: adversaries with access to a portion of the shared programmable logic fabric could remotely try to attack a system from inside the FPGA chip by constructing attacking circuits in proximity of its target. Whereas designs from different users should be logically isolated, some interactions are unavoidable due to the shared nature of the silicon substrate and power distribution network (PDN). Previous work has already exposed new threats in this multi-tenancy scenario. Gnad et al. [10] succeeded in crashing an FPGA by creating supply voltage drops, induced by a circuit with a high dynamic current consumption in the FPGA fabric. Such voltage drops were also used by Krautter et al. in [11] to mount a fault attack against an AES implementation and by Mahmoud and Stojilović in [12] to attack a self-timed ring TRNG by creating timing violations in the digitisation circuit. More passive attack scenarios are also available, as was shown by Acar and Ergun in [13]. Here, correlation between the output of two ring oscillator based TRNGs was observed when the lookup tables (LUTs) were shared between the oscillators.

In this work, we further explore the possibilities of attacking TRNGs in these multi-tenant scenarios. Three different attack scenarios: *voltage manipulation*, *ring oscillator locking*, and *observation* are investigated and the effectiveness on two TRNG designs is analysed. To the best of our knowledge, this is the first work presenting a successful attack on the entropy source of a TRNG circuit in a multi-tenant scenario.

The remainder of this paper is structured as follows. Section II introduces the threat model and presents the three types of attack scenarios together with the experimental setup used to evaluate them. Section III briefly presents the implementation of the first target TRNG: an elementary ring oscillator (ERO) TRNG and evaluates its vulnerability against the proposed attack scenarios. Section IV presents the implementation of the second target TRNG: a transition effect ring oscillator (TERO) TRNG and compares its vulnerability to the voltage manipulation attack with that of the ERO TRNG. Section V concludes this work and proposes future research directions.

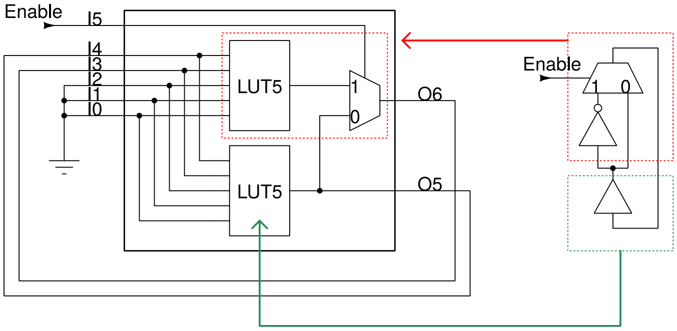

Fig. 1. Schematic of a ring oscillator (right) used in the supply voltage manipulating attack scenario and its mapping to a LUT6_2 (left).

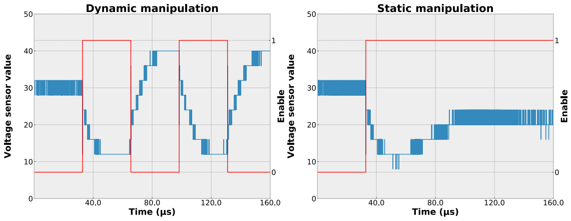

Fig. 2. Measured supply voltage (blue) and oscillator activation signal (red) for the the dynamic (left) and the static (right) supply voltage manipulation scenario both using 400 slices filled with oscillators.

II. Attack setup

A. Threat model

In this work, we assume the multi-tanant FPGA threat model proposed by Provelengios et al. [14]:

- Independent users are assigned parts of the FPGA fabric that share a common power distribution network (PDN).
- Users are logically isolated from each other, there is no user defined logic making a physical connection to circuitry defined by other users.
- Users do not have physical access to the FPGA device.
- No weakness is assumed in the software to design and interface with the FPGA.
- No restrictions apply to what the user can implement, other then what is physically possible in the FPGA fabric.

Note that for the attack scenarios carried out in this work, the adversary is assumed to have knowledge of the exact location of the victim TRNG circuit. Additionally, this adversary is able to place and route logic in arbitrarily close proximity of this victim TRNG circuit. While the second requirement is necessary for the attacks to work, the first one can be alleviated by trying out many locations and selecting one that results in a large attack response.

B. Scenario 1: Supply voltage manipulation

In the first attack scenario the local power supply voltage is manipulated, similar to an under-power attack. The goal is to degrade the quality of the generated random bits, not by causing timing violations, but by targeting the underlying entropy source. Equivalent to works of [10], [11] and [12], the attack circuit consists of a large amount of ring oscillators. These ring oscillators are implemented using only one 6-input, 2-output lookup table (LUT6_2). Fig. 1 depicts the schematic of such an oscillator and also illustrates its mapping to a LUT6_2.

The increased dynamic power consumption caused by activating all these oscillators at once, creates a localised voltage drop in the FPGA PDN. This drop in supply voltage is caused by the increase in current flow through the distributed inductance of the PDN [10].

To measure the effect of the attack circuit on the PDN voltage, we use a method proposed by Zick et al. in [15]. It comprises an on-chip voltage measurement sensor based on the voltage dependent propagation delay of a chain of logic gates. This voltage sensor produces measurement values between 0 and 63 representing lower and higher voltages respectively. The sensor is always implemented in close proximity of the attack circuitry.

In this work, we make a distinction between dynamic and static supply voltage manipulation. Both approaches use the same attack circuit, comprising of 400 slices filled with ring oscillators.

1) Dynamic supply voltage manipulation: The ring oscillators in the attack circuit are periodically activated at 15.24 kHz. This frequency induces a resonance effect in the PDN voltage. It will cause both drops and overshoots higher than the original power supply voltage, as can be seen in Fig. 2 (left).

2) Static supply voltage manipulation: If the attack ring oscillators remain active, the initial transient voltage drop disappears. However, the voltage remains lowered compared to the original supply voltage as shown in Fig. 2 (right). This effect remains as long as the attack ring oscillators stay activated. This effect is caused by the now stable increased current flow through the distributed resistance of the PDN.

C. Scenario 2: Locking

In the second attack scenario, the oscillating ring of the TRNG is locked to a signal in the neighbouring FPGA fabric. Locking the TRNG entropy source prevents random jitter from accumulating, thus reducing the entropy of the generated random bits. Mureddu et al. previously showed in [16] that different types of oscillating rings, used in TRNGs, can lock to a signal injected by other circuitry in the FPGA fabric. The proposed attack additionally generates the injected signal on-chip, which removes the requirement for any external equipment. In this work, the following two approaches to lock the TRNG entropy source are presented:

1) Identical ring oscillator approach: The first approach uses an attack circuit consisting of oscillating rings, implemented identically, and placed close to the target TRNG. Due to the identical design of the attack ring oscillator, the oscillation frequency should be tuned close to the oscillation frequency of the entropy source in order to increase the chance of locking.

2) Frequency matching approach: Since the lock-in range for a ring oscillator can be quite small, as was shown in [16], this second approach consists of injecting a signal with a frequency as close as possible to the natural frequency of the TRNG oscillator. The injected signal is generated by an oscillating ring that is identically designed to the targeted ring and placed at a carefully chosen location so that their frequencies are as similar as possible. Equivalent to the method used in [16], this injected oscillating signal is brought in close to the targeted ring using delay lines, implemented in proximity of the TRNG. This method provides more freedom to choose a suitable location for the attacker ring oscillator to generate the injected signal. This approach targets a specific implementation of a TRNG and requires the attacker to precisely know the frequency of the oscillating ring in the TRNG.

D. Scenario 3: Replica observation

In this final attack scenario, the adversary does not actively reduce the statistical quality of the bits produced by the target TRNG, instead it improves the chances of correctly predicting the TRNG output by implementing a replica observing TRNG circuit close to it. Due to interactions that can exist between the two TRNG circuits, the replica TRNG might be able to provide some additional information about the behaviour of the target TRNG. To verify whether there exists some dependency between the random bits generated by both TRNGs, the bits are combined by bitwise XORing and subsequently analysing the resulting bitstream for bias. A positive bias would indicate that it is more likely for a bit generated by the replica TRNG to be different from the one generated by the original TRNG whereas a negative bias would indicate that they are more likely to be equal.

E. Experimental setup

We used the *Zybo Zynq-7000* trainer board by *Digilent* containing a *Zynq XC7Z010* system on chip (SoC) as a target for all experiments. The circuits described in every attack scenario are accompanied by the experimental setup, shown in Fig. 3. In this setup, the generated random bits are temporarily stored in an on-chip block RAM memory before being sent to a computer via an UART transmitter. The block memory has a size of 64 KiB which allows gathering 2^{19} consecutive bits from the TRNG. The setup for the replica observation scenario additionally contains a second RAM block as it needs to collect data from two TRNGs simultaneously.

Since the experiments affect the operating conditions inside the FPGA, every experiment is preceded by an isolation test. This test verifies that the data gathering logic functions correctly under the attack conditions and that it doesn't corrupt the TRNG data during transmission.

III. Case study: ERO

Our first TRNG target is the ERO design proposed in [1]. This TRNG architecture is shown in Fig. 4. In this design an output signal of the jittery ring oscillator is sampled after a certain accumulation time to generate random bits. The implemented ERO contains a ring oscillator consisting of a NAND gate and two buffers. An additional buffer at the output of this oscillator allows for consistent routing of the internal signals. Each logic gate is implemented using one six-input LUT and the entire ERO TRNG fits exactly in one four-LUT slice. Frequency measurements of the ring oscillator for different locations within the FPGA show frequencies ranging from 1.023 GHz to 1.125 GHz.

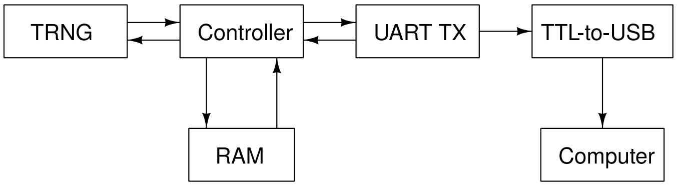

Fig. 3. Schematic of the experimental setup.

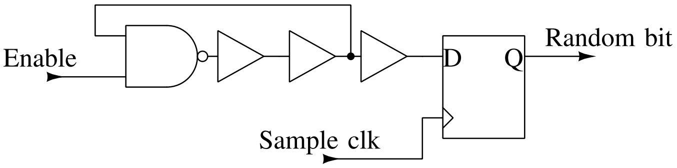

Fig. 4. Schematic of the ERO TRNG architecture.

We used the differential delay line methodology from [6] to measure the jitter strength of the ring oscillators and obtained a minimum jitter accumulation rate (σ^2/t) of 20.43 fs. This value represents the rate at which the variance (σ^2), of the random variable describing the duration until an edge occurs in the oscillator's signal, accumulates over the time t.

According to the statistical model in [1], a generator with these parameters and an accumulation time of 2^9 clock cycles at 125 MHz should provide 0.99992 bits of entropy per generated bit. Statistical testing of an actual implementation using this accumulation time showed that it produces bits with a bias larger than predicted by the model, which did not disappear when increasing the accumulation time. This bias is likely not due to a lack of accumulated jitter, but rather due to timing violations in the sampling flipflop. we target this implementation in order to verify whether the proposed attacks can influence the design in some way.

To evaluate the effectiveness of the attacks we look at the autocorrelation of non-overlapping segments of generated bits. This metric was proposed in [1] to detect a weakened behaviour of the ERO. It is calculated as

$$C_1 = \sum_{i=1}^{n-1} \frac{1}{n-1} (-1)^{b_i + b_{i+1}} ,$$

where n is the segment length and b_i is the i-th bit in the segment.

A. Scenario 1: Supply voltage manipulation

Both voltage manipulation experiments described in Sect. II-B were used to target the ERO design. The dynamic

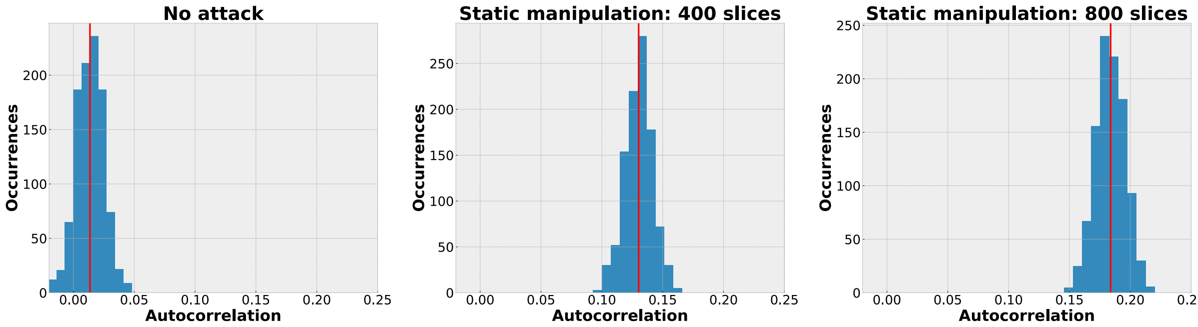

Fig. 5. Autocorrelation histograms for bitstream segments generated by the ERO under standard conditions (left) and when attacked using static voltage manipulation for different sizes of the attack circuit (centre and right). The vertical red lines indicate the measured average value.

supply voltage manipulation attack, for a range of different sizes of the attack circuit and for different activation frequencies, did not seem to have any measurable negative effect on the ERO design. This might be caused by the low bit rate of the ERO TRNG, as only a few consecutively generated bits are affected by a single voltage drop. The ERO generates a bit every 4.1 μs while the duration of the voltage drops, such as in the left part of Fig. 2, is only around 32 μs. This means that a single voltage drop can only affect up to 8 consecutive bits. Decreasing the toggling frequency below 16.6 kHz is not benificial in this experiment because, as shown in the right part of Fig. 2, the voltage already recovers partially after around 60 μs, causing a situation similar to that of the static supply voltage manipulation.

Even though the effect of the static manipulation on the supply voltage is smaller compared to that of the dynamic manipulation, the static manipulation does manage to affect the ERO design by increasing the autocorrelation of the generated bits. Fig. 5 illustrates this increase by showing autocorrelation histograms calculated from 1024 extracted bitstream segments of 8192 bits each, both under standard operating conditions, and during the static supply voltage manipulation attack. Additionally, two different sizes of the attack circuit have been tested: 400 and 800 slices. As can be seen from the figure, the average autocorrelation from both attacking circuit sizes shifts to a higher value during the attack, indicating that the neighbouring bits in a sequence are more likely to be equal than to be different. This increase is more pronounced for a larger attack circuit, as it induces a lower supply voltage. Using a circuit consisting of 800 slices (approximately 18% of the available slices) filled with oscillators, this attack was able to increase the average observed autocorrelation from 1.3% to 18.3%.

In order to make an estimate of the entropy loss, we first estimate the probability $p_i = Pr(X = i)$ of each byte value. Min-entropy per byte can then be estimated as:

$$H_\infty(X) = -log_2(\max_{0 \leq i \leq 255}(p_i)) \,. \tag{1}$$

The results show that the generated min entropy per byte drops from 7.1 for normal conditions to 6.07 and 5.48 during the attacks. Upon closer inspection of data under attack, we find that the generator is more likely to produce longer strings of consecutive zeros or ones. For example, a sequence of consecutive 9 zeros is twice as likely to appear in a data produced under attack. The longest run of consecutive zeros detected under normal operating conditions is 21, whereas this number grows to 33 under attack.

B. Scenario 2: Locking

The identical ring oscillator approach was evaluated by placing varying amounts of ring oscillators with an identical design as the target TRNG ring oscillator around it. None of the experiments using this approach showed any statistical effect on the generated bits. A possible explanation for this is that the frequencies of neighbouring oscillators, even though they have an identical design, are still not similar enough to achieve locking with the limited coupling that exists between them.

For the frequency matching approach, we were able to find a ring oscillator implementation providing a frequency very close to the oscillating frequency of the ERO TRNG. This approach affects the ERO TRNG, and its effect is visible as an increase in the autocorrelation of the generated bits. Fig. 6 illustrates this effect by showing autocorrelation histograms, calculated from 1024 bitstream segments of length 8192 bits each, collected from the ERO with surrounding delay lines. This approach was able to increase the average observed autocorrelation from 1.3% to 10.5% when the ERO was surrounded by 6 delay lines. The resulting min-entropy per byte is 6.21. Fig. 6 additionally shows that the effect decreases when too many delay lines are used. A possible explanation for this effect is that the injected signal can not arrive at all delay lines at the exact same time, which causes a phase difference between the signals and prevents the added delay lines from contributing constructively.

C. Scenario 3: Replica observation

In this attack scenario, we examine two identical TRNGs, implemented in close proximity of each other. Next to the two TRNGs, two delay lines are implemented. These delay lines are driven by an oscillator with similar frequency as the

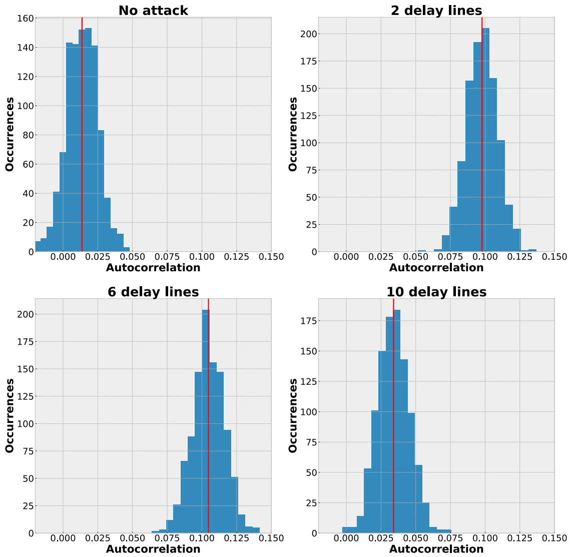

Fig. 6. Autocorrelation histograms for bitstream segments generated by the ERO under standard operating conditions (top left) and when attacked using the frequency matched locking approach for varying amounts of delay lines next to the ERO. The vertical red lines indicate the measured average value.

ERO rings. The purpose of these delay lines is to force both ring oscillators to lock, to increase the coupling between both TRNGs to potentially cause an increased dependency between their generated bits.

The produced bits of both TRNGs were analysed to examine whether the replica can leak some information about the original TRNG. However, analysing the XORed bitstreams did not reveal any bias. Consequently, the results of this experiment do not allow to conclude that observing an ERO, using nearby attacker controlled logic, can increase the probability of correctly guessing its output.

IV. Case study: TERO

In the second case study we attempt to attack a TERO design, originally proposed in [4], using the supply voltage manipulation attack scenario. The TERO architecture is depicted in Fig. 7. It contains a bi-stable loop that can be triggered to oscillate temporarily by toggling its control signal. The amount of oscillations that occur before it reaches a stable state again is unpredictable, as it is influenced by noise in its logic gates. The least significant bit of this oscillator count is a random bit.

The TERO implementation targeted here consists of two branches, each with one XOR gate, one AND gate and six buffers each. Each logic gate is manually mapped to individual LUTs in the FPGA and the interconnections between them are routed as symmetrically as possible. Multiple placement locations of the TERO TRNG were evaluated based on the distribution of oscillation counts and by testing with the NIST SP 800-22 [17] statistical test suite. Some locations that passed these tests were subsequently subjected to the voltage manipulation attacks.

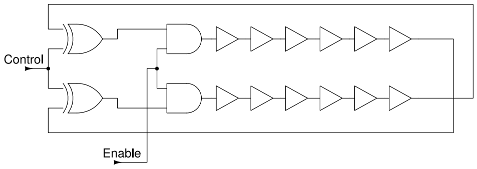

Fig. 7. Schematic of the TERO TRNG architecture

A. Scenario 1: Supply voltage manipulation

Both the dynamic and static voltage manipulation attack approaches were used to target the TERO TRNG, and both were able to alter its oscillation count distribution, as illustrated in Fig. 8. The static supply voltage manipulation primarily causes a shift of the distribution to a different average value. On the other hand, the dynamic supply voltage manipulation results in a bimodal distribution with one mode corresponding to voltage drops and the other one corresponding to voltage overshoots. Although both attack approaches clearly affect the oscillation count distribution, we were not able to measure a negative effect on the statistical quality of the generated TERO TRNG bits.

The next experiment investigates if the voltage manipulation attacks can have an effect when applied to a TERO TRNG implementation with already weakened characteristics. This experiment targets an implementation with a narrow oscillation count distribution centred around a low average that generates bits that barely pass statistical tests under standard operating conditions. The analysis of the bitstreams produced by this implementation when applying the dynamic supply voltage manipulation attack only looks at the bits generated during voltage minima, as they seem to have degraded the most. Table I contains the results of analysing these bits using a selection of tests from the NIST test suite that have a recommended sequence length which is not too long. These results show that the manipulated supply voltage indeed has an effect on the generated bits of a poorly implemented TERO TRNG. Upon closer inspection of data under attack, we find that the generator produces uniformly distributed bits most of the time, with occasional bursts of consecutive ones or zeros lasting between 16 and 32 bits. Approximately 1% of the data consists of these bursts.

As stated by [5] and confirmed in our own experiments, most placement locations on FPGA produce a low quality TERO TRNG. Only a search procedure over the FPGA area can produce a well functioning TERO TRNG. Designers should therefore be careful, and examine the oscillation count distribution, when selecting a location for a TERO TRNG

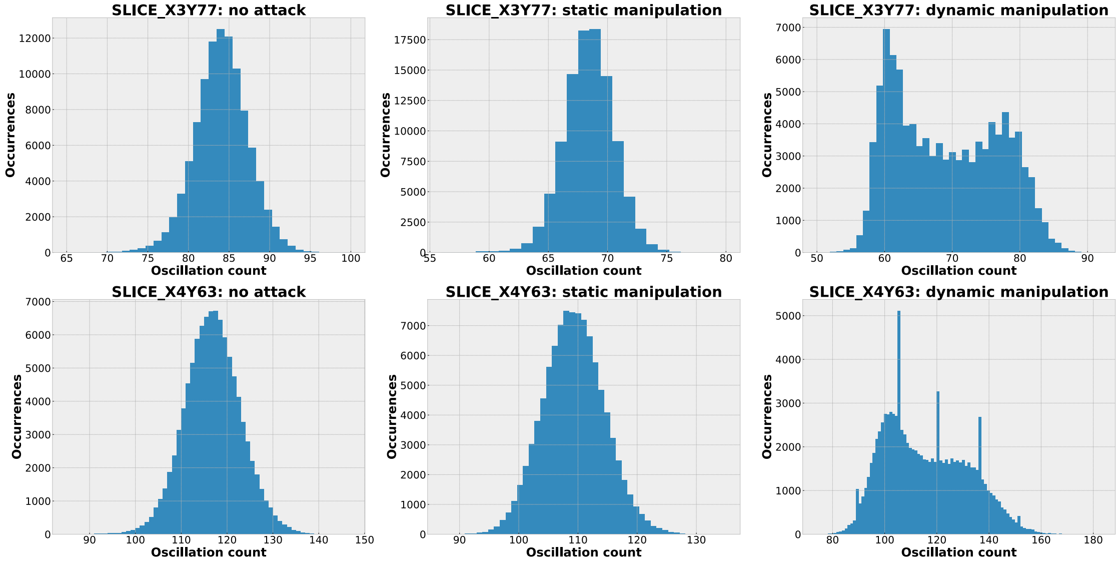

Fig. 8. Oscillation count histograms constructed from 100 000 measurements for a TERO TRNG implemented at two different locations, under standard operating conditions (left) under the static supply voltage manipulation approach (centre) and under the dynamic supply voltage manipulation approach (right).

TABLE I
STATISTICAL TEST PASS RATE FOR 100 SEQUENCES OF 80 000 BITS COLLECTED FROM A WEAK TERO LOCATION.

Test name	Pass rate standard	Pass rate attack
Frequency	97/100	97/100
Block frequency (M = 128)	98/100	3/100
Cumulative sums	97/100	95/100
Runs	93/100	1/100
Longest run	100/100	58/100
Rank	98/100	100/100
FFT	99/100	97/100
Serial (m = 14)	99/100	0/100

implementation that should be resilient against this type of attack.

V. Conclusion and Future Work

This paper proposed three different types of scenarios to attack TRNG entropy sources in a multi-tenant FPGA scenario. The effectiveness of each attack on the two TRNG designs under study is summarised in Table II. Two of these scenarios showed a significant reduction of randomness when evaluated in the presented case studies. The supply voltage manipulation attempts were able to negatively influence both the ERO TRNG and, to some extent, the TERO TRNG designs. Autocorrelation of the generated bits increased for the ERO TRNG and caused the TERO TRNG to fail statistical testing, when the designer is careless in selecting the TRNG location on the FPGA. The ERO TRNG proved to be more vulnerable to the static supply voltage manipulation whereas the TERO TRNG proved to be more vulnerable to the dynamic supply voltage manipulation. The attempted locking attack was only able to influence the ERO TRNG when the injected signal's frequency was extremely close to the natural frequency of the ERO TRNG ring oscillator. Finally, the replica observation experiment was not able to indicate a possibility to increase the probability of correctly predicting the generated bits of an ERO TRNG design, using an identically implemented replica TRNG next to it. These results confirm that attacks mounted entirely from within the FPGA itself can have negative effects on the behaviour of both the ERO TRNG and the TERO TRNG architecture.

TABLE II
ATTACK SCENARIO EFFECTIVENESS SUMMARY.

Architecture	Scenario 1 effective	Scenario 2 effective	Scenario 3 effective
ERO TRNG	Yes	Yes	No
TERO TRNG	Yes	-	-

This work proposed several attack scenarios that may be considered as unrealistic in real world multi user FPGA cloud computing applications, e.g. *Microsoft Azure* or *AWS FPGA Cloud service*. Currently, these services prohibit multiple users to be assigned to the same physical FPGA chip, which prevents the proposed attacks from being deployed. However, when the available logic in these cloud computing FPGA devices scales up in the future, this scenario might become available to improve FPGA logic utilisation [18]. This work should therefore serve as a cautionary note to future multi tenancy cloud FPGA systems. To accommodate this, we assembled a list of good practices when designing these systems:

- Physical separation between different users: the proposed attack scenarios require close proximity of adversary and victim to be successful. Users should therefore be physically separated to reduce the attack effectiveness.
- PDN sharing: FPGA devices tailored for use in multi tenancy scenarios should accommodate separate PDN islands for each user, but at the same time allow one design to make use of several PDN islands. This will reduce electromagnetic coupling between designs of different users.
- Attack injection via the substrate: as the silicon substrate is shared by multiple users, noise can be injected by an adversary and influence victim circuitry. FPGA devices tailored for use in multi tenancy scenarios should allow for substrate isolation between different users to avoid this noise injection.
- Design restrictions: several FPGA cloud providers (e.g. *AWS FPGA Cloud service*) prohibit the use of combinatorial loops. This prevents the creation of ring oscillators and therefore also prevents the proposed attack scenarios from being deployed. However, this restriction also makes it impossible to implement jitter based entropy sources for use in TRNGs. Furthermore, it has been shown in [19] that even with this restriction, ring oscillators can still be constructed by introducing a transparent latch in the ring. The recommendation is not to restrict the submitted user designs, as this will harm applications with a legitimate use of combinatorial loops and does not prohibit malicious ring oscillators from being constructed.
- Attack detection: when designing security critical circuits on these multi tenancy systems, the designer should incorporate sensor structures to detect potential incoming attacks. An example of such a sensor is the on-chip voltage monitor, introduced by [15].
- Bitstream scanning: a tool to scan the FPGA configuration bitstream for malicious circuits is available [20]. Among others, this tool can detect large fan-out gates, which can indicate a large aray of ring oscillators being constructed.

A possible subject of future research could be to perform the attack scenarios on other TRNG designs as well. In particular, the voltage manipulation attack might be viable to attack other designs that are known to be sensitive to under-powering.

Acknowledgment

This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work is supported in part by the Hercules Foundation AKUL/11/19, and by the European Commission through the Horizon 2020 research and innovation programme Cathedral ERC Advanced Grant 695305. Adriaan Peetermans is funded by an FWO fellowship and Vladimir Rožić is an FWO post-doctoral fellow under grant number 12Y4919N.

References

[1] M. Baudet, D. Lubicz, J. Micolod, and A. Tassiaux, "On the security of oscillator-based random number generators," *Journal of Cryptology*, vol. 24, pp. 398–425, 04 2011.

[2] M. Majzoobi, F. Koushanfar, and S. Devadas, "Fpga-based true random number generation using circuit metastability with adaptive feedback control," in *International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 6917, 2011, pp. 17–32.

[3] B. Sunar, W. Martin, and D. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 109–119, 2007.

[4] M. Varchola and M. Drutarovsky, "New high entropy element for fpga based true random number generators," in *International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 6225, 2010, pp. 351–365.

[5] O. Petura, U. Mureddu, N. Bochard, V. Fischer, and L. Bossuet, "A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices," in *26th International Conference on Field - Programmable Logic and Applications* , ser. 26th International Conference on Field - Programmable Logic and Applications, Lausanne, Switzerland, Aug 2016, pp. 1 – 10. [Online]. Available: https://hal-ujm.archives-ouvertes.fr/ujm-01570124

[6] Y. Cao, V. Rozic, B. Yang, J. Balasch, and I. Verbauwhede, "Exploring active manipulation attacks on the tero random number generator," in *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, vol. 0. IEEE, 2016, pp. 1–4.

[7] A. Markettos and S. Moore, "The frequency injection attack on ring-oscillator-based true random number generators," in *International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 5747, 2009, pp. 317–331.

[8] P. Bayon, L. Bossuet, A. Aubert, V. Fischer, F. Poucheret, B. Robisson, and P. Maurine, "Contactless electromagnetic active attack on ring oscillator based true random number generator," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2012, pp. 151–166.

[9] A. Khawaja, J. Landgraf, R. Prakash, M. Wei, E. Schkufza, and C. J. Rossbach, "Sharing, protection, and compatibility for reconfigurable fabric with amorphos," in *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, 2018, pp. 107–127.

[10] D. Gnad, F. Oboril, and M. Tahoori, "Voltage drop-based fault attacks on fpgas using valid bitstreams," in *2017 27th International Conference on Field Programmable Logic and Applications, FPL 2017*. Institute of Electrical and Electronics Engineers Inc., 2017.

[11] J. Krautter, D. R. E. Gnad, and M. B. Tahoori, "Fpgahammer: Remote voltage fault attacks on shared fpgas, suitable for dfa on aes," *Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 3, 2018. [Online]. Available: https://doaj.org/article/c10a099e833d4304933b6fceeffbe8a6

[12] D. Mahmoud and M. Stojilović, "Timing violation induced faults in multi-tenant fpgas," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2019, pp. 1745–1750.

[13] B. Acar and S. Ergun, "Correlation-based cryptanalysis of a ring oscillator based random number generator," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2018, pp. 1050–1053.

[14] G. Provelengios, D. Holcomb, and R. Tessier, "Characterizing power distribution attacks in multi-user fpga environments," in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2019, pp. 194–201.

[15] K. Zick, M. Srivastav, W. Zhang, and M. French, "Sensing nanosecond-scale voltage attacks and natural transients in fpgas," in *Proceedings of the ACM/SIGDA international symposium on field programmable gate arrays*, ser. FPGA '13. ACM, 2013, pp. 101–104.

[16] U. Mureddu, N. Bochard, L. Bossuet, and V. Fischer, "Experimental study of locking phenomena on oscillating rings implemented in logic devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 7, pp. 2560–2571, 2019.

[17] A. Rukhin, J. Soto, and J. Nechvatal, *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, ser. Computer security. Gaithersburg: United States department of commerce. National institute of standards and technology, 2000.

[18] A. Vaishnav, K. D. Pham, D. Koch, and J. Garside, “Resource elastic virtualization for fpgas using opencl,” in *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2018, pp. 111–1117.

[19] I. Giechaskiel, K. B. Rasmussen, and J. Szefer, “Measuring long wire leakage with ring oscillators in cloud fpgas,” in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2019, pp. 45–50.

[20] K. Matas, T. La, N. Grunchevski, K. Pham, and D. Koch, “Invited tutorial: Fpga hardware security for datacenters and beyond,” in *The 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2020, pp. 11–20.