SPICE Simulation of a "Provably Secure" True Random Number Generator

Markus Dichtl, Bernd Meyer, Herrmann Seuschek

Siemens Corporate Technology, Munich, Germany

Abstract. In their paper "A Provably Secure True Random Number Generator with Built-in Tolerance to Active Attacks", B. Sunar, W. Martin, and D. Stinson propose a design for a true random number generator. Using SPICE simulation we study the behaviour of their random number generator and show that practical implementations result in a too high frequency signal to be processed with current CMOS technology.

Key words: Random number generation, provable security, SPICE simulation.

1 Introduction

In [6], Berk Sunar, William Martin, and Douglas Stinson presented a new design for a true random number generator (TRNG) and claimed it to be provably secure.

The suggested TRNG consists of a large number of ring oscillators (ROs) of equal lengths whose outputs are XORed together. The output of the XOR is sampled periodically. The sampled bits are then post-processed by a resilient function to produce the output bits.

In [2] Markus Dichtl and Jovan Golić pointed out that the security proof is based on several highly unrealistic assumptions and hence is not relevant for practical implementations of the TRNG design.

This paper uses SPICE simulation to elaborate further on one of the objections of [2], namely that the XOR of the many RO outputs is impossible to compute as it would result in a too high frequency signal to be processed with current technology.

This is not the first paper using SPICE simulation to study the TRNG design of [6], in [9] the XOR of four RO outputs was studied. However, these results on the XOR do not allow conclusions on practical implementations of the TRNG design suggested in [6] with many more RO outputs XORed. There, a sample design with parameters claimed to be realistic is suggested. This sample design uses 114 ROs of length 13.

2 Tools and Parameters Used for the SPICE Simulation

To make the simulation results easily reproducible, we used the freely available SPICE simulation tool LTspice, which is part of SwitcherCAD III by Linear Technology [3].

The parameters for the CMOS transistors were taken from the 250 nm library described in [4] which is available online [5]. For NMOS transistors, the parameters L = 250 nm and W = 375 nm were used, for PMOS the parameters were L = 250 nm and W = 1125 nm . The supply voltage for the simulations was 2.5 V as suggested in [4].

Now a 250 nmCMOS technology may be considered quite outdated by some, but the results of our paper do not depend too much on which generation of CMOS technology is used. Of course smaller transistors are faster, but as both the ring oscillators producing the signals and the XOR tree processing the signals get faster on the same scale, the principal behaviour remains the same. We verified this by using a 250 nm CMOS SPICE library, which lead to similar results.

3 How the XOR Tree was Simulated

The XOR gates for the SPICE simulation used the standard 12 transistor CMOS design for XOR gates [1]. The 113 two input XOR gates for the computation of the XOR of 114 RO outputs were arranged in a balanced tree.

As the SPICE simulation of the 114 ring oscillators with 13 inverters each, which results in 2964 transistors, is very time consuming, a different approach was taken. Another drawback of simulating the 114 ring oscillators would be that all have the same frequency and phase. So instead of simulating all the 114 ring oscillator, we simulated only one in a first step. A RO of length 13 resulted in an oscillation frequency of 1.275 GHz. In practical implementations, not all ROs of length 13 will have exactly the same frequency. We modelled this by assuming that the frequencies of the 114 ROs are normally distributed around the mean frequency of 1.275 GHz with a standard deviation of 127.5 MHz. In the SPICE simulation, we used 114 voltage sources with frequencies determined in this way and uniformly distributed initial phases. The sweep times of the voltage sources between 0 V and 2.5 V were 10 ps.

4 Simulation Results for the Sample Design

Figure 1 shows a small section of the SPICE simulation result of the sample design of [6]. In total, 100 ns were simulated. In these 100 ns we observed 359 transitions of the XOR value through the voltage level 2.5 V. According to the model of [6], there should be 29070 transitions within 100 ns. This means that only 1.2 percent of the transitions of the ring oscillators appear at the output of the XOR tree. The theoretical model of [6] assumes that all of them are available for sampling at the XOR output. Hence, the model is inadequate for describing the behaviour of the sample design.

What happened to the remaining 98.8 percent of the ring oscillator transitions? Somewhere on their way through the XOR tree, they came too near to another transition. Therefore, these transitions cancelled out.

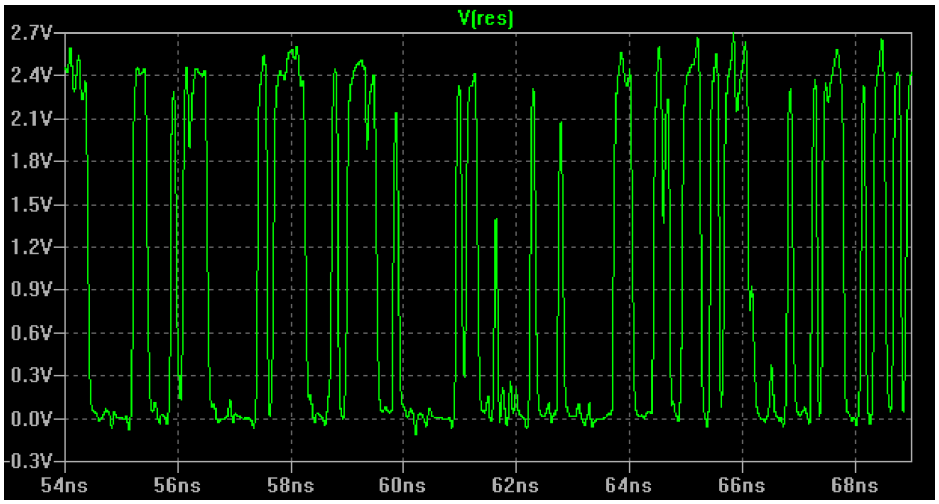

Fig. 1. XOR of the 114 ring oscillators of length 13 in the SPICE simulation. According to the theoretical model of [6], there should be 4360 transitions through the voltage level 1.25 V within the 15 ns shown, but only a very small fraction of them occurs in the SPICE simulation.

As 98.8 percent of the ring oscillator transitions are lost in the XOR tree, using 4360 transistors in the 114 ring oscillators is a waste of resources. Just two ring oscillators of length 13 with a total of 52 transistors would result in more exploitable jitter events at the output of the XOR than the 114 ROs.

In [6] an urn model is used to analyze the distribution of jitter events. For the sample design, the RO period is split up into 100 urns and it is claimed that 60 percent of these urns contain a transition. In [9], it is claimed that lost signals in the XOR tree can reduce this filling rate at most by a factor of two. The argument given there is that an even number of hits in an urn carries a risk of cancellation, whereas an odd number of hits in an urn is claimed to result in a remaining jitter event in this urn. However, this argument is not correct, the XOR tree hardware is not able to determine the parity of numbers of transitions in an urn. According to this incorrect claim, the filling rate of the urns of the sample design of [6] could drop to 30 percent in the worst case of signal cancellation in the XOR tree. The results of our SPICE simulation show that the filling rate falls to a value below two percent, clearly falsifying the claim from [9].

5 Suggestions to Overcome the Problems of the XOR Tree

In [7] B. Sunar comments on the speed problem of the XOR tree pointed out in [2]. He suggests solving it by reducing the sampling rate. This approach can not work. However low one chooses the sampling rate, the filling rate of the sampled urn remains below the two percent level. According to the urn model of [6], reducing the sampling rate has no effect whatsoever.

In [9] two obvious ways to overcome the speed problems of the XOR tree are suggested, namely to lower the frequencies of the ROs or to reduce the number of ROs. In the next section we will investigate how low the frequencies of the ROs must be made in order to have an acceptably low rate of signals cancelled out in the XOR tree.

6 How Slow Must the Design be Made for the XOR Tree to Work Properly?

When we reduce the oscillation frequencies of the 114 ROs by a factor of 1000 by building each RO from 13000 inverters instead of 13, still some of the transitions get lost in the XOR tree. The SPICE simulation resulted in 7.8 percent of the RO transitions getting lost in the XOR tree. This means that, according to the urn model, the resilient postprocessing function suggested in [6] has more corrupted input bits. In [6] it is assumed that 103 of the 256 input bits of the resilient function are deterministic. But in the simulation 7.8 percent, that is 11.9 transitions of the 156 input bits considered valid in [6] get lost in the XOR tree. Now the sample design of [6] allows some of the 156 bits to be corrupted in order to thwart active attacks. However, only up to 9 bits may be corrupted, whereas 11.9 bits get lost in the XOR tree. This means that even slowing down the ROs by a factor of 1000 is not sufficient to overcome the speed problems of the XOR tree.

As a consequence, the lengths of the ROs of the sample design from [6] must be increased by a factor of more than 1000 in order to overcome the speed problems of the XOR tree. The increased lengths of the 114 ROs lead to a design which uses almost three million transistors. This does not compare too favourably with the 70 transistors in the metastable ring oscillator of the recent random number generator design [8].

But the speed of random bit generation should also be considered. In [6] it is suggested to sample once per ring oscillator period. For the parameters of the SPICE model we used, this means a random bit output frequency below 70.5 kHz. However, the urn model would also allow sampling not only one urn per period, but all of them, which would increase the output rate by a factor of 100.

7 Conclusion

We have shown that the sample design for a true random number generator from [6], whose parameters are claimed to be realistic, leads to signal rates three orders of magnitude too fast to be processed in current CMOS technology. Contrary to the claim, the choice of parameters has turned out to be highly unrealistic. Using 114 ring oscillators as in the sample design turned out to be a waste of resources, using only two ring oscillators would result in more randomness in the output.

The claim of [9], that in the urn model signal loss in the XOR tree can at most reduce the urn fill rate by a factor of two, was also falsified by the SPICE simulation. The factor of loss in the urn fill rate was higher than 30.

The speed problems of the XOR tree shown in this paper as well as the additional problems pointed out in [2] make it clear that the true random number generator from [6] is not well suited for practical implementations.

References

1. R. J. Baker, H. W. Li, and D. E. Boyce, "CMOS, Circuit Design , Layout, and Simulation", IEEE Press, 1997
2. M. Dichtl and J. Golić, "High-Speed True Random Number Generation with Logic Gates Only", Cryptographic Hardware and Embedded Systems - CHES 2007, *Lecture Notes in Computer Science*, vol. 4727, pp. 45-62, 2005
3. Linear Technology, `http://ltspice.linear.com/software`
4. J. Rabaey, A. Chandrakasan, and B. Nilolić, "Digital Integrated Circuits, A Design Persective, Second Edition", Prentice Hall, 2003
5. J. Rabaey, A. Chandrakasan, and B. Nilolić, `http://bwrc.eecs.berkeley.edu/IcBook`
6. B. Sunar, W. Martin, and D. Stinson, "A Provably Secure True Random Number Generator with Built-in Tolerance to Active Attacks", *IEEE Trans. Computers*, vol. 56(1), pp. 109–119, Jan. 2007
7. B. Sunar, "Response to Dichtl's Criticism", `http://ece.wpi.edu/~sunar/preprints/comment.pdf`
8. I. Vasyltsov, E. Hambardzumyan, Y. Kim, B. Karpinskyy, "Fast Digital TRNG Based on Metastable Ring Oscillator", Preprint, Accepted for CHES 2008
9. S. Yoo, B. Sunar, D. Karakoyunlu, B. Birand, "A Robust and Practical Random Number Generator", `http://ece.wpi.edu/~sunar/preprints/rings.pdf`