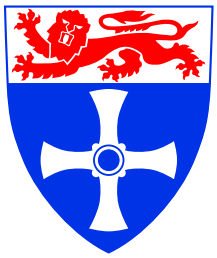
School of Electrical, Electronic & Computer Engineering

UNIVERSITY OF
NEWCASTLE UPON TYNE



Securing On-Chip Operations against Timing Attacks

C.A. Hoggins, C. D'Alessandro, D.J. Kinniment and A. Yakovlev

Technical Report Series

NCL-EECE-MSD-TR-2005-108

September 2005

Contact:

c.a.hoggins@ncl.ac.uk

crescenzo.d'alessandro@ncl.ac.uk

david.kinniment@ncl.ac.uk

alex.yakovlev@ncl.ac.uk

Supported by EPSRC grant GR/S81421

NCL-EECE-MSD-TR-2005-108

School of Electrical, Electronic & Computer Engineering,
Merz Court,
University of Newcastle upon Tyne,
Newcastle upon Tyne, NE1 7RU, UK

http://async.org.uk/

Securing On-Chip Operations against Timing Attacks

C.A. Hoggins, C. D'Alessandro, D.J. Kinniment and A. Yakovlev

September 2005

Abstract

Cryptosystem designers frequently assume that secrets will be manipulated in closed, reliable computing environments. Unfortunately, actual microchips leak information about the operations they process. Cryptosystems often take slightly different amounts of time to process different inputs and also internal signals.

By carefully measuring the amount of time required to perform private key operations, attackers may be able to discover secret exponents or keys to break the cryptosystem [2, 4]. A technique to hide the time differences from attackers using random delays is shown in this report.

1 Introduction

Secure encryption is dependent on having sources of cryptographically secure random numbers for key generation. The use of pseudo-random numbers is common, but this makes a system more susceptible to attack as the number of states which need to be searched is reduced [4].

For high-speed cryptosystems, random numbers need to be generated at a high enough rate so that reuse of random numbers is not possible. Most hardware random number generators depend primarily on a source of external thermal noise which is amplified in some manner. These types of generators are vulnerable to attacks which can alter the distribution of the output, therefore post-processing hardware is needed to continuously monitor the quality of the output, which in turn can affect the rate at which random numbers can be produced.

An alternative method of random number generation is to use a bistable device in metastability, shown in section 2 of this paper. This approach is capable of producing random bits at 100MHz, and is simple to integrate. It has been shown that the final state, of two which are equally likely, of a bistable device in metastability is affected by noise within the device. If this noise is caused by thermal motion, then it will be of a random nature. Therefore, it follows that by continuously clocking a bistable device in metastability, a sequence of bits, determined by the polarity of the noise at the time will be produced [3]. One problem of using this method to create random bits is how to ensure that the offset of the initial bias point is smaller than the noise level in the bistable. If the offset is larger than the noise level, the output will be dependant on this, and not the noise, thus removing the random nature of the output.

To use the random numbers produced to vary the timing of a signal, a variable delay element (VDE) is needed. These variable delay elements can be implemented in several ways, as shown in section 3 of this paper. These variable delay elements could be used to add a random delay to secure signals in several ways, either through adding a small random delay to the signal after processing, thus removing the timing

difference information which can provide information about the value of, and operations performed on a signal. Alternatively, by modulating several levels of a secure circuit with a random clock signal, timing information.

Other techniques such as dual-rail encoding, return-to-spacer protocols and energy balancing [8] could be used to further improve security in conjunction with this approach, which could therefore limit the possibilities of both timing and power analysis attacks.

2 Random Number Generator

The bistable device used is the same as one used by Kinniment [3], shown in Figure 1.

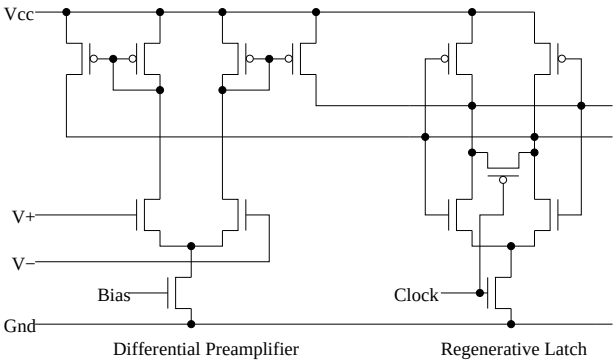


Figure 1: Bistable Device consisting of a Differential Preamplifier and Regenerative Latch

This device is composed of a differential preamplifier and a regenerative latch [5]. The analogue input stage drives the bistable with a small current difference, which affects the polarity of the output. This allows the distribution of the output values of the generator to be post-processed, and provide feedback to the analogue input stage.

With the device constructed by Kinniment [3] in a AMS 0.6μM process, the noise measured in the bistable when it is in metastability was approximately 1mV. Therefore, to take advantage of the random nature of this noise, the initial offset must be controlled to lower than 1mV. To accomplish this and also to ensure equal distribution of output values, some post-processing of the output is necessary so that feedback can be provided. This can be accomplished in several ways, such as the switched capacitor network as used by Kinniment [3], or by a more simple digital integrator circuit built of a counter with up/down inputs controlled by the output of the RNG.

To convert the digital signal from the counter, an analogue to digital converter is necessary. To provide the correct offsets for the differential preamplifier, scaled inverters are used. The circuit shown in figure 2 gives a range of values for the offset varying between 4.3μV and 28mV. These values depend on the sizing of the transistors used, and can be varied depending on the noise levels present in the bistable device due to varying manufacturing processes and layout. It can be seen that the counter must be initialised with its most significant bit being high and all others low, so that the minimum offset is provided at the beginning of operation.

The offsets to the differential preamplifier are varied as the counter counts up and down, which is dependent on the current output of the RNG. Therefore, if several outputs are high in sequence, then the

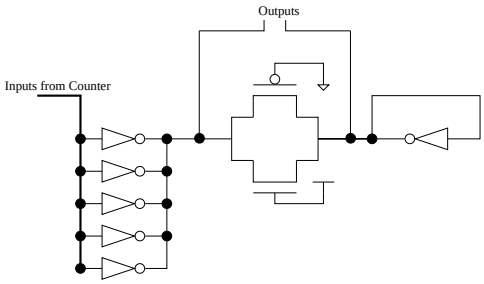


Figure 2: Circuit to create variable offset values

counter will count down, reducing the offset, and making a low output more likely, and vice-versa. This should ensure that the distribution of the output is fairly equal. The full circuit is shown in figure 3.

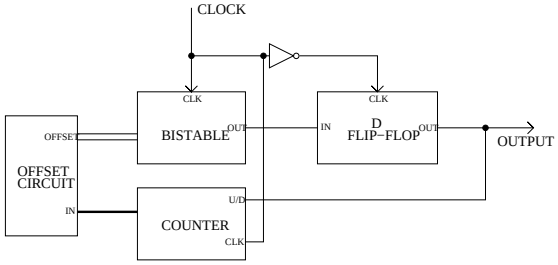


Figure 3: Random Number Generator Circuit

This random number generator is capable of providing sequential random bits at a high rate. To convert these random bits into random numbers, either several RNGs must be used in parallel, or a SIPO device is needed. Both of these approaches are capable of providing random numbers which are valid for a fixed amount of time, and then updated. To provide random numbers which are valid for a random amount of time, the variable delay elements described in section 3 can be used in conjunction with the RNGs. A cascaded setup, and the proposed output of such a circuit is shown in figure 4, where the time δt is random. This approach could of course be extended to include more levels.

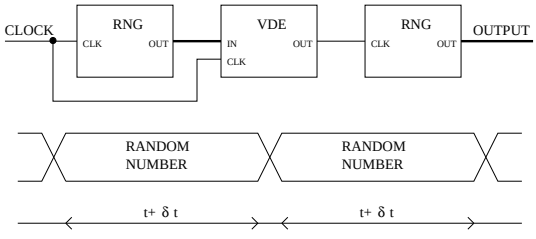


Figure 4: Cascaded RNG, and the proposed output of such a configuration

3 Variable Delay Elements

There are several ways to vary the delay of a signal. Perhaps the most simple way is to provide two different paths which a signal could take, one with a delay, and one without. A select signal could then be used to multiplex which path the signal should take. This approach is a good way of providing coarse delays to signals, but if many levels of these delay elements are needed then the delay can become very large and therefore adversely affect the timing, and possibly operation, of a circuit.

An alternative approach is to vary the dynamic characteristics of an inverter to provide differing delay to a signal [1, 6], by using an array of transistors above (or below) an inverter, as shown in figure 5(a). By using this approach, which effectively controls the delay by means of supply voltage control, the falling edge (or rising edge for figure 5(b)), is subject to a varying delay which is altered by the control signals providing inputs to the transistors. One transistor in the array is always on to provide a minimum/maximum delay for the element, according to the design.

To provide equal delays to both rising and falling edge of the input signal, arrays of transistors could be used both above and below the inverter, but this creates problems as the P and N type transistors have to matched, and also the control signals have to be inverted. Another option is to simply use two of the VDEs with the arrays of transistors either above or below chained together (without the output inverters), as shown in figure 5(c). This would ensure that both the rising and falling edge of the input signal pick up equal delays. This approach is able to provide much finer delays than the multiplexed option, but to provide a linear reaction to the control signals, the arrays of transistors must be scaled precisely. In this particular application, the non-linear reaction to the control signals, or imprecise scaling of the transistors, should not be a problem as the delays needed are of a random nature anyway.

4 Securing On-Chip Operations

The use of variable delay elements could help to disguise the time taken for a cryptosystem to process different signals, and therefore help disguise the data values and operations associated with this timing information. It is possible to do this in several ways, either adding a random delay to a signal after an operation has been completed, thus removing the timing information associated with the operation that has been performed, or by modulating a random clock signal with several layers of a circuit. Of course, some systems may have critical timing requirements, and therefore where delays can be used is design specific.

In some synchronous systems, or systems where timing requirements are critical, it may be possible to translate the circuit into a self-timed design. This could have several advantages, such as being able to implement other security approaches such as dual-rail encoding of signals, which could in turn lead to return-to-spacer protocols or energy balancing techniques being implemented [8, 7]. This could potentially help to secure the system against power analysis attacks as well as timing attacks.

Figure 6 shows how several layers of a circuit can be modulated with a random clock signal. This kind of approach could be useful for the generation or verification of keys, where keeping the key information secret is vital for the overall security of the system. When using a system such as this, one must ensure that the delay provided by the VDEs are at least as long as the time it takes for a circuit level to evaluate, to ensure correct operation.

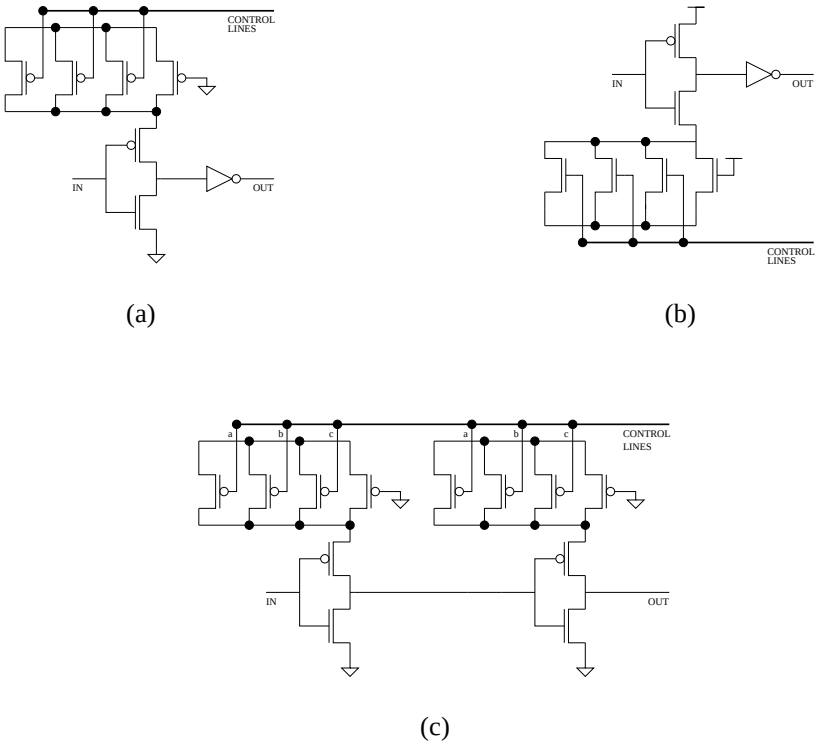


Figure 5: (a) VDE with N-type transistor array, (b) VDE with P-type transistor array, (c) VDE with equal delay for rising and falling signal

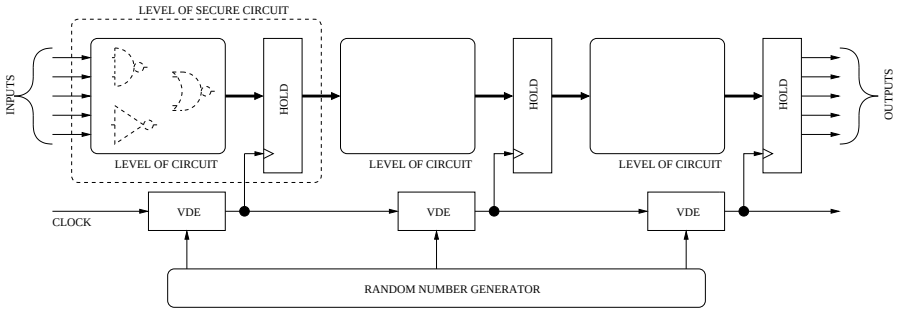


Figure 6: Levels of secure circuit modulated using random clock signal

5 Simulated Results

Although the system has not as yet been tested on actual cryptographic hardware, a simple verilog block capable of performing several operations was used to simulate such a system. These operations take different times to complete, depending on the actual operation, the values of the inputs and the value of the output, as in most systems.The set up was similar to that shown in figure 6, where a D-type flip-flop was used as the hold element and the unsecured circuit element implemented was used as the 'level of circuit'. The clock input to the circuit was delayed twice, once for the ma ximum amount of time that it could take the output of the verilog block to evaluate, and then an additional small random delay. The input and

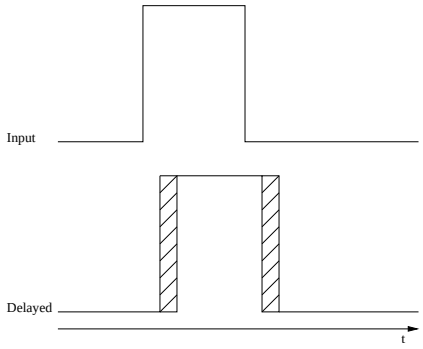
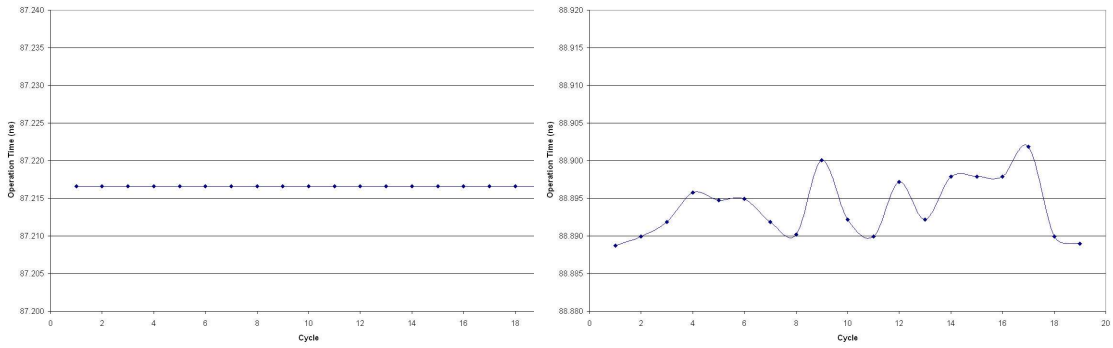


Figure 7: Input and randomly delayed clock signals

delayed clock signals are shown in figure 7.

To remove the timing information associated with the data values and operations performed, the output needs to be held for a random time before becoming available. This is done using the randomly delayed clock signal, and a hold element (a D-type flip-flop used for testing). By running the system for several cycles while holding the inputs and operation steady, we can look for correlations between the time taken between the input clock rising and the output becoming available. This effectively shows the total time that the operation has taken to complete. The timing results for a particular operation are shown in figure 8. This shows that the cycles take random times to complete, therefore calculating which operation has run with which data values has now become very difficult, which can be seen when the timing information for several different operations is compared, shown in figure 9.



(a) (b)

Figure 8: Time between clock being asserted and output becoming valid for an operation. (a) Unsecured, (b) Secured

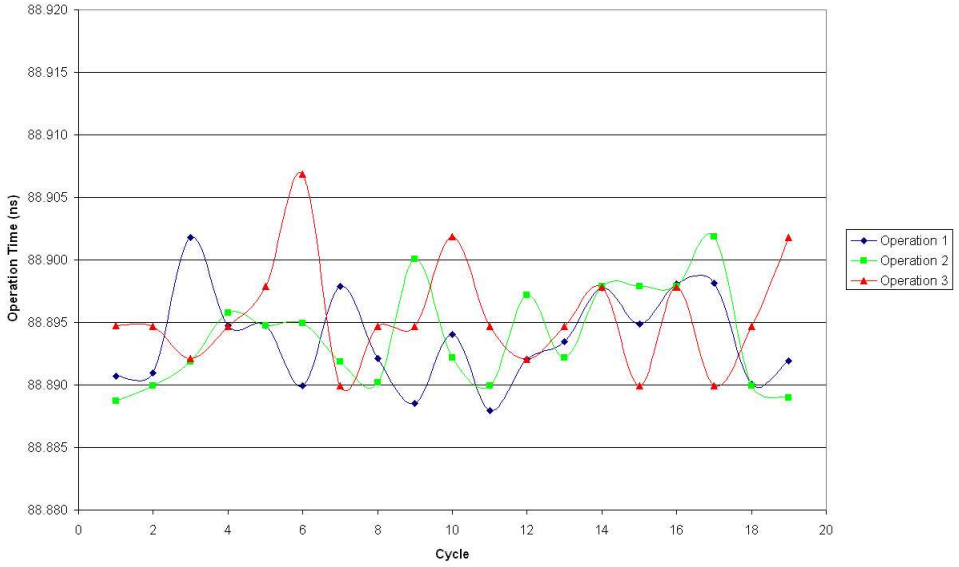


Figure 9: Comparison of timing for secured operations

6 Conclusions

We have shown how it could be possible to hide the timing information associated with the values of signals and the operations performed on these signals using random numbers and variable delay elements. Other approaches are able to be used in conjunction with this method, and could provide additional security benefits, such as resistance to power analysis attacks.

A random number generator based on metastability which is able to provide random bits at a high bandwidth has been used, this is particularly suited to on-chip integration, and by accompanying this with a variable delay element, then it is possible to add a random delay to a signal.

A method of modulating a clock signal which is delayed by a random amount of time between different levels of a circuit has been proposed, which should reduce the possibility of timing attacks on such systems. Although not tested on an actual cryptographic system as yet, simulated results which highlight the benefits of such a system are shown.

References

[1] C. D'Alessandro, K. Gardiner, D. Kinniment, and A. Yakovlev. On-chip sub-picosecond phase allignment. In *Second UK Embedded Forum 2005*, 2005.

[2] J. F. Dhem, F. Koeune, P. A. Leroux, P. Mestré, J. J. Quisquatar, and J. L. Willems. A practicle implementation of the timing attack. Technical report, Université catholique de Louvain, 1998.

[3] D. J. Kinniment and E. G. Chester. *Design of an On-Chip Random Number Generator using Metastability*. School of Electrical, Electronic and Computer Engineering, University of Newcastle upon Tyne.

[4] P. C. Kocher. *Timing Attacks on Implementations of Diffie-Helleman, RSA, DSS and Other Systems.* Cryptography Research, Inc.

[5] F. J. Martinez. Design and fabrication of a comparator for an asynchronous flash a/d converter. Master's thesis, School of Electrical, Electronic and Computer Engineering, University of Newcastle upon Tyne, 1998.

[6] A. S. Sedra and K. C. Smith. *Microelectronics Circuits.* Oxford University Press, 4th edition, 1998.

[7] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev. Design and analysis of dual-rail circuits for security applications. *IEEE Transactions on Computers*, 54(4), 2005.

[8] D. Sokolov, J. P. Murphy, A. Bystrov, and A. Yakovlev. Improving the security of dual-rail circuits. Technical report, School of Electrical, Electronic and Computer Engineering, University of Newcastle upon Tyne, 2004.