Inventer

André Brisson Whitenoise Laboratories Canada Inc. #701–1736 West 10th Street, Vancouver British Columbia V6J 2A6, Canada Email: abrisson@wnlabs.com

Name

Whitenoise secure circuit design implementation techniques to prevent Power Analysis attacks and other side channel attacks, secure other physical cryptosystem implementations, and implementation of Whitenoise into low cost micro processing and smart components retaining one-time-pad characteristics.

Abstract:

Abstract—Two circuit design techniques are examined which improve the robustness of Whitenoise encryption algorithm implementation against side-channel attacks based on dynamic and/or static power consumption. The first technique conceals the power consumption and has linear cost. The second technique randomizes the power consumption and has quadratic cost. These techniques are not mutually exclusive; their synergy provides a good robustness against power analysis attacks. Other circuit-level protection can be applied on top of the proposed techniques, opening the avenue for generating very robust implementations.

TECHNICAL FIELD

The invention relates to the fields of security, cryptography and circuit design for microprocessors, circular shift registers, line feed shift registers, peripheral interface controllers, SIM cards, RFID tags, counters and other smart components in order to secure communications and data storage in hardware implementations of the Whitenoise algorithm for hardware based trust and identity.

BACKGROUND

The most widely used method for providing security online for authentication, encryption and other network security controls is using asymmetrical systems of the public key design where authentication and encryption rely on certificates issued by certificate servers and other readily accessible processes and services. Public Key Infrastructure (PKI) systems have known security vulnerabilities such as being susceptible to Man-in-the-Middle [MiM] attacks, because they are often implemented improperly and because public keys are always available for

factoring and because there is always key negotiation to initiate a session. PKI is characterized by its difficulty and often inability to sustain identity and data provenance throughout network sessions and in storage.

The overhead of the PKI system is high, not just because of all the steps involved in the architecture, but also their choice of cryptography. The key strengths used by the PKI have been called into question recently. Public keys are compound primes and they are always available for attack. There have been significant strides in prime numbers and factoring theory. New techniques exist to factor compound primes. Fast computers factor compound primes by simplified techniques like the “sieve” method, so that what used to take years now can be done in hours. Using progressively stronger keys with public key systems becomes progressively more difficult because of the additional computational overhead introduced as keys get stronger (longer). Additionally, with the advent of quantum computing all public keys will be easily factored and broken because of fixed key sizes.

The deployment of cryptosystems into hardware complicates security implementations further by introducing new physical variables that might be available to attackers. Any physical implementation of a cryptosystem provides side-channel information that attackers can use to try to reveal the secret keys or secret key lengths and fixed data therein (in this example with circular shift registers). As the encryption activity depends on the secret key, attacks based on power analysis exploit the correlation between the data, operations, and power consumption. Field-Programmable Gate Arrays (FPGA), for example, are notable for their large power consumption, and that leads to vulnerable implementations. It is worth mentioning that the cost needed to mount power attacks ranges from thousands of dollars in the case of simple cryptosystems to tens or hundreds of thousands of dollars for more complex cryptosystems. In any case, this vulnerability is a major concern for the builders of cryptosystems.

Need

Networks globally are under threat from nation state cyber warfare, stateless activistists, terrorists, hacktivists, criminal gangs and a broader spectrum of cyber threats than ever before.

Side channel attacks are very expensive and are usually conducted by nation states and hacktivists. Compromising utilities, energy, communications, transportation and other critical infrastructures to deny service and resources at a societal level is motive enough for those groups. For example, one successful attack can shut down the power for entire regions and generate cascading negative consequences.

At the same time there is exploding interconnectivity where more and more components and devices are becoming networked. The Internet of Everything, smart grids, utilities, communications and critical infrastructures are adding billions if not trillions of components and devices with communication capabilities (regardless of how limited) that become vulnerable points of egress to our networks. Many of these components are mobile and the majority of

them communicate machine-to-machine which complicates sustaining endpoint identity of the authorized user and provenance for hardware based trust.

Traditional RSA-style, asymmetric cryptographic implementations with current AES-style encryption algorithms cannot be implemented properly, reliably or even at all in computing environments and components with limited power, computational and overhead resource availability. These include common components that we find in almost all manufactured electronic products and include chips, ASICs, field programmable gate arrays, sensors, circular shift registers, line feed shift registers, counters and peripheral interface controllers to name some.

Any topology or technologies created to provide the highest level of network security must address issues of secure key management, key creation, key exchange, authentication, intrusion detection, revocation and authorizations. Secure networks must also be able to assure continual authentication of all persons, components, and devices on a network as well as impose provenance on all data.

There is a need for a key based network security control, protocol, process and framework where there is never any transfer of key or offset information during sessions, after one-time pre-distribution and pre-authentication of users and endpoints following accepted identity proofing techniques for person and non-person entities.

There is a need for a system where there is never a shared secret transmitted in session, where there is never a public key which can be factored or broken because of improved factoring techniques or quantum computing, and where there is no reliance on asymmetric key exchange or negotiation which always has security flaws if used in isolation.

There is a need for large scale, distributed authentication systems where there is only partial sharing of credentials.

There is a need to secure low-cost hardware components that are becoming prevalent in our networks and critical infrastructures that must be able to authenticate themselves and secure their data and communications.

There is a need to secure hardware components against side channel attack classes like power analysis attacks and generalize the solution to other kinds of side channel attacks.

There is a need to secure widely deployed existing cryptographic protocols like AES NI (new instructions) to make them side channel attack resistant and remove other inherent vulnerabilities they have.

The foregoing examples of the related art and limitations related thereto are intended to be illustrative and not exclusive. Other limitations of the related art will become apparent to those of skill in the art upon a reading of the specification and a study of the drawings.

SUMMARY

The security of all crypto systems, whether asymmetric or symmetric, is determined by key creation, key management, key distribution and exchange, key storage, the nature of the keys, and how the keys are used.

In this summary, we will look at two unique and inventive key creation and deployment techniques for secure circuit design using a deployment in circular shift registers as the exemplary context but the invention is not restricted to this context.

The preferred embodiment uses Whitenoise Super keys (patent: Boren Brisson 10/299847 granted) or other exponential, one-time-pad keys for additional key generation and for all security functions including encryption. The encryption function may be accomplished with any deterministic random (pseudo random) data source and any encryption algorithms. Adoption of secure network topologies also relies in some contexts on its ability to leverage existing technologies. As such, hybrid approaches are envisioned.

The following embodiments and aspects thereof are described and illustrated in conjunction with systems, tools and methods which are meant to be exemplary and illustrative, not limiting in scope. In various embodiments, one or more of the above-described problems have been reduced or eliminated, while other embodiments are directed to other improvements.

Two dynamic distributed key and identity management circuit designs are provided in which circular shift registers are assigned a random number of Whitenoise subkey lengths and populated with deterministic randomized data from a Whitenoise distributed master key.

BRIEF DESCRIPTION OF DRAWINGS

Exemplary embodiments are illustrated in referenced figures of the drawings. It is intended that the embodiments and figures disclosed herein are to be considered illustrative rather than restrictive.

FIG. 1 illustrates a common implementation of the Whitenoise algorithm, which consists of prime-length Circular Shift Registers (CSR) storing randomized subkey data, whose outputs are XOR'd and sent to an S-box for delinearization. The number of CSRs, their lengths, and their byte values are all configurable, being populated and re-populated from a one-time deployable Master Key.

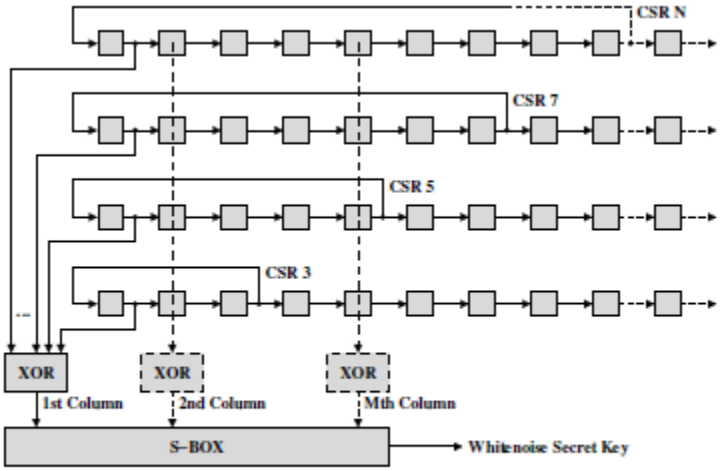

Fig. 1: Common Whitenoise algorithm implementation.

142

143 FIG. 2 illustrates the evolution of a 7-cell CSR over seven time cycles. It assumes for illustration
144 purposes that the maximum configurable length of a subkey in the implementation is 11. Also
145 assume a circular shift register with seven cells, whose configured byte values are A, B, C, D, E,
146 F, and G (the letters are used in ascending order for clarity only, as they do not represent static
147 values).

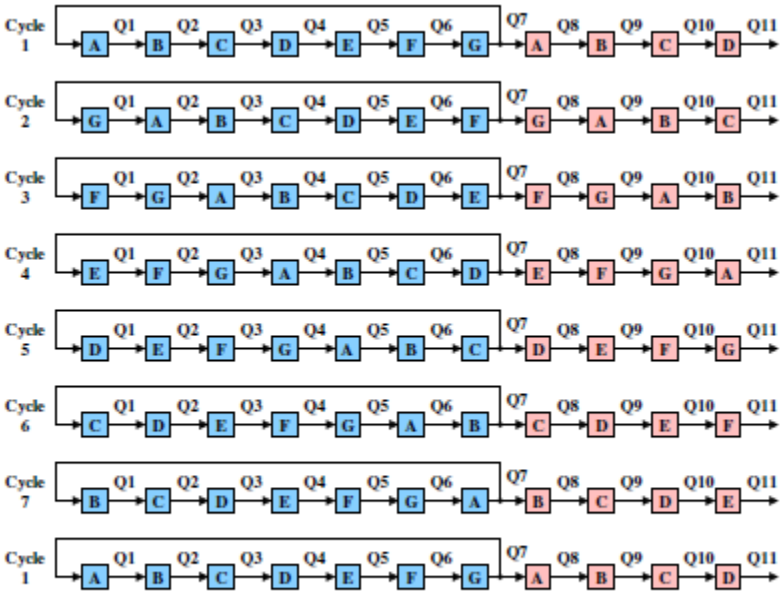

Fig. 2: A 7-cell circular shift.

148

149 FIG. 3 illustrates a technique to conceal the periodic component of the power consumption of
150 the red stub. A second (green) stub, such that the total number of cells in the red and green
151 stubs equals the number of cells in the blue (main) shift register is deployed. It is apparent that
152 the byte values in the red+green part are identical to those in the blue part; thus, the power

consumption of the red and green part equals the power consumption of the blue part, which, as described, is itself a constant due to the circular operation.

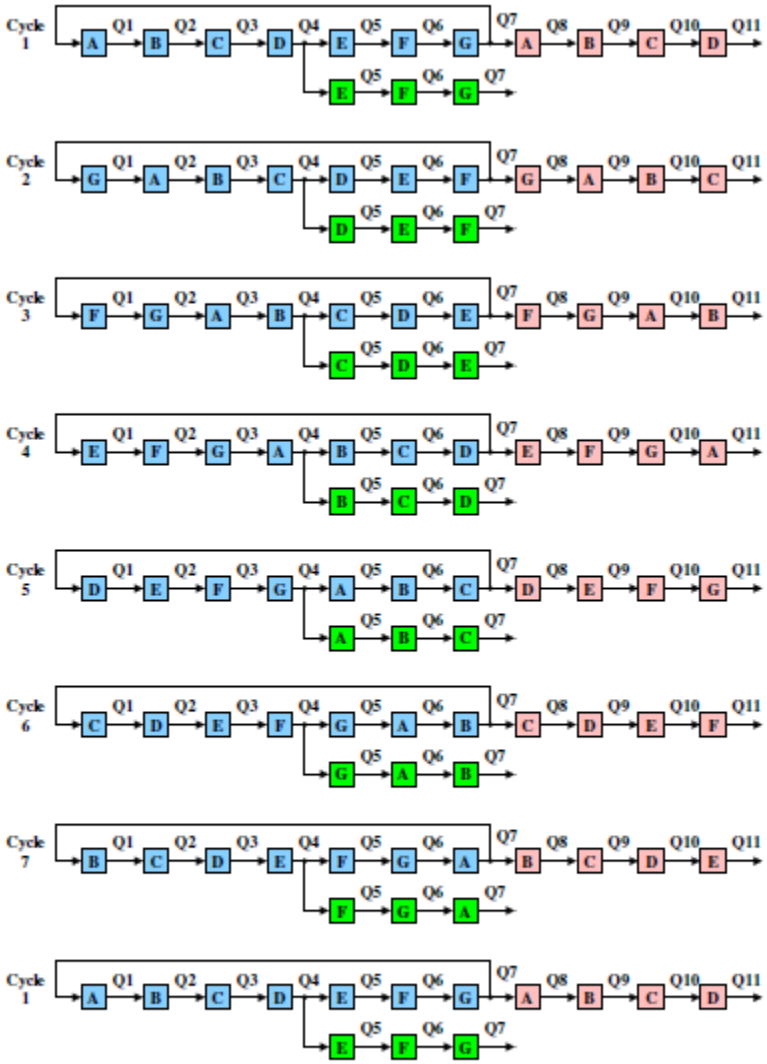

Fig. 3: A secured 7-cell circular shift register.

FIG. 4 illustrates that a way to conceal both the dynamic and static power consumptions is by driving the red+green part with inverted signals (Fig. 4). This way, since the number of cells in logic '1' equals the number of cells in logic '0' at any time instance, the dependence of the power consumption on Hamming distance is removed with no additional hardware. At this point, the attacker is no longer able to obtain side-channel information by, for example, stopping the clock and measuring the leakage.

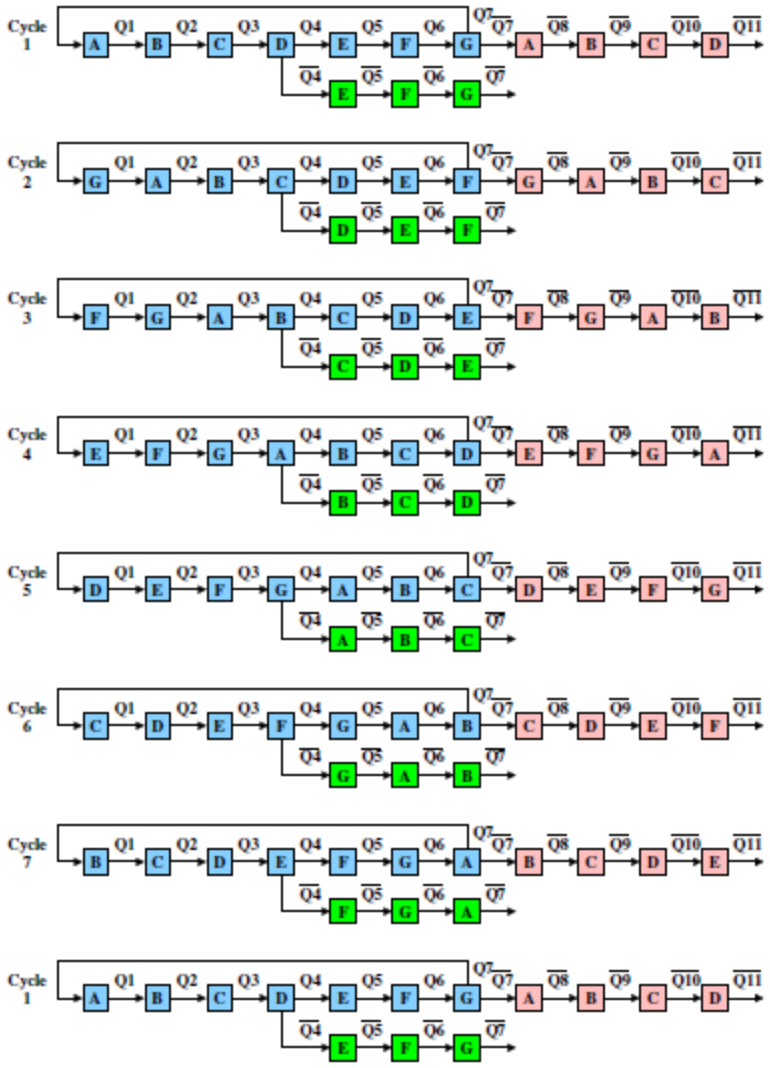

Fig. 4: A highly secured 7-cell circular shift register.

FIG. 5 illustrates a technique replicating each register N −1 times (where N is the total number of registers), such as the original register runs its own subkey to produce real randomized data, whereas its replicas run the subkeys of the other registers to produce fake randomized data. This is exemplified where Registers 1 and 2 are replicated. Register 1 replica runs the key bytes of the original Register 2, and Register 2 replica runs the key bytes of the original Register 1.

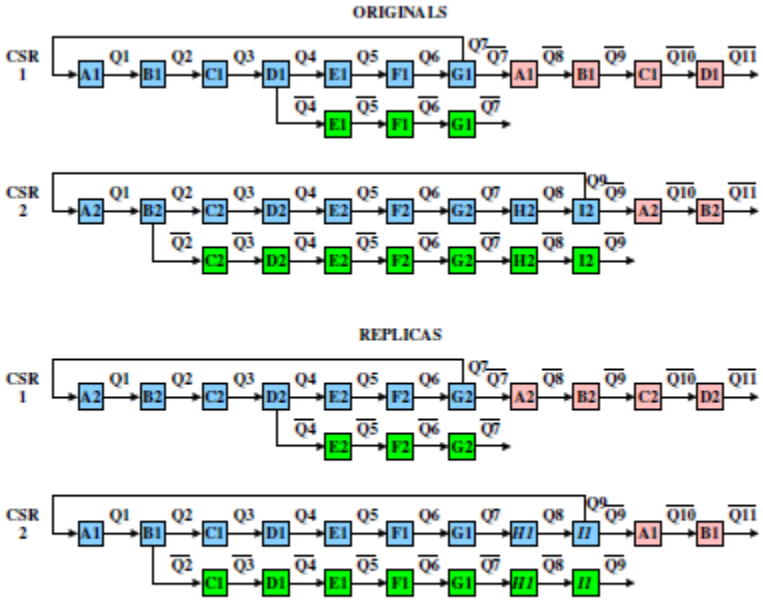

Fig. 5: A masked implementation.

In addition to the exemplary aspects and embodiments described above, further aspects and embodiments will become apparent by reference to the drawings and by study of the following detailed descriptions.

Throughout the following description specific details are set forth in order to provide a more thorough understanding to persons skilled in the art. However, well known elements may not have been shown or described in detail to avoid unnecessarily obscuring the disclosure. Accordingly, the description and drawings are to be regarded in an illustrative, rather than a restrictive, sense.

FIG. 1 illustrates a Whitenoise deployment in a circular shift register (CSR).

FIG. 2 illustrates the evolution of a 7-cell CSR over seven time cycles.

FIG. 3 illustrates a technique to conceal the periodic component of the power consumption.

FIG. 4 illustrates a technique of replicating each register.

FIG. 5 illustrates a technique of masking.

Description

I. INTRODUCTION

In the challenge of securing Internet communications, Whitenoise Laboratories has a patented stream-cipher algorithm. This technology can be used in applications such as identity management, secure network access, dynamic authentication, intrusion detection, automatic revocation, as well as encryption. The Whitenoise algorithm is rather simple; thus, its implementation will need only reduced silicon area or a low-end microcontroller (such as, PIC from Microchip Tech, and exhibits low power consumption, being particularly suitable for use in portable electronics. The Whitenoise encryption algorithm survived a large number of brute force attacks mounted on a computer array during a performance analysis conducted by Traore and Liu at the University of Victoria, Canada. As described by Wagner, no mathematical attack breaking Whitenoise was found in his security analysis at the University of California at Berkeley. As a result, the algorithm is considered to be highly secure.

The deployment of any cryptosystem into hardware introduces new physical variables; thus, any cryptosystem's physical implementation provides side-channel information that attackers can use to reveal secret keys, the secret key lengths and fixed data therein. As the encryption activity depends on the secret key, attacks based on power analysis exploit the correlation between the data, operations, and power consumption. Field-Programmable Gate Arrays (FPGA) are notable for their large power consumption, and that leads to vulnerable implementations.

There are two main techniques to increase the robustness of cryptographic implementations [10]: (i) hiding (or concealing), which makes the power consumption independent relative to the processed data and/or operations, and (ii) masking, which randomizes the power consumption.

In this invention both techniques are considered in improving the security of the Whitenoise implementation in customized hardware or as an adjunct to mask other cryptosystems such as AES NI from Intel or PAX from Princeton University.

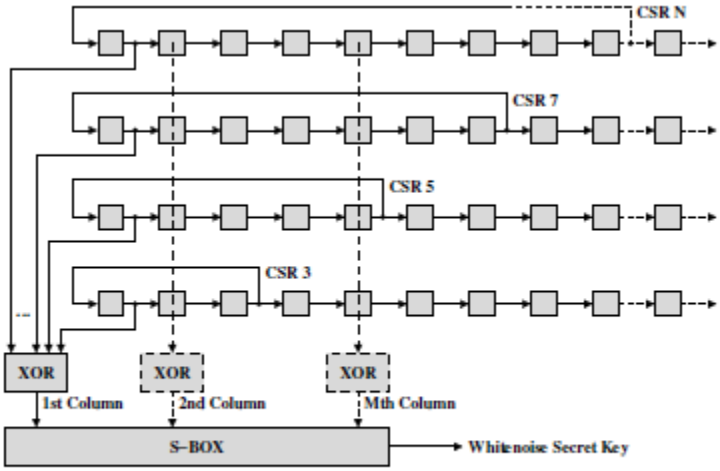

Fig. 1: Common Whitenoise algorithm implementation.

Figure 1 shows a common implementation of the Whitenoise algorithm, which consists of prime-length Circular Shift Registers (CSR) storing randomized subkey data, whose outputs are XOR'd and sent to an S-box for delinearization. The number of CSRs, their lengths, and their byte values are all configurable, being populated and re-populated from a one-time deployable Master Key. It is a concern that this Whitenoise implementation might be vulnerable to attacks based on the CSR power consumption and Hamming distance. This invention gives circuit implementation techniques that remove this kind of potential weakness, rendering an encryption implementation with increased robustness against such attacks. The invention contributions are as follows.

1) A design technique based on cell replication, which conceals the power consumption of the CSRs.

2) A design technique based on signal polarity inversion, which further removes the power consumption dependence on Hamming distance.

3) A design technique based on fake keys, which randomizes the power consumption of the CSRs.

II. WHITENOISE ENCRYPTION ALGORITHM

The implementation of the Whitenoise algorithm shown in Figure 1 is based on Circular Shift Registers (CSR), which store randomized subkeys whose number, lengths, and byte values are

defined by the secret master key. For clarity only four registers with 3, 5, 7, and N cells, respectively, are shown.

These registers loop infinitely with their outputs being XOR'd. The XOR outputs are sent to an S-box for delinearization. In a common hardware implementation scenario, each CSR is manufactured with the largest number of cells allowed by the algorithm and cost of the chip.

Based on randomized data from the master key, a set of selectors (not figured) will define the length and byte values of each CSR.

For the sake of presentation, assume that the maximum configurable length of a subkey in the implementation is 11. Also assume a circular shift register with seven cells, whose configured byte values are A, B, C, D, E, F, and G (the letters are used in ascending order for clarity only, as they do not represent static values). Figure 2 presents the evolution of the 7-cell CSR over seven time cycles. Due to the circular operation only the position of the bytes stored in the main (blue) loop change, not their values. Thus, its total power consumption does not change from cycle to cycle.

On the other hand, the stub of four (red) cells, which operates in parallel with the main (blue) loop, exhibits a periodic power consumption (ABCD, GABC, FGAB, EFGA, DEFG, CDEF, BCDE) having the period equal to the length of the circular shift register. This is a major leak of side-channel information that can be used to reveal the number and the lengths of the circular shift registers, and ultimately a static secret key [11].

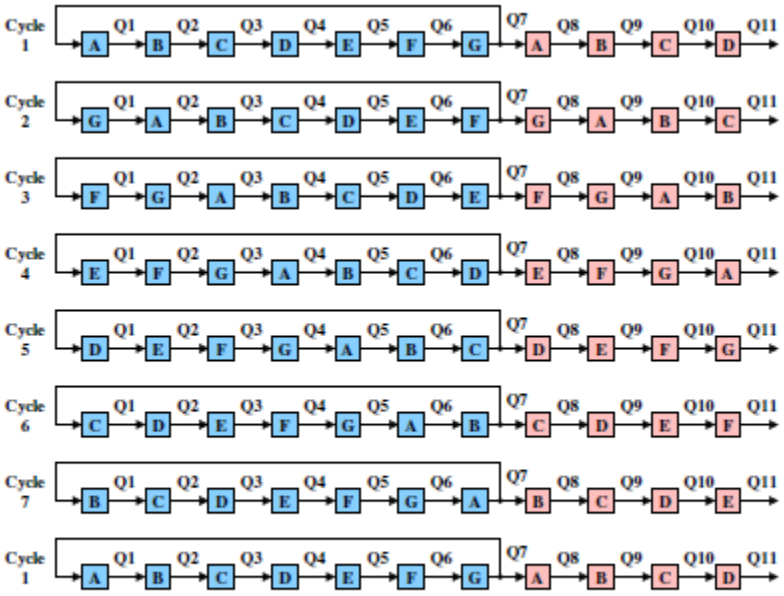

Fig. 2: A 7-cell circular shift.

Fig. 2: A 7-cell circular shift.

By deactivating the red stub (for example, by forcing it in an idle state), the periodic component of the power consumption is eliminated. However, this approach only slightly improves the robustness against power attacks, as the levels of the dynamic and static (leakage) power consumptions of the main (blue) loop still depend on the Hamming distance.

In particular, attacks based on leakage have been proven highly successful. As leakage power reaches a large fraction of the total power consumption in modern technologies, this problem cannot be neglected.

In the process section, we present a circuit technique that conceals both the static and dynamic power consumption. It will have strong beneficial effects in Whitenoise robustness against attacks based on power consumption.

Process

III. TECHNIQUES TO CONCEAL POWER CONSUMPTION

To conceal the periodic component of the power consumption of the red stub, we deploy a second (green) stub, such that the total number of cells in the red and green stubs equals the number of cells in the blue (main) shift register. This technique is illustrated in Figure 3. It is apparent that the byte values in the red+green part are identical to those in the blue part; thus, the power consumption of the red and green part equals the power consumption of the blue part, which, as described, is itself a constant due to the circular operation. Unfortunately, both the static power consumptions of the blue and red+green parts depend on Hamming distances, being a source of information leak. As mentioned in the previous section, this is a serious security threat.

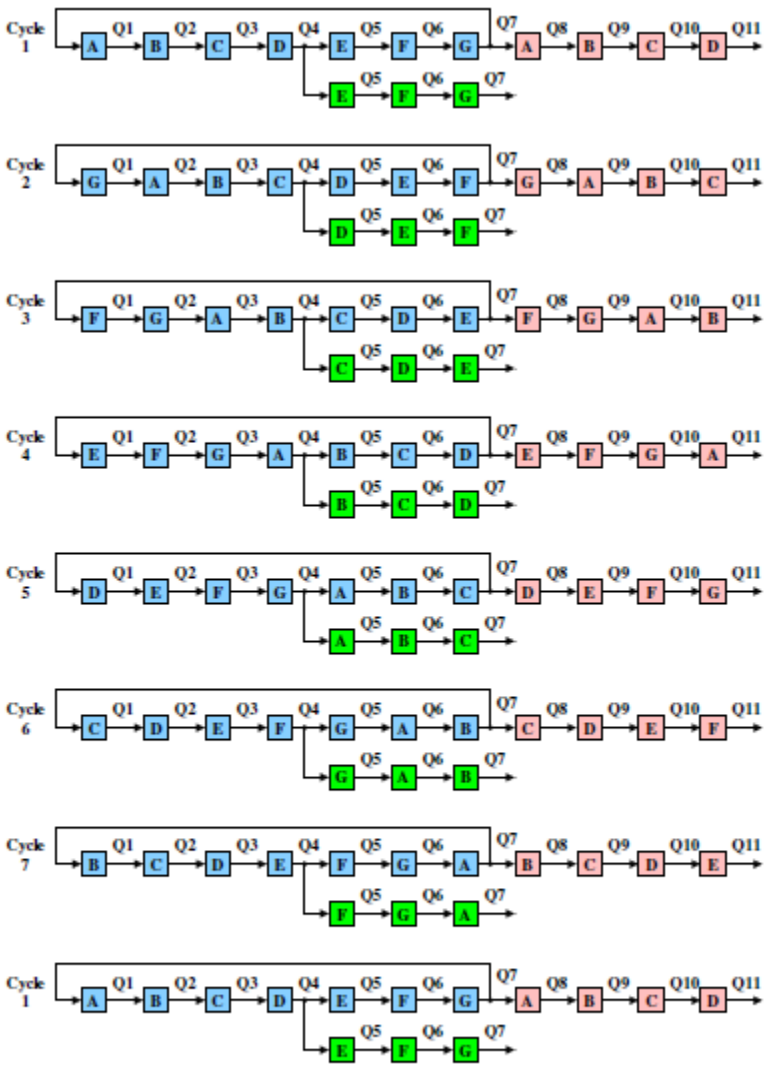

Fig. 3: A secured 7-cell circular shift register.

282

283 Fig. 3: A secured 7-cell circular shift register.

284 A way to conceal both the dynamic and static power consumptions is by driving the red+green
285 part with inverted signals (Fig. 4). This way, since the number of cells in logic ’1’ equals the
286 number of cells in logic ’0’ at any time instance , the dependence of the power consumption on
287 Hamming distance is removed with no additional hardware. At this point, the attacker is no
288 longer able to obtain side-channel information by, for example, stopping the clock and
289 measuring the leakage.

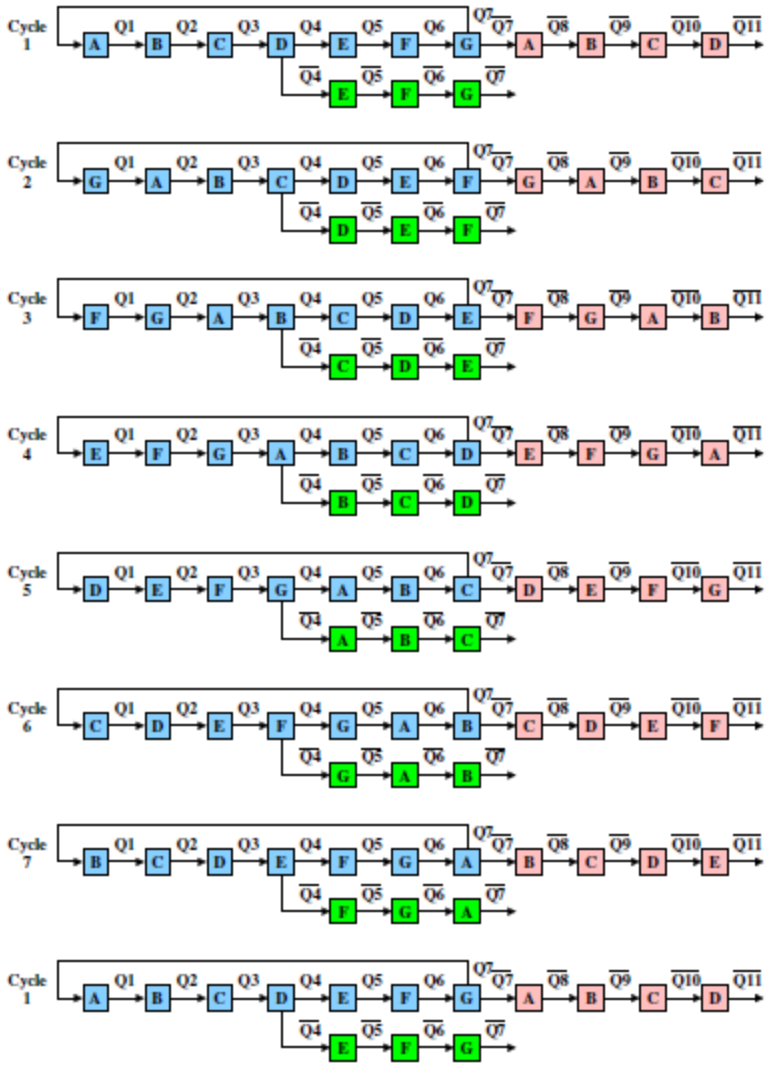

Fig. 4: A highly secured 7-cell circular shift register.

Fig. 4: A highly secured 7-cell circular shift register.

The manufacturing technology exhibits variations with process and temperature. This means that different cells in the implementation do not have equal power consumptions; mismatches will always exist. As a result, security techniques based on power concealment suffer from technological limitations. The next section addresses this issue, and proposes an additional layer of protection based on power masking.

IV. TECHNIQUES TO MASK POWER CONSUMPTION

This technique consists in replicating each register N −1 times (where N is the total number of registers), such as the original register runs its own subkey to produce real randomized data, whereas its replicas run the subkeys of the other registers to produce fake randomized data. This technique is exemplified in Figure 5, where Registers 1 and 2 are replicated (Register 1

replica runs the key bytes of the original Register 2, and Register 2 replica runs the key bytes of the original Register 1). Note that Bits H1 and H1 in the Register 2 replica (typed in slanted fonts) are dummy data; they are needed since the Register 2 feedback is longer than the Register 1 feedback.

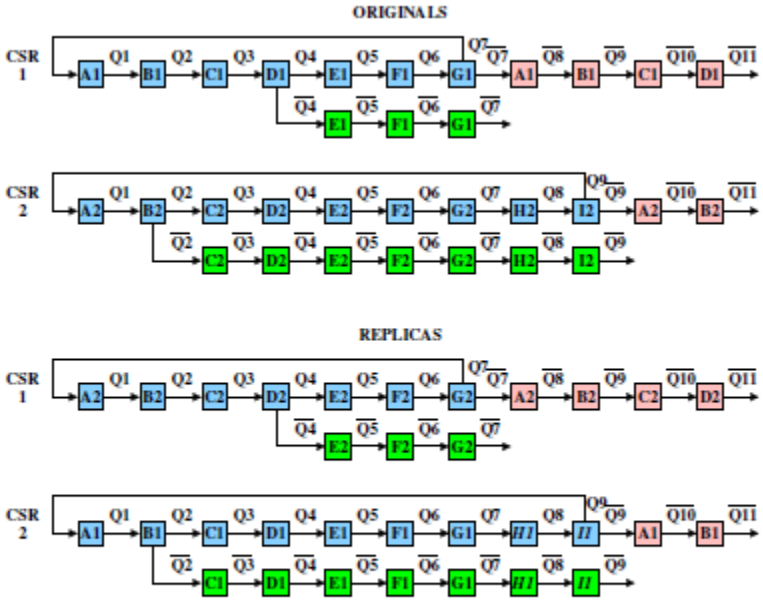

Fig. 5: A masked implementation.

Fig. 5: A masked implementation.

It should be observed that the attacker does not have access to any of the replicas' outputs, as they are not routed to the circuit's pins. Without knowledge on these outputs, it is very difficult (if not impossible) to launch side-channel attacks on replicas themselves; this makes the masking technique highly secure.

Assume N = 10. Then, there will be N ×N = 10×10−10 = 90 registers running fake data. As a result, the power consumption will generate 90% fake side-channel information misleading the attacker. This represents an order of magnitude improvement in robustness achieved at a quadratic cost. For increased security adding the masking technique on top of the hiding technique is envisioned. Other security techniques at the circuit level can be used on top of the presented invention techniques. For example, dual-rail logic can be an option for more advanced implementations.

To summarize, the following physical transformations of the invention are implemented to build a robust Whitenoise implementation.

• The addition of a second (green) stub to force the total power consumption of the red+green part constant. This technique aims to conceal the power consumption of the circular shift registers.

• The change in signal polarity for both stubs, which aims to remove the dependence on both dynamic and static power consumptions on Hamming distance.

• The replication of the circular shift registers, such that the replicas run fake subkeys. This technique aims to mask the power consumption.

Retaining one-time-pad characteristics

The only mathematically provable unbreakable encryption algorithm is a one-time-pad. In components like circular shift registers and other low-cost electronic components the single, one-time use of the data source is rapidly exhausted before the data source begins to repeat itself.

Side channel attacks in the context presented can only provide information about the subkey lengths and the static, deterministically randomized data therein. It has no information and can tell us nothing about the distributed master key which populates those subkey lengths.

This invention tracks when the data source would begin repeating itself, flushes out the static data in the subkey lengths, and beginning at the master key's last current dynamic offset or appropriate index repopulates the subkey lengths with unused, deterministically random data from the next unused portion of the master key.

In this manner, even a very limited environment like counters and circular shift registers, the process is made dynamic and retains the necessary characteristics of a one-time-pad which make these inexpensive and not complex components highly secure.

In cryptographic processes like Whitenoise, one time pad characteristics are retained by not reusing key stream portions (tokens). Related to this is the ability to track indexes, setup the component to an index point, and begin the generation of a specific key length beginning at that point to recreate the correct portion of the key that was used for a specific operation.

Because of the nature of components that can exploit this invention, and the design of this invention, we have a finite set of orientations between the subkeys and registers. Much like spinning the dials on a simple lock to open a suitcase, we can create current dynamic offsets based on the orientation of the registers.

When the data source of the registers is exhausted and would begin to repeat with continued use, they are flushed out. They are repopulated with specific deterministic random data from the Master key and the master key offsets are tracked so that there is no re-use of any key stream portion.

The master key current dynamic offset is used in conjunction with the register orientation to identify how those registers and subkey lengths were filled, and then one step further, what the data orientation within the register is to create an index to identify the starting point of the first register in this invention for the next fresh key segment.

In this manner, very small restricted components can have access to a limitless data source and operate dynamically and like a one-time-pad. Because these characteristics are imposed in this context, even small dynamic tokens can be used for authentication without fear of compromise.

AES NI as example of cryptosystems to secure

There are countless cryptosystems like AES NI (new instructions) that are vulnerable to side channel attacks as well as other attacks. Addition of the current invention to those cryptosystems to randomize, mask and hide output will remove side channel attack vulnerability and increase the security of those systems by orders of magnitude.

VI. CONCLUSION

This invention shows circuit design techniques preventing side channel attacks based on power consumption analysis and which can be generalized to preventing other side channel attack techniques. Thematically, Side Channel attacks are prevented because all operations when using the preferred embodiment of Whitenoise keys are order 1 after key load so there is no change in output to make meaningful correlations and because there is no access to the key.

Claims

1. A method of using Whitenoise based on cell replication, which conceals the power consumption of the Circular Shift Registers. The method conceals the periodic component of the power consumption of the red stub by deploying a second (green) stub, such that the total number of cells in the red and green stubs equals the number of cells in the blue (main) shift register. This technique is illustrated in Figure 3. It is apparent that the byte values in the red+green part are identical to those in the blue part; thus, the power consumption of the red and green part equals the power consumption of the blue part, which, as described, is itself a constant due to the circular operation.

2. A method of using Whitenoise signal polarity inversion, which further removes the power consumption dependence on Hamming distance.

3. A method of using Whitenoise based on fake keys, which randomizes the power consumption of the Circular Shift Registers. This method consists in replicating each register N −1 times (where N is the total number of registers), such as the original register runs its own subkey to produce real randomized data, whereas its replicas run the subkeys of the other registers to produce fake randomized data. This technique is exemplified in Figure 5, where Registers 1 and 2 are replicated (Register 1 replica runs the key bytes of the original Register 2, and Register 2 replica runs the key bytes of the original Register 1). Note that Bits H1 and H1 in the Register 2 replica (typed in slanted

fonts) are dummy data; they are needed since the Register 2 feedback is longer than the Register 1 feedback.

4. A method of using Whitenoise of tracking when the data source would begin repeating itself at which point the data is flushed out of the register and repopulated with the next portion of unused key stream data from a master key to which a hacker never has access or knowledge. The process involves tracking current dynamic offsets and ensures that one-time-pad characteristics are maintained in low-cost, low-power, low-computationally resourced components requiring security and identity by never repeating a key or portion of a key stream even in a highly restrictive environment.

5. A method of using Whitenoise of implementing the invention with other cryptosystem implementations like AES NI to truly randomize their output, make them act like one-time-pads, and secure components deploying those other cryptosystems from side channel attacks.

6. A method of using Whitenoise of securing low-cost, low-computational components with limited power and overhead resource availability to drive the implementation on national security level cryptographic technology to components where traditional RSA-styled-public-key-technologies and current cryptographic algorithms cannot be implemented because of cost and overhead restrictions. This is comprised of implementing claims 1-5 in low cost computational components such as Circular Shift Registers, counters, key rings, line feed shift registers, field programmable arrays, subscriber identity modules and other low cost microprocessors and chips.