Keymill: Side-Channel Resilient Key Generator
A New Concept for SCA-Security by Design

Mostafa Taha[1], Arash Reyhani-Masoleh[1], and Patrick Schaumont[2]

[1] Department of Electrical and Computer Engineering
Western University, London, ON N6A 5B9, Canada
mtaha9@uwo.ca, areyhani@uwo.ca
[2] Secure Embedded Systems
Center for Embedded Systems for Critical Applications
Bradley Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA 24061, USA
schaum@vt.edu

Abstract. In the crypto community, it is widely acknowledged that any cryptographic scheme that is built with no countermeasure against side-channel analysis (SCA) can be easily broken. In this paper, we challenge this intuition. We investigate a novel approach in the design of cryptographic primitives that promotes inherent security against side-channel analysis without using redundant circuits. We propose `Keymill`, a new keystream generator that is immune against SCA attacks. Security of the proposed scheme depends on mixing key bits in a special way that expands the size of any useful key hypothesis to the full entropy, which enables SCA-security that is equivalent to the brute force. Doing so, we do not propose a better SCA countermeasure, but rather a new one. The current solution focuses exclusively on side-channel analysis and works on top of any unprotected block cipher for mathematical security. The proposed primitive is generic and can turn any block cipher into a protected mode using only 775 equivalent NAND gates, which is almost half the area of the best countermeasure available in the literature.

1 Introduction

Side-Channel Analysis (SCA) is a major threat to the embedded implementation of cryptographic schemes. It is an implementation attack, where the adversary exploits side-channel outputs in order to recover information about secret values. Side-channel outputs include power consumption, electromagnetic radiation, execution time, and more. SCA targets the underlying implementation rather than the mathematical structure of the scheme. Its concept depends on predicting changes in the behavior of a crypto module using key hypotheses. Then, the hypothesis can be confirmed or rejected based on the actual behavior of the module. There are many variations in the details involved in applying this attack, but the overall concept remains the same.

Traditional countermeasures comes in three categories: Masking, Hiding, and Leakage Resiliency. Masking depends on blinding the internal operations using

a random variable. The effect of randomness should be removed at the end of computation to retrieve the legitimate output. This countermeasure prevents correct prediction about the power consumption. Hiding depends on minimizing the signal-to-noise ratio in the leakage using a complement processing module, shuffling, dedicated noise generator or other means. This countermeasure prevents internal operations from affecting the power consumption. Leakage Resiliency depends on updating the secret value after every operation to prevent aggregating unbounded information against the same secret.

After much research in this field, it was acknowledged that protecting an already designed cryptographic algorithm can become very costly in terms of area and clock cycles. Hence, Medwed et al. [12] proposed using two different primitives: one to achieve security against side-channel analysis, while the other is used to protect the design against mathematical cryptanalysis, as shown in Fig. 1(a). They proposed modular multiplication between the key and a random number (function $g_k(r)$) that can be easily protected against SCA using masking and shuffling. The random number works as an Initialization Vector for block cipher modes, and should be sent to the other party. The output is a unique secret that can be used to encrypt plain data using any block cipher. Essentially, they proposed separation of duties while still depending on the common SCA countermeasure techniques. They acknowledged that any cryptographic primitive that is built with no sound SCA countermeasure can be easily broken.

In this paper, we challenge this intuition by proposing a new primitive that is secure against SCA attacks inherently by design without requiring any redundant circuit. We follow the separation guidelines of Medwed et al. [12] to better focus on side-channel properties. However, we propose a keystream generator that can encrypt plain data of any length, as shown in Fig. 1(b). The keystream generator, `Keymill`, depends on a special class of NLFSRs, augmented with some implementation aspects that are hardware specific. Security of the proposed scheme depends on mixing the key bits in a novel way so that no key hypothesis that is smaller than 128 bits can break the system. Our design can be implemented using 775 GEs in 130 nm CMOS technology.

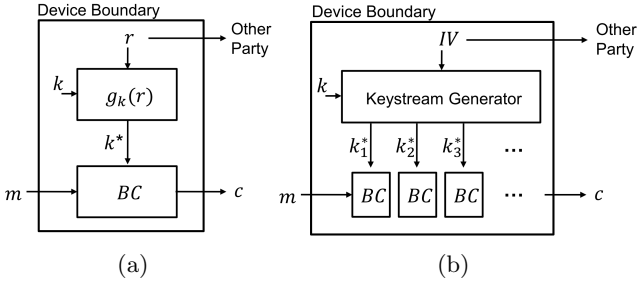

Fig. 1. (a) Domain separation proposed in [12]. (b) Generalized domain separation, as followed in this paper. *BC* denotes a block cipher.

The paper is organized as follows: Sec. 2 reviews some background about NLFSR and some insights about SCA that are mandatory for this research. Sec. 3 proposes a new definition about SCA-security and highlights the problem statement. Sec. 4 introduces some toy models that will be helpful in the analysis of the full system. Sec. 5 shows the proposed design, while its security analysis is highlighted in Sec. 6. The implementation cost and comparison with previous techniques are discussed in Sec. 7. We conclude the paper in Sec. 8.

2 Background

2.1 Nonlinear Feedback Shift Registers

An NLFSR is a common component in cryptographic stream ciphers. NLFSRs are known to be challenging targets for SCA [7], while having high performance at small implementation cost [15]. An NLFSR consists of n binary storage units called stages. In each cycle, the register is shifted by one bit, while the new value of the first bit is the output of the feedback function $f(S)$; where $f(S)$ is a non-linear function computed over the state of the register with mapping $f : \{0,1\}^n \rightarrow \{0,1\}$. The output of the NLFSR is the sequence that shows at the last stage. The period of an NLFSR is the length of the longest cyclic output that it can produce. The NLFSR that can generate the full period of $2^n - 1$ (excluding the zero state) is called a primitive NLFSR, where n is the length of the register.

2.2 Taxonomy of SCA

SCA depends on recovering information from leakage traces. The number of points that are involved in recovering complete information about any piece of the secret key determine the attack class.

For example, Simple Power Analysis (SPA) works by recovering information from a single point in the trace. Differential Power Analysis (DPA) works by combining information from a selected trace point across many different inputs. Higher Order DPA (HO-DPA) works by computing higher order moments before applying regular DPA attacks. Finally, a Multi-Variate DPA attack combines information from different trace points along the time at different input patterns.

2.3 SCA's Divide-and-Conquer

SCA works only for its ability to break complexity of the secret value. The typical trend in designing cryptographic algorithms is to mix the secret key in its original format at full entropy with the input data. We understand that, mathematically speaking, there is no reason to reduce entropy of the key before using it. However, this is exactly where the hardware fails, as the adversary can control (or monitor) the input data and observe the hardware's behavior as the input data interacts with the secret value. Usually, the input data width is equal

to or larger than the width of the secret value, giving so much flexibility in the attacker's hands to isolate, test, and collect unbounded information about small parts of the secret key. This is known as the divide-and-conquer principle of SCA.

For example, in the `AES` encryption algorithm, 8-bits of the key can be isolated and recovered at the output of the SBox by controlling 8-bits of the input data. Similarly, 4-bits can be isolated by controlling 4-bits of the input in the `PRESENT` cipher. Also, in the typical implementation of RSA, singular key-bits can be isolated by monitoring changes in the entire input data. In these examples (and many others), increasing mathematical security of the algorithm does not affect its side-channel security, as a longer key length can be broken by recursively recovering smaller segments.

In this regard, our design has two features as detailed later. We shrink the input data width to only 1-bit, which is the smallest possible. Also, we reduce entropy of the key before interacting with the input data, which preserves its secrecy.

3 Design Goals

3.1 A new definition: SCA-security

In this paper, we propose a new definition for SCA-security.

SCA-security is the minimum size of key hypothesis (in bits) such that the leakage-model using the correct key correlates to the measured leakage significantly higher than the leakage-model using any other key.

Let L_s be the leakage-model using s bits of the secret key:

$$L_s = f(x, |k|_s),$$

where x is the known public data (IV), and $|k|_s$ represents s bits of the secret key. Let L_s^* represents the leakage using the correct secret key (k^*). Also, let M be the measured leakage using the same input data set.

The SCA-security can be defined as the minimum value of s so that:

$$\rho(L_s^*, M) >> \rho(L_s, M),$$

where ρ is the Pearson product-moment correlation coefficient.

Under this definition, SCA-security of the regular square-and-multiply algorithm of `RSA` is only 1-bit. SCA-security of `AES` is 8-bits, while that of `PRESENT` cipher is 4-bits.

The goal in this paper is to design a cryptographic primitive with SCA-security that equals its brute force.

3.2 Practical Applications

The `AES` encryption modes CBC, CFB, OFB, and CTR, and the authenticated encryption modes CCM, GCM and OCB [1, 5] are equally vulnerable to SCA attacks as they use one fixed key k in every call to the underlying block cipher.

The direct application of our proposal as highlighted in Fig. 1(b) is to convert any of the aforementioned modes into SCA-secure. This is possible by using the secret key k along with the Initialization Vector (IV) as a seed for the random number generator. Each 128 bits of the pseudorandom output should be used only once to encrypt a message using any block cipher (denoted BC in the figure). In this case, the public input data that can be monitored (or controlled) by the adversary is the IV. In the following sections, we assume that the IV is 128 bits while the proposed keystream generator can be used with any other length.

4 Introductory Toy Models

A Simple Power Analysis against the internal state of a linear feedback shift register was proposed in [3]. They observed that the difference between power consumption following the Hamming Distance model of two consecutive clock cycles depends only on the edge bits, as the effect of internal bits will cancel out. This attack is not directly applied to the NLFSR case as proposed here. Zadeh et al. [17] concluded that an SCA attack can reduce complexity of the secret key only if the adversary can detect changes within the underlying gates in the non-triggering edge of the clock cycle. This condition is very tricky and has never been tested through actual power traces. Hence, in the following toy models, we will focus on Differential Power Analysis.

4.1 Toy Model I: One 8-bit NLFSR

Without loss of generality, we focus on a toy example of NLFSR that holds some security properties that are similar to the proposed structure. In this example, we study an 8-bit shift register shown in Fig. 2 where all the taps are connected to the feedback function. The structure is initialized with 8-bits of secret key, and the public data is added one-bit at a time by `xor`ing with the feedback function.

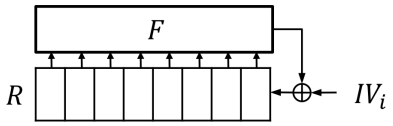

Fig. 2. Toy Model I: One 8-bit NLFSR

We denote the state at clock i by S^i, with its internal bit number j as s_j, ($j \in [0:7]$). *In the first clock cycle*, the power leakage following the Hamming

Distance power model is:

$$\begin{aligned} L &= HD(S^0, S^1), \\ &= HW(S^0 \oplus S^1), \\ &= HW((s_0, s_1, s_2, ..., s_6, s_7) \oplus (s_1, s_2, s_3, ..., s_7, F(S^0) \oplus i_0)). \end{aligned}$$

where HD is the Hamming Distance function, which is the number of bit-flips between its two inputs. HW is the Hamming Weight function, which is the number of set bits in the binary representation of its input. $F(S)$ is the feedback function. i_x is the input bit number x.

At a fixed secret value, the first terms are not data-dependent $((s_0, s_1, s_2, ..., s_6) \oplus (s_1, s_2, s_3, ..., s_7))$, hence their power consumption will not change by changing the input data and their effect will be canceled out by correlation. Hence, the data-dependent power leakage will be:

$$L = s_7 \oplus F(S^0) \oplus i_0,$$

where, the HW function was removed as its input is only one bit.

This equation shows a linear relationship between the measurable power consumption and one-bit of the input data, which does not reveal any information to the attacker. The reason is that the leakage will directly follow changes in the input ($L = 1$ at $i_0 = 1$) or the exact opposite ($L = 0$ at $i_0 = 1$) with equal probabilities of 50%, i.e. no advantage to the adversary.

In the second clock cycle, if we keep the register isolated with no other connected registers, the power leakage will depend on:

$$L = s_7 \oplus F(S^0) \oplus i_0 + F(S^0) \oplus i_0 \oplus F(S^1) \oplus i_1.$$

Here i_0 becomes part of $F(S^1)$, interacting non-linearly with other key bits to generate the output. This equation reveals information leakage that can be used by the adversary to break the system. Here, $F(S^1)$ compromises SCA-security of the system but the system is not completely broken yet. The reason is that the attacker can control only one bit-input of an 8-to-1 non-linear function. If function $F(S^1)$ is balanced over its input bits, the input data sequence will cause the output to flip in 50% of the cases, i.e. half of the secret space will be equally ranked first in the analysis, hiding the original secret.

The adversary can aggregate knowledge about the first two registers by addressing both $F(S^0)$ and $F(S^1)$ as highlighted in the equation above, to further reduce SCA-security of the system. In principle, the adversary makes a hypothesis over the initial state of the register (the key), and predicts the output of the feedback function in each clock cycle based on the input data. Then, he focuses on clock cycle number x to predict data-dependent power variations affecting all the taps $[1 : x]$. Increasing the number x by one reduces complexity of the

unknown secret by at least one bit. The exact SCA-security reduction depends on nonlinearity of the feedback function. In this example, attacking clock cycle number 8 can uniquely determine the secret key.

To conclude, any regular NLFSR that is similar to *Model I* can be broken by SCA, regardless of complexity of the feedback function. Next, we will propose a novel modification in order to improve security of the structure.

4.2 Toy Model II: Two 8-bit NLFSRs with rotating cross-connect

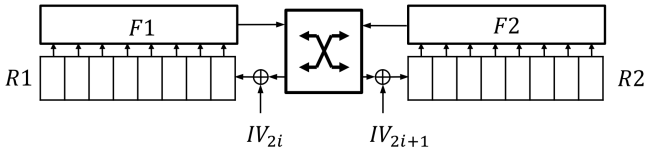

Fig. 3. Toy Model II: Two 8-bit NLFSRs with rotating cross-connect

In this model, we use two 8-bit registers, $R1$ and $R2$, each with its own function $F1(S1)$ and $F2(S2)$, as shown in Fig. 3. Here, $F1(S1)$ (or $F2(S2)$) is a non-linear function over the state bits of register $R1$ (or $R2$, respectively). Here, the feedback functions are connected to the register using a rotating cross-connect. In the odd clock cycles, the output of each function is normally connected back to its own register. In the even clock cycles, the output of function $F1(S1)$ is connected to $R2$, while $F2(S2)$ is connected to $R1$. Also, the system accepts two fresh IV bits per clock. Each bit is `xor`ed with the output of the non-linear functions before being stored in the first tap of the register.

Analysis of the system in the first clock cycle is equivalent to the previous model. Although the algorithmic noise here is higher, this noise alone cannot support sound security against SCA attacks.

In the second clock cycle, the data-dependent power leakage will be:

$$\begin{aligned} L = {} & s1_7 \oplus F1(S1^0) \oplus i_0 + F1(S1^0) \oplus i_0 \oplus F2(S2^1) \oplus i_2 \\ & + s2_7 \oplus F2(S2^0) \oplus i_1 + F2(S2^0) \oplus i_1 \oplus F1(S1^1) \oplus i_3. \end{aligned}$$

where $s1, s2$ are the state bits of register $R1$ and $R2$, respectively. Similarly, $S1$ and $S2$ represent the state of registers.

The first part of this equation represents power variations in the first register $R1$, while the other part represents $R2$. The equation shows the effect of using two registers with a cross-connect. If the adversary predicts the initial state one register $S1$ (or $S2$), $F2(S2)$ (or $F1(S1)$ respectively) will act as a source of data-dependent noise. In this case, the adversary can still break the system, but only with a hypothesis over the entire secret space (16 bits in this particular example).

To the best of our knowledge, this is the first cryptographic structure that can combine the effect of two non-linear functions, while being immune against the divide-and-conquer principle of SCA.

One may think that the adversary may try a specially crafted input sequence to focus on manipulating only one register. One possible choice is to switch between one random bit and one fixed bit (i_0 0 i_2 0 i_4 0...). However, this will not result in any better attack. Register $R2$ will still show data-dependent variations brought by the other feedback function ($F1(S1^1)$) as highlighted in the equation above.

To conclude, more than one non-linear register can be combined using a rotating cross-connect to increase the secret space of the structure.

4.3 Toy Model III: Two 8-bit registers with 4-bit feedback function

In this section, we try to answer the interesting question: Should SCA-security depend on the register length or the number of bits involved in evaluating the feedback function? In the previous models, the two numbers were identical. In Model III, we keep the length of registers as 8-bits, but we use 4-to-1 nonlinear feedback functions. For instance, we assume that only the odd numbered taps are connected to the feedback function.

Here, the data-dependent power leakage in the second clock cycle will be:

$$\begin{aligned} L = {} & s1_7 \oplus F1(S1^0_{odd}) \oplus i_0 + F1(S1^0_{odd}) \oplus i_0 \oplus F2(S2^1_{even}) \oplus i_2 \\ & + s2_7 \oplus F2(S2^0_{odd}) \oplus i_1 + F2(S2^0_{odd}) \oplus i_1 \oplus F1(S1^1_{even}) \oplus i_3. \end{aligned}$$

where S_{odd} represents the odd numbered taps, while S_{even} represents the even numbered taps.

The equation shows that the adversary needs a correct hypothesis over the entire register in order to correctly model the power consumption. This is true if knowledge about the value of some taps does not help in predicting the output of the feedback function in the next clock cycle. This is best achieved by connecting the feedback function over the odd taps of the register.

To conclude, we do not have to find a nonlinear feedback function that covers all the register taps. Interestingly, a feedback function that connects only half of the taps can provide the same level of SCA-security. Feedback functions with a smaller number of inputs will have a degraded level of SCA-security.

5 `Keymill`, The Proposed Design

Our first option for a full system with 128-bits of secret key is to combine 16 8-bit registers using `AES` SBox as a non-linear function. However, the size of the rotating cross-connect circuit will be significant. More importantly, the combined secret space will be less-optimal. One full rotation of the cross-connect will require 16 cycles, while the IV input will vanish in only 8 clock cycles (assuming 16-bits input per clock with IV of 128 bits).

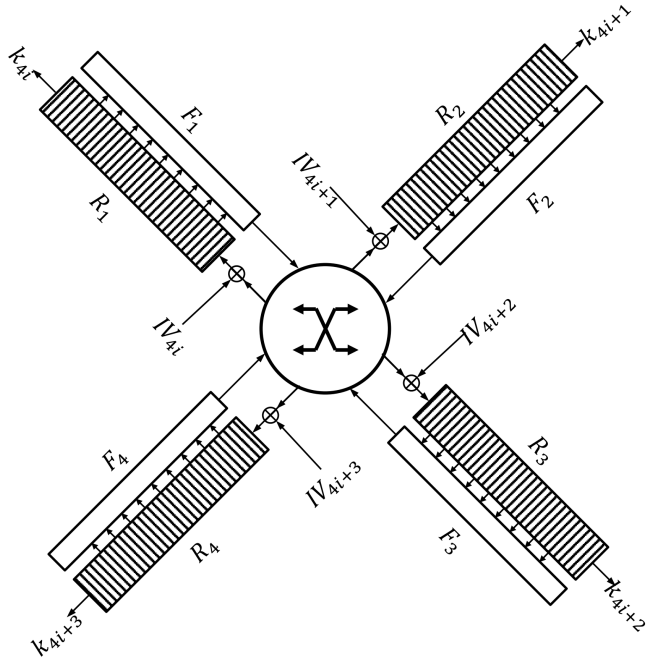

Fig. 4. Structure of `Keymill`, the proposed keystream generator

Hence, we propose to use only four registers featuring 8 full rotations while accepting new IV bits, which will be done within $128/4 = 32$ clock cycles.

Unfortunately, there is no theory on how to construct NLFSRs with good cryptographic properties and long period. However, there are many constructions in the literature that have been carefully designed for cryptography with a guaranteed maximum period. Hence, we will not design a new NLFSR. Instead, we will focus on how to use one of the established NLFSRs in the proposed construction.

Achterbahn [8] is a cryptographic stream cipher that was designed as part of the eSTREAM competition. An innovative part of Achterbahn stream cipher is the design of 13 primitive NLFSRs of different sizes (21 bits to 33 bits). Although Achterbahn did not advance to the eSTREAM portfolio due to some limitations in the combining function, which we are not using here, the NLFSRs are still a valuable contribution. Next, we will use three of the Achterbahn NLFSRs to build the proposed scheme.

The proposed construction is composed of four NLFSRs, as shown in Fig.4. Register $R1$ is a 31-bit register. Registers $R2$ and $R3$ are 32 bits each. Register $R4$ has 33 bits. We use feedback functions from Achterbahn stream cipher [8]

(where they are named A_{10}, A_{11}, A_{12}) with the following equations:

$$\begin{aligned}F1(S) =& s_0 + s_2 + s_5 + s_6 + s_{15} + s_{17} + s_{18} + s_{20} + s_{25} + s_8 s_{18} \\ &+ s_8 s_{20} + s_{12} s_{21} + s_{14} s_{19} + s_{17} s_{21} + s_{20} s_{22} + s_4 s_{12} s_{22} + s_4 s_{19} s_{22} \\ &+ s_7 s_{20} s_{21} + s_8 s_{18} s_{22} + s_8 s_{20} s_{22} + s_{12} s_{19} s_{22} + s_{20} s_{21} s_{22} \\ &+ s_4 s_7 s_{12} s_{21} + s_4 s_7 s_{19} s_{21} + s_4 s_{12} s_{21} s_{22} + s_4 s_{19} s_{21} s_{22} \\ &+ s_7 s_8 s_{18} s_{21} + s_7 s_8 s_{20} s_{21} + s_7 s_{12} s_{19} s_{21} + s_8 s_{18} s_{21} s_{22} \\ &+ s_8 s_{20} s_{21} s_{22} + s_{12} s_{19} s_{21} s_{22}.\end{aligned}$$

$$\begin{aligned}F2(S) = F3(S) =& s_0 + s_3 + s_{17} + s_{22} + s_{28} + s_2 s_{13} + s_5 s_{19} + s_7 s_{19} + s_8 s_{12} \\ &+ s_8 s_{13} + s_{13} s_{15} + s_2 s_{12} s_{13} + s_7 s_8 s_{12} + s_7 s_8 s_{14} + s_8 s_{12} s_{13} \\ &+ s_2 s_7 s_{12} s_{13} + s_2 s_7 s_{13} s_{14} + s_4 s_{11} s_{12} s_{24} + s_7 s_8 s_{12} s_{13} \\ &+ s_7 s_8 s_{13} s_{14} + s_4 s_7 s_{11} s_{12} s_{24} + s_4 s_7 s_{11} s_{14} s_{24}.\end{aligned}$$

$$\begin{aligned}F4(S) =& s_0 + s_2 + s_7 + s_9 + s_{10} + s_{15} + s_{23} + s_{25} + s_{30} + s_8 s_{15} \\ &+ s_{12} s_{16} + s_{13} s_{15} + s_{13} s_{25} + s_1 s_8 s_{14} + s_1 s_8 s_{18} + s_8 s_{12} s_{16} \\ &+ s_8 s_{14} s_{18} + s_8 s_{15} s_{16} + s_8 s_{15} s_{17} + s_{15} s_{17} s_{24} + s_1 s_8 s_{14} s_{17} \\ &+ s_1 s_8 s_{17} s_{18} + s_1 s_{14} s_{17} s_{24} + s_1 s_{17} s_{18} s_{24} + s_8 s_{12} s_{16} s_{17} \\ &+ s_8 s_{14} s_{17} s_{18} + s_8 s_{15} s_{16} s_{17} + s_{12} s_{16} s_{17} s_{24} + s_{14} s_{17} s_{18} s_{24} \\ &+ s_{15} s_{16} s_{17} s_{24}.\end{aligned}$$

Hence, the internal state of the structure is 128-bits, similar to the common length of `AES`' key. The feedback functions are mixed using rotating cross-connect as follows. Assuming that $i = [1, 5, 9, 13, ...]$,

Clock cycle i: F1 $\rightarrow$ R1, F2 $\rightarrow$ R2, F3 $\rightarrow$ R3, F4 $\rightarrow$ R4
Clock cycle $i+1$: F1 $\rightarrow$ R2, F2 $\rightarrow$ R3, F3 $\rightarrow$ R4, F4 $\rightarrow$ R1
Clock cycle $i+2$: F1 $\rightarrow$ R3, F2 $\rightarrow$ R4, F3 $\rightarrow$ R1, F4 $\rightarrow$ R2
Clock cycle $i+3$: F1 $\rightarrow$ R4, F2 $\rightarrow$ R1, F3 $\rightarrow$ R2, F4 $\rightarrow$ R3

The structure starts by loading the secret key into the internal state. Then on each clock cycle, four bits from the IV, one for each register, are added to the feedback functions. After $128/4 = 32$ clock cycles, adding the IV should be completed. Then, the structure goes free running without inputs or outputs for 33 clock cycles, equivalent to the length of the longest register. From this point forward, the NLFSRs will generate 4 bits in each clock cycle, one from each register.

Every 128 bits of the output should be used only once to encrypt plain data with the `AES` block cipher.

6 Security Analysis

First of all, we selected 4 registers with a total number of taps equal to 128 bits in order to preserve the entropy of the secret key. Also, the output is not generated until after 33 clock cycles from the acceptance of the last IV bit. This number allows the last input bit to go through all the taps of the longest register, which allows the structure to distribute its effect over all the internal state. This lets each unique IV generate a unique bit stream.

Regarding SCA-security, the number of taps that are connected to functions $F1(S)$, $F2(S)$ and $F3(S)$ is 17 bits each. The number of connected taps to $F4(S)$ is 18 bits. The unique taps that are connected to each register are as follows:

$$\begin{aligned} F1(S) &: 0, 2, 4, 5, 6, 7, 8, 12, 14, 15, 17, 18, 19, 20, 21, 22, 25 \\ F2(S), F3(S) &: 0, 2, 3, 4, 5, 7, 8, 11, 12, 13, 14, 15, 17, 19, 22, 24, 28 \\ F4(S) &: 0, 1, 2, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 23, 24, 25, 30 \end{aligned}$$

The number of connected taps in each register is slightly higher than half the register length, with a good distribution that is close to the distribution recommended in Model III (see Sec. 4.3). Hence, it is very reasonable to declare that the registers will have SCA-security equivalent to their length (31, 32, 32, and 34 bits).

Moreover, the feedback functions are mixed with a rotating cross-connect that is similar to Model II (see Sec. 4.2). Hence, SCA-security of the system will be equivalent to the aggregated length of the involved registers, which is (31+32+32+34 = 128 bits). Essentially, the adversary cannot make an accurate estimate about the data-dependent power changes in the structure unless he makes a correct hypothesis over the entire secret key.

To the best of our knowledge, this is the first cryptographic structure ever proposed that has an SCA-security that equals its brute force security.

6.1 Failure of other NLFSRs

NLFSRs have long been used in the design of stream ciphers including, `KeeLoq` [6] and `Grain` [10] as notable examples. One of the design principles that we followed in this proposal is to limit the attacker's control on the internal behavior by shrinking the data-width of public data that is used in each operation. In this regard, `KeeLoq`'s design features control the entire internal state, while the key bits are used one at a time [6]. `Grain` makes use of a non-linear function (called H) that involves four bits of the input data and one bit key [7]. Another key difference is that the output of the aforementioned non-linear functions are feedback to the register, which originally holds the input data. Hence, the previous state of the HD power model is known to enable an easy recovery of the new state.

6.2 Similarity to GGM Structures

GGM is a tree-based structure, named after its inventors [9], that is used to realize pseudorandom functions from any pseudorandom generator. It was reintroduced in many recent contributions as a structure that is capable of initializing leakage resilient primitives in an SCA-secure manner [2, 4, 16].

The GGM structure starts from the secret key and inserts the IV one bit at a time followed by a randomization step. The value of each bit determines the next branch in the tree. The randomization step is used to distribute the effect of the inserted bit over the entire internal state. Hence, the attacker will face a new secret at each step, which renders DPA attacks almost impossible. Randomization can be realized using block ciphers [13] or hashing functions [11].

The proposed `Keymill` is similar to the GGM in accepting one bit of the IV at each clock cycle, however the core concept for SCA-protection is different. GGM employs a *leaky* function, that is not secure against SCA attacks, to build an SCA-secure algorithm. Here, the randomization primitive used in the GGM (block cipher or hashing function) is still vulnerable to SCA attacks, while protection is achieved by preventing the adversary from aggregating information across different executions. On the contrary, SCA-security of the proposed `Keymill` depends on expanding the size of key hypotheses to the full size of the secret key. Hence, `Keymill`, as an isolated primitive, is inherently secure against SCA attacks without being part of any special algorithm.

6.3 Cautionary Notes

There are a couple of cautionary notes that come with this new protection mechanism.

Proposing a new masking or hiding countermeasure must be evaluated with actual power consumption traces. This is a typical requirement in order to ensure that the engineering defects (glitches and balanced routing) are resolved. On the contrary, we found it difficult to evaluate the SCA-security of our scheme on the same grounds. The reason is that our countermeasure depends on expanding the size of key hypothesis to 128 bits. Hence, we could not enumerate all the possible 2^{128} cases in order to measure feasibility of the proposed scheme. Rather, we built our security on mathematical modeling. In fact, this is in line with leakage resilient schemes that depend on updating the secret key after each run.

Another similarity with leakage resilient schemes is that the proposed countermeasure slightly changes the algorithm. Hence, the same algorithm needs to be applied in the two sides of communication even if one side is physically protected (server or so).

One last note is that high entropy of the input key is required in order to generate high-entropy keystream. In other words, if the input key is all zeros, the output bit stream will also be zeros. This limitation is inherited from the Achterbahn NLFSRs [8]. Here, we focused exclusively on SCA-security and we did not add any cryptographic part to break symmetry of the scheme, which can be a topic for future improvement.

Table 1. Comparison against similar schemes

Contribution	Area (GE)	Clock cycles
Modular Mul of [12]	7,300	562
Minimum SP network of [2]	5,302	61
The proposed `Keymill`	775	97

7 Hardware results

Using the hardware budget of the individual NLFSRs as discussed in [8], the hardware cost of the proposed structure at a low-Vt 1.5V standard cell library targeting 130 nm CMOS technology is:

$$Area = 608 + 125.5 + 41.5 = 775\ GE$$

The $(4.75 \times 128) = 608$ GE covers the internal state of the registers. The $(31.75 + 31 + 31 + 31.75) = 125.5$ GE covers the feedback functions $F1$, $F2$, $F3$, and $F4$ respectively. The 41.5 GE covers the rotating cross-connect. The rotating cross-connect can be implemented very efficiently using four 4-to-1 multiplexers ($4 \times 7.5 = 30$ GE) and a 2-bit counter (11.5 GE).

A comparison between the hardware cost of the proposed scheme and that of the previous work is shown in Table 1. The results of [12] are taken at the first-order masked implementation, while the results of the minimum SP network are taken at the lightest implementation of [2]. The table shows superiority of the proposed scheme in terms of both area and clock cycles.

Also, the proposed scheme shows superior performance over typical masking schemes. The smallest threshold implementation of `AES` (to prevent leakage caused by glitches) requires 8,393 GE of area overhead [14].

Although there is no initialization required for threshold implementations, the initialization overhead of our scheme requires only 65 clock cycles. Then, one key is generated every 32 clock cycles.

8 Conclusion

In this paper, we have proposed a new solution to SCA attacks. Our solution depends on a new cryptographic structure that expands the size of key hypothesis, and breaks the divide-and-conquer principle of SCA. Our structure is generic, lightweight and can turn any block cipher into an SCA-secured encryption mode.

Acknowledgment

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under the Discovery and Discovery Accelerate Supplement (DAS) Grants awarded to A. Reyhani-Masoleh.

References

1. Information technology, security techniques, authenticated encryption. In *ISO/IEC 19772:2009*. Retrieved March 12, 2013.
2. S. Belaid, F. D. Santis, J. Heyszl, S. Mangard, M. Medwed, J.-M. Schmidt, F.-X. Standaert, and S. Tillich. Towards fresh re-keying with leakage-resilient PRFs: Cipher design principles and analysis. Cryptology ePrint Archive, Report 2013/305, 2013.
3. S. Burman, D. Mukhopadhyay, and K. Veezhinathan. LFSR based stream ciphers are vulnerable to power attacks. In *Progress in Cryptology–INDOCRYPT 2007*, pages 384–392. Springer, 2007.
4. Y. Dodis and K. Pietrzak. Leakage-resilient pseudorandom functions and side-channel attacks on feistel networks. In *CRYPTO*, pages 21–40, 2010.
5. M. Dworkin. NIST special publication 800-38A, recommendation for block cipher modes of operation: Methods and techniques.
6. T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani. On the power of power analysis in the real world: A complete break of the KeeLoq code hopping scheme. In *Advances in Cryptology, CRYPTO 2008*, page 203:220. Springer, 2008.
7. W. Fischer, B. M. Gammel, O. Kniffler, and J. Velten. Differential power analysis of stream ciphers. In *Topics in Cryptology–CT-RSA 2007*, pages 257–270. Springer, 2007.
8. B. M. Gammel, R. Göttfert, and O. Kniffler. Achterbahn-128/80. In *eSTREAM, ECRYPT Stream Cipher Project*. 2006.
9. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.
10. M. Hell, T. Johansson, and W. Meier. Grain: a stream cipher for constrained environments. *International Journal of Wireless and Mobile Computing*, 2(1):86:93, 2007.
11. P. Kocher. Complexity and the challenges of securing SoCs. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, page 328331, 2011.
12. M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni. Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In *Progress in Cryptology, AFRICACRYPT 2010*, page 279:296. Springer, 2010.
13. M. Medwed, F.-X. Standaert, and A. Joux. Towards super-exponential side-channel security with efficient leakage-resilient PRFs. In *Cryptographic Hardware and Embedded Systems, CHES 2012*, page 193:212. Springer, 2012.
14. A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang. Pushing the limits: a very compact and a threshold implementation of AES. In *Advances in Cryptology, EUROCRYPT 2011*, page 69:88. Springer, 2011.
15. M. Robshaw and O. Billet. New stream cipher designs: the estream finalists. volume 4986. Springer, 2008.
16. F.-X. Standaert, O. Pereira, Y. Yu, J.-J. Quisquater, M. Yung, and E. Oswald. Leakage Resilient Cryptography in Practice. In *Towards Hardware-Intrinsic Security*, pages 99–134. 2010.
17. A. A. Zadeh and H. M. Heys. Simple power analysis applied to nonlinear feedback shift registers. *Information Security, IET*, 8(3):188–198, 2014.