Mutual Information Analysis for Three-Phase Dynamic Current Mode Logic against Side-Channel Attack

Hyunmin Kim, Dong-Guk Han, and Seokhie Hong

To date, many different kinds of logic styles for hardware countermeasures have been developed; for example, SABL, TDPL, and DyCML. Current mode–based logic styles are useful as they consume less power compared to voltage mode–based logic styles such as SABL and TDPL. Although we developed TPDyCML in 2012 and presented it at the WISA 2012 conference, we have further optimized it in this paper using a binary decision diagram algorithm and confirmed its properties through a practical implementation of the AES S-box. In this paper, we will explain the outcome of HSPICE simulations, which included correlation power attacks, on AES S-boxes configured using a compact NMOS tree constructed from either SABL, CMOS, TDPL, DyCML, or TPDyCML. In addition, to compare the performance of each logic style in greater detail, we will carry out a mutual information analysis (MIA). Our results confirm that our logic style has good properties as a hardware countermeasure and 15% less information leakage than those secure logic styles used in our MIA.

Keywords: TPDyCML, mutual information analysis, side-channel attack, information theoretic analysis.

Manuscript received Mar. 10, 2014; revised Dec. 1, 2014; accepted Jan. 28, 2015.

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1003) supervised by the IITP (Institute for Information & communications Technology Promotion).

Hyunmin Kim (hmkim.sec@gmail.com) and Seokhie Hong (corresponding author, shhong@korea.ac.kr) are with the Center for Information Security Technologies, Korea University, Seoul, Rep. of Korea.

Dong-Guk Han (christa@kookmin.ac.kr) is with the Department of Mathematics, Kookmin University, Seoul, Rep. of Korea.

I. Introduction

Modern cryptosystems, though mathematically proven to be secure, are potentially vulnerable to *physical* attacks. Physical attacks, such as side-channel analysis (SCA), exploit additional information via side channels; hence, they are often referred to as side-channel attacks (SCAs) — the first of which was introduced by P. Kocher and others [1].

A power analysis attack is an SCA that reveals secret data by monitoring the power consumption of a CMOS cryptographic device, which is closely related to the internal state of the device. Recently developed hardware countermeasures against this type of SCA have been designed to eliminate such a vulnerability. For a hardware countermeasure to be efficient, it should address the transistor level. The most commonly used hardware design methods adopt a top-down design flow and automatic logic synthesis. As indicated in [2], automatic logic synthesis is vulnerable to power analysis due to an unbalanced load capacitance. Therefore, it is important to design a logic style that is both suitable for combining it with automatic logic synthesis and secure against power analysis in this design method. Additionally, unexpected side-channel vulnerabilities can appear in hardware implementations due to the data-dependent power consumption of logic gates. For these reasons, various countermeasures against SCAs are designed at the transistor level. Secure logic styles with data-independent power consumption can be classified into the following two categories: dual-rail precharge (DRP) logic style and current mode logic (CML) style.

The DRP logic style has a precharge phase and utilizes

complementary wires on dual outputs. The most important characteristic of this logic style is that the power consumption is independent of the processed signal values if all dual outputs have the same capacitive load. The representative logics of this category include sense amplifier–based logic (SABL) and three-phase dual-rail pre-charge logic (TDPL) [2]–[3].

CML style operates using a small voltage swing at the outputs and produces a constant current at the internal nodes. Because of the small swing operation, these logic styles are suitable for low power circuit design. The current mode–based logic styles consume 50% less power than DRP logic styles [4].

MOS CML (MCML) [5] can be used as a secure CML style against power analysis. However, high static power consumption and difficulty in making the impedance of the dual rail balanced are its main drawbacks. For these reasons, dynamic CML styles, such as dynamic current mode logic (DyCML) [4], [6]–[7], have been proposed as an alternative.

However, most of the proposed secure logic styles suffer from the problem of unbalanced capacitance, which is a direct consequence of routing, implementation environment, and process condition. The resulting asymmetry is a weak point that makes all hardware designs vulnerable to power analysis [8]–[9]. For this reason, it is necessary to compensate for unbalanced capacitive loads.

Since its development, SABL has been modified and improved to increase its ability to resist SCAs; in particular, it has been developed so as to create a more balanced capacitance [10]. Although a symmetrical internal rail dividing method using back annotation or a fat wire layout [11] has been proposed, it doesn't have a perfectly balanced capacitive load due to process variation and coupling capacitance [12]. To overcome this problem more efficiently, TDPL based on the DRP logic style was introduced.

In comparison to other DRP logic styles, TDPL utilizes an additional third phase; that is, a discharge phase. This discharge phase results in constant energy consumption per clock cycle. Even though this logic style has high side-channel security, it consumes twice as much power as the other DRP logic styles. This increased power consumption limits the use of TDPL in small cryptographic devices such as RFID tags and smart cards.

In [13], we presented a novel, secure logic style — three-phase dynamic current mode logic (TPDyCML) — that makes use of a three-phase technique.

The proposed logic style is based on DyCML, and as a result, it produces a low power consumption due to its small voltage swing operation. This advantage, which originates from the CML logic style, can be combined with the three-phase technique for small cryptographic devices against SCA. This logic style is robust against an unbalanced load capacitance; thus, it can be used in an automated design environment without additional routing restrictions. However, even though TPDyCML has good normalized energy deviation (NED) and normalized standard deviation (NSD), this is not enough to confirm that it has good characteristics as a hardware countermeasure. The fact that the NED and NSD, which are used in [13] as evaluation criteria, only take into account a few particular adversaries means that worst-case scenarios are not considered; however, these criterion can provide a good starting point for those wishing to develop a hardware countermeasure against SCAs. An NED represents the amount of variation in power consumption per cycle. If the variation is small, then an adversary needs more measurements through precise measuring devices to exploit any side-channel information. Thus, in such a case, an adversary who can accurately measure and specifically target dynamic logic can predict side-channel information with higher probability. The other aforementioned evaluation criterion, the NSD, indicate the sum of various independent identical distributions. The NSD will be close to that of a normal Gaussian distribution. This distribution is confirmed assuming practical noisy implementation and a measuring environment that have a particular noise level (no noise, a small amount of noise, and so on). Especially, NSD only takes into consideration an adversary that has knowledge of every environment and the fact that the power consumptions of environments are distributed normally at the same noise level. Therefore, these are only appropriate for this particular type of adversary [4], [8].

In this paper, we have further optimized our logic style using the BDD algorithm [14]. The BDD algorithm is able to make our logic style more compact through the use of an NMOS logic tree. It improves the performance of our logic style and reduces its implementation area. It is also possible to share certain components, such as a self-timing buffer, on the same logic level. For more practical testing and confirmation of the improved security functionality of TPDyCML, we simulated the power consumptions of AES S-boxes, which were constructed from a compact NMOS tree and optimized by a BDD algorithm, with different logic styles, such as SABL, TDPL, DyCML, TPDyCML, and CMOS.

For the method of analysis, we first adopt a correlation power attack (CPA) [15] using a Hamming distance model to the AES S-box, because such an attack is normally used in SCAs. However, as in [4] and [8], it is difficult to compare secure logic styles when using Hamming distance and Hamming weight models.

At a simulation using an AES S-box with CMOS logic style, we can find the right key at just using 3,500 measurement traces. But, using an AES S-box with secure logic styles such as SABL, TDPL, DyCML, and TPDyCML, we can't find the right key even if we were to apply 20,000 measurement traces.

Therefore, CPAs with a Hamming distance and Hamming weight model can't be applied to a hardware countermeasure with a secure logic style, because the implementation further increases the nonlinearity between power consumption and internal state due to the dual-rail configuration [8].

To solve this issue, a unified framework, which is normally used to compare secure logics, was introduced in [16]. A mutual information analysis (MIA) that uses an information-theoretic analysis [17] is a more efficient method of comparing secure logics with a dual-rail configuration. This method exploits information leakages from implementations of each secure logic style and quantifies the information. So, it is possible to compare the performance quality of hardware countermeasure logic styles independent of whether an adversary has knowledge of the attack model and of what the implemented configuration is (for example, CMOS, secure logics, and so on). This is because an MIA only considers an information leakage to be from an implementation, regardless of how to configure it.

To be specific, the power consumption of CMOS logic is dynamic and only depends on the probability of output transitions of a transistor (intermediate value). Thus, adversaries have to consider output capacitance loads without being aware of the design. As such, it is possible to precisely evaluate side-channel information just by predicting intermediate value. In this case, an adversary knows that the attack model is Hamming distance, the implementation consists of CMOS logic, and that they are to assume that the environment has the same noise level. We call such an adversary a *strong* adversary, because the correlation between power consumption and intermediate value can be increased according to the accuracy of the power consumption model used for the prediction of security information.

In contrast with CMOS logic, in the case of a dual-rail logic style, such as secure logic with dynamic and differential configurations, output capacitance is independently loaded with input transitions and intermediate values. The differences that occur in the power consumptions of such logic styles are closely related to the parasitic capacitances of the corresponding hardware designs. Thus, if we apply the previous analysis method to these secure logics, then accurate prediction of secure information is very hard without specific transistor-level knowledge. The previous analysis method is only useful in the best-case scenario; that is, where the adversary has specific and deep knowledge about transistor-level operation and knows the power consumption precisely (that is, a strong adversary).

In contrast, an MIA is considered a worst-case scenario, in which adversaries don't know the particular implementation and measurement setting, as various noise levels compare with what NED and NSD consider to be a particular adversary who knows the implementation of the target and measurement environments. Such an adversary also doesn't have a deep knowledge of transistor-level operation, because the analysis just uses leaked information from the implementation. Therefore, an MIA is very useful for comparing security characteristics with secure logics because an MIA considers the various environments and measurement environments and does not target specific implementations such as CMOS [8].

As a result of our own MIA, we confirm that our TPDyCML reduces information leakage by 15% compared to other secure logic styles. To improve even further, we can apply several techniques to BDD optimization. In [9], they recommend balancing methods to improve the resistance characteristics of the implementation with secure logic against SCAs. First, the same load has to be configured in each of the output nodes of the NMOS tree. Second, the independent numbers of series transistors with input transition should be connected to each of the output nodes. Third, the symmetry layout of the NMOS tree and balanced routing guarantees a balanced interconnected capacitance. Therefore, those methods provide a more greatly improved resistance to side-channel analysis. Based on evaluation through MIA, TPDyCML has a security advantage as a hardware countermeasure against SCAs.

The remainder of this paper is as follows: Section II reviews our proposed TPDyCML. Section III describes the method we use to optimize the AES S-box using the BDD algorithm. In addition, the CPA simulation results between CMOS and TPDyCML are provided in Section IV. In Section V, an MIA is conducted on an AES S-box implementation with secure logics and CMOS to compare the performance as a hardware countermeasure against SCAs. Finally, Section VI concludes the paper and highlights its contribution. Also, we suggest further work that needs to be done in the conclusion section.

II. TPDyCML

In [13], we developed TPDyCML. This logic style has an improved security functionality compared with other secure logic styles. In the previous paper regarding TPDyCML, we confirmed the characteristics in an unbalanced capacitance from an unexpected environment. In this section, we simply review TPDyCML.

1. Structure of TPDyCML

Figure 1 shows the basic architecture of TPDyCML. The TPDyCML gate consists of the following: a differential logic tree, precharge circuits (P1, P4), virtual ground circuits (C1–C2), a p-type latch to preserve a logic value after evaluation (P2, P3), the dynamic operation part of TPDyCML (Q1, Q2),

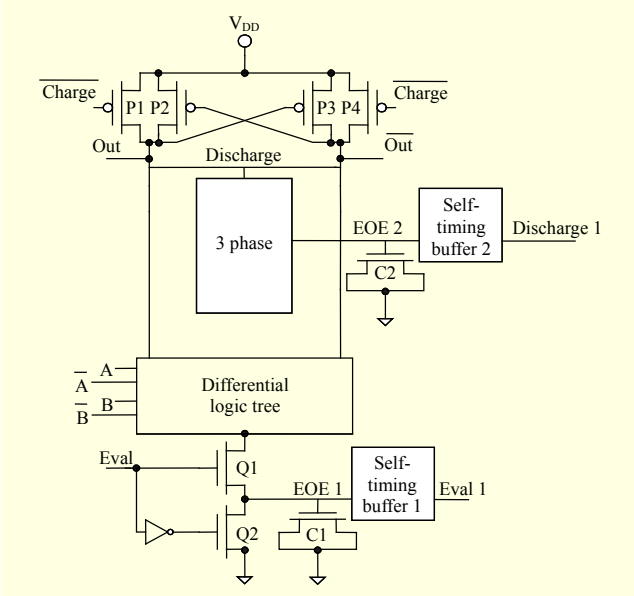


Fig. 1. Structure of TPDyCML.

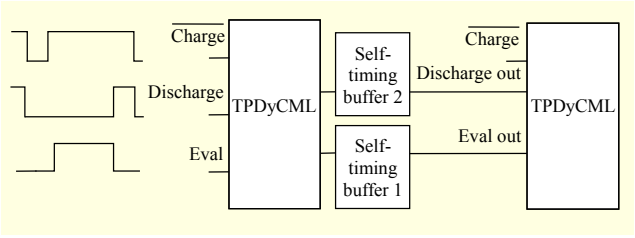


Fig. 2. Cascaded mechanism of TPDyCML.

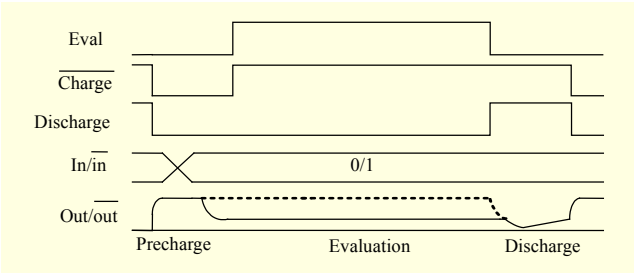


Fig. 3. Timing diagram of TPDyCML.

the three-phase part (additional discharge), and one additional self-timing buffer that is used to enable another clk (that is, discharge 1 and discharge 2, and so on) delay making TPDyCML operation more robust.

Discharge phase control transistors are added to enable TPDyCML to operate in a three-phase mode, just as in TDPL. The additional discharge phase makes the consumption of energy balanced. Thus, the following operation occurs: first, during the evaluation phase, the appropriate line is discharged to the GND in accordance with processed data. Second, the discharge control transistors are switched on just before the end of the clock cycle. Finally, the unbalanced charge at the output nodes discharges simultaneously. Thus, the amount of energy consumed over one clock cycle becomes constant.

2. Operation of TPDyCML

A. Precharge Phase

At the beginning of every clock cycle, output node capacitances are charged to a high logic value. The upper PMOS transistors (P1, P4) of the circuit switch on when the charge signal is at high level and the power source charges output node capacitances. In addition, transistor Q2 switches on and the charge stored in the virtual ground flows to the GND because the Eval signal is also set to a low level.

B. Evaluation Phase

In this phase, the PMOS transistor (P1, P4) switch is set to "OFF." The charge stored in the output node flows in the source of transistor Q1 through a path in the logic block that corresponds to input values. Since the Eval signal is high, transistor Q1 switches on, and a discharge current then flows to the virtual ground.

C. Discharge Phase

In this phase, the residual charge stored in the output node is removed through transistors Q3 and Q4 in the three-phase block. Even if TPDyCML-style gates have an unbalanced load capacitance, this operation balances power consumption. Figures 2 and 3 show the cascaded mechanism of TPDyCML and a timing diagram, respectively.

III. AES S-Box Implementation

We optimize the AES composite field, and basic Boolean arithmetic units are configured to compact the NMOS tree. It is compacted using a binary decision diagram (BDD). We referred to [18]–[19] to make the optimized AES composite field (see Fig. 4). The AES composite field with BDD optimization increases the operating speed and reduces the power consumption of the AES compared to other AES implementations that are done without BDD optimization.

1. BDD

A Boolean function is a function that is related to only Boolean values. In the context of this paper, a Boolean value is dependent upon which input assignment is inserted into it. Boolean functions are described using a truth table and a Boolean expression, which are formulae of a basic Boolean operation. Furthermore, in a graph, a Boolean function is

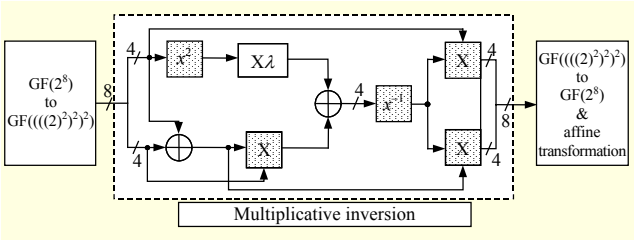


Fig. 4. AES S-box.

assigned to each path that represents one assignment of input variables, such as in a binary decision tree.

To further optimize the Boolean function, it is important to not to raise the number of inputs too quickly and to make it compact using a compact NMOS tree at logic implementation. Additionally, knowing how to find the appropriate assignment of the input variable through evaluating the Boolean function correctly is important to efficiently manipulating logic [14].

Since S.B. Akers, in [14], first introduced BDD using a binary decision tree, BDD has been used in lots of research fields, including cryptographic hardware for logic optimization. BDD is based on a recursive manipulation, such as that of recursively dividing into function F point to function F(X=0) and F(X=1), in accordance with Shannon's expansion theorem. Also, BDD is often used to optimize and minimize the Boolean representation fast and efficiently.

The most important characteristics of BDD are as follows. In the same function, the edge of BDD can be shared, and on each path in the diagram, it can easily order the variables, especially input variables (for example, fix the order of input variables). BDD uses the If-Then-Else algorithm to optimize and unite different Boolean functions so that it can convert very large circuits into more compact ones with improved performance in terms of speed and area. Excluding the above-mentioned, there are also additional advantages to designing hardware countermeasures against SCAs through making an NMOS tree balanced using BDD.

2. AES Composite Field Optimization Using BDD

Our design doesn't count the number of normalized capacitances of each input sequence using BDD as mentioned in [20]. However, we can obtain a BDD using the four steps explained in the subsections below. Now, we explain in a step-by-step manner how to use BDD to implement inversion functions of the AES composite field.

A. Identification

First of all, the complex function has to be identified. For example, an inversion block, GF(2)^4, of the AES S-box is constructed from the following four inversion functions, explained in [19]:

- $q_0^{-1} = q_3q_2q_1 \oplus q_3q_2q_0 \oplus q_3q_1 \oplus q_3q_1q_0 \oplus q_3q_0 \oplus q_2 \oplus q_2q_1 \oplus q_2q_1q_0 \oplus q_1 \oplus q_0$
- $q_1^{-1} = q_3 \oplus q_3q_2q_1 \oplus q_3q_1q_0 \oplus q_2 \oplus q_2q_0 \oplus q_1$
- $q_2^{-1} = q_3q_2q_1 \oplus q_3q_2q_0 \oplus q_3q_0 \oplus q_2 \oplus q_2q_1$
- $q_3^{-1} = q_3 \oplus q_3q_2q_1 \oplus q_3q_0 \oplus q_2$

B. Finding Optimal Logic Depth

Second, record the truth table of that function (the inversion functions here) using the Karnough Map and use it to find the optimal logic depth. As per our design, the maximum logic depth is 20. It can be operated at 1 MHz or 100 kHz for smart cards and RFIDs.

C. Describe BDD Diagram

Third, make a BDD diagram as explained in [20]. When the two child vertices of a vertex v have the same value, the child of v is replaced by the mother vertex v. In other words, the BDD represents only the essential variables. The reduced vertex has the same function as the vertex before the modification. If all of the replaced vertices and their children have the same function, then they will have to remove and substitute a particular vertex.

D. Mapping

Fourth, map the BDD to a DPDN network as explained in [21]. In this step, the isomorphism between BDD and DPDN is extracted and implemented to a compact DPDN network. This characteristic makes optimization using BDD easy and efficient. Figure 5 shows this final compact NMOS network for all four inversion functions.

IV. CPA

1. Simulation Attack Using CPA

The CPA is a standard method for evaluating the robustness of a cryptographic device against a side-channel attack. It uses a Hamming weight model or a Hamming distance model to make an estimation of a right key of a cryptographic device. Particularly in the case of transistor level analysis, as in our simulation, the Hamming distance model is more frequently used. The reason for this is because transistor switching behavior is influenced by input transitions according to transistor type (NMOS or PMOS). The model $H_{i,R} = H(M_i \oplus R)$ used to estimate the right correlation factor ρ_{wH} of the estimation formula is described in detail in [15]. Equation (1)

ETRI Journal, Volume 37, Number 3, June 2015
http://dx.doi.org/10.4218/etrij.15.0114.0297

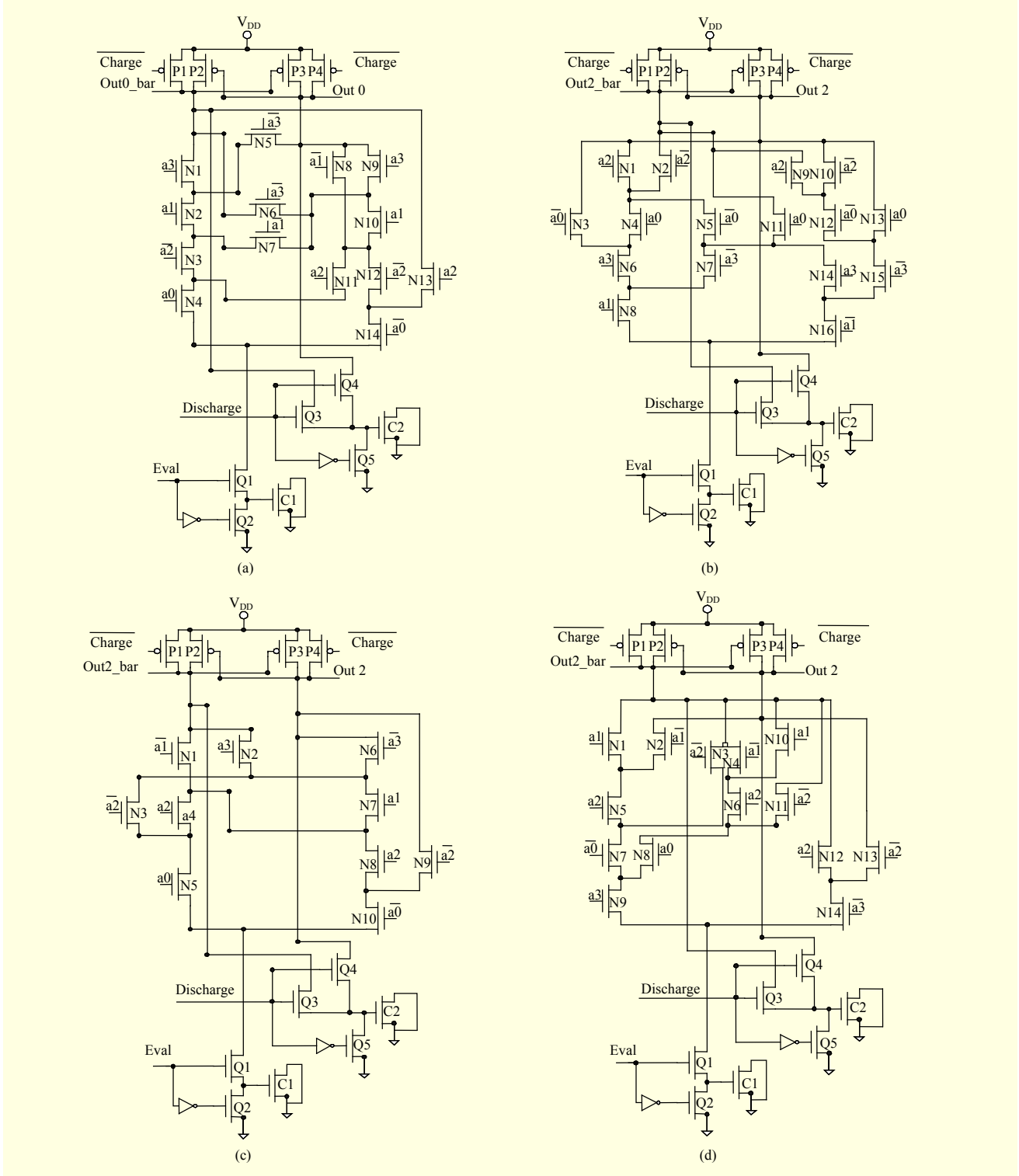


Fig. 5. Structure of inversion functions: (a) q_0^{-1}, (b) q_1^{-1}, (c) q_2^{-1}, and (d) q_3^{-1}.

below is the estimation formula for CPA.

$$\hat{\rho}_{wH}(R) = \frac{N\Sigma W_i H_{i,R} - \Sigma W_i \Sigma H_{i,R}}{\sqrt{N\Sigma W_i^2 - (\Sigma W_i)^2}\sqrt{N\Sigma H_{i,R}^2 - (\Sigma H_{i,R})^2}}. \quad (1)$$

In (1), there is a set of N power curves; W_i and N are associated with random data words M_i. In addition, in (1), we have a reference state R of known data words that produce a set of N predicted Hamming distances $H_{i,R}$. When the estimation

factor ρ_{wH} is at its highest peak, the estimation is closed to prediction.

2. CMOS versus TPDyCML

We target this reference state to 8-bit AES S-box output bits for every 256 input transitions after the logical summation of an input value and a certain secret key value. The AES S-box for DUT mentioned in the previous section is implemented by UMC 0.13μm technology. Figure 6 describes a DUT circuit for CPA.

Before confirming TPDyCML's improved performance as a hardware countermeasure, we have to first compare it with previous secure logic styles and CMOS logic; hence, we conduct a CPA simulation attack on the AES S-box with CMOS and four secure logics that have already been developed; namely, SABL, TDPL, DyCML, and TPDyCML. Through these simulations, we want to confirm that CPA is not appropriate for comparing secure logic styles, as mentioned in [8], [17].

In this simulation, we fix the right key to a value of 36 as a decimal so that we can check the simulation results easily. The estimation value approaches the highest peak when it has a right key value of 36. Figure 7 shows the graphs of CPA results for CMOS and TPDyCML in 256 input transitions. In this case, we can find the right key, 36, at just using 3,500 measurement traces. But, in the case of TPDyCML, we can't get the highest peak even if we applied 20,000 measurements to the AES S-box. The other secure logic styles have similar results with TPDyCML. In other words, it is difficult to find a right key at the DUT circuit of the AES S-box with secure logic styles. However, as far as we can tell, the right peak can be seen in CMOS logic using only a small number of traces.

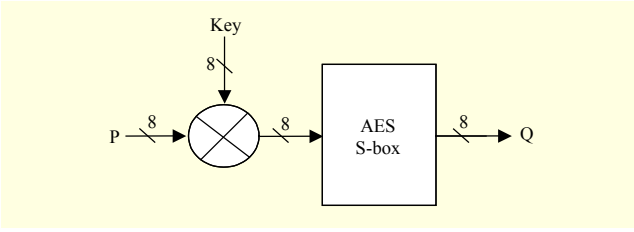


Fig. 6. DUT circuit for CPA.

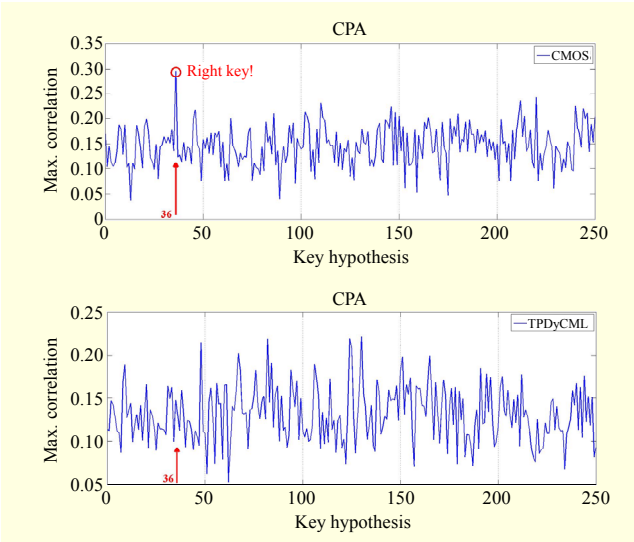


Fig. 7. CPA results of CMOS and TPDyCML in 256 input transitions.

This phenomenon has already been mentioned in [8]. The dual-rail logic style increases nonlinearity more than CMOS logic. It is because dual-rail logic style operates in a way that is both dynamic and differential, and it doesn't just depend on the input transitions of the transistor. Therefore, a CPA method that uses a summation of the Hamming distance value is more useful in analyzing those CMOS logics that have a single-rail configuration and a dynamic logic operation. From this CPA simulation, we can confirm that the security evaluation of secure logic styles with dual-rail logic must be done using a different method, since the dual-rail logic has dynamic and differential logic operations and high nonlinearity.

In the next section, we use an information theoretical metric for obtaining leaked information from secure logics and evaluate our logic style, TPDyCML, according to whether it has reduced leakage information compared with other secure logic styles.

V. MIA

In this section, we first exploit leaked information from the AES S-box with CMOS and secure logics using information theoretic analysis, and then we compare the performance of the secure logics. When we introduced our logic style, TPDyCML, at the WISA 2012 conference, we used the characteristics NED and NSD to confirm our logic style as being one that is secure. However, as already stated in the introduction of this paper and [4], [8], and [17], these characteristics are not enough to guarantee that it can be used in hardware countermeasures and be resistant to side-channel attacks, though it does provide a good starting point. A more exact method for verifying whether a logic style is secure is to use an information theory and entropy analysis for secure dual-rail configuration.

The correct method for comparing the characteristics of a hardware countermeasure implementation is to use an information theoretic analysis of the leaked information from the implementation. The leaked information is used when comparing an implementation with other logic styles. Of course, to perform a more exact security evaluation, the security analysis has to be achieved by comparing the success

rate of the template attack (mentioned in [16]) on the information theoretic analysis. Therefore, we need to keep the security analysis for future work after making the ASIC chip, because we will try to provide a security analysis for a real measured power trace and carry out a real attack. Even if our results about this information theoretic analysis is based on a simulation, it is meaningful to an extent for comparing our TPDyCML with other logic styles.

In this paper, we don't use the entropy from basic gates, as was done in [17], but instead implement a real cryptographic module that has strong nonlinear AES S-boxes with either secure logics and CMOS. Thus, we obtain leaked information to quantify the information leakage of every logic styles and compare the amounts of information leakages.

1. Information Theoretic Analysis

F.-X. Standaert and others [22] proposed an information theoretic metric for exploiting information leakage that was independent of the ability of an adversary. After the proposition, a lot of research papers were published, and it was confirmed to be a good tool for comparing implementations. In [16], an adaptation for securing logic gates was introduced, and the authors accurately described how to use information theoretic analysis for secure logics that have a dual-rail configuration.

At first, NED and NSD were normally used to evaluate the performance of secure logics and compare fluctuations in power consumption. However, in this case, the evaluation method was only adapted to one particular adversary, as mentioned previously (see Section IV). In short, in contrast with the information theoretic analysis, NED and NSD cannot be applied to all implementations independent of whether an adversary has knowledge of the configuration of the target, such as whether the implementation consists of CMOS logic, or whether the particular attack only considers Hamming distance and Hamming weight models.

For a unified method, we use an information theoretic metric of leaked information that is to be quantified for MIA. Therefore, in this paper, we adeptly compare the performance of the AES S-box with other secure logics that have robustness against SCAs through MIA by using information leakage.

Equation (2) below shows the conditional entropy for quantifying information leakage in a side-channel analysis.

$$H[S_g \mid L_{s_g}^q] = E_{s_g} E_{l_{s_g}^q} - \log_2 \Pr[S = s_g \mid L_{s_g}^q = l_{s_g}^q]. \quad (2)$$

In (2), we calculate information quantified from a side-channel leakage using conditional entropy. The given entropy of a side-channel leakage is $H[S_g \mid L_{s_g}^q]$, and this leakage information can be used to compare the amount of information for MIA. Table 1 illustrates slightly altered forms of (2) for different logic styles.

Table 1. Conditional entropy equations by logic style.

	Conditional entropy
CMOS	$H[S_g \mid L_{s_g}^q] = -\sum_{s_g} \Pr[s_g] \int \Pr[l^q \mid s_g] \cdot \log_2 \Pr[s_g \mid l^q] dl$
Secure logics	$H[S_g \mid L_{s_g}^q] = -\sum_{s_g} \Pr[s_g] \sum_t \Pr[t] \int \Pr[l^q \mid s_g, t] \cdot \log_2 \Pr[s_g \mid l^q] dl$

In applying an MIA, we assume that a power trace has a Gaussian noise distribution; thus, in the leakage model, we may consider the variance of the noise to be a Gaussian distribution. In Table 1, $L_{s_g}^q$ is leaked information, which is a random vector that contains a correct key class, s_g.

To apply (2) to the MIA, we use (3), which indicates information entropy without knowing the leakage model, and (4), which follows a Gaussian distribution.

$$H[S_g] = E_{s_g} - \log_2 \Pr[S_g = s_g], \quad (3)$$

$$l_{s_g}^q = d_{s_g}^q + n^q. \quad (4)$$

According to this information theoretic method, we can compare the amount of information leakages between several logic styles that have a Gaussian leakage model.

In (3) and (4), the following notation is used:

- $H[S_g]$ is the entropy of key class S_g before activating SCA.
- S_g and s_g are the correct target signals of the key value and particular key candidates, respectively.
- $l_{s_g}^q$ is information leakage.
- $d_{s_g}^q$ is the deterministic value of intermediate leakage.
- n^q is noise variance, which has a Gaussian distribution.

2. MIA Results for Secure Logics

From the information leakages, which are configured from conditional entropy ((2), (3), and (4)), we derive mutual information, as in the case of (5) below.

$$I(S_g; L_{s_g}^q) = H[S_g] - H[S_g \mid L_{s_g}^q]. \quad (5)$$

We plan to analyze a compact AES S-box with secure logic by using an information theoretic method. An MIA will describe quantified information leakages of an AES S-box having a secure logic at a variable noise level. It will show the hardware countermeasure characteristics, independent of whether the ability of the adversary is strong, by an information theoretic metric with a conditional entropy; an information theoretic metric is uncorrelated with a particular implementation

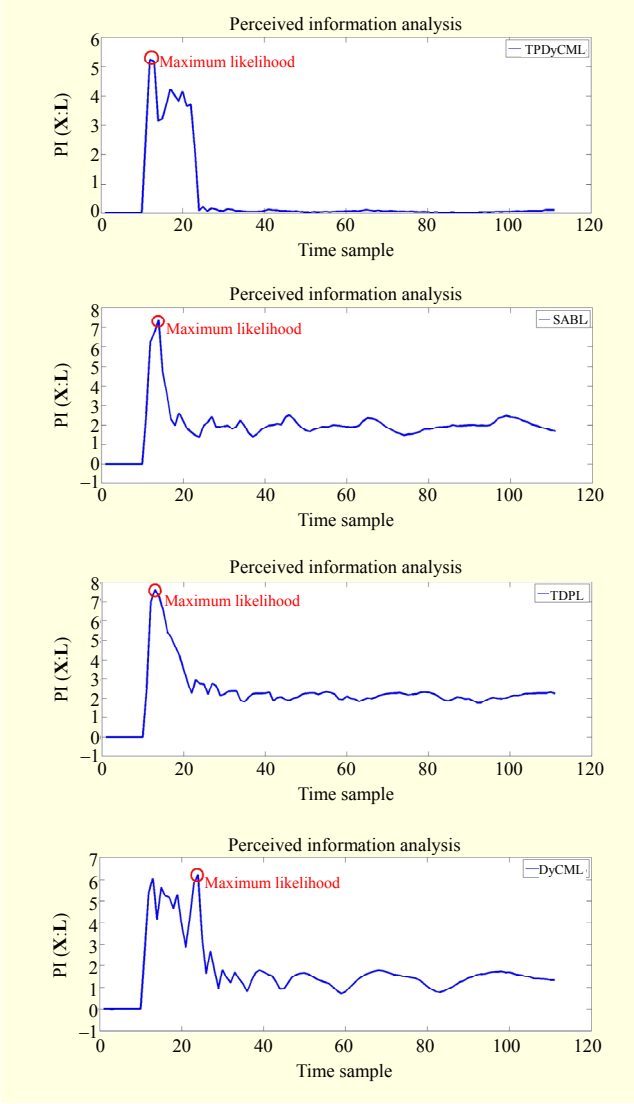


Fig. 8. Maximum likelihood graph of secure logics.

[16]. For example, this particular implementation means a masked countermeasure inserted in the target or a random clock added in it. Also, the measuring environment can be altered so as to contain no noise or a small amount of noise. In other words, generally, lots of analysis methods except a method using information theoretic metric assume that the measuring environment is noise-free or has only a small amount of noise.

$$\tilde{x} = \arg\max_{x^*} \hat{\Pr}_{\text{model}}[x^* \mid l]. \quad (6)$$

MIA uses the maximum likelihood function described by (6). It chooses a time sample that maximizes the perceived amount of information. After this, the analysis evaluates and exploits the information leakage using an information theoretic metric at a different noise level. Figure 8 shows a maximum likelihood graph of the secure logics used in this simulation.

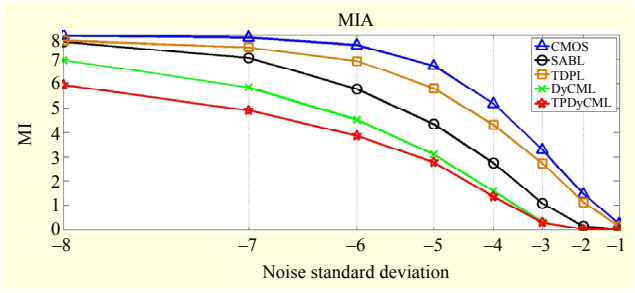


Fig. 9. MIA for AES S-box with CMOS and secure logics.

Practically, the MIA proceeds in two steps as a side-channel analysis. The first phase is the preparation phase. In this step, we profile the leaked information and characterize the adversaries who know the noise level (measuring environment) of the target. For instance, we average the intermediate power trace and calculate the variance of the data. If we use simulation data, then we can skip this phase, because the intermediate value already has a deterministic value and we can assume that the noise variance is a normal Gaussian distribution. The second phase is the exploitation phase or the classification phase. In this phase, we actually recover the right key value from an estimation of the intermediate value. We apply a maximum likelihood of each secure logic to a key candidate and calculate the information leakage. Finally, we use (5) for an MIA and evaluate comparisons with each of the logic styles.

From this analysis, we can confirm that our logic style, TPDyCML, has 15% less leaked information than other secure logic styles. Figure 9 compares the mutual information of each of the secure logic styles, including CMOS, SABL, TDPL, DyCML, and TPDyCML, according to several noise levels.

We have confirmed a HSPICE simulation result in this paper. As mentioned in [17], the evaluation method with MIA is a good analyze method at which to judge whether TPDyCML is a good hardware countermeasure and whether it has good security characteristics before undertaking a real attack for security analysis.

It is because the previous results, [4], [8]–[9], and [17], prove the simulation results that the results have demonstrated a similar tendency as the real security analysis through a template attack.

Therefore, an MIA is a useful procedure for evaluating TPDyCML, which is very robust as a hardware countermeasure compared with previous secure logic styles that worked against SCAs.

VI. Conclusion and Further work

At WISA 2012, we first introduced TPDyCML. In this paper,

we have optimized our secure logic style using the BDD algorithm and evaluated TPDyCML more practically than in [13]. We have conducted a CPA and an MIA. The former is usually used to attack a Hamming-weight-model device and a Hamming-distance-model device. In addition, from the latter, we can exploit the leakage information of implementations with secure logic, which are independent of adversaries.

From the results, we can confirm that TPDyCML is appropriate as a hardware countermeasure against SCAs. Even though it has some disadvantages, the area of the TPDyCML is 1.5 times bigger and consumes more power than DyCML as per our design, it still reduces power consumption compared with DRP logic styles such as SABL and TDPL. Moreover, it can be improved further with respect to the design of an implementation with a compact NMOS tree, because part of the current source can be shared with other circuits and it is possible to make the AES S-box functions more compact.

In spite of the few aforementioned weaknesses, TPDyCML demonstrated that it reduces information leakage by 15% in comparison with other logic styles, even DyCML, against SCAs.

In later works, we will try to do security analyses using a template attack with a success rate. Also, we will try to perform one that evaluates the real measuring of a power trace. In this security analysis, we will find certain points; for example, the most important interesting point in time period. We will also acquire the success rate of the template attack on the basis of several messages.

In addition, in accordance with on-the-fly stochastic attacks and template attacks [8], we will be improving the evaluation performance, because a stochastic analysis (non-profiled attack) can be applied to the most linear part at small noise levels and a template attack (profiled attack) can be applied to the part that is nonlinearity dominated. From this practical security analysis, we can definitely confirm the performance of secure logic styles as a hardware countermeasure against SCAs in a more practical manner.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Annual Int. Cryptography Conf.*, Santa Barbara, CA, USA, Aug. 15–19, 1999, pp. 388–397.

[2] M. Bucci et al., "Three-Phase Dual-Rail Pre-Charge Logic," *Workshop Cryptographic Hardware Embedded Syst.*, Yokohama, Japan, Oct. 10–13, 2006, pp. 232–241.

[3] K. Tiri, M. Akmal, and I. Verbauwhede, "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards," *European Solid-State Circuits Conf.*, Florence, Italy, Sept. 24–26, 2002, pp. 403–406.

[4] F. Mace et al., "A Dynamic Current Mode Logic to Counteract Power Analysis Attacks," *Conf. Des. Circuits Integr. Syst.*, Bordeaux, France, Nov. 24–26, 2004, pp. 186–191.

[5] F. Regazzoni et al., "Evaluation Resistance of MCML Technology to Power Analysis Attacks Using a Simulation-Based Methodology," in *Trans. Comput. Sci.*, Berlin Heidelberg: Springer, 2009, pp. 230–243.

[6] M.W. Allam and M.I. Elmasry, "Dynamic Current Mode Logic (DyCML): A New Low-Power High Performance Logic Style," *IEEE J. Solid-State Circuits*, vol. 36, no. 3, Mar. 2001, pp. 550–558.

[7] T. Sundstrom and A. Alvandpour, "A Comparative Analysis of Logic Styles for Secure IC's Against DPA Attacks," *Nordic Microelectron. Conf.*, Oulu, Finland, Nov. 21–22, 2005, pp. 297–300.

[8] M. Renauld et al., "Information Theoretic and Security Analysis of a 65-nanometer DDSLL AES S-Box," *Workshop Cryptographic Hardware Embedded Syst. Conf.*, Nara, Japan, Sept. 28–Oct. 1, 2011, pp. 223–239.

[9] D. Kamel et al., "Analysis of Dynamic Differential Swing Limited Logic for Low-Power Secure Applications," *J. Low Power Electron. Appl.*, vol. 2, no. 1, Mar. 2012, pp. 98–126.

[10] K. Tiri and I. Verbauwhede, "Design Method for Constant Power Consumption of Differential Logic Circuits," *Des. Autom. Test Europe Conf.*, Munich, Germany, Mar. 7–11, 2005, pp. 628–633.

[11] K. Tiri and I. Verbauwhede, "Place and Route for Secure Standard Cell Design," *Smart Card Res. Adv. Appl.*, Toulouse, France, Aug. 22–27, 2004, pp. 143–158.

[12] L. Lin and W. Burleson, "Analysis and Mitigation of Process Variation Impacts on Power-Attack Tolerance," *Des. Autom. Conf.*, San Francisco, CA, USA, July 26–31, 2009, pp. 238–243.

[13] H. Kim, V. Rozic, and I. Verbauwhede, "Three-Phase Dynamic Current Mode Logic: A More Secure DyCML to Achieve a More Balanced Power Consumption," *Int. Workshop Inf. Security Appl.*, Jeju, Rep. of Korea, Aug. 16–18, 2012, pp. 68–81.

[14] S.B. Akers, "Binary Decision Diagrams," *IEEE Trans. Comput.*, vol. C-27, no. 6, June 1978, pp. 509–516.

[15] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," *Workshop Cryptographic Hardware Embedded Syst.*, Cambridge, MA, USA, Aug. 11–13, 2004, pp. 16–29.

[16] F.-X. Standaert, T.G. Malkin, and M. Yung, "A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks," *Annual Int. Conf. Theory Appl. Cryptographic Techn.*, Cologne, Germany, Apr. 26–30, 2009, pp. 443–461.

[17] F. Mace, F.-X. Standaert, and J.-J. Quisquater, "Information Theoretic Evaluation of Side-Channel Resistant Logic Styles," *Workshop Cryptographic Hardware Embedded Syst.*, Vienna, Austria, Sept. 10–13, 2007, pp. 427–442.

[18] N. Mentens et al., "Systematic Evaluation of Compact Hardware Implementation for the Rijdael S-Box," *Cryptographers' Track RSA Conf.*, San Francisco, CA, USA, Feb. 14–18, 2005, pp. 323–333.

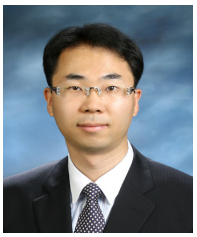
[19] X. Zhang and K.K. Parhi, "High-Speed VLSI Architectures for the AES Algorithm," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 12, no. 9, Sept. 2004, pp. 957–967.

[20] H.R Anderson, *An Introduction to Binary Decision Diagrams*, The IT University of Copenhagen, Lecture Notes for Efficient Algorithms and Programs, Fall 1999. Accessed Mar. 24, 2015. http://www.cmi.ac.in/~madhavan/courses/verification-2011/andersen-bdd.pdf

[21] J. Cortadella, "Mapping BDDs into DCVSL Gates," UPC/DAC (Universitat Politecnica de Catalunya), Barcelona, Spain, Tech. Rep. No. RR 95/04, Feb. 1995.

[22] F.-X. Standaert, T.G. Malkin, and M. Yung, "A Formal Practice-Oriented Model for the Analysis of Side-Channel Attacks," Cryptology ePrint Archive (http://eprint.iacr.org), Rep. 2006/139, 2006.

Hyunmin Kim received his MS degree in engineering in information security from Korea University, Seoul, Rep. of Korea, in 2011. In December 2005, he joined Samsung Electronics, Yongin, Rep. of Korea, where he was engaged in the development of Flash and DRAM memory as a process engineer until December 2008. From 2010 to 2012, he was an international scholar and pre-doctoral student at COSIC of Katholieke Universiteit Leuven, Belgium. Since 2013, he has been working toward his PhD degree in engineering at the Center for Information Security Technologies, Korea University. His research interests include public-key cryptography and its efficient FPGA and ASIC implementation; secure chips; PUFs; and hardware countermeasure design against side-channel attacks.



Dong-Guk Han received his BS and MS degrees in mathematics from Korea University, Seoul, Rep. of Korea, in 1999 and 2002, respectively. He received his PhD degree in engineering in information security from Korea University, in 2005. He was a postdoctoral researcher at Future University Hakodate, Hokkaido, Japan. After finishing his doctoral course, he was then an exchange student with the Department of Computer Science and Communication Engineering, Kyushu University, Japan, from April 2004 to March 2005. From 2006 to 2009, he was a senior researcher at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. He is currently working as an associate professor with the Department of Mathematics, Kookmin University, Seoul, Rep. of Korea. He is a member of KIISC, IEEK, and IACR.

Seokhie Hong received his MA and PhD degrees in mathematics from Korea University, Seoul, Rep. of Korea, in 1997 and 2001, respectively. He worked for Security Technologies Inc., Seoul, Rep. of Korea, from 2000 to 2004. From 2004 to 2005, he worked as a postdoctoral researcher with COSIC, Katholieke Universiteit Leuven, Belgium. Since 2005, he has been with Korea University, where he is now a professor at the Center for Information Security Technologies, Korea University. His research interests include the design and analysis of symmetric-key cryptosystems; public-key cryptosystems; side-channel analysis; and digital forensic systems.