Differential Side Channel Analysis Attacks on FPGA Implementations of ARIA

ChangKyun Kim, Martin Schläffer, and SangJae Moon

In this paper, we first investigate the side channel analysis attack resistance of various FPGA hardware implementations of the ARIA block cipher. The analysis is performed on an FPGA test board dedicated to side channel attacks. Our results show that an unprotected implementation of ARIA allows one to recover the secret key with a low number of power or electromagnetic measurements. We also present a masking countermeasure and analyze its second-order side channel resistance by using various suitable preprocessing functions. Our experimental results clearly confirm that second-order differential side channel analysis attacks also remain a practical threat for masked hardware implementations of ARIA.

Keywords: DPA, DEMA, ARIA, FPGA, side channel attacks, countermeasure.

I. Introduction

Side channel analysis attacks are used to investigate the security of the implementation of cryptographic devices. Since its invention by Kocher and others [1], many methods have been published and different algorithms attacked. One of the most powerful types of attacks is the differential side channel analysis (DSCA) attack, which requires little knowledge about the cryptographic device but requires a large number of traces instead. DSCA attacks analyze how the power consumption or electromagnetic (EM) radiation depends on the processed data at fixed moments in time. For the attack, the intermediate results of the algorithm, which depend on the secret information (the secret key), are chosen. Then, the power consumption is measured and compared to the hypothetical power consumption of these intermediate values.

Many papers that assess DSCA resistance of hardware and software implementations have been published. Most of these papers are related to smart card implementations of the AES block cipher. Other papers analyze ASIC and FPGA implementations of AES [2]-[5]. In this paper, we investigate the differential side channel resistance of the block cipher ARIA, which is a national Korean standard algorithm [6]. ARIA has been implemented in hardware and software for various applications. However, previous publications have only focused on the side channel resistance of *software* implementations of ARIA [7], [8]. To our knowledge, there has been no previous study that considers the side channel resistance of *hardware* implementations of ARIA; therefore, it is necessary to check the strength of these implementations as well.

Most of the techniques presented in this paper have already been applied to various implementations of a similar block cipher, AES. Using these techniques, we provide a systematic

Manuscript received May 24, 2007; revised Nov. 19, 2007.

This research was partially supported by the MIC, Rep. of Korea, under the ITRC support program supervised by the IITA (IITA-2007-C1090-0701-0026), and the Austrian Science Fund (FWF) under grant number P18321.

ChangKyun Kim (phone: + 82 42 870 2122, email: kimck@etri.re.kr) is with Institute Attached to ETRI, Daejeon, Rep. of Korea.

Martin Schläffer (email: martin.schlaeffer@iaik.tugraz.at) is with the Institute for Applied Information Processing and Communications, Graz University of Technology, Graz, Austria.

SangJae Moon (email: sjmoon@ee.knu.ac.kr) is with the School of Electrical Engineering and Computer Science, Kyungpook National University, Daegu, Rep. of Korea.

evaluation of the side channel resistance of hardware implementations of ARIA. We implement different variants of ARIA on our FPGA platform. The unprotected implementations are successfully attacked using first-order DSCAs, such as differential power analysis (DPA) [1] and differential EM analysis (DEMA) [9] in the near and far fields. In addition, we analyze various protected implementations using masking to assess their second-order DSCA resistance. Finally, we analyze different practical preprocessing functions to perform successful higher-order DSCA attacks. Our experiments clearly show that second-order DSCA attacks are practical for masked hardware implementations of ARIA as well.

The remainder of this paper is organized as follows. In section II, we give a short description of ARIA and the investigated hardware implementations. In section III, we present first-order DSCA attacks of our unprotected ARIA implementations. In section IV, we first present a masked hardware implementation of ARIA. Then, the masked implementation and some masking variations are analyzed to assess their second-order DSCA resistance. Finally, we conclude this paper in section V.

II. Hardware Implementation of ARIA

In this section, we first give a short description of the block cipher ARIA. Then, we discuss two different FPGA implementations of the cipher, which are analyzed regarding their side channel resistance.

1. ARIA Block Cipher

ARIA is a symmetric block cipher which encrypts 128-bit blocks of data. The possible key sizes are 128-, 192-, or 256-bit and the numbers of rounds are 12, 14, or 16, respectively. The cipher is an involution, substitution, and permutation encryption network, and each round consists of three parts.

- Roundkey addition (RA): a 128-bit data block is XORed with the 128-bit roundkey.
- Substitution layer (S-boxes): the (nonlinear) substitution layer applies four different S-boxes to the previous values.
- Diffusion layer (DL): the output of the substitution layer is used in a (linear) binary 16×16 matrix multiplication.

The substitution layer of ARIA uses the S-boxes S_1 and S_2 together with their inverses S_1^{-1} and S_2^{-1}. The S-box used in AES is S_1. Each S-box represents an affine transformation of a high degree power function over $GF(2^8)$. S_1 is defined by $S_1(x) = A \cdot x^{-1} \oplus a$, and S_2 is defined by $S_2(x) = B \cdot x^{247} \oplus b$. The matrices A and B and the vectors a and b are defined by (1) and (2). Note that the even rounds have a slightly different ordering of the S-boxes to perform the cipher involution.

The roundkeys are generated by the key schedule of ARIA. The key schedule is first initialized by a three-round Feistel cipher. It then generates the roundkeys by a sequence of XOR, rotate-right, and rotate-left operations.

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \text{ and } a = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad (1)$$

$$B = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \text{ and } b = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \quad (2)$$

2. FPGA Implementation of ARIA

To analyze the side channel resistance of hardware implementations of ARIA, we implemented the cipher on an FPGA. In the unprotected reference implementation, we use a one-round loop architecture that uses five clock cycles per round. Four S-boxes are computed in one cycle; therefore, 4 cycles are needed for the substitution layer. One cycle is used by the diffusion layer.

To investigate different practical implementations of ARIA, we consider two general cases of S-box implementations. In the first implementation, each S-box is based on a multiplicative inverter over the composite field $GF(((2^2)^2)^2)$ [10], and in the second implementation each S-box consists of a table look-up implemented in ROM.

By looking at the algebraic definition of the different S-boxes, the memory size of their implementation can be reduced. Since $x^{247} = (x^{-1})^8$ in $GF(2^8)$, each S-box can be expressed by an affine transformation of the inversion over $GF(2^8)$ [11], [12]. Therefore, only one multiplicative inverter or one inversion table is needed to compute all four S-boxes. Equations (3) to (6) show the representation of these S-boxes, where C is the 8×8 matrix, which takes an element to its 8th power in $GF(2^8)$:

$$S_1(x) = A \cdot x^{-1} \oplus a\,, \tag{3}$$

$$S_2(x) = BC \cdot x^{-1} \oplus b\,, \tag{4}$$

$$S_1^{-1}(x) = (A^{-1} \cdot x)^{-1} \oplus a^{-1}\,, \tag{5}$$

$$S_2^{-1}(x) = ((BC)^{-1} \cdot x)^{-1} \oplus b^{-1}\,. \tag{6}$$

Moreover, to reduce the number of gates and the critical path delay, the affine transformations A, BC, A^{-1}, and $(BC)^{-1}$ are combined with the isomorphism function into the composite field $GF(((2^2)^2)^2)$.

To perform the second-order DSCA attacks on ARIA, we implemented a simple masking countermeasure where the S-box table of the original cipher has been replaced by a masked S-box table. The sizes of our different implementations of ARIA are shown in Table 1.

Table 1. Sizes of our FPGA implementations of ARIA.

	Logic cells	Memory bits
Table look-up	5,126	8,192
Multiplicative inverter	5,453	-
Masked table look-up	7,089	8,192

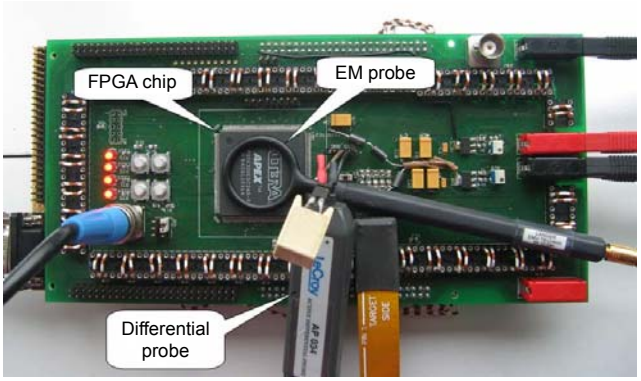


Fig. 1. Measurement setup for DSCA attacks on FPGA implementations of ARIA.

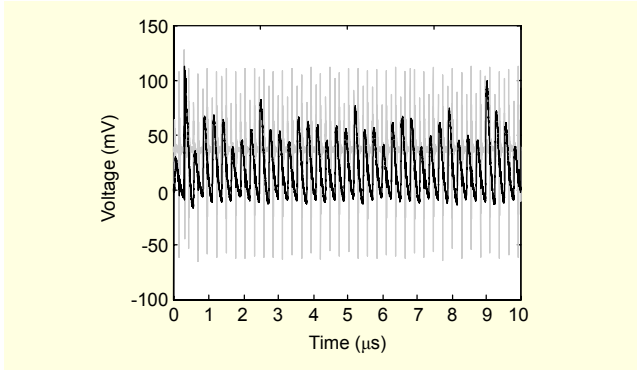


Fig. 2. Single power trace (black) and EM trace (grey) of an FPGA implementations of ARIA.

III. First-Order DSCA Attacks on ARIA

In this section, we investigate our reference implementations regarding first-order DSCA attacks. We compare DPA attacks with DEMA attacks in the near and far fields for both S-box implementations. Finally, the provided experimental results show that an unprotected FPGA implementation of ARIA is still vulnerable to all these attacks.

To attack FPGA implementations of ARIA, we used the Altera EP20K300EQC240-3 device of the APEX 20K family and the measurement setup shown in Fig. 1. The EM and power traces were measured simultaneously using the same trigger to ensure a more accurate comparison. For the attack, 10,000 random plaintexts and one fixed (but random) key were used.

A singe power trace and EM trace of one computation of ARIA is shown in Fig. 2. The black trace corresponds to the power trace and the gray trace corresponds to the EM trace. To characterize the noise of the power and EM traces, we performed 100 measurements. For each measurement, the same input data and the same key was used to avoid data and key dependent variations. Since the noise is normally distributed at each sampling point, we can characterize the

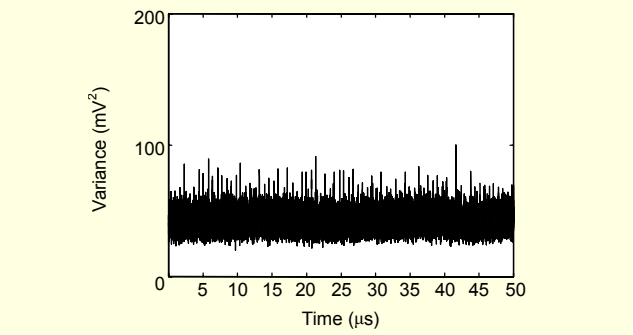


Fig. 3. Noise variance of the power measurements.

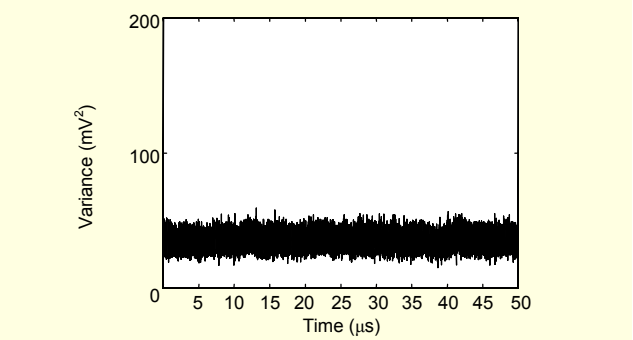


Fig. 4. Noise variance of the EM measurements.

noise of the power and EM traces by the standard deviation σ or the variance σ^2. Figures 3 and 4 show the noise variance of the power and EM traces, respectively.

The DSCA attacks performed in this study are correlation attacks. These attacks examine the linear relationship between side channel leakage and the hypothetical leakage of the data being processed by the cryptographic algorithm [13]. For this attack, it is important to build a hypothetical model based on assumptions about the energy dissipation of the device. After building a reasonable hypothetical model, an attacker computes the correlation coefficient between the measurement signals and the hypothesis. In a successful attack, only the correct key hypothesis leads to a high correlation coefficient.

We assume that the power consumption of our implementation depends on the transitions that occur in the circuit (Hamming distance model); therefore, an attacker needs to predict the transitions of the intermediate values. Generally, transitions of values stored in registers are selected to predict the side channel leakage of a device because these transitions have a high influence on the data dependent power consumption. Recently, Standaert and others [5] demonstrated that the power consumed by an FPGA depends on the amount of resources used by a design as well.

1. Attacking the S-box Output

In our unprotected implementation, four S-boxes are computed in each cycle. Therefore, the output of the i-th S-box and the output of the (i+4)th S-box are consecutively stored in the same register. Because our device leaks the Hamming distance (HD) of consecutive values, we can model the power consumption of the first round according to the Hamming weight (HW):

$$HW(S(P_i \oplus K_i) \oplus S(P_{i+4} \oplus K_{i+4})), \quad (7)$$

where P_i and K_i are the i-th plaintext and roundkey bytes, respectively. Since the HD of the intermediate values depends on two roundkey bytes, we need 2^{16}=65536 key hypotheses to determine K_i and K_{i+4}.

2. DPA Attacks on ARIA

We performed the DPA attacks on the substitution layer, which was implemented as a multiplicative inverter and as a table look-up. For both implementations, we computed the correlation coefficient between the power traces and the hypotheses.

A. Multiplicative Inverter

For the visualization of the results, we fixed the key K_i to the correct key. The resulting DPA traces for the 256 key hypotheses of K_{i+4} are plotted in Fig. 5. The black trace corresponds to the correct key, while the gray traces correspond to the 255 incorrect key hypotheses. There are no significant gray peaks, and there is only one peak in the black trace. Because this peak corresponds to the correct key, we know that our attack was successful.

Figure 6 illustrates the highest correlation coefficients of the 256 key hypotheses as a function of the number of traces. The correlation coefficient for the correct key hypothesis converges to about 0.15, while all other correlation coefficients converge to values below 0.1. Therefore, the correct key hypothesis can be clearly separated from the wrong key hypotheses.

B. Table Look-up

The DPA traces of the attack on the S-box table look-up are plotted in Fig. 7. As in the previous attack, the black trace corresponds to the correct key, and the gray traces correspond to the incorrect key hypotheses. Figure 8 shows that the correct key can already be distinguished from the incorrect keys after 200 traces. The correlation coefficient for the correct key hypothesis is about 0.22 which is significantly higher than attacking the multiplicative inverter. This is because the multiplicative inverter is composed of a more complicated combinational logic than the table look-up, and more signals that do not correlate with the hypothesis occur. Therefore, the signal-to-noise ratio (SNR) of the table look-up is higher, and a better result can be obtained.

Note that the highest correlation coefficients of these two implementations are lower than the highest correlation coefficients of software-based implementations (see [8]). However, both results (Figs. 5 and 7) show that unprotected FPGA implementations of ARIA are vulnerable to DPA attacks, as stated for software implementations in [7], [8].

3. Near-Field DEMA Attacks on ARIA

EM signals are another source of unintentional leakage of

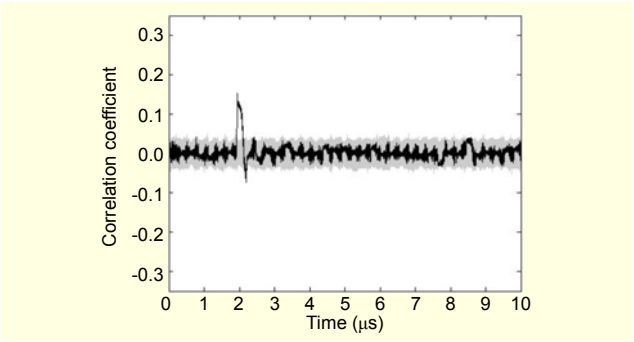


Fig. 5. Multiplicative inverter: DPA traces for the S-box output of all 256 roundkey hypotheses.

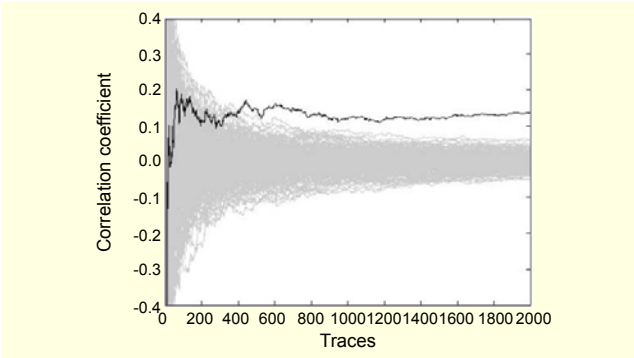


Fig. 6. Multiplicative inverter: correlation coefficient for various numbers of traces.

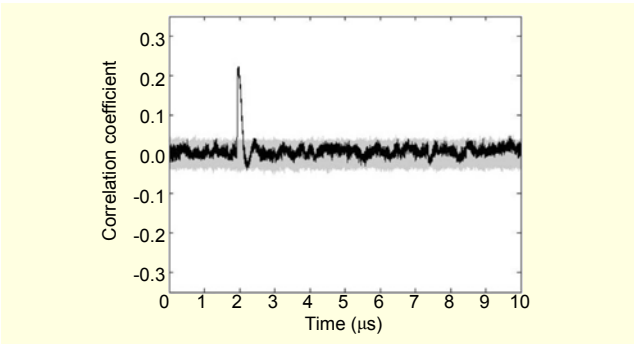


Fig. 7. Table look-up: DPA traces for the S-box output of all 256 roundkey hypotheses.

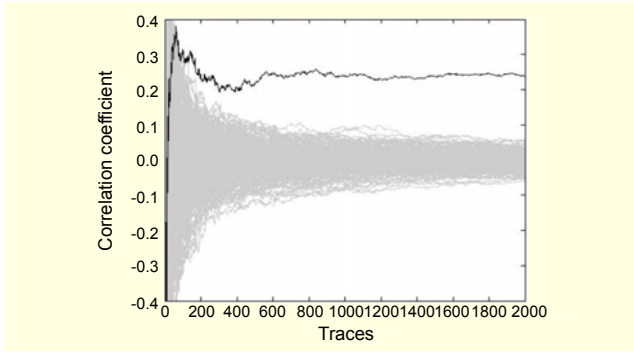


Fig. 8. Table look-up: correlation coefficient for various numbers of traces.

information by the device during its cryptographic computations. EM attacks can be compared to power attacks. Using power measurements, one has only access to the global power consumption, while EM traces are confined to a small and specific area of the target device. Therefore, we can obtain EM traces with a higher SNR and a higher correlation. So far, many papers have shown that EM attacks are as powerful as power attacks [9], [13]-[16].

In this section, we present a near-field DEMA attack on an unprotected implementation of ARIA. To measure the near-field EM traces, we used the LANGER RF-R 400-1 EM probe

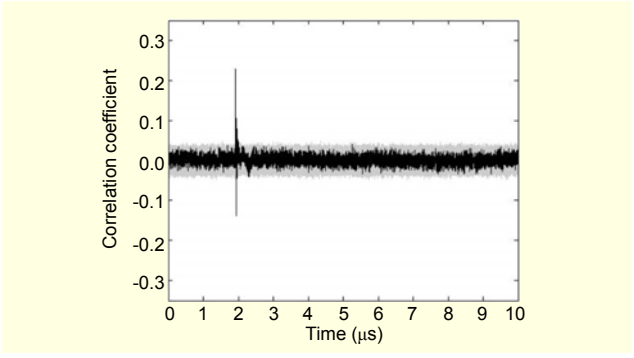


Fig. 9. Multiplicative inverter: DEMA traces for the S-box output of all 256 roundkey hypotheses.

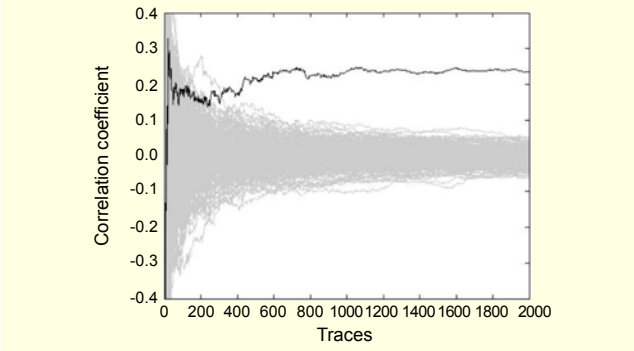


Fig. 10. Multiplicative inverter: correlation coefficient for various numbers of traces.

and a preamplifier that can amplify weak EM traces with high spatial resolution [17]. The measurement probe was positioned a few millimeters from the FPGA device (see Fig. 1). For the near-field DEMA, we attacked the same implementations of the substitution layer as in the DPA attack. We also used the same hypotheses to compare the different results with the previous DPA attacks.

A. Multiplicative Inverter

Figure 9 shows the result of a DEMA attack on an S-box based on a multiplicative inverter. The resulting DEMA trace of the correct key is plotted in black, while the gray traces correspond to the wrong hypotheses. Figure 10 shows that the correlation coefficient converges to 0.22, and the number of traces needed for a successful attack is lower than in the DPA attacks.

B. Table Look-up

The black trace in Fig. 11 shows the correct hypothesis of an S-box based on a table look-up. The correlation coefficient is about 0.32, which is, again, higher than attacking the multiplicative inverter. Figure 12 shows the correlation coefficient as a function of the number of measured traces.

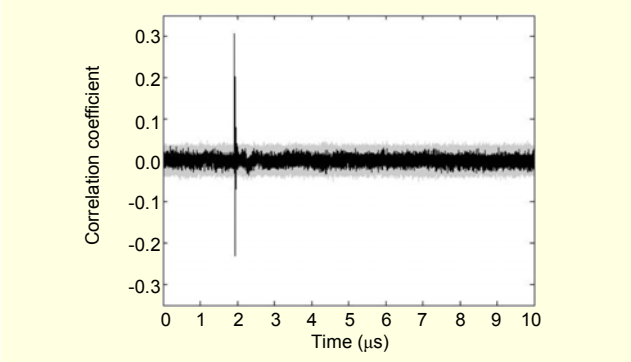


Fig. 11. Table look-up: DEMA traces for the S-box output of all 256 roundkey hypotheses.

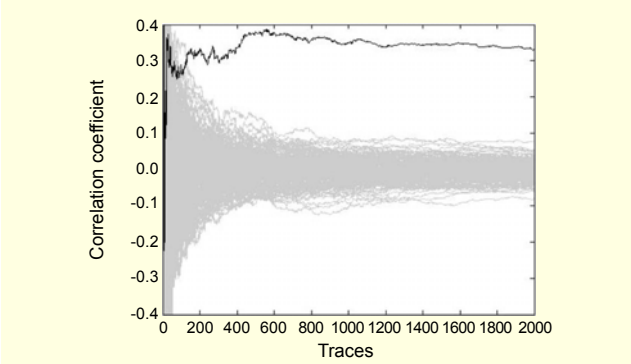


Fig. 12. Table look-up: correlation coefficient for various numbers of traces.

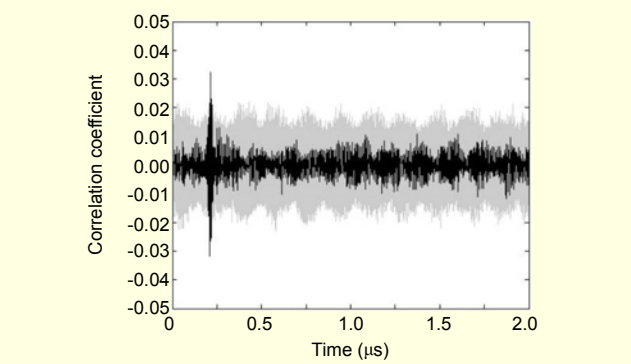


Fig. 13. Far-field DEMA traces for the S-box output of all 256 roundkey hypotheses.

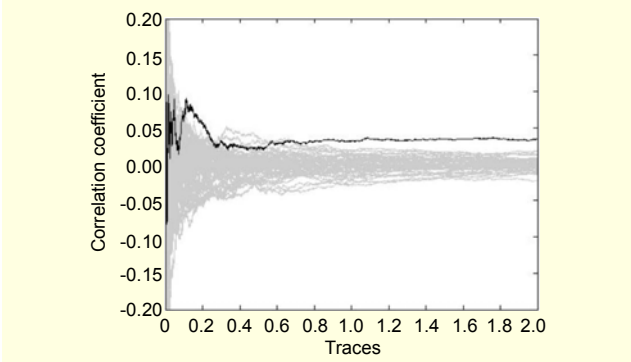


Fig. 14. Correlation coefficient for various number of traces.

4. Far-Field DEMA Attacks on ARIA

The electromagnetic emissions of a device can also be measured in the far field; however, EM attacks in the far field are usually more difficult to perform than attacks in the near field because the emissions in the far field include much more noise. Some noise sources are radio signals or the radiated emissions of other electronic devices, located in the reception area of the antenna.

To demonstrate that even far-field EM traces of an unprotected FPGA implementation of ARIA contain enough information to perform successful attacks, we built a simple measurement setup. We placed our measurement in a non-shielded environment to perform an attack under realistic conditions. The far-field EM traces were measured with a directional antenna with a frequency range from 200 MHz to 1 GHz. We connected the antenna via a preamplifier (30 dB) to a digital oscilloscope. No filter was used between the antenna and the oscilloscope. In this case, the captured traces contain a wide range of frequencies, mainly in the bandwidth of the antenna. For these frequencies, we can roughly compute the area (d) where the near field changes into the far field as follows:

$$d = \frac{\lambda}{2\pi} = \frac{c}{2\pi f} = \frac{3 \cdot 10^8 \,(\text{m/s})}{2\pi \cdot 2 \cdot 10^8 \,\text{Hz}} \approx 0.24 \text{ m}, \qquad (8)$$

where λ is the wavelength, f is the frequency, and c is the velocity of light. In our attack, we placed the antenna at a distance of 1 meter from the FPGA board.

To obtain a trigger signal for the oscilloscope, a probe was connected to an I/O pin of the FPGA board. In practice, it is difficult to obtain a trigger signal without connecting to the FPGA board. However, it is still possible to use a smart trigger, which triggers at specific patterns of the far-field EM signal. If this method does not provide good results, an attacker can apply alignment techniques and discard outliers, to be able to perform far-field EM attacks without connecting a trigger signal directly to the board.

Since the far field contains a lot of noise, only one S-box was implemented as a table look-up to improve the SNR. Note that the highest correlation values of a DPA and a DEMA attack on this implementation were about 0.6 and 0.8, respectively.

Figures 11 and 12 show the results of the DEMA attack in the far field. The highest correlation coefficient for the attack at a distance of 1 meter was about 0.032. Due to the high amount of noise, these values are very low compared to those of the DPA and near-field DEMA attacks. Nevertheless, we have shown that

DEMA attacks can also be conducted in the far field.

5. Comparison of Results

An interesting result of side channel attacks is the number of traces needed to distinguish the correct hypothesis from all wrong hypotheses. Mangard demonstrated in [18] that the number of traces needed to perform side channel analysis attacks can be computed using the correlation coefficient ρ between the correct predictions and the traces. This relationship is defined as

$$N = 3 + 8\left(\frac{Z_\alpha}{\ln\frac{1+\rho_{\max}}{1-\rho_{\max}}}\right)^2, \quad (9)$$

where the confidence interval Z_α determines the distance between the distributions of $\rho = 0$ and $\rho = \rho_{\max}$, and the probability α determines the confidence level.

To attack the whole cipher, a significant peak of the correct key is needed; therefore, we set the confidence level to $\alpha = 0.9999$ as suggested by the rule of thumb of [29]. We get $Z_\alpha = 3.719$ and can calculate the maximum number of traces needed for the different attacks and implementations. Table 2 shows that in our implementations the near-field EM attack on a table look-up leads to the lowest number of traces.

We also calculated the number of traces needed for a successful DEMA attack in the far field. As shown in Table 3, a DEMA at a distance of 1 meter needs twice as many traces as a DEMA at a distance of 0.5 meter. The further the distance between the FPGA board and the antenna is, the more traces that are needed for an attack.

Table 2. Numbers of traces needed for DPA and near-field DEMA attacks.

	DPA		DEMA	
	Inverter	Table	Inverter	Table
$\rho_{\max}$	0.1521	0.2222	0.2289	0.3068
$t_{\min}$	1180	545	512	278

Table 3. Numbers of traces needed for far-field DEMA attacks.

	DEMA in the far field	
	0.5 m	1 m
$\rho_{\max}$	0.0461	0.0324
$t_{\min}$	13000	26335

Note that we can improve the SNR of the traces by using a spectrum analyzer which can remove the frequencies with a low data dependency. Moreover, an EM attack could be performed in a shielded environment to reduce external noise.

IV. Second-Order DSCA Attacks on a Masked ARIA Implementation

In the previous section, we demonstrated that an unprotected hardware implementation of ARIA is vulnerable to first-order DSCA attacks. In this section, we investigate the second-order DSCA resistance of a masked hardware implementation of ARIA.

1. Masking ARIA

Masking schemes are popular methods to protect block ciphers against first-order DSCA attacks. Using masking, the intermediate values that occur during the computation are concealed by a random value (the mask). Thus, the power consumption should be independent of the unmasked values. In our analysis, we use additive masking where the mask is XORed with the intermediate value:

$$a_m = a \oplus m = P_1 \oplus K_1 \oplus m .$$

However, most hardware countermeasures based on masking schemes are insecure and susceptible to first-order DPA attacks due to the effect of glitches in nonlinear combinational logics [19]-[22]. Therefore, we have targeted an implementation using a masked table look-up of the S-boxes. This implementation has a higher hardware requirement but has been resistant to first-order DSCA attacks in our experiments.

The S-box table S of the original cipher has been replaced by a masked S-box table S_m such that $S_m(a \oplus m_i) = S(a) \oplus m_o$ for variable input and output masks, m_i and m_o. To reduce the number of stored masked S-box tables, the same mask can be used for all S-boxes. Only one stored masked table of the inversion over $GF(256)$ is needed. This table is then used to compute the four S-boxes using affine transformations (see section II.2).

Particular care has to be taken to avoid unintentional cancellation of masks. If a device leaks the HD, and the output of two masked S-boxes with the same mask are stored subsequently in the same register, the HD of the intermediate values will leak. The algorithm can then be attacked by a first-order DSCA using the following power (or EM) model:

$$HD(a \oplus m, b \oplus m) = HW(a \oplus b), \quad (10)$$

where $b = P_4 \oplus K_4$.

Even if some S-boxes of the substitution layer are masked

with the same mask, but not stored in the same register, the diffusion layer can unmask the intermediate values. Therefore, different masks must be used for the diffusion layer to ensure that all intermediate values stay masked.

2. Second-Order DSCA Attacks

Messerges was the first to show that a simple masking scheme is vulnerable to second-order DSCA attacks in practice [23]. Since then, many practical and improved second-order DSCA attacks for masked software and hardware implementations have been published [24]-[28].

Second-order DSCA attacks exploit the leakage of two intermediate values, a_m and b_m, which are related to the same mask m. The attack can be divided into a preprocessing step and an evaluation step. In the preprocessing step, an attacker chooses an interval in which the values a_m and b_m are processed in the device. Then, each pair of points of the power (or EM) trace in this interval is combined using a preprocessing function to get the preprocessed trace.

In the evaluation step, similar to first-order DSCA attacks, an attacker calculates the correlation between the preprocessed traces and the hypothetical power consumption $HW(a \oplus b)$. Therefore, it is important to choose a preprocessing function which maximizes the correlation between $HW(a \oplus b)$ and the measured power consumption of the masked values a_m and b_m:

$$\rho(HW(a \oplus b), pre(HW(a_m), HW(b_m))). \quad (11)$$

Table 4 lists some possible preprocessing functions and their corresponding correlation coefficients in the case of a 1-bit and 8-bit scenario [29]. Usually, the best result is obtained by subtracting the two power consumptions and taking the absolute value of the difference [29].

Table 4. Correlation coefficients for various preprocessing functions of the traces. In row 5, E represents the expected value of $HW(a_m) + HW(b_m)$.

Preprocessing		Value				ρ	
						1 bit	8 bit
a_m		0	0	1	1		
b_m		0	1	0	1		
$HW(a_m \oplus b_m)$		0	1	1	0		
1	$HW(a_m) \cdot HW(b_m)$	0	0	0	1	-0.57	-0.09
2	$\lvert HW(a_m) - HW(b_m) \rvert$	0	1	1	0	1	0.24
3	$HW(a_m) + HW(b_m)$	0	1	1	2	0	0
4	$(HW(a_m) + HW(b_m))^2$	0	1	1	4	-0.33	-0.04
5	$\lvert HW(a_m) + HW(b_m) - E \rvert$	1	0	0	1	-1	-0.24

Note that we can still perform second-order DSCA attacks if the two masked intermediate values are processed at the same time. In this case, the device adds up the two power consumptions; therefore, there is less freedom in choosing the precomputation function. However, by applying nonlinear functions a second-order DSCA attack is still possible. In the following subsection, we will investigate these two second-order DSCA scenarios using our masked ARIA implementations.

A. Different Clock Cycles

In this case, we attack an implementation in which two S-boxes use the same mask but are processed in different cycles. To avoid implicit cancellation of the mask, the initial values of the registers are set to zero. Another method would be to use different masks or to fill the registers with random values instead. However, in most practical environments, it is difficult to generate many random values in each cycle. The power consumption of the device corresponds to $P1 \approx HW(S(P_1 \oplus K_1) \oplus m)$ when the first S-box is processed and $P2 \approx HW(S(P_4 \oplus K_4) \oplus m)$ when the second S-box is processed. Therefore, the resulting hypothesis for the second-order DSCA attack is:

$$HW(S_1(P_1 \oplus K_1) \oplus S_2(P_4 \oplus K_4)). \quad (12)$$

The results of the second-order DPA and DEMA attacks using this hypothesis are given in Figs. 15 and 17, respectively. Significant peaks using the preprocessing function $|P1 - P2|$ are seen. As before, the trace that corresponds to the correct key is plotted in black, and the traces that correspond to the incorrect keys are plotted in gray. Figures 16 and 18 show how the correlation coefficient develops over an increasing number of traces. According to these figures, we can calculate the number of traces needed to distinguish the correct key from the wrong keys. We have also computed the correlation using the preprocessing function $|P1 - P2|^{\beta}$ for different values of β [24], [28], and the results are given in Table 5. While the

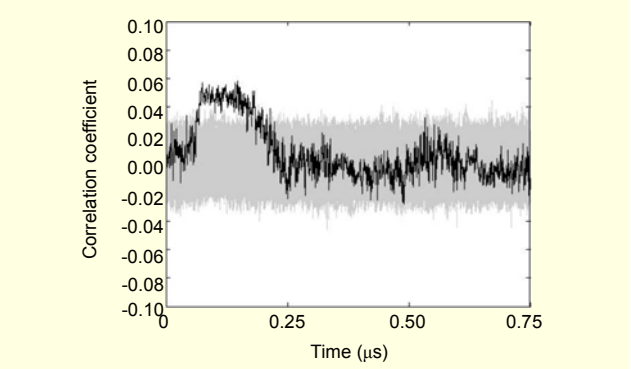


Fig. 15. Second-order DPA traces of all 256 key hypotheses.

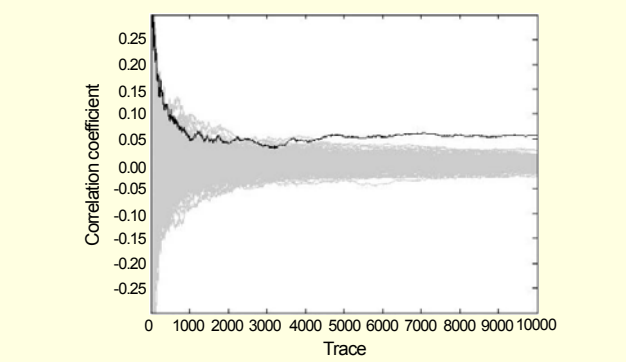


Fig. 16. Second-order DPA: correlation coefficient for various numbers of traces.

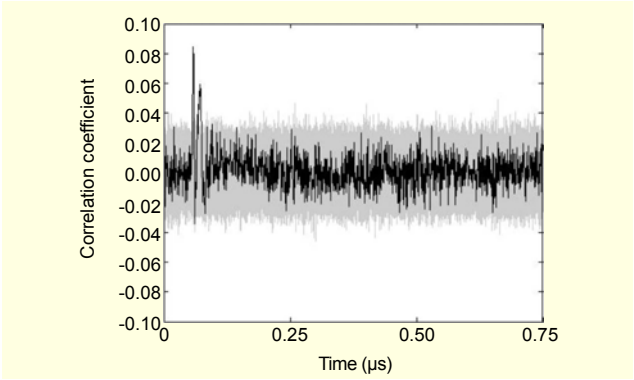


Fig. 17. Second-order DEMA traces of all 256 key hypotheses.

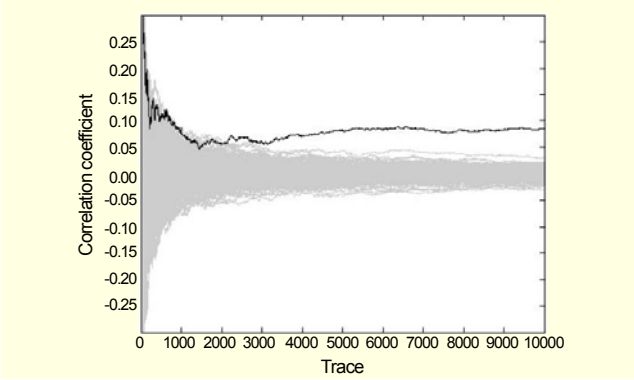


Fig. 18. Second-order DEMA: correlation coefficient for various numbers of traces.

Table 5. Correlation coefficient for different values of β.

β	1	2	3	4	5
DPA	0.0547	0.0573	0.0592	0.0605	0.0614
DEMA	0.0791	0.0843	**0.0864**	0.0861	0.0839
β	6	7	8	9	10
DPA	**0.0618**	0.0616	0.0611	0.0602	0.0589
DEMA	0.0801	0.0753	0.0698	0.0640	0.0581

maximum correlation coefficient of the second-order DPA attack can be obtained with β=6, the correlation coefficient of the second-order DEMA attack has its maximum at β=3.

B. Same Clock Cycle

In the previous section, the two masked values, a_m and b_m, concealed by the same mask m, were processed at different clock cycles. However, in hardware implementations, masked values are usually processed at the same time. If two masked values are concealed by the same mask but processed in two parallel circuits, the combined power consumption can still be susceptible to second-order DSCA attacks.

To find a suitable preprocessing function for this case, we performed second-order DSCA attacks on a simple masked S-box and measured 100,000 power traces and 100,000 EM traces simultaneously. Two masked S-boxes were concealed by the same mask in our reference implementation. The two outputs of the S-boxes are $a_m = S(x_1) \oplus m$ and $b_m = S(x_2) \oplus m$, where x_1 and x_2 are the inputs for each S-box and m is the common mask.

The device under attack leaks the Hamming weight of the intermediate values. Therefore, we can model the overall

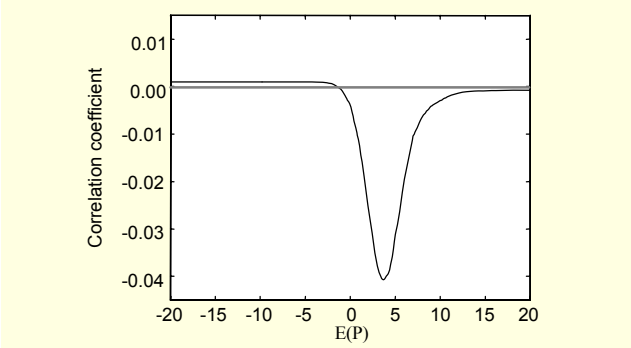


Fig. 19. Correlation coefficient for different offsets of the mean $E(P)$ in case of the EM traces.

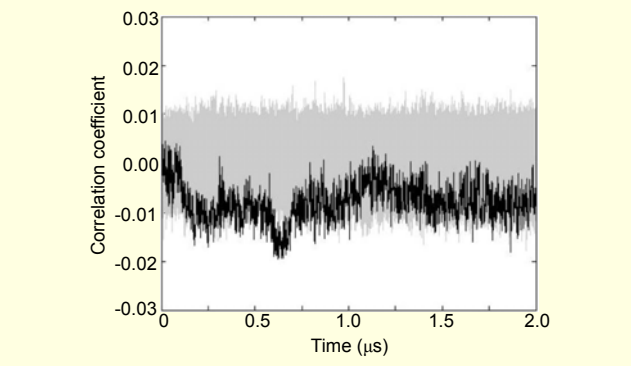


Fig. 20. Result of the second-order DPA attack using the preprocessing function $pre(P) = |P - E(P)|$.

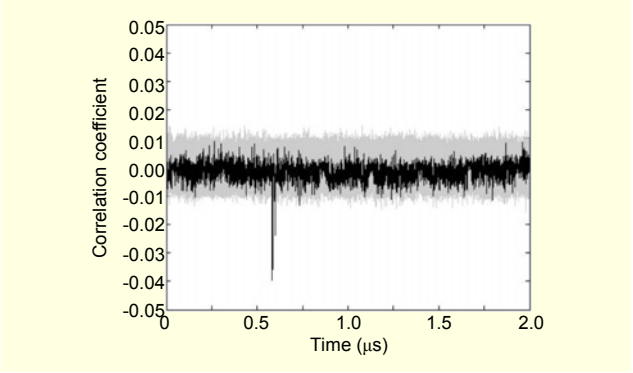


Fig. 21. Result of the second-order DEMA attack using the preprocessing function $pre(P) = |P - E(P)|$.

power consumption by adding the power consumption of the two parallel circuits: $P \approx HW(a_m) + HW(b_m)$. We calculated the correlation coefficients between $HW(a \oplus b)$ and the traces after applying the preprocessing functions $|P|$ and P^2 of [24]. However, even with 100,000 measurements we were not able to perform a successful second-order DSCA attack for any of these preprocessing functions. We did not see any significant peak for the unmasked values $S(x_1)$ and $S(x_2)$ either.

The problem is that we are limited in the choice of good preprocessing functions because the power consumptions are added implicitly by the device. Therefore, we have been looking for a preprocessing function which yields a similar correlation coefficient as the function $|P1 - P2|$. Taking the absolute value is only useful if there are positive and negative values in the traces. The impact on the correlation after adding an offset to the traces and then applying the absolute value is shown in Fig. 19. The absolute value works best if the traces are centered around zero; therefore, we removed the mean of the traces before taking the absolute value.

We computed the correlation coefficient using $|P - E(P)|$ (see Table 4), and with this preprocessing function we were able to perform successful second-order DPA and DEMA attacks. The highest correlation coefficient of the second-order DPA attack is –0.0196 and that for the DEMA attack is –0.0396. As the correlation coefficient for $|P - E(P)|$ shown in Table 4, the two correlation coefficients have a negative value. The mean values of the power traces and EM traces at the corresponding points in time are 4.5 mV and 4.16 mV, respectively. Figures 20 and 21 show the traces of the second-order DSCA attacks using our preprocessing function. In each attack, the peak in the black trace corresponds to the correct key.

V. Conclusion

In this paper, we investigated the side channel resistance of various hardware implementations of ARIA. For this purpose, we implemented different unprotected and masked variants of ARIA on an FPGA without other hardware countermeasures. We demonstrated that an unprotected hardware implementation of ARIA is vulnerable to first-order DPA and DEMA attacks. The secret key can be recovered with a low number of power or near-field EM measurements. In the far field, it is more difficult but still possible to perform a successful attack. Note that there are hardware countermeasures, such as Faraday cages and the like, which can normally be used where an FPGA is used; therefore, the application of these attacks is most likely in an embedded system, such as a smartcard containing an ASIC, where hardware countermeasures are somewhat limited.

We also implemented different masking variants to protect our implementation against first-order DSCA attacks in practice. We successfully analyzed these implementations regarding second-order DSCA attacks. Using suitable preprocessing functions and hypotheses, we were able to attack parallel masked S-box implementations using the same mask. Our experimental results show that second-order DSCA attacks are a realistic and practical threat for masked hardware implementations of ARIA as well.

Although masking allows an increase in the number of needed traces, it is not sufficient to prevent side channel analysis attacks completely. Moreover, masked hardware implementations of ARIA need significantly more resources than unprotected implementations. Therefore, we conclude that further research is needed to develop efficient and secure hardware implementations of ARIA.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Proc. CRYPTO*, LNCS 1666, 1999, pp. 388-397.

[2] S.B. Örs, E. Oswald, and B. Preneel, "Power-Analysis Attacks on an FPGA: First Experimental Results," *Proc. CHES*, LNCS 2779, 2003, pp. 35-50.

[3] S.B. Örs, F. Gurkaynak, E. Oswald, and B. Preneel, "Power-Analysis Attack on an ASIC AES Implementation," *Proc. ITCC*, vol. 2, 2004, pp. 546-552.

[4] F. Standaert, S. Örs, and B. Preneel, "Power Analysis of an FPGA Implementation of Rijndael: Is Pipelining a DPA Countermeasure?" *Proc. CHES*, LNCS 3156, 2004, pp. 30-44.

[5] F.X. Standaert, F. Mace, and J.J. Quisquater, "Updates on the Security of FPGAs against Power Analysis Attacks," *Proc. CHES*, LNCS 3985, 2006, pp. 335-346.

[6] D. Kwon, J. Kim, S. Park, S. Sung, Y. Sohn, J. Song, Y. Yeom, E. Yoon, S. Lee, J. Lee, S. Chee, D. Han, and J. Hong, "New Block Cipher: ARIA," *Proc. ICISC'03*, LNCS 2971, 2004, pp. 432-445.

[7] J. Ha, C. Kim, S. Moon, I. Park, and H. Yoo, "Differential Power

Analysis on Block Cipher ARIA," *Proc. HPCC*, LNCS 3726, 2005, pp. 541-548.

[8] H. Yoo, C. Herbst, S. Mangard, E. Oswald, and S. Moon, "Investigations of Power Analysis Attacks and Countermeasures for ARIA," *Proc. WISA'06*, LNCS 4298, 2007.

[9] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," *Proc. CHES*, LNCS 2162, 2001, pp. 251-261.

[10] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," *Proc. ASIACRYPT*, LNCS 2248, 2001, pp. 239-254.

[11] A. Satoh and S. Morioka, "Unified Hardware Architecture for 128-bit Block Cipher AES and Camellia," *Proc. CHES*, LNCS 2779, 2003, pp. 304-318.

[12] S. Yang, J. Park, and Y. You, "The Smallest ARIA Module with 16-Bit Architecture," *Proc. ICISC*, LNCS 4296, 2006, pp. 107-117.

[13] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," *Proc. CHES*, LNCS 3156, 2004, pp. 16-29.

[14] D. Agrawal, B. Archambeault, J.R. Rao, and P. Rohatgi, "The EM Side-Channel(s)," *Proc. CHES*, LNCS 2523, 2002, pp. 29-45.

[15] D. Agrowal, B. Archambeault, S. Chari, P. Rohatgi, and J. Rao, "Advances in Side-Channel Cryptanalysis, Electromagnetic Analysis and Template Attacks," *Cryptobytes*, vol. 6, no. 1, 2003, pp. 20-32.

[16] C.H. Gebotys, S. Ho, and C.C. Tiu, "EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA," *Proc. CHES*, LNCS 3659, 2005, pp. 250-264.

[17] LANGER EMV-Technik, http://www.langer-emv.de/en/produkte/prod_rf2.htm/.

[18] S. Mangard, "Hardware Countermeasures against DPA: A Statistical Analysis of Their Effectiveness," *Proc. CT-RSA*, LNCS 2964, 2004, pp. 222-235.

[19] S. Mangard, T. Popp, and B.M. Gammel, "Side-Channel Leakage of Masked CMOS Gates," *Proc. CT-RSA*, LNCS 3376, 2005, pp. 351-365.

[20] S. Mangard, N. Pramstaller, and E. Oswald, "Successfully Attacking Masked AES Hardware Implementations," *Proc. CHES*, LNCS 3659, 2005, pp. 157-171.

[21] W. Fischer and B.M. Gammel, "Masking at Gate Level in the Presence of Glitches," *Proc. CHES*, LNCS 3659, 2005, pp. 187-200.

[22] S. Mangard and K. Schramm "Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations," *Proc. CHES*, LNCS 4249, 2006, pp. 76-90.

[23] T. Messerges, "Using Second-Order Power Analysis to Attack DPA Resistant Software," *Proc. CHES'00*, LNCS 1965, 2004, pp. 238-251.

[24] J. Waddle and D. Wagner, "Towards Efficient Second-Order Power Analysis," *Proc. CHES*, LNCS 3156, 2004, pp. 1-15.

[25] F. Standaert, E. Peeters, and J. Quisquater, "On the Masking Countermeasure and Higher-Order Power Analysis Attacks," *Proc. ITCC*, vol. 1, 2005, pp. 562-567.

[26] M. Joye, P. Paillier, and B. Schoenmakers, "On Second-Order Differential Power Analysis," *Proc. CHES*, LNCS 3659, 2005, pp. 293-308.

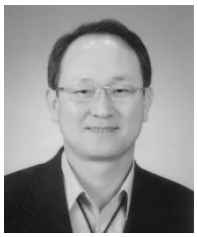
[27] E. Peeters, F. Standaert, N. Donckers, and J. Quisquater, "Improved Higher Order Side-Channel Attacks with FPGA Experiments," *Proc. CHES*, LNCS 3659, 2005, pp. 309-323.

[28] E. Oswald, S. Mangard, C. Herbst, and S. Tillich, "Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers," *Proc. CT-RSA*, LNCS 3860, 2006, pp. 192-207.

[29] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smartcards*, Springer-Verlag, 2007.

ChangKyun Kim received his BE and ME degrees in electronics from Kyungpook National University, Daegu, Rep. of Korea, in 2001 and 2003, respectively. Currently, he is a PhD student at Kyungpook National University. Since December 2004, he has been a researcher with the Institute Attached to Electronics and Telecommunications Research Institute. He took part in the Korea Certificate-Based Digital Signature Algorithm using Elliptic Curves (EC-KCDSA) Standard project. His current research interests include side channel cryptanalysis on hardware crypto devices and secure implementations of cryptographic algorithms.

Martin Schläffer received his BS and MS degrees in computer science at the Graz University of Technology, Graz, Austria. He is currently working for his PhD degree at the Institute for Applied Information Processing and Communications (IAIK) from Graz University of Technology. His research interests are cryptographic algorithms and their secure implementations.



SangJae Moon received his BE and ME degrees in electronics from Seoul National University, Rep. of Korea, in 1972 and 1974, respectively. He received his PhD in communication engineering from the University of California, Los Angeles, USA, in 1984. Currently, he is a professor with the School of Electrical Engineering and Computer Science, Kyungpook National University, Rep. of Korea, and the director of the Mobile Network Security Technology Research Center (MSRC). He is also an honorary president of the Korea Institute of Information Security and Cryptology. His current research interests are information security in mobile, ubiquitous, and RFID networks including the physical security of smart IC cards. He took part in the Korea Certificate-Based Digital Signature Algorithm (KCDSA) Standard project. He has a number of issued patents and more than one hundred technical publications in international journals and conferences in the area of information security.