On the Security of Balanced Encoding Countermeasures

Yoo-Seung Won[1,2], Philip Hodgers[2], Máire O'Neill[2], Dong-Guk Han[1,3⋆]

[1]Department of Financial Information Security, Kookmin University, Seoul, Korea
mathwys87@kookmin.ac.kr
[2]Center for Secure Information Technologies, Queen's University, Belfast, UK
p.hodgers@qub.ac.uk
m.oneill@ecit.qub.ac.uk
[3]Department of Mathematics, Kookmin University, Seoul, Korea
christa@kookmin.ac.kr

Abstract. Most cryptographic devices should inevitably have a resistance against the threat of side channel attacks. For this, masking and hiding schemes have been proposed since 1999. The security validation of these countermeasures is an ongoing reserach topic, as a wider range of new and existing attack techniques are tested against these countermeasures. This paper examines the side channel security of the balanced encoding countermeasure, whose aim is to process the secret key-related data under a constant Hamming weight and/or Hamming distance leakage. Unlike previous works, we assume that the leakage model coefficients conform to a normal distribution, producing a model with closer fidelity to real-world implementations. We perform analysis on the balanced encoded PRINCE block cipher with simulated leakage model and also an implementation on an AVR board. We consider both standard correlation power analysis (CPA) and bit-wise CPA. We confirm the resistance of the countermeasure against standard CPA, however, we find with a bit-wise CPA that we can reveal the key with only a few thousands traces.

Keywords: Balanced Encoding, bit-wise CPA, PRINCE block cipher

1 Introduction

With the advent of the Internet of Things, an increasing number of cryptographic devices enter our daily lives. Most of these devices are accessible, and therefore, cannot avoid the threat of side channel attacks [1]. Thus, it is essential that these devices are protected with side channel countermeasures, such as masking and/or hiding [14, 17, 19, 20, 24, 28]. From the point of view of a software implementation, masking countermeasures aim to break the relationship between the power leakages and the intermediate variables. However, masking is still vulnerable to higher-order attacks [31]. Hiding countermeasures, on the other hand, aim to reduce the statistical linkage between the trace samples and the intermediate

⋆ The corresponding author.

variables. In general, the cost of a hiding countermeasure is lower than that of a masking countermeasure. Due to the complementary security features of these countermeasures, their combination is often regarded as advantageous compared to the implementation of either in isolation.

In order to improve protection against side channel analysis, the idea of producing a constant leakage for any intermediate variables in software implementations was first proposed by P.Hoogvorst *et al.* [23], suggesting the concept of Dual-rail with Precharge Logic [5, 7, 10–12, 21]. However, their work did not include an implementation for security evaluation. Subsequently, Maekawa *et al.* [26] proposed an improved scheme in terms of memory-cost, referred to as Symbolic Representation. Similarly they did not evaluate the security of their proposal in terms of implementation. That is, nobody knows for sure whether the implementation of the constant leakage countermeasure is secure or not. Recently, these approaches have been re-proposed with a security validation by Servant *et al.* [32], claiming that their scheme is resistant to side channel analysis. More precisely, they showed that the leakage of the constant Hamming weight countermeasure remains fixed for constant weighting coefficients of a polynomial leakage function, regardless of the variation of intermediate variables. Thus, this countermeasure appears to be more secure than other previously proposed countermeasures. In support of this, they showed that the success rate for key recovery was less than 10%, when performing correlation power analysis (CPA) [8] from a simulated leakage model.

Most prior works, which deal with constant weight countermeasures, only consider the Hamming weight model. However, some leakage of the Hamming distance between intermediate variables is still a possibility, even when a cryptographic algorithm is implemented in software. Ideally, a scheme should provide resistance against both Hamming distance and Hamming weight leakage. Recently, C.Chen *et al.* [30] proposed such a scheme for the first time, calling it balanced encoding. They showed that their scheme was not vulnerable to first-order CPA in an AVR microcontroller board. They also claimed that their scheme ensures security against linear and balanced leakage models.

In this paper, we conduct a security evaluation for the balanced encoding countermeasure proposed in [30]. For this, we employ the polynomial of higher degree leakage model as with previous works. The coefficients of leakage model in [29,32] are regarded as 0 or 1. However, since these ideal values may be difficult to realize in practice, we therefore consider a leakage model where the coefficients deviate from purely constant values. For this purpose, we assume that the coefficients comform to a normal distribution. We therefore develop a balanced and imbalanced leakage model of the PRINCE block cipher [25], performing CPA on both simulated and real traces obtained from an 8-bit AVR microcontroller implementation. This is the same environment as prior work [30], allowing for a direct comparison. Crucially, we will show that the leakage of a balanced encoding countermeasure is not constant in simulated and implemented environments, when analyzed with a bit-wise CPA, demonstrating that the countermeasure in [30] is not secure against standard side channel analysis techniques.

The remainder of this work is structured as follows: The balanced encoding countermeasure is summerized in Section 2. The leakage model and the result of the security evaluation for simulated traces are reported in Section 3. Section 4 presents the result of the security evaluation for real traces implemented on an AVR microcontroller board, with our conclusions in Section 5.

2 Related Works

This section provides an overview of the constant leakage countermeasure for the balanced encoded PRINCE block cipher.

2.1 Dual-rail with Precharge Logic

Dual-rail with precharge logic, which has been proposed as a hardware-based countermeasure, is a combination of dual-rail logic and precharge logic. In contrast with single-rail logic, where each wire carries a single bit-value, dual-rail logic uses two wires to carry each bit. Furthermore, in precharge logic all signals in the circuit are set to a precharge value of either 0 or 1. In a software implementation, each bit of an intermediate variable is replaced with either 01 or 10. The purpose of this concept is to retain a constant Hamming weight value. If either 00 or 11 is selected, this rule will be broken.

2.2 Balanced Encoding Countermeasure for PRINCE block cipher

In this section, we introduce the balanced encoding countermeasure for PRINCE, proposed in [30]. The purpose of all such encoding schemes is to produce constant weight bytes by inserting the relevant complementary bits for any position. Fig.1 shows part of a balanced encoding scheme for the PRINCE block cipher. The two columns on the left-hand-side of Fig.1 show the Hamming weight and Hamming distances of the states. Because the unencoded state of PRINCE is 4-bits (nibble), the balanced encoded state is a byte. In Fig.1, the basic encoding scheme is called Encode I, and additionally, in order to perform the **S-layer** operation, Encode II and III are also required.

The first state nibble can be simply extended to one byte without inserting complementary bits. That is, $x_3x_2x_1x_0 \rightarrow x_3x_3x_2x_2x_1x_1x_0x_0$. The **KeyAddition** operation is then calculated after K_0 is encoded with Encode I. The other round key and round constant values are likewise extended to one byte without inserting complementary bits, retaining a consistant encoding representation. Encode I cannot be applied to **S-layer** and therefore Encode II, whose encoding operation is given in the lower right-hand side of Fig.1, is applied. That is, $EncII[S(x_3x_2x_1x_0)] = S'(EncI[x_3x_2x_1x_0])$ where $S(\cdot)$ and $S'(\cdot)$ represent the S-box and the new S-box, respectively. In this way, a constant Hamming weight and Hamming distance value is preserved. In order to input the **M-layer** (**MixColumns**) we need to go back to Encode I. However, Encode III is required since it is not possible to go directly from Encode II to Encode I as this would

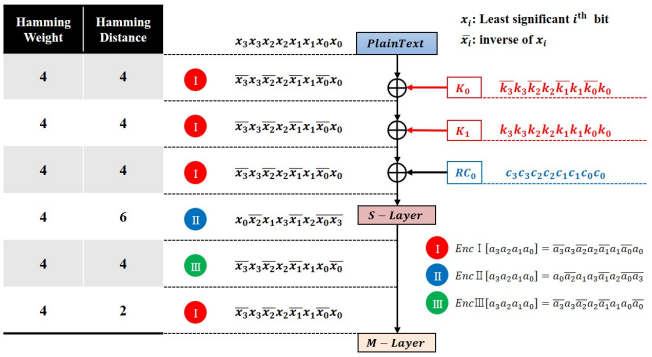


Fig. 1: Part of the structure for the balanced encoding of PRINCE

generate a non-constant Hamming distance value. The scheme is described in further detail in [30]. The above explanation of Encode I, Encode II, and Encode III is just one example of a balanced encoding scheme and there are other potential ways to design such a scheme.

We now turn out attention to the security of the balanced encoding countermeasure of [30].

3 Security Evaluation based on Polynomial Leakage Model

In this section, we conduct a security evaluation with a simulated leakage model. By performing standard and bit-wise CPA, we can examine whether the balanced encoding ensures security under both analysis methods or not. First, however, we need to define the leakage model. As with prior works [29, 32], we assume that the leakage function follows a polynomial form as Eqn. (1).

$$L(X) = \sum_{i} a_i \cdot x_i + \sum_{i,j} b_{i,j} \cdot (x_i \cdot x_j) + \sum_{i,j,k} c_{i,j,k} \cdot (x_i \cdot x_j \cdot x_k) \tag{1}$$

where x_i indicates the i^{th} bit of the sensitive value X and a_i, $b_{i,j}$ and $c_{i,j,k}$ are some weighting coefficients, and $a_i \sim \mathcal{N}(\mu_i, \sigma_i)$, $b_{i,j} \sim \mathcal{N}(\mu_{i,j}, \sigma_{i,j})$ and $c_{i,j,k} \sim \mathcal{N}(\mu_{i,j,k}, \sigma_{i,j,k})$, where μ_i and σ_i denote mean and variance respectively.

The difference between our model and prior works [29, 32] is that we assume the coefficient follows a normal distribution. We make this assumption to create a more realistic model. Also, we consider that the mean of the coefficients should be **close to one**, because most cryptographic devices conform to a Hamming weight model when implemented in software. In other words, we consider that each distribution of coefficients will likely be equal or very similar.

Previous work [32] has modeled coefficients of leakage following $\mathcal{N}(1, \sigma)$ and as such we consider the case which satisfies $\mu_i = \mu_{i,j} = \mu_{i,j,k} = 1$. Under this assumption, prior work [30] declare that the balanced encoding countermeasure is

secure. But we do not consider this to be a realistic model since in the real-world the value of μ_i is **close to but never exactly 1**. We therefore try to consider a more realistic leakage model. In order to investigate the effect of higher degree coefficients, we can assign some weighting values for the coefficients. However, we limit the change of the coefficients on a small scale, to maintain the Hamming weight assumptions. For example, $\mu_i = 1.00 < \mu_{i,j} = 1.01 < \mu_{i,j,k} = 1.02$.

We therefore consider two conditions. The first condition is the fundamental approach where the weighting coefficients of the polynomial functions are **identical** for the degree of the polynomial, *i.e.* $\mu_i = \mu_a, \mu_{i,j} = \mu_b, \mu_{i,j,k} = \mu_c$, $\sigma_i = \sigma_a, \sigma_{i,j} = \sigma_b$, and $\sigma_{i,j,k} = \sigma_c$, and $\mu_a \approx 1, \mu_b \approx 1, \mu_c \approx 1$. The second condition considers that the weighting coefficients of the polynomials are **different from each other**. These models are called balanced and imbalanced leakage models, respectively. Also, these approaches are used to evaluate the balanced encoded PRINCE traces generated from a simulation tool such as Matlab.

To assess their security, we target the sensitive value X from the **S-layer** output of the encoding scheme in Fig.1. We focus on Encode II in this work because it is directly applied to the output of the **S-layer**. That is, $O = L(EncII(X))$. We apply both standard CPA and bit-wise CPA during the security evaluation of the balanced encoding countermeasure. We now briefly introduce standard and bit-wise CPA.

3.1 Standard and Bit-wise CPA

In order to perform a CPA, after predicting the leakage model, the adversary will calculate the correlation between the acquired traces and the guessed intermediate variables. However, in case of a balanced encoding countermeasure, the guessed intermediate variables are constant (assuming a perfect Hamming weight model). We therefore wish to consider bit-wise CPA to find the leaked information from the balanced encoding countermeasure. When performing standard CPA, we denote the guessed intermediate variable including the Hamming weight leakage model $H(\cdot)$ and the intermediate variable X as $H(X)$. In the PRINCE block cipher, it is obvious that $H(X)$ is represented as the Hamming weight value of the nibble. And, when performing bit-wise CPA, we denote that X_i is the least significant i^{th} bit, and can use $H(X_i)$, corresponding to the guessed intermediate variable $(i = 0, 1, 2, 3)$.

3.2 Security metrics

Before we conduct the security evaluation for the balanced encoding countermeasure, and in order to quantify the effectiveness of our attack with a security metric, we employ guessing entropy [13,16]. Let $\mathbf{g}_q$ be the vector including the key candidates sorted according to the evaluation result, after side channel analysis has been performed: $\mathbf{g}_q := [g_1, g_1, ..., g_{|S|}]$ (S denotes the set of key candidates), and $\mathbf{L}_q$ be the random vector of the observations generated with q queries to the target physical device, and $\mathbf{l}_q = [l_1, l_2, ..., l_q]$ be a realization of this vector. We define the index of a key class s in side channel analysis as: $\mathsf{l}_s(\mathbf{g}_q) = i$ such that

$g_i = s$. The guessing entropy is then the average position of s in this vector, as shown Eqn. (2):

$$\mathbf{GE}_S = \mathop{\mathbf{E}}_{s} \mathop{\mathbf{E}}_{\mathbf{l}_q} \mathsf{l}_s(\mathbf{g}_q) \tag{2}$$

Intuitively, the guessing entropy means the average number of key candidates required for successful evaluation after the side channel analysis has been performed. Additionally, guessing entropy will be averaged over 1,000 trials in this work.

3.3 Security Evaluation for Balanced Leakage Model

We conduct the security evaluation for the balanced encoding countermeasure when assuming the fundamental approach. Because section 4 will present the results of performing CPA for the real traces, the guessing entropy is adopted as the security metric. We consider some specific cases for assessing the security as follows.

- Case 1 : $\mu_c = \mu_b = \mu_a > 0, \sigma_c = \sigma_b = \sigma_a > 0$
- Case 2 : $\mu_c > \mu_b > \mu_a > 0, \sigma_c = \sigma_b = \sigma_a > 0$
- Case 3 : $\mu_c = \mu_b = \mu_a > 0, \sigma_c > \sigma_b > \sigma_a > 0$
- Case 4 : $\mu_c > \mu_b > \mu_a > 0, \sigma_c > \sigma_b > \sigma_a > 0$

In Case 1, the mean and variance are constant, reflecting the approach of previous works [29, 32]. The main purpose of Case 2 is to examine the variation of the means, and Case 3 to examine the variation of the variances, of the coefficients in Eqn.(1). Case 4 is a combination of Case 2 and Case 3. The cases consider that the mean and the variance of the coefficients of the leakage functions are non-zero in the real world and we assume that the leakage model conforms to the Hamming weight model, where $\mu_a \approx 1, \mu_b \approx 1, \mu_c \approx 1$, and $\mu_a \neq \mu_b \neq \mu_c$. In other words, the above cases are more realistic in practice than previous works [29, 32]. We give a high weighting to coefficients of higher degree since they exhibit higher mean and variance values. Without variance in the coefficents there would be no leakage and therefore in pursuit of a more realistic model, it is an inevitable parameter to select.

Fig.2 represents the result of security evaluations for the balanced encoding countermeasure. Against our expectations, all cases have a similar result. That is, the leakage of information is independent of the mean and variance of the leakage distribution when assuming a balanced leakage model. However, we feel that it is hard to perfectly conceal sensitive variables from the balanced encoding countermeasure, since the guessing entropy has under 6 candidates when performing bit-wise CPA.

In Fig.2, all cases seem to be secure, since guessing entropy is between 10 or 12 when performing standard CPA. Like prior work [30], the balanced encoding countermeasure is shown to resist standard CPA. The complexity of an exhaustive attack of the PRINCE round key is about 32bit or 41bit, if we consider guessing entropy 4 or 6. In other words, the round key complexity can be reduced by about 50% because the round key is 64bit.

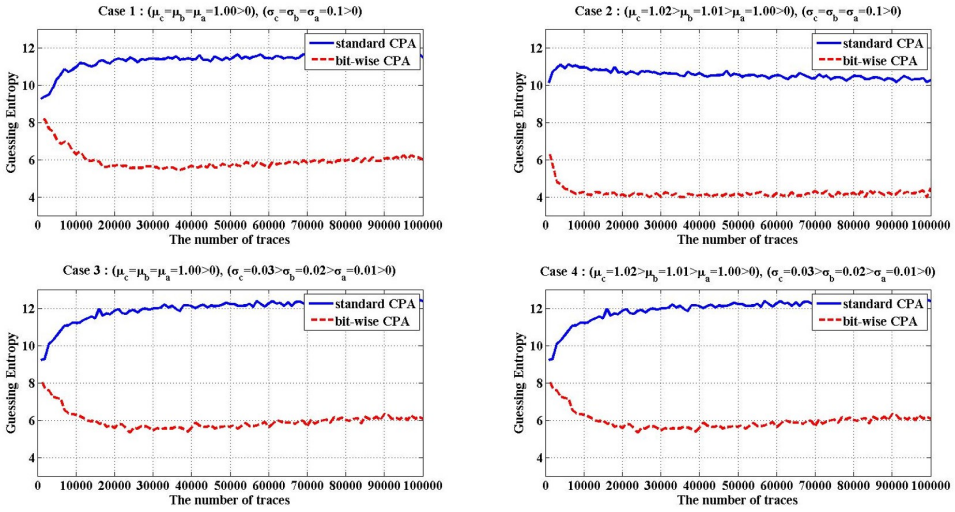


Fig. 2: The result of security evaluations for the balanced encoding countermeasure when assuming a balanced leakage model

There is of course a tolerance against basic side channel analysis when properties of the device significantly conform to the assumption. We consider, however, that there are few cryptographic devices which would have these ideal properties. Therefore, in the next subsection, we introduce a more realistic assumption than the just considered fundamental approach.

3.4 Security Evaluation for Imbalanced Leakage Model

We now consider the scenario of an imbalanced leakage model. As previously mentioned, we assign a different-weight for the coefficients of a polynomial leakage function. However, we only control the first degree of the polynomial, since this is enough to create a more realistic model, although higher degree coefficients could be considered. That is, $b_{i,j} = c_{i,j,k} = 0$. We therefore adopt a linear model, however, we expect that it can reflect a more realistic model, because the distributions of the coefficients are different from each other. Likewise as per the balanced leakage model, we assume that $\mu_i \approx 1$, but $\mu_i \neq \mu_j$.

For investigation of the security, we make the assumption that only a few bits have a biased distribution. For example, if only signal bit has a biased distribution, $\mu_0 = 1.01, \mu_i = 1.00$, with i not equal to zero. The reason why we set this distribution is to examine the security of the balanced encoding countermeasure, when the leakage model is extremely biased. Additionally, we cannot handle all cases, because there would be too many to consider. In this work, we therefore only examine two cases, namely a single biased bit and two biased bits.

Because the state of unprotected PRINCE is represented as a nibble, it is possible to insert a biased distribution for at most 4 positions of the state for

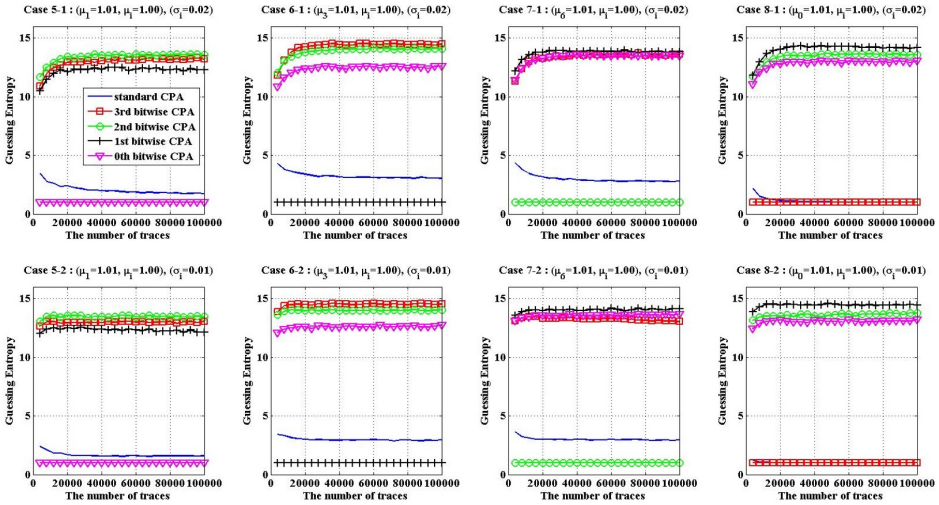


Fig. 3: The result of security evaluation for the balanced encoding countermeasure when assuming single bit-weighted imbalanced leakage model

the balanced encoding countermeasure in the case of a biased single bit. Since $EncII[x_3x_2x_1x_0]$ is represented as $x_0\bar{x_2}x_1x_3\bar{x_1}x_2\bar{x_0}\bar{x_3}$, we can insert a biased distribution for any position. But when inserting a biased distribution for x_i or $\bar{x_i}$, the same result is expected. Therefore, we have to select the position for insertion of the biased distribution; the original bit or complementary bit. In this work we choose the complementary bit. Then, we can find matching bits between the pre-conversion nibble and the post-conversion byte. For instance, pre-conversion bit x_3 is matched with post-conversion least significant 0^{th} bit. That is, a_0 only has a different distribution: $a_0 \sim \mathcal{N}(\mu_0, \sigma_0)$, $a_i \sim \mathcal{N}(\mu, \sigma)$, where $i \neq 0$. Also, we assign different variances for all cases, in order to investigate the effect of variance.

The result of performing standard CPA and bit-wise CPA is illustrated in Fig.3. Also, the mean of the distribution of coefficients is set to **almost 1**. As previously mentioned, our premise is that the real model almost follows the Hamming weight model. That is, $a_i \sim \mathcal{N}(1.01, \sigma)$, $a_j \sim \mathcal{N}(1.00, \sigma)$, where $0 \leq j \leq 8$, $i \neq j$, $\sigma = 0.01$ or 0.02. As explained earlier, we consider that i is 0 or 1 or 3 or 6. As expected, the sensitive variables is not concealed for all cases when performing bit-wise CPA. However, excluding Case 8, it seems to be secure when performing standard CPA. It is hard to explain the result of Case 8[1]. In any case, another countermeasure is required as all other cases exposed the key information. More precisely, we examine the effect of variance of distribution. For this, we set σ to 0.01 or 0.02. In case of a large variance, recovering the

[1] It is still not sure the cause of result. However the result slightly depends on the correlation between data. $Corr(x_3x_2x_1x_0, x_3) \approx 0.87$ but other cases are lower than 0.5. ($Corr(A, B)$: correlation coefficient between A and B)

key requires more traces. For example, in Case 5-1, it requires 5,000 traces to recover the key when performing 0^{th} bit-wise CPA, but it only requires a few hundred in Case 5-2. Therefore, as well as the mean of the coefficient, the variance has influence on leakage information. As in prior work [30], leakage information cannot be detected via performing standard CPA. However, we find the balanced encoding countermeasure leaks information when conducting bit-wise CPA.

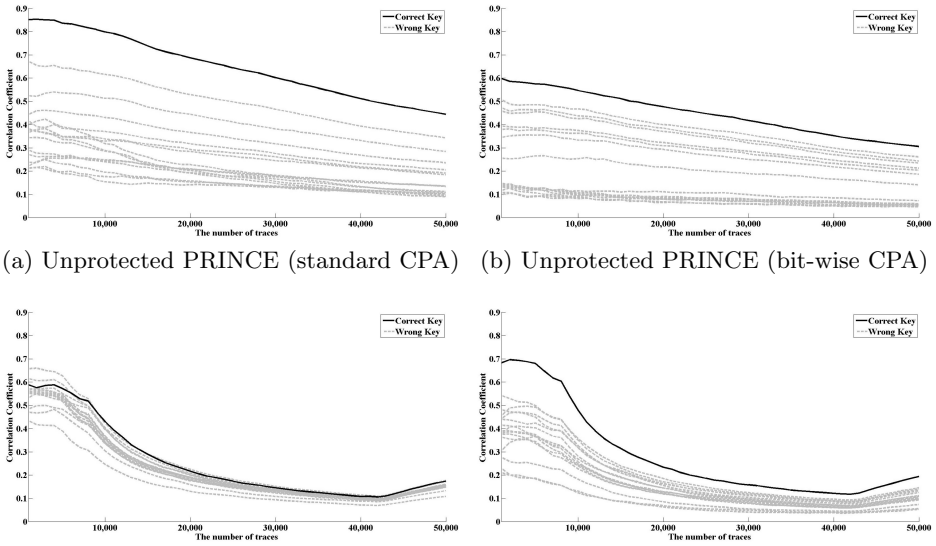
It may be that the leakage of the balanced encoding countermeasure is revealed, thanks to the constrained rule. Therefore the former rule is extended. As already stated, we examine that biased-weight is two bits. Similar to former rule, when (i, j) is $(0, 1)$ or $(0, 3)$ or $(0, 6)$ or $(1, 3)$ or $(1, 6)$ or $(3, 6)$, we give some weightings. More precisely, $a_i \sim \mathcal{N}(0.99, \sigma)$, $a_j \sim \mathcal{N}(1.01, \sigma)$, $a_k \sim \mathcal{N}(1.00, \sigma)$, where $0 \leq k \leq 8$, $i \neq j \neq k$, $\sigma = 0.01$ or 0.02. As expected, this gives results which are analogous to former results as provided in Appendix A. Namely, the balanced encoding countermeasure cannot ensure security in any situation. The success of the attack may be a natural result because we give some weightings for the distribution. However, surprisingly, it fails to recover the key when performing standard CPA for all cases. It means that it cannot guarantee security for bit-wise CPA, only for performing standard CPA.

Therefore, it is important that future constant leakage countermeasures should be resilient against both standard and bit-wise CPA. We note that as the variance of the distribution of the coefficient increases so does its resistance.

4 Experimental Results for an AVR Implementation

In this section, we conduct a security evaluation for the balanced encoding countermeasure on an AVR microcontroller board. For this, the power consumption of the AVR microcontroller chip has been measured using a 1 GHz LeCroy WaveRunner 104MXi 8-bit digital-storage oscilloscope. For all experiments, we used a sampling rate of 500MS/s. The target of attack is the output of **S-layer**, and a total of 500,000 power traces with random plaintext inputs were acquired. Additionally, for calculating the guessing entropy, many power traces are required.

We performed CPA on unprotected and balanced encoded PRINCE. As mentioned earlier, in order to compare between standard CPA and bit-wise CPA, we only consider the single output of the S-box. The results are represented in Fig.4. As authors of prior work claimed [30], the correlation between power traces and assumed leakage model is significantly reduced in the balanced encoding countermeasure. Again, the correlation of unprotected PRINCE has about 0.6 in Fig.4 (a), but one of balanced encoded PRINCE has about 0.2 in Fig.4 (c). That is, the correlation is reduced by a third. However, in case of the result of performing bit-wise CPA, the key is directly revealed although there is not much change in correlation. In fact, when comparing between Fig.4 (a) and Fig.4 (b), bit-wise CPA doesn't allow a better result than standard CPA. The result represented as Fig.4 shows that the balanced encoding countermeasure is definitely more vulnerable to bit-wise CPA. As stated in section 3, it directly reveals the vulnerability



(a) Unprotected PRINCE (standard CPA) (b) Unprotected PRINCE (bit-wise CPA)

(c) Balanced PRINCE (standard CPA) (d) Balanced PRINCE (bit-wise CPA)

Fig. 4: The standard and bit-wise CPA results for the unprotected PRINCE (a), (b), and balanced encoding PRINCE (c), (d), respectively. The y-axis indicates the absolute correlation coefficient, and the x-axis indicates the number of real traces used.

if the real power model tends to be slightly biased-weight at any bit location. Fig.5 illustrates how vulnerable the balanced encoding countermeasure is.

When performing bit-wise CPA, we can get the sensitive information using only a few thousand power traces. However, it's not enough to get sensitive information when only performing standard CPA. As seen in Table 1, in order to get satisfactory guessing entropy, many power traces are required. However, the guessing entropy for a few thousand power traces is not much different than for one requiring thousands. For example, in case of standard CPA, the guessing entropy ranges from 3.75 to 3 in Fig.5. In conclusion, it allows us to easily retrieve round key candidates which are included in correct round key.

Note that the result is analogous to our simulated leakage model. In comparision with single-weighted leakage model, the trend of guessing entropy has some similarity with the result of Case 5 in Fig.3. In other words, the guessing entropy is very poor when performing standard CPA. Additionally, it is vulnerable when performing 0^{th} bit-wise CPA. But, some parts do not match with the result of simulation; the result of 2^{nd} and 3^{rd} bit-wise CPA. Also, some parts are analogous to Case 9 of Fig.6 and Fig.7 in Appendix A; the result of 0^{th} and 1^{st} bit-wise CPA. Nevertheless, real traces revealed the sensitive information when conducting bit-wise CPA. However, assuming the balanced leakage model, it doesn't perfectly reveal the sensitive information, even when performing bit-

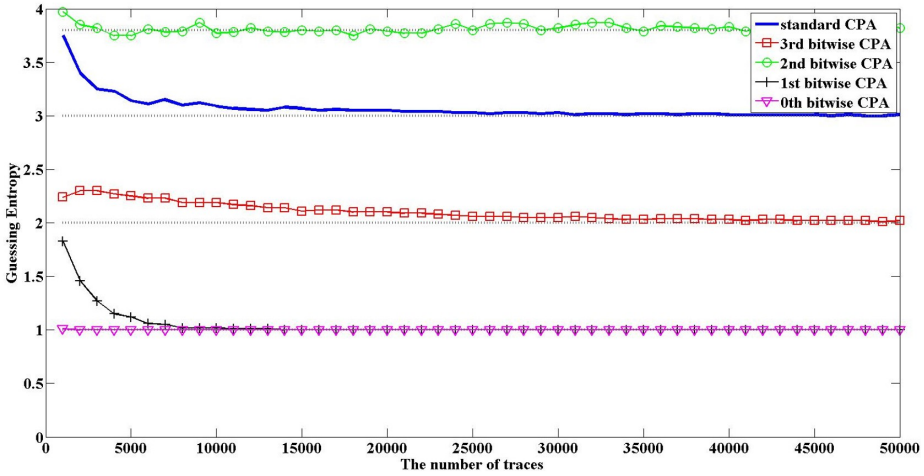


Fig. 5: The result of security evaluation for balanced encoding against real traces in an AVR board

wise CPA. It means that the balanced leakage model is not matched with the real model. As mentioned earlier, the security is ensured if the real model is matched with a balanced leakage model.

Attack	The minimum number of traces	Asymptotic guessing entropy
standard CPA	45,000	3
3^{rd} bit-wise CPA	50,000	2
2^{nd} bit-wise CPA	4,000	3.8
1^{st} bit-wise CPA	15,000	1
0^{th} bit-wise CPA	2,000	1

Table 1: The result of security evaluation for the balanced encoding against real traces in an AVR board. In order to get asymptotic guessing entropy, it illustrates the minimum number of traces.

5 Conclusion and Future Work

This work conducts a security evaluation of a balanced encoding countermeasure. Before we can examine whether the countermeasure is effective, it is necessary to build simulated leakage models; balanced and imbalanced leakage models. Unlike prior works [29,32], in accordance with Hamming weight model, we assume that

the coefficients of polynomial obey a normal distribution. In all cases, it allows us to identify the leak of the balanced encoding countermeasure from the simulated leakage model. Also, we demonstrate that it reveals the vulnerability of the balanced encoding countermeasure from real traces. In conclusion, from both simulated traces and real traces, we can detect the sensitive information when only using a basic side channel analysis approach.

In [32], it was suggested that the balanced encoding scheme could be combined with masking and/or randomization countermeasures, since they are independent from each other. However, this combination of countermeasures needs to be re-examined in light of this work.

Acknowledgements. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2013R1A1A2A10062137). The authors would like to thank Dooho Choi at ETRI for supporting us with SCARF boards (http://www.k-scarf.or.kr/). The SCARF boards were supported by the KLA-SCARF project, the ICT R&D program of ETRI.

References

1. Kocher, P., Jaffe, J., and Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999, LNCS, vol. 1666, pp. 388-397. Springer, Heidelberg (1999)
2. Satoh, A., Morioka, S., Takano, K., and Munetoh, S.: A Compact Rijndael Hardware Architecture with S-box Optimization. In: Boyd, C. (ed) ASIACRYPT 2001, LNCS, vol. 2248, pp. 239-254. Springer, Heidelberg (2001)
3. Bevan, R. and Knudsen, E.: Ways to Enhance Differential Power Analysis. In: Lee, P-.J., Lim, C-.H. (eds.) ICISC 2002, LNCS, vol. 2587, pp. 327-342. Springer, Heidelberg (2002)
4. Messerges, T-.S., Dabbish, E-.A., and Sloan, R-.H.: Examining Smart-Card Security under the Threat of Power Anlaysis Attacks. In: IEEE Transactions on Computers, vol. 51, pp. 541-552. (2002)
5. Tiri, K. and Verbauwhede, I.: Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology. In: Walter, C-.D., Koc, C-.K., Paar, C. (eds.) CHES 2003, LNCS, vol. 2779, pp. 125-136. Springer, Heidelberg (2003)
6. Ishai, Y., Sahai, A., and Wagner, D.: Private Circuits: Securing Hardware against Probing Attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463-481. Springe,r Heidelberg (2003)
7. Sokolov, D., Murphy, J., and Bystrov, A.: Improving the Security of Dual-Rail Circuits. In: Joye, M., Quisquater, J-.J. (eds.) CHES 2004, LNCS, vol. 3156, pp. 282-297. Springer, Heidelberg (2004)
8. Brier, E., Clavier, C., and Olivier, F.: Correlation Power Analysis with a Leakage Model. In: Joye, M., Quisquater, J-.J. (eds.) CHES 2004, LNCS, vol. 3156, pp. 16-29. Springer, Heidelberg (2004)
9. Waddle, J. and Wanger, D.: Towards Efficient Second-Order Power Analysis. In: Joye, M., Quisquater, J-.J. (eds.) CHES 2004, LNCS, vol. 3156, pp. 1-15. Springer, Heidelberg (2004)

10. Tiri, K., and Verbauwhede, I.: A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In: DATE 2004: Proceedings of the conference on Design, automation and test in Europe, pp. 246-251. IEEE Computer Society, Washington (2004)
11. Popp, T. and Mangard, S.: Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints. In: Rao, J-.R., Sunar, B. (eds) CHES 2005, LNCS, vol. 3659, pp. 172-186. Springer, Heidelberg (2005)
12. Suzuki, D. and Saeki, M.: Security Evaluation of DPA Countermeasures Using Dual-rail Pre-charge Logic Style. In: Goubin, L., Matsui, M. (eds) CHES 2006, LNCS, vol. 4249, pp. 255-269. Springer, Heidelberg (2006)
13. Standaert, F.-X., Malkin, T.G., and Yung, M.: A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. Cryptology ePrint Archieve, Report 2006.139.
14. Schramm, K. and Paar, C.: Higher Order Masking of the AES. In: Pointcheval D. (ed) CT-RSA 2006, LNCS, vol. 3860, pp. 239-254. Springer, Heidelberg (2006)
15. Mangard, S., Oswald, E., and Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer (2007)
16. Standaert, F.-X., Gierlichs, B., and Verbauwhede, I.: Partition vs. Comparison Side-Channel Distinguishers: An Empiricl Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices. In: Lee, P. and Cheon, J. (eds.) ICISC 2008, LNCS, vol. 5461, pp. 253-267. Springer, Heidelbert (2009)
17. Rivain, M., Dottax, E., and Prouff, E.: Block Ciphers Implementations Provably Secure Against Second Order Side Channel Analysis. In: Nyberg, K. (ed.) FSE 2008, LNCS, vol. 5086, pp. 127-143. Springer, Heidelberg (2008)
18. Pan, J., Hartog, J.I.d., and Lu, J.: You Cannot Hide behind the Mask: Power Analysis on a Provably Secure S-Box Implementation. In: Yourm, H-.U., Yung, M. (eds.) WISA 2009, LNCS, vol. 5932, pp. 178-192. Springer, Heidelberg (2009)
19. Furnaroli, G., Martinelli, A., Prouff, E., and Rivain, M.: Affine Masking against Higher-Order Side Channel Analysis. In: Biryukov, A., Gong, G., Stinson, D-.R. (eds.) SAC 2010, LNCS, vol. 6544, pp. 262-280. Springer, Heidelberg (2010)
20. Rivain, M. and Prouff, E.: Provably Secure Higher-Order Masking of AES. In: Mangard, S., Standaert, F-.X. (eds.) CHES 2010, LNCS, vol. 5086, pp. 127-143. Springer, Heidelberg (2010)
21. Guilley, S., Sauvage, L., Flament, F., Vong, V-.N., Hoogvorst, P., and Pacalet, R.: Evaluation of Power Constant Dual-Rail Logics Countermeasures against DPA with Design Time Security Metrics. IEEE Trans. Computers, vol. 59, pp. 1250-1263. (2010)
22. Goubin, L. and Martinelli, A.: Protecting AES with Shamir's Secret Sharing Scheme. In: Preneel, B., Takagi, T. (eds.) CHES 2011, LNCS, vol. 6917, pp. 79-94. Springer, Heidelberg (2011)
23. Hoogvorst, P., Duc, G., and Danger, J-.L.: Software Implementation of Dual-Rail Representation. COSADE 2011, pp. 73-81. (2011)
24. Kim, H-.S., Hong, S., and Lim, J.: A Fast and Provably Secure Higher-Order Masking of AES S-box. In: Preneel, B., Takagi, T. (eds.) CHES 2011, LNCS, vol. 6917, pp. 95-107. Springer, Heidelberg (2011)
25. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E. B., Knezevic, M., Knudsen, L. R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S. S., and Yalçin, T.: PRINCE A Low-Latency Block Cipher for Pervasive Computing Applications. In Wang, X. and Sako, K. (eds.) ASIACRYPT 2012 LNCS, vol. 7658, pp.208-225. Springer, Heidelberg (2012)

26. Naekawa, A., Yamashita, N., Tsunoo, T., Minematsu, K., Suzuki, T., and Tsunoo, Y.: Tamper-Resistance Techniques Based on Symbolic Implementation Against Power Analysis. SCIS 2013, pp.73-81. (2013)
27. Tunstall, M., Whitnall, C., and Oswald, E.: Masking Tables-An Underestimated Security Risk. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 425-444. Springer, Heidelberg (2014)
28. Coron, J-.S., Prouff, E., Rivain, M., and Roche, T.: Higher-Order Side Channel Security and Mask Refreshing. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 410-424. Springer, Heidelberg (2014)
29. Grosso, V., Standaert, F.-X., and Pouff, E.: Low Entropy Masking Schemes, Revisited. In: Francilon, A. and Rohatgi, P. (eds.) CARDIS 2013. LNCS, vol. 8419, pp. 33-43. Springer, Heidelberg (2014)
30. Chen, C., Eisenbarth, T., Shahverdi, A., and Ye, X.: Balanced Encoding to Mitigate Power Analysis: A Case Study. In: Joye, M., Moradi, A. (eds.) CARDIS 2014, LNCS, vol. 8968, pp. 49-63. Springer, Heidelberg (2014)
31. Ding, A.A., Zhang, L., Fei, Y., Luo, P.: A statistical model for higher order DPA on masked devices. In: Batina, L. and Robshaw, M. (eds.) CHES 2014, LNCS, vol. 8731, pp. 147-169. Springer, Heidelberg (2014)
32. Servant, V., Debande, N., Maghrebi, H., and Bringer, J.: Study of a Novel Software Constant Weight Implementation. In: Joye, M., Moradi, A. (eds.) CARDIS 2014, LNCS, vol. 8968, pp. 35-48. Springer, Heidelberg (2014)

A The result of security evaluation for the **balanced encoding** countermeasure when assuming multiple bit-weighted imbalanced leakage model

Fig.6 and Fig.7 represent the result of conducting standard CPA and bit-wise CPA when assuming imbalanced leakage model. As previously stated, we assume that two bits of the coefficients have some weighted-value. That is, (i, j) is $(0, 1)$ or $(0, 3)$ or $(0, 6)$ or $(1, 3)$ or $(1, 6)$ or $(3, 6)$. For example, in case of $(i, j) = (0, 1)$, $\mu_0 = 0.99, \mu_1 = 1.01, \mu_i = 1.00$ $(i \neq 0, 1)$.

Also, in order to examine the effect of variance of leakage model, variance is considred as 0.01 or 0.02. In more detail, it does not conceal the key information when performing 0^{th} and 1^{st} bit-wise CPA at Case 9-1 in Fig.6 and Case 9-2 in Fig.7. The success of the attack may be a natural result because we give some weight for distribution of a_3 and a_1. Additionally, when the value of variance is 0.02, resistance strength to standard CPA is stronger. That is, the slope in Fig.6 gentle. Anyhow, this countermeasure doesn't have a resistance to basic side channel analysis any more. Additionally, in all cases, standard CPA doesn't work. Note that it seems to be secure when only conducting standard CPA. In conclusion, if the real leakage model obeys roughly Hamming weight model (the coefficients **nearly equal to 1**), the vulnerability of this countermeasure is revealed. Most devices may be difficult to conform perfectly Hamming weight model. Thus, our assumption is not unrealistic.

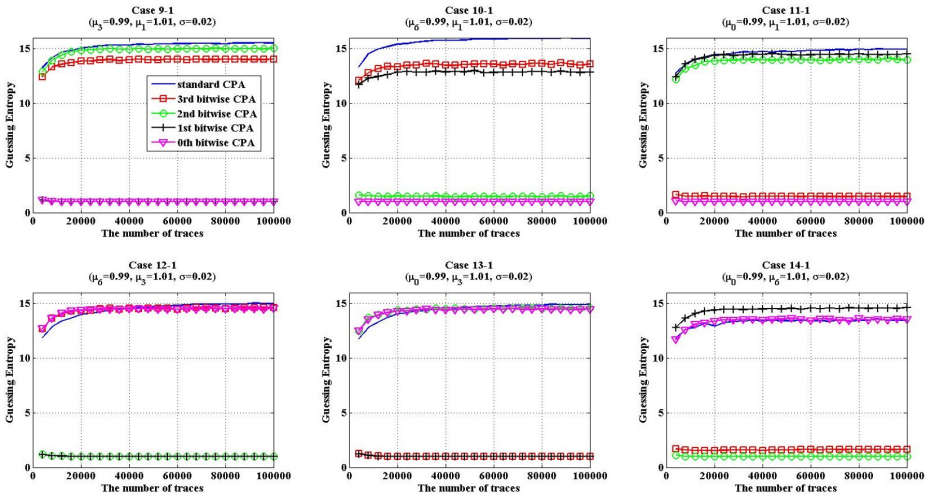


Fig. 6: The result of security evaluation for the balanced encoding countermeasure when assuming multiple bit-weighted imbalanced leakage model (σ=0.02)

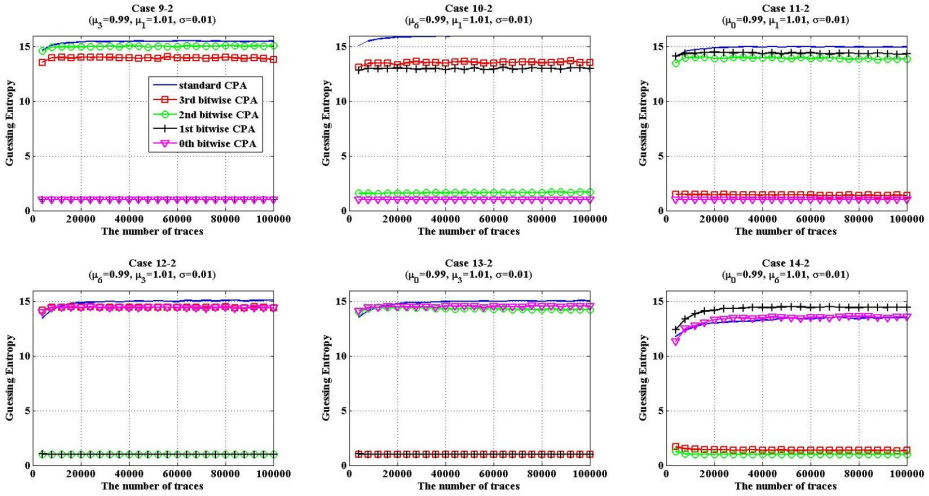


Fig. 7: The result of security evaluation for the balanced encoding countermeasure when assuming multiple bit-weighted imbalanced leakage model (σ=0.01)