Templates as Master Keys

Dakshi Agrawal[1], Josyula R. Rao[1], Pankaj Rohatgi[1], and Kai Schramm[2]

[1] IBM Watson Research Center, P.O. Box 704
Yorktown Heights, NY 10598, USA
{agrawal, jrrao, rohatgi}@us.ibm.com
[2] Communication Security Group, Ruhr-Universität Bochum
Universitaetsstrasse 150, 44780 Bochum, Germany
schramm@crypto.ruhr-uni-bochum.de

Abstract. We introduce two new attacks: the *single-bit template attack* and the *template-enhanced DPA attack*. The single-bit template attack can be used very effectively to classify even *single* bits in a single side channel sample with a high probability of correctness. The template-enhanced DPA attack, combines traditional DPA with single-bit template attacks to show that if an adversary has access to a test card with even a slightly biased RNG, then he/she can break *protected* cryptographic implementations on a target card even if they have perfect RNGs. In support of our claim, we report results from experiments on breaking two implementations of DES and AES *protected by the masking countermeasure* running on smartcards of different manufacturers.

In light of these results, the threat of template attacks, generally viewed as intrinsically difficult to mount, needs to be reconsidered.

1 Introduction

Several side channel cryptanalytic techniques, such as those based on measuring timing, power consumption and electromagnetic emanations have been used effectively to launch a wide range of attacks such as simple power analysis (SPA), differential power analysis (DPA), higher order DPA, template attacks and multi-channel attacks [Koc96, KJJ99, AARR02, CRR02, ARR03] against a wide variety of cryptographic devices. While countermeasures, even provably secure ones, have been developed for some attacks such as DPA, the perceived difficulty (in terms of the work effort required by an adversary) of launching other attacks has led developers to discount their feasibility.

This is particularly true for template attacks. For instance, the very high successful classification results that can often be achieved with the analysis of a single side channel sample, make template attacks the ideal choice to attack ciphers, such as stream ciphers, which use ephemeral keys. However, until now all published works [CRR02, RO04] used template attacks to classify the state of a byte, e.g., a key byte in RC4. This makes the process of creating templates quite tedious since 256 templates need to be created for each byte. Further, templates for the full attack cannot be precomputed as the templates for a subsequent key

J.R. Rao and B. Sunar (Eds.): CHES 2005, LNCS 3659, pp. 15–29, 2005.

byte need to be created for each likely hypothesis for the earlier key bytes, i.e., the template building process can only be guided by partial attack results. In this paper, we show that this apparent difficulty is not intrinsic and present two new attack techniques to surmount it.

1.1 Contributions

Our first contribution is the *single-bit template attack* technique. For a given bit, this attack *uses DPA to build templates*. It relies on our experimental observation that templates can be built from peaks observed in a DPA attack and these templates can predict the value of a *single DPA-targeted bit* in a *single side channel sample* with high probability. Thus, even though the specific computation yielding the single sample uses byte sized variables, the template can predict a single bit from those variables.

This technique immediately yields attacks where an adversary precomputes a large number of single-bit templates using several different DPA attacks on a test device and uses these precomputed templates and their classification probabilities to attack a single sample from an identical target device. These templates provide the best guess for each of the DPA-targeted bits and the template classification probabilities can be used to guide a weighted brute force search for the key. With enough precomputed templates, the entropy of the key is reduced substantially making the weighted brute force search practical. For example, in an experiment on a DES implementation, just attacking the 32-bits of S-box output in the first round, reduced the key entropy by over 16 bits. Clearly, by building templates, for DPA attacks carried on other variables in other rounds, the key entropy could be further reduced.

Reflecting further on the single-bit template attack, it should be evident, that knowledge of a single-bit template is comparable to having some partial knowledge about the key used in a card. Possession of several such single-bit templates is akin to having a master key that can be used to break any of a collection of cards from the same mask. This is true even for cards that are protected by DPA countermeasures such as secret sharing and random masking [GP99, CJR+99, AG01], if single-bit templates for the bits being processed in such cards can be built.

The second major contribution of this paper is to introduce *template-enhanced DPA* attacks which can be used to attack DPA protected cards under some assumptions. The problem with such cards is that single-bit templates (as described earlier) cannot be built, since in principle, the DPA protection renders DPA (the first step in building single-bit templates) infeasible. However, in practice, this is not a limitation, as there are multiple ways to get hold of a test card with a (slightly) biased RNG. For example, an adversary in collusion with the designers, testers and maintainers of card software may have hooks to add code to disable specific RNG registers on their own test cards while changes to deployed cards in the field may be much more tightly controlled and impossible for an adversary. Some production cards may fail the RNG tests at fabrication time and may be discarded only to be picked up by an adversary.

In our experience, we have sometimes encountered even production cards with slight RNG bias (to the tune of 3–4%). Therefore, if cards are not tested or tested to wide tolerance limits, then it is highly likely that several cards in the field may have slightly biased RNGs. As a last resort, an attacker could mount an intrusive attack to disable the RNG on his own test card.

Given a test card with a (slightly) biased RNG, an adversary can successfully perform multiple DPA attacks on the test card to build single-bit templates. The DPA peaks in these attacks would occur at locations where the masked value of the predicted variable bit (such as an S-box output bit) occur, since the masking is imperfect. Single-bit templates built using these DPA peaks would then be able to classify corresponding bits of the masked variables used in any card, including cards that have a perfect RNG. The *template-enhanced DPA* attack works by setting the DPA selector function to be the XOR of the standard DPA selector function (e.g., an S-box output bit for a key hypothesis) and the classification obtained by the single-bit templates (such as the masked S-box output bit). Depending on the effectiveness of the template classification, this DPA selector function will have high correlation with the mask bit being used. Thus for the right key hypothesis, this attack will show DPA peaks at locations where the random mask is being used.

We demonstrate such a single-bit template attack for two DPA protected implementations: a protected DES implementation on a 6805 based smartcard and a protected AES implementation on an AVR architecture. We also report a surprising result that indicates that in practice, the bias of the RNG in the test card has little relevance to the effectiveness of the *template-enhanced DPA* attack. The RNG bias only affects the effort required to build single-bit templates. The classification error with single-bit templates built using a slightly biased RNG is not significantly worse than the classification error using templates built using a completely broken (fixed at 0) RNG.

The paper is organized as follows: In Section 2, we introduce single-bit template attacks. In Section 3, we introduce the template-enhanced DPA attacks and show how it can be uses to attack two smartcards of different architecture[1], which run protected implementations of DES and AES.

2 Single-Bit Template Attack

We extend earlier work on template attacks [CRR02, RO04] that focused on classifying a byte in a computation, e.g., a byte of key used in RC4, by showing how template attacks can be applied to classify single bits in a computation from a single side channel sample.

A template attack begins by selecting variables occurring in the computation for which templates would be built. Furthermore, it requires a selection of significant points for each of the selected variables that are included in the corresponding template. Having a good selection criterion for significant points

[1] Smartcard A is an ST19 based on the 6805 architecture and smartcard B is an Atmel ATmega163 based on the AVR architecture.

is critical to the success of template attacks and this problem has been well studied: ideally, the significant points should have high variance with respect to the particular variable of interest. For example, Bohy et al. [BNSQ03] suggest Principal Component Analysis (PCA) while Rechberger et al. [RO04] suggest a simpler and computationally less expensive approach that resembles classical DPA. For the single-bit template attack, we let DPA attacks guide the selection of both the bits in the computation for which templates are built and significant points included in these templates. Templates are built for the bits for which a DPA attack is successful and the significant points included in a template are the points with the top N highest DPA-peaks.

We illustrate the attack by means of an example. Consider an unprotected implementation of DES on smartcard A. Consider the 32 s-box output bits of the DES computation in round one. For the unprotected DES implementation, one can easily perform DPA for each of the 32 output bits. Correspondingly, we built a pair of templates for each output bit corresponding to the bit being equal to 0 and 1 respectively. In order to build these templates, we performed a DPA of each output bit using the improved DPA metric described in [ARR03] which results in a higher signal-to-noise ratio (SNR) than the standard DPA. The improved metric is computed by using the following formula:

$$M_{H_i} = \frac{\left(\mu_{H_i} - \mu_{H_v}\right)^2}{\frac{\sigma^2_{H_v,0}}{N_0} + \frac{\sigma^2_{H_v,1}}{N_1}} - \ln\left(\frac{\frac{\sigma^2_{H_i,0}}{N_0} + \frac{\sigma^2_{H_i,1}}{N_1}}{\frac{\sigma^2_{H_v,0}}{N_0} + \frac{\sigma^2_{H_v,1}}{N_1}}\right) \tag{1}$$

where μ_H is the difference of sample means of signals in the 0-bin and the 1-bin respectively for a hypothesis H. Similarly, $\sigma^2_{H,0}$ and $\sigma^2_{H,1}$ are the sample variances of the signals in the 0-bin and the 1-bin respectively for a hypothesis H. H_i denotes a hypothesis where a subkey is assumed to be i, and H_v is a special hypothesis (null hypothesis) where signals are partitioned in the 0-bin and the 1-bin randomly.

Figure 1 displays the improved metric of s-box 1, bit 0. The figure reveals several points in time that clearly correlate with the selected s-box output bit. In

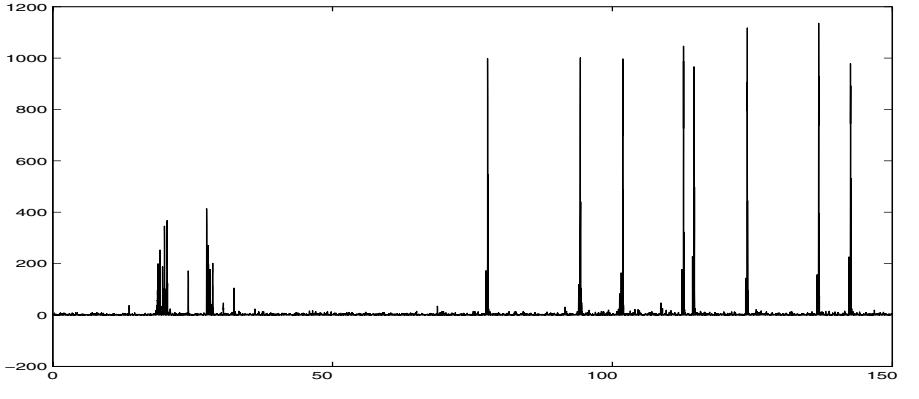

Fig. 1. Improved DPA metric of s-box 1, bit 0 of the test device. Time in μs.

Table 1. *s*-box output bit classification success rates and entropy loss

	s-box 1	**s-box 2**	**s-box 3**	**s-box 4**	**s-box 5**	**s-box 6**	**s-box 7**	**s-box 8**
bit 0	1.00	0.91	0.88	0.93	0.77	0.72	0.80	0.84
bit 1	0.98	0.88	0.92	0.94	1.00	0.92	0.97	0.77
bit 2	0.75	0.89	0.99	0.92	0.95	0.83	0.90	0.79
bit 3	0.90	0.91	0.72	0.85	0.83	0.86	1.00	0.89
entropy loss	2.57	2.10	2.13	2.30	2.28	1.50	2.61	1.35

our experiments, we chose the 50 highest peaks from this DPA metric to select significant points and built a pair of templates for these points for each *s*-box output bit using a single set of 1400 side channel samples.

To estimate classification success rate, we classified the state of the 32 *s*-box output bits using a single set of another 100 random side channel samples measured from the same device. The classification success rates $\eta_{S_i b_j}$ for the i-th *s*-box and j-th bit, $1 \leq i \leq 8$ and $0 \leq j \leq 3$, together with the corresponding entropy loss are shown in Table 1. The classification success rates ranged from 0.72 to 1.00; in the worst case *s*-box 3, bit 3 and *s*-box 6, bit 0 were predicted correctly for only 72 of the 100 samples. From these results, the probability that the entire 32-bit output of all *s*-boxes is classified correctly is $\prod_{i=1}^{8}\prod_{j=0}^{3}\eta_{S_i b_j} = 0.0154$ which although small is still 66-million times higher than a random guess.

These results can also be viewed in terms of entropy loss. For a particular bit, if the classification success rate is p, then its corresponding entropy loss is given by $1 + (1-p)\log_2(1-p) + p\log_2(p)$. To compute the entropy loss for multiple bits we can add the individual losses (this corresponds to the worst case where classification of different bits is independent). From this formula, we can see that 16.8-bits of entropy has been lost from the 48-bits of the DES key used in the first round (out of a maximum possible loss is 32-bits if the classification was perfect). The loss of entropy of the keyspace can be translated into reduced expected computational cost of a guided exhaustive search through the entire keyspace that examines more likely keys earlier than the less likely keys.

For DES implementations, the attack can be improved substantially. Templates can be built not just for round 1, *s*-box output bits but also for other bits such as the data bits fed to the second round. These templates will further narrow down the possibilities for the 48 key bits used in the first round. In addition, templates can be built for the corresponding DPA attacks on the last two rounds of DES (which utilize another 48-bit size subset of the key) and so on. Depending on the implementation, single-bit templates can also be built directly for the key bits that are likely to be highly effective since the same key bits show up in multiple locations in a round and across multiple rounds.

To summarize, single-bit template attacks are capable of classifying a single bit in a single side channel sample with high probability even though the influence of a single bit on the side channel signal is generally very little at

a particular instance of time, and is superimposed by several sources of noise including that from other adjacent bits. Cryptographic algorithms with high contamination properties [CRR02], such as DES, are ideally suited for single-bit classification. Multiple precomputed single-bit templates can lead to practical guided keyspace search algorithms using only a single sample from the target device. Moreover, single-bit attacks when combined with other attacks can result in much more devastating attacks as we show in the next section.

3 Attacking the Masking Countermeasure: Template-Enhanced DPA

The proposed attack consists of two steps: a *profiling phase* and a *hypothesis testing phase*. In the profiling phase, the adversary, who is in possession of a test card with a biased RNG, builds templates, and in the *hypothesis testing phase*, the adversary uses these prebuilt templates to mount a DPA-like attack on a target card which is identical to the test card, but has a perfect RNG.

3.1 Profiling Phase

We assume that the adversary has a test card with a biased RNG that produces 0 bits with some biased probability $\nu \neq 0.5$, and that the adversary only faces masking countermeasures such as the duplication method [GP99][2][3]. A masking countermeasure generally blinds all intermediate key-dependent variables with randomly generated masks. The original values of the intermediate variables can be recovered from their blinded values by applying the inverse mask. Non-linear functions such as the s-boxes in DES and AES cannot be dealt with this way; they are typically handled by creating masked tables in RAM. While the unmasked s-box output $s(x \oplus k)$ never occurs as a run-time variable during the execution of the algorithm, both the masked output $s(x \oplus k) \oplus m$ and the mask m do occur and thus leak in the side channel sample.

As an illustration, consider the upper plots of Figures 2 and 3 that show DPA attacks on two test cards, one with a protected DES implementation, and another with a protected AES implementation. The target of both attacks was the bit 0 of s-box 1 in round one. The differential samples were obtained by switching off the RNGs of the test cards ($\nu = 1$). Both plots show peaks at points in time when the masked s-box output bit leaks. Note that the differential trace from the AES implementation contains less peaks compared to the DES implementation due to the lower contamination properties of AES.

The first step in the profiling stage is to perform exhaustive DPA attacks on the test card using as many samples as possible. In a card with a biased RNG, where the mask is not perfectly random, such an attack will succeed since the

[2] We make this simplifying assumption just for the sake of exposition, the attacks would work if bad RNG has different biases for different bits in a random byte.

[3] We assume other countermeasures, such as the desynchronization of side channel samples due to random wait states etc., have been removed using signal processing.

DPA prediction of an algorithmic bit (e.g., s-box output bit $s(x \oplus k)$) would be correlated with the masked value of that bit. A successful DPA attack will give us the subkey k (in fact we will get all the subkeys) and also reveal the points of time t^* when targeted masked algorithmic bit (e.g., masked s-box output bit $s(x \oplus k) \oplus m$) leaks.

The second step of the profiling phase is to create single-bit templates based on each of the DPA attacks. For each DPA attack, the adversary builds a pair of templates for the masked bit being 0 and 1 by using the collected samples at the points where the DPA peaks appear. It may seem that building the template pairs will require that the adversary knows which of the N collected samples have the masked bit 0 and which have the masked bit 1. This is *not possible* in general, unless the RNG is completely broken in a known way (e.g., fixed at 0). Instead the adversary blindly assumes that the bit is exactly the same as the DPA prediction and builds the templates anyway.

Clearly, if the RNG is not fixed at 0, but has a probability ν of outputing a 0 bit, the templates built by an adversary have significant errors. For example when $\nu > 0.5$, then the 0-bit template will be built using roughly $\nu * N/2$ samples that are actually 0 samples and roughly $(1 - \nu) * N/2$ samples that are actually 1's. When $\nu < 0.5$, then the templates are inverted: the 0 template is built using more 1 samples than 0 samples. Such templates are equally useful since they will *consistently* predict the bit incorrectly with high probability. When $\nu = 0.5$, DPA will not work and the templates as described here cannot be built.

We will show later in the paper that even though significant errors are introduced in the templates when the RNG is very slightly biased, i.e., when ν is close to 0.5, *if enough signals are used to build these templates*, then the performance of *the template-enhanced DPA attack is not significantly impacted*—the attack works almost as well as an attack using perfect templates ($\nu = 1$).

3.2 Hypothesis Testing Phase

Once the adversary has built templates to classify masked s-box output bits in DES or AES using a test device with imperfect RNG, he/she is given a target device to attack that is identical to the test device, except for the fact that its RNG is perfect.

The adversary can make a hypothesis regarding the secret key k used in the target device, and for a particular side channel sample, use the key hypothesis to predict the unmasked output bit $s(x \oplus k)$. Furthermore, the adversary can use template classification to predict the masked output bit $s(x \oplus k) \oplus m$. These two together can be used to predict the mask bit m itself[4]

$$m = \underbrace{[s(x \oplus k)]}_{\texttt{prediction}} \quad \oplus \quad \underbrace{[s(x \oplus k) \oplus m]}_{\texttt{template classification}} \tag{2}$$

Since the mask bit m is an intermediate variable in the algorithm, it will leak at some instances of time in the side channel sample. The idea is to perform

[4] We assume that boolean masking is used.

a DPA-like attack on the prediction of m according to the equation above. If the hypothesized value of k is correct, peaks will show up in the corresponding differential trace at points in time when the mask bit m leaks.

The number of samples required to perform this attack depends on two main factors: the number of samples required to perform a DPA attack based on a perfect prediction for m and the template classification error probability ϵ. The first factor is a function of the leakage properties of m in the smart-card, while the second factor is dependent on the quality of single-bit templates. A higher value of ϵ results in worse SNR of the differential sample since classification errors make the predicted and actual values of the mask m less correlated. To estimate the impact of classification error ϵ, we modified the SNR model proposed by Messerges et al. in [MDS99] to account for the additional noise caused by the misclassification (details are given in the Appendix). Table 2 shows the impact of ϵ on the proposed template-enhanced DPA attack. This table assumes that a certain SNR ratio is obtained using 100 side-channel samples with perfect classification and computes how many side-channel samples would be needed to achieve the same SNR with different values of ϵ. Given the classification results obtained for single-bit templates in the earlier section, where all error probabilities were less than 0.3 and many were under 0.1, it would be reasonable to assume that the template-enhanced DPA attacks would be a factor of 1.5 to 6 more expensive (in terms of the number of required samples) than the regular DPA attacks.

Table 2. Number of measurements N required to achieve a constant SNR in a template-enhanced DPA attack for different template classification errors ϵ

ϵ	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.48	0.49
N	100	123	156	204	278	400	625	1,111	2,500	10,000	62,500	250,000

3.3 Results

We performed the proposed template-enhanced DPA attack on two smardcards: a protected DES implementation on the smartcard A and a protected AES implementation on the smartcard B. For each smartcard, in the profiling phase, the templates were built with the RNG turned off ($\nu = 1$). In the hypothesis testing phase, traces were obtained with the RNG on and working perfectly ($\nu = 0.5$). For the smartcard A, the lower plot in Figure 2 shows the differential trace of the template-enhanced DPA attack on the hypothesized mask bit m. A similar differential trace for the smartcard B is shown in the lower plot of Figure 3. Both plots contain distinct peaks even though the masking protection was fully functional. For completeness, Figure 4 shows a template-enhanced DPA trace for a false key hypothesis for smartcard B, which shows no peaks.

If the RNG in the test card during the profiling phase is just slightly biased instead of being broken, then the templates obtained from the test card would

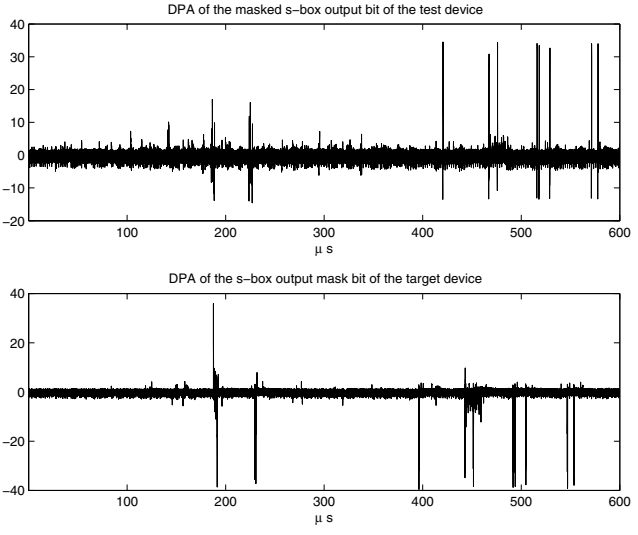

Fig. 2. Smartcard A: DPA of the masked s-box output bit using the test device and DPA of the mask bit using the target device

have significant cross-contamination. One may conjecture that as a result, the probability of error ϵ would be higher as the bias in RNG becomes smaller. However, this is not the case—in the appendix, we prove the following counter-intuitive result.

Theorem 1. *If the noise covariance matrix of side channel traces is the same for two values of a mask bit and enough traces are available from a test card with a biased RNG ($0.5 < \nu < 1.0$), then the templates prepared from such traces give the same probability of error as the templates obtained from a test-device with broken RNG ($\nu = 1$).*

In our experiments, we found that the noise covariance matrices of side channel traces for different values of mask bit are nearly the same. For the actual covariance matrices obtained in one of our experiments, we performed a Monte Carlo simulation of how well the signal classification works when templates are built using different numbers of samples from the test card with different RNG biases. In this simulations, the samples were generated by sampling from the noise probability distributions and the RNG bias was simulated by randomly misclassifying samples into the bins used to build templates. We also performed an actual experiment where 1000 samples were obtained from the test card and templates were build for different RNG biases (again simulated by putting samples randomly in incorrect bins). The results of these experiments are shown in Figure 5. Three plots are derived from the Monte Carlo simulation involving 1000, 10,000, and 100,000 traces from a simulated test card with biased RNG to build templates. These three plots show that as the number of traces from the test card increases, the probability of classification error becomes insensitive to

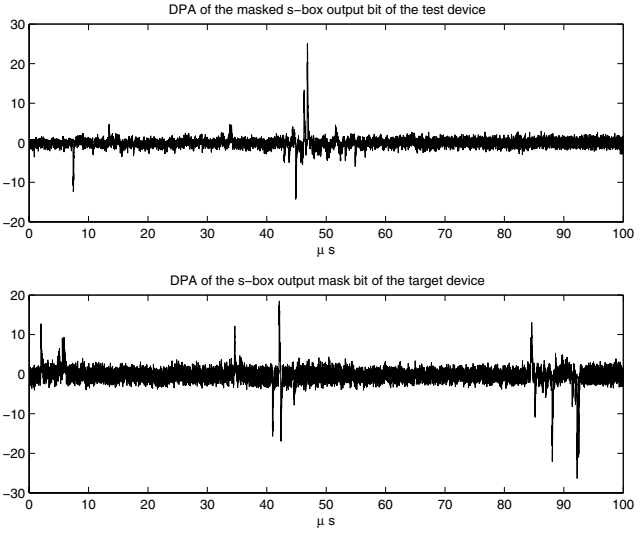

Fig. 3. Smartcard B: DPA of the masked s-box output bit using the test device and DPA of the mask bit using the target device

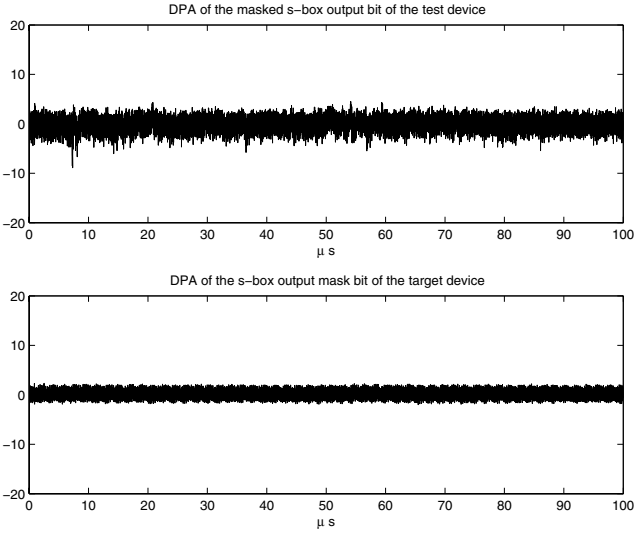

Fig. 4. Smartcard B: DPA of the masked s-box output bit using the test device and DPA of the mask bit using the target device (both with wrong hypothesis)

the RNG bias. The fourth plot is the experimental using 1000 samples from a test card to build templates. The experimental curve is in excellent agreement with our analytical results.

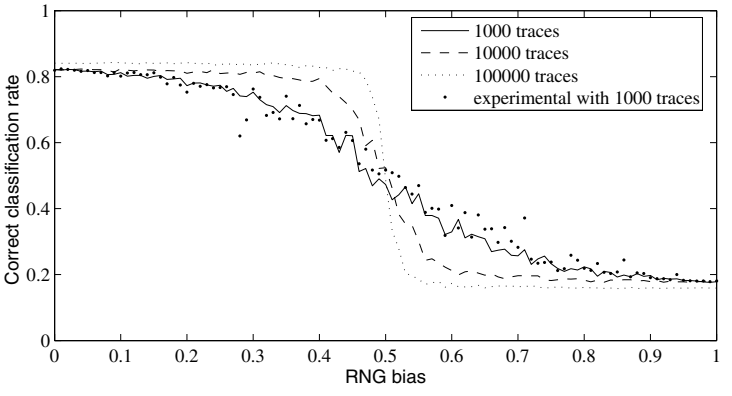

Fig. 5. Probability of correct classification versus RNG bias.

In summary, even with a test card with very small RNG bias, it is possible to mount template-enhanced DPA attacks; the only effect of a small bias is that many more samples are needed to build templates that are as good as template built from a card with completely broken RNG.

Acknowledgments: We would like to thank Helmut Scherzer for providing us with a protected DES implementation with a switchable RNG on smartcard A and Andreas Krügersen for the AES implementation with switchable RNG on smartcard B.

References

[AARR02] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side – Channel(s). In B.S. Kaliski, Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2002*, volume 2535, pages 29–45. Springer-Verlag, 2002.

[AG01] M.-L. Akkar and C. Giraud. An Implementation of DES and AES Secure against Some Attacks. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2001*, volume LNCS 2162, pages 309–318. Springer-Verlag, 2001.

[ARR03] D. Agrawal, J. R. Rao, and P. Rohatgi. Multi-channel Attacks. In C. D. Walter, Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2003*, volume 2779, pages 2–16. Springer-Verlag, 2003.

[BNSQ03] L. Bohy, M. Neve, D. Samyde, and J.-J. Quisquater. Principal and Independent Component Analysis for Crypto-systems with Hardware Unmasked Units. In *e-Smart 2003*, 2003.

[CJR+99] S. Chari, C. S. Jutla, J. R. Rao, , and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *Advances in Cryptology — CRYPTO '99*, volume LNCS 1666, pages 398 – 412. Springer-Verlag, August 1999.

[CRR02] S. Chari, J.R. Rao, and P. Rohatgi. Template Attacks. In B.S. Kaliski, Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2002*, volume LNCS 2523, pages 13–28. Springer-Verlag, 2002.

[GP99] L. Goubin and J. Patarin. DES and Differential Power Analysis: the Duplication Method. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 1999*, volume LNCS 1717, pages 158–172. Springer-Verlag, 1999.

[KJJ99] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis: Leaking Secrets. In *Advances in Cryptology — CRYPTO '99*, volume LNCS 1666, pages 388–397. Springer-Verlag, 1999.

[Koc96] P. Kocher. Timing Attacks on Implementations of Diffie-Hellmann, RSA, DSS, and Other Systems. In *Advances in Cryptology — CRYPTO '96*, volume LNCS 1109, pages 104–113. Springer-Verlag, 1996.

[MDS99] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX Workshop on Smartcard Technology*, pages 151–162, 1999.

[MDS02] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Examining Smart-Card Security under the Threat of Power Analysis Attacks. *IEEE Transactions On Computers*, 51(4):1–12, April 2002.

[RO04] C. Rechberger and E. Oswald. Practical Template Attacks. In *Workshop on Information Security Applications — WISA 2004*, August 2004.

[Tre68] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I.* John Wiley & Sons. New York, 1968.

A Sensitivity of Probability of Success on Bias

Let H_0 and H_1 denote two hypotheses corresponding to the target bit being equal to 0 and 1 respectively. Let p_{H_0} and p_{H_1} model the distribution of captured side-channel emanations under H_0 and H_1, respectively. Assume that p_{H_0} and p_{H_1} are multivariate Gaussian distributions [CRR02, ARR03] with means $\mathbf{m}_0$ and $\mathbf{m}_1$, and variances $\mathbf{\Sigma}_0$ and $\mathbf{\Sigma}_1$, respectively.

Let α be the mixing factor, that is, the samples collected for H_0 are distributed according to the Gaussian mixture distribution $(1-\alpha)p_{H_0} + \alpha p_{H_1}$, and the samples collected for H_1 are distributed according to the Gaussian mixture distribution $(1-\alpha)p_{H_1} + \alpha p_{H_0}$. As a result of mixing, the mean and covariance of samples collected for H_0 is given by

$$\widetilde{\mathbf{m}}_0 = \int \mathbf{s}\Big((1-\alpha)p_{H_0}(\mathbf{s}) + \alpha p_{H_1}(\mathbf{s})\Big)d\mathbf{s} = (1-\alpha)\mathbf{m}_0 + \alpha\mathbf{m}_1 \tag{3}$$

$$\begin{aligned}\widetilde{\mathbf{\Sigma}}_0 &= \int (\mathbf{s}-\widetilde{\mathbf{m}}_0)(\mathbf{s}-\widetilde{\mathbf{m}}_0)'\Big((1-\alpha)p_{H_0}(\mathbf{s}) + \alpha p_{H_1}(\mathbf{s})\Big)d\mathbf{s} \\ &= (1-\alpha)\mathbf{\Sigma}_0 + \alpha\mathbf{\Sigma}_1 + \alpha(1-\alpha)\Delta\mathbf{m}\Delta\mathbf{m}'\end{aligned} \tag{4}$$

where A' denotes the transpose of the matrix A. We note that derivation of (4) requires tedious but straight-forward algebraic manipulations. Similarly, the mean and covariance of samples collected for H_1 is given by

$$\widetilde{\mathbf{m}}_1 = (1-\alpha)\mathbf{m}_1 + \alpha\mathbf{m}_0 \tag{5}$$

$$\widetilde{\mathbf{\Sigma}}_1 = (1-\alpha)\mathbf{\Sigma}_1 + \alpha\mathbf{\Sigma}_0 + \alpha(1-\alpha)\Delta\mathbf{m}\Delta\mathbf{m}' \tag{6}$$

During the hypothesis testing phase, an adversary would use distorted templates based on (3)–(6) to classify the target bit from a captured side-channel emanation $\mathbf{s}$. Specifically, the decision criterion is given by

$$(\mathbf{s}-\widetilde{\mathbf{m}}_0)'\widetilde{\mathbf{\Sigma}}_0^{-1}(\mathbf{s}-\widetilde{\mathbf{m}}_0) - (\mathbf{s}-\widetilde{\mathbf{m}}_1)'\widetilde{\mathbf{\Sigma}}_1^{-1}(\mathbf{s}-\widetilde{\mathbf{m}}_1) > \log(|\widetilde{\mathbf{\Sigma}}_1|) - \log(|\widetilde{\mathbf{\Sigma}}_0|) \tag{7}$$

where a decision is made in favor of H_1 if the above inequality is true, and in favor of H_0 otherwise.

By assuming $\mathbf{\Sigma}_0 = \mathbf{\Sigma}_1 = \mathbf{\Sigma}$[5], (7) can be reduced to the following [Tre68]

$$(\widetilde{\mathbf{m}}_1 - \widetilde{\mathbf{m}}_0)'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{s} > \frac{1}{2}\Big(\widetilde{\mathbf{m}}_1'\widetilde{\mathbf{\Sigma}}^{-1}\widetilde{\mathbf{m}}_1 - \widetilde{\mathbf{m}}_0'\widetilde{\mathbf{\Sigma}}^{-1}\widetilde{\mathbf{m}}_0\Big) \tag{8}$$

By using (3) and (5) along with the symmetry of inverses of covariance matrices to cancel common terms, we can further simplify (8) to

$$\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{s} > \frac{1}{2}\Big(\mathbf{m}_1'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_1 - \mathbf{m}_0'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_0\Big) \tag{9}$$

Note that $\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{s}$ is a linear combination of Gaussian variables. As a result, under the hypothesis H_0, $\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{s}$ is Gaussian distributed with the following mean and variance

$$E[\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{s}] = \Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_0 \tag{10}$$

$$V[\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{s}] = \Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m} \tag{11}$$

Let $Q(x), x \geqslant 0$ denote the probability of a Gaussian random variable with mean 0 and variance 1 being larger than x. Under the hypothesis H_0 (and by symmetry, under the hypothesis H_1), the probability of error incurred by using the distorted templates is given by

$$P(\text{error}) = Q\Big(\frac{|\frac{1}{2}\big(\mathbf{m}_1'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_1 - \mathbf{m}_0'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_0\big) - \Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_0|}{\sqrt{\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m}}}\Big) \tag{12}$$

We can express the numerator of $Q(\cdot)$ in the above equation solely in terms of $\Delta\mathbf{m}$ by realizing that $\mathbf{m}_1'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_0$ is one dimensional and therefore it equals to its transpose $\mathbf{m}_0'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_1$.

$$\begin{aligned}
&\frac{1}{2}\big(\mathbf{m}_1'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_1 - \mathbf{m}_0'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_0\big) - \Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_0 \\
&\qquad = \frac{1}{2}\mathbf{m}_1'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_1 + \frac{1}{2}\mathbf{m}_0'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_0 - \frac{1}{2}\mathbf{m}_1'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_0 - \frac{1}{2}\mathbf{m}_0'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{m}_1 \\
&\qquad = \frac{1}{2}\mathbf{m}_1'\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m} - \frac{1}{2}\mathbf{m}_0'\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m} \\
&\qquad = \frac{1}{2}\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m}
\end{aligned}$$

[5] In our experiments, this assumption holds well.

Thus, probability of error can be expressed as

$$P(\text{error}) = Q\Big(\frac{\frac{1}{2}|\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m}|}{\sqrt{\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m}}}\Big) \tag{13}$$

Our task is to prove that the argument of $Q(\cdot)$ in the above equation is independent of α, and therefore, the probability of error in hypothesis testing phase is independent of the RNG bias. Our strategy is to factorize the numerator and denominator of the argument of $Q(\cdot)$ in (13), and show that factors involving α cancel each other out. The first step towards this factorization is to obtain an expression for $\widetilde{\mathbf{\Sigma}}^{-1}$ in terms of $\mathbf{\Sigma}^{-1}$ by using the matrix inversion lemma. The matrix inversion lemma states that for arbitrary matrices A, U, C, and V, with the only restriction that inverses of A and C exist and the product UCV and the sum $A + UCV$ are well-defined, the following holds true

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1} \tag{14}$$

Substituting $A = \mathbf{\Sigma}, U = \alpha(1-\alpha)\Delta\mathbf{m}, C = 1$, and $V = \Delta\mathbf{m}'$, we obtain

$$\begin{aligned}\widetilde{\mathbf{\Sigma}}^{-1} &= (\mathbf{\Sigma} + \alpha(1-\alpha)\Delta\mathbf{m}\cdot 1\cdot\Delta\mathbf{m}')^{-1} \\ &= \mathbf{\Sigma}^{-1} - \alpha(1-\alpha)\mathbf{\Sigma}^{-1}\Delta\mathbf{m}\Big(1 + \alpha(1-\alpha)\Delta\mathbf{m}'\mathbf{\Sigma}^{-1}\Delta\mathbf{m})\Big)\Delta\mathbf{m}'\mathbf{\Sigma}^{-1}\end{aligned}$$

Let $\beta = \Delta\mathbf{m}'\mathbf{\Sigma}^{-1}\Delta\mathbf{m}$. Since β is a one dimensional quantity, it can be factored out to obtain

$$\widetilde{\mathbf{\Sigma}}^{-1} = \mathbf{\Sigma}^{-1} - \alpha(1-\alpha)(1 + \alpha(1-\alpha)\beta)\mathbf{\Sigma}^{-1}\Delta\mathbf{m}\Delta\mathbf{m}'\mathbf{\Sigma}^{-1} \tag{15}$$

Now we are ready to factor the numerator.

$$\begin{aligned}\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m} &= \Delta\mathbf{m}'\mathbf{\Sigma}^{-1}\Delta\mathbf{m} - \alpha(1-\alpha)(1+\alpha(1-\alpha)\beta)\Delta\mathbf{m}'\mathbf{\Sigma}^{-1}\Delta\mathbf{m}\Delta\mathbf{m}'\mathbf{\Sigma}^{-1}\Delta\mathbf{m} \\ &= \beta(1 - \alpha\beta(1-\alpha)(1+\alpha(1-\alpha)\beta))\end{aligned} \tag{16}$$

Similarly, to factorize the denominator, we perform the following steps.

$$\begin{aligned}&\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m} \\ &\quad= \Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\Big(I - \alpha(1-\alpha)(1+\alpha(1-\alpha)\beta)\Delta\mathbf{m}\Delta\mathbf{m}'\mathbf{\Sigma}^{-1}\Big)\Delta\mathbf{m} \\ &\quad= (\Delta\mathbf{m}'\widetilde{\mathbf{\Sigma}}^{-1}\Delta\mathbf{m})\Big(1 - \alpha\beta(1-\alpha)\big(1+\alpha(1-\alpha)\beta\big)\Big) \\ &\quad= \beta\Big(1 - \alpha\beta(1-\alpha)\big(1+\alpha(1-\alpha)\beta\big)\Big)^2\end{aligned} \tag{17}$$

Using (16) and (17), the numerator and denominator of (13) can be simplified to give the following expression for probability of error

$$P(\text{error}) = Q\Big(\frac{1}{2}\sqrt{\beta}\Big) \tag{18}$$

Note that since $\mathbf{\Sigma}^{-1}$ is a positive definite matrix, $\beta > 0$. Furthermore, β only depends on the statistics of emanations under H_0 and H_1. In particular, it does not depend on α.

B Impact of Classification Error on SNR

Let ν be the probability of correct classification in a DPA attack for a bit X. If $\nu \neq 1$, then the erroneous classification of the bit X can be interpreted as an additional noise source in the differential trace. If the bit X leaks at times t^* and δ denotes the average difference in amplitude of two l-bit wide operands separated by the Hamming distance one, the expected values of the zero-bit and one-bit partition are[6]:

$$E[p_i(t^*)|X = 0] = a + \frac{l-1}{2} \cdot \delta + (1 - \nu) \cdot \delta \tag{19}$$

$$E[p_i(t^*)|X = 1] = a + \frac{l-1}{2} \cdot \delta + \nu \cdot \delta \tag{20}$$

where a denotes some operand independent voltage offset and the term $\frac{l-1}{2} \cdot \delta$ denotes the average algorithmic noise caused by the remaining $l - 1$ bits of the operand. The differential trace $\Delta(t^*)$ can then be given as

$$\Delta(t = t^*) = E[p_i(t^*)|X = 1] - E[p_i(t^*)|X = 0] = (2 \cdot \nu - 1) \cdot \delta \tag{21}$$

whereas $\Delta(t \neq t^*)$ approximates zero. A possible expression of the SNR of a differential trace in DPA attacks was given by Messerges et al. in [MDS99]. We enhance their SNR description with the additional noise factor $(2 \cdot \nu - 1)$ caused by the misclassification, which yields

$$SNR = \frac{(2 \cdot \nu - 1) \cdot \delta \cdot \sqrt{N}}{\sqrt{8 \cdot \sigma^2 + \delta^2 \cdot (\alpha \cdot l + l - 1)}} \tag{22}$$

where N denotes the number of measured side channel traces, α denotes the percentage of algorithmic noise[7] at times $t \neq t^*$ and σ^2 denotes the variance of non-algorithmic time-invariant noise contained in a single trace. Let us assume that in case of perfect classification at $\nu = 1$, an adversary would have to measure $N = 100$ traces to obtain a differential trace with an acceptable SNR. From the above formula, it follows that for an arbitrary error ν the adversary will need $(\frac{10}{2\nu-1})^2$ samples to obtain the same SNR. Table 2 provides the number of traces needed for different values of $\epsilon = 1 - \nu$, using this formula.

[6] For simplicity we assume that the power signal is linear proportional to the Hamming weight of the leaked operand x, i.e. $p_i(t) = a + \delta \cdot HW(x)$.

[7] according to [MDS02] α can be often neglected.