Empirical Evaluation of Multi-Device Profiling Side-Channel Attacks

Neil Hanley and Maire O'Neill
Center for Secure Information Technologies,
ECIT, Queen's University Belfast,
Belfast, BT3 9DT, UK.
Email: {n.hanley, maire.oneill}@qub.ac.uk

Michael Tunstall†
Cryptography Research, Inc.,
425 Market Street, 11th Floor,
San Francisco, CA 94105, USA.
Email: michael.tunstall@cryptography.com

William P. Marnane
Dept. Electrical & Electronic Eng.,
University College Cork,
Cork, Ireland.
Email: l.marnane@ucc.ie

***Abstract*—Side-channel analysis of cryptographic systems can allow for the recovery of secret information by an adversary even where the underlying algorithms have been shown to be provably secure. This is achieved by exploiting the unintentional leakages inherent in the underlying implementation of the algorithm in software or hardware. Within this field of research, a class of attacks known as *profiling attacks*, or more specifically as used here *template attacks*, have been shown to be extremely efficient at extracting secret keys. Template attacks assume a strong adversarial model, in that an attacker has an identical device with which to *profile* the power consumption of various operations. This can then be used to efficiently attack the target device. Inherent in this assumption is that the power consumption across the devices under test is somewhat similar. This central tenet of the attack is largely unexplored in the literature with the research community generally performing the profiling stage on the same device as being attacked. This is beneficial for evaluation or penetration testing as it is essentially the *best case scenario* for an attacker where the model built during the profiling stage matches exactly that of the target device, however it is not necessarily a reflection on how the attack will work in reality. In this work, a large scale evaluation of this assumption is performed, comparing the key recovery performance across 20 identical smart-cards when performing a profiling attack.**

I. Introduction

Traditionally, attacks on cryptographic primitives have focused on analysing inputs and outputs of systems, however the introduction of timing [1] and power [2] attacks showed that the implementation of an algorithm must also be taken into account, especially in the context of embedded security where an attacker might have direct access to a device. This was followed up with further research showing that electromagnetic analysis (EMA) could also recover secret key information [3], [4].

Power analysis attacks work on the premise that the power consumption of a device while it is processing some data is dependent on that data. In an non-profiled scenario, an adversary seeks to use some leakage model $\mathcal{L}$ to estimate the power consumption $\hat{x}$ for some intermediate value that is a function $\mathcal{F}$ of some known input p and secret s, *i.e.* $\hat{x} = \mathcal{L}(\mathcal{F}(p, s))$. As the secret s is unknown, the hypothetical leakage of each element $s^* \in \mathcal{S}$ is calculated, with some statistical distinguisher used to compare $\hat{x}$ with the actual power consumption x to determine the most likely key $\hat{s}$. Commonly used distinguisher's include the difference of means [2], Pearson's linear correlation coefficient [5] and mutual information analysis [6]. While any arbitrary function can be used for $\mathcal{L}$, it is generally based on some engineering intuition of the device under attack. Two models which have been shown to perform well for a wide range of devices are the Hamming weight, and Hamming distance models [7], which are commonly used when attacking software and hardware devices respectively.

The field of side-channel attacks (SCAs) is not purely of academic interest, and there have been multiple examples of attacks on real-world devices such as the KeeLoq remote entry system [8], the bit-stream encryption in Xilinx FPGAs [9] and Mifare DESFire contactless payment cards [10], to name but a few. Hence many embedded cryptographic devices now ship with countermeasures against SCA such as the randomisation of intermediate values through methods such masking [11], [12], the use of dummy operations [13], or through hiding the data dependent power consumption with the use of secure logic styles [14]. Countermeasures come at a cost however, with increased execution time, power consumption, and area (memory or silicon) requirements, depending on the chosen countermeasures and target platform.

The paper is organised as follows, in Section II an overview of profiling attacks is given, with a particular emphasis on template attacks (TAs) in Section II-A as utilised in this work. In Section III the experimental analysis is performed, with separate subsections on the target algorithm in Section III-A, the experimental setup in Section III-B, and the trace preprocessing steps performed in Section III-C. Finally conclusions are drawn and future work suggested in Section IV.

II. Profiling Attacks

The concept of a profiling SCA was originally introduced by Fahn and Pearson in [15], where they proposed inferential power analysis (IPA) to make a detailed model of the power consumption of a device prior to an attack. TAs or quadratic discriminant analysis (QDA), subsequently introduced by Chari *etal.* [16], and its variants, are among the most popular and effective methods to perform a profiling attack. However

† Work undertaken while the author was employed at the University of Bristol.

many machine learning based algorithms can be used, and recent research has looked to exploit the large body of work from the statistical learning community. For example, support vector machines (SVMs) [17], [18], random forests (RFs) [18], or Stochastic methods (which are linear regression based) [19] are all viable alternatives to TAs. However given an unbounded training phase, *i.e.* an unrestricted number of training samples, then TAs can be viewed as optimal in an information theoretic sense [16] for devices where the distribution of noise on the power traces is Gaussian.

An advantage of profiling attacks is that they allow for secret key recovery with few or only a single power trace, allowing the circumvention of many re-keying countermeasures designed to restrict the number of traces an adversary can acquire for a given key. This comes with the trade-off of the stronger attacker model compared to non-profiling attacks, which generally require a much larger number of traces for key recovery, in that an identical or similar device is available to the adversary to model the power consumption prior to the attack. How much control or knowledge of the key they are assumed to have over said device is open to interpretation, hence this assumption is not as restricting it may first seem. For example, in [10] a non-profiling attack was first used to recover a key prior to subsequently using the broken device to build templates. In [20], it was shown how a device with a faulty random number generator (RNG) suffices to build templates, while in [21] it was shown how two devices with different unknown keys could be used. It was also suggested in [22] that public verification functions could be used to build templates using the device under attack itself. These are outside the scope of this paper however, and here we assume that the adversary has full control of the profiling device(s).

One of the first detailed studies which looked at the effect of building templates on a different device to that being attacked was provided in [23]. Here the authors studied power variability issues when dealing with nano-scale devices. They introduced the concept of perceived information (PI) to quantify the difference between the modelled and actual leakage from a device. However they select three features for their analysis based on a heuristic examination of traces with known inputs, hence any error due to choosing the points of interest in the target device based on an analysis of the training device is not accounted for. They also suggested the use of $d > 1$ distinct devices when profiling, to attack device $d + 1$. The work in [24], while using the same device, looks at the effect of building templates when the acquisitions are separated in time (by a period of 4 years), and when the supply voltage is reduced. As is the case here, this work uses attack metrics rather than the information theoretic metrics as used in [23]. Multiple PIC devices are used in [25] where the authors perform a EMA based TA. However, their analysis requires multiple attack traces for key recovery in an *amplified* TA in order to separately normalise the testing data. In [26], three different microprocessors with different architectures and fabrication processes are examined, with three separate devices for each micro-controller. Our work is most comparable to this study, as they also examine a *real attack* context in that the synchronisation of the traces and location of points of interest cannot be assumed, but they do not extend the building of templates with $d > 1$ devices.

A. Template Attacks

A TA is a two stage attack, the first stage consisting of a supervised machine learning problem where the trace data acquired from the identical device with known labels (where the label corresponds to some intermediate value or leakage model) known as the training data, is used to build a model of the power consumption. The second, attack, stage involves estimating the most likely key from the target trace based on what template best fits it. Note that while the profiling stage can be time-consuming in order to achieve optimal key recovery, the same set of templates can subsequently be used to attack many devices. Generally, the secret key is divided into smaller, more manageable "chunks" which are then attacked independently to recover the entire key.

1) Training Stage: The first stage of a TA is the training or profiling stage. A set of m power traces x, of length n are collected with their corresponding plaintext p and key s inputs. The target key space is given by $\mathcal{S}$ and contains $|\mathcal{S}|$ elements. The traces are assigned to a class $y \in \mathcal{K}$ such that $y = \mathcal{F}(p, s)$. The function $\mathcal{F}$ is chosen such that it maps y to a secret s given p. This does not necessarily have to be a unique mapping (*i.e.* it could include the leakage power model $\mathcal{L}$), however unless it is bijective the classification stage will require more than one target trace to recover the secret. The unique values in the set $\mathcal{K}$ are denoted by, $o^{(1)}, o^{(2)}, \ldots, o^{(|\mathcal{K}|)}$. If the noise on the traces is additive and follows a Gaussian distribution, the traces can be assumed to be drawn from the multivariate normal distribution as given in Equation 1.

$$\mathcal{N}\left(x \mid \mu^{(i)}, \Sigma^{(i)}\right) = \left((2\pi)^n \left|\Sigma^{(i)}\right|\right)^{-\frac{1}{2}} \times e^{-\frac{1}{2}\left(x-\mu^{(i)}\right)\left(\Sigma^{(i)}\right)^{-1}\left(x-\mu^{(i)}\right)^{\top}} \quad (1)$$

Where $\mu^{(i)}$ and $\Sigma^{(i)}$ represent the mean vector and noise covariance matrix of the class $o^{(i)}$, and $^{\top}$ represents the transpose operation. The training stage then consists of empirically estimating the mean vector $\hat{\mu}^{(i)}$ and noise covariance matrix $\hat{\Sigma}^{(i)}$ pair, for each instance of $o^{(i)}$, as defined in Equation 2 and Equation 3. Here $i \in \{1, \ldots, |\mathcal{K}|\}$, and $x^{(j,i)}$ represents the j^{th} acquisition of the class $o^{(i)}$, where $j \in \{1, \ldots, m^{(i)}\}$ and $m^{(i)}$ is the number of traces available for $o^{(i)}$ such that $\sum_{i=1}^{|\mathcal{K}|} m^{(i)} = m$.

$$\hat{\mu}^{(i)} = \frac{1}{m^{(i)}} \sum_{j=1}^{m^{(i)}} x^{(j,i)} \quad (2)$$

$$\hat{\Sigma}^{(i)} = \frac{1}{m^{(i)}} \sum_{j=1}^{m^{(i)}} \left(x^{(j,i)} - \hat{\mu}^{(i)}\right)\left(x^{(j,i)} - \hat{\mu}^{(i)}\right)^{\top} \quad (3)$$

This estimated mean vector and noise covariance matrix pair $\left(\hat{\mu}^{(i)}, \hat{\Sigma}^{(i)}\right)$ is then the template associated with $o^{(i)}$ and completely specifies its noise distribution.

2) Testing Stage: To recover key information a test trace is required from the device under attack, preferably recorded under the same conditions. The trace must first be reduced in size and processed using the same steps that were used when generating the templates. For each possible class $o^{(i)} \in \mathcal{K}$, the likelihood of the trace corresponding to it is calculated using the multivariate Gaussian distribution from Equation 1, and plugging in the estimated values of $\left(\hat{\mu}^{(i)}, \hat{\Sigma}^{(i)}\right)$. The likelihood of $o^{(i)}$ can then be converted to a probability by applying Bayes' theorem as given in Equation 4.

$$\Pr\left(o^{(i)} \mid x\right) = \frac{p\left(x \mid o^{(i)}\right) \Pr\left(o^{(i)}\right)}{\sum_{j=1}^{|\mathcal{K}|} p\left(x \mid o^{(j)}\right) \Pr\left(o^{(j)}\right)} \tag{4}$$

Here $\Pr\left(o^{(i)}\right)$ is the prior probability of the class occurring, and $p\left(x \mid o^{(i)}\right)$ is given by $\mathcal{N}\left(x \mid \mu^{(i)}, \Sigma^{(i)}\right)$. Applying the maximum likelihood principal, the key guess is then given by Equation 5. If all operations are equiprobable then the application of Bayes' theorem is unnecessary as it simply scales the likelihood values, and Equation 5 can be applied directly.

$$\hat{s} = \mathcal{F}\left(\arg\max_{o} \Pr\left(o^{(i)} \mid x\right), p\right)^{-1} \tag{5}$$

The success of the attack is increased if a set of power traces for a constant secret key is available such that $m > 1$ for the attack traces, allowing an *amplified* TA. In this scenario, Bayes' theorem can be applied iteratively if the power traces are statistically independent thereby increasing the power of the attack as given in Equation 6 [27]. Note this is equivalent to Equation 4 when $m = 1$. Once again the maximum likelihood principal can be used to return the estimated key $\hat{s}$.

$$\Pr\left(o^{(i)} \mid x\right) = \frac{\left(\prod_{k=1}^{m} p\left(x^{(k)} \mid o^{(i)}\right)\right) \cdot \Pr\left(o^{(i)}\right)}{\sum_{j=1}^{|\mathcal{K}|} \left(\left(\prod_{k=1}^{m} p\left(x^{(k)} \mid o^{(j)}\right)\right) \cdot \Pr\left(o^{(j)}\right)\right)} \tag{6}$$

3) Linear Discriminant Analysis: Note that the attack as described is equivalent to the application of QDA as described in statisical learning literature such as [28]. The accurate estimation of $\hat{\Sigma}^{(i)}$ in Equation 3 can require a large number of training traces for each class $y^{(i)}$. It has been suggested that *reduced templates* can be used, where the features are assumed independent and only variances are considered which is equivalent to Naïve Bayes learning, or that the covariance matrix is replaced by an identity matrix which can be viewed as a Euclidean distance classifier [7]. This no longer makes full use of the leakage however, and poorer classification performance can be expected.

Another alternative is the use of a pooled covariance matrix or linear discriminant analysis (LDA). The advantages of this method with regards to the number of traces required for estimation were outlined in [29]. The noise covariance matrix $\hat{\Sigma}^{(i)}$ for each class $y^{(i)}$, is now replaced with a single $\hat{\Sigma}$ for all $o^{(i)}$ as given in Equation 7, with each template now defined by $\left(\hat{\mu}^{(i)}, \hat{\Sigma}\right)$.

$$\hat{\Sigma} = \frac{1}{m} \sum_{i=1}^{|\mathcal{K}|} \sum_{j=1}^{m^{(i)}} \left(x^{(j,i)} - \hat{\mu}^{(i)}\right) \left(x^{(j,i)} - \hat{\mu}^{(i)}\right)^{\top} \tag{7}$$

Intuitively, the use of a pooled covariance matrix to model the noise, fits with the underlying assumption that the noise of each trace follows a zero-mean Gaussian distribution. Hence after the empirical mean is removed to calculate the noise vector for a given trace, there is no reason to expect it would be any different from a noise vector for a different class. Hence, for the experiments that follow, references to the building of templates refers to LDA rather than QDA.

III. Experimental Analysis

As previously mentioned, attack metrics rather than the information theoretic metrics of [23] are used here. The aim of the study is to examine the feasibility of profiling on one (or more) device, when performing the attack on a different device. The target algorithm is the widely used Advanced Encryption Standard (AES), and all templates are built to allow recovery of a key byte with only a single attack trace. Hence all results are given as the expected error rate when averaged over a large number of independent testing traces. The choice of AES is due to its widespread use in practice, however the experiment could equally have been performed on any other algorithm.

A. Advanced Encryption Standard

In 2001, the block cipher Rijndael by Joan Daemen and Vincent Rijmen, was selected via a public competition by the National Institute of Standards and Technology (NIST) to become the AES [30] as a replacement for the outdated Data Encryption Standard (DES) algorithm. It is an substitution-permutation network (SPN) based iterative block cipher which acts on plaintext blocks of 128-bits and supports significantly larger key sizes than DES, *i.e.* 128-bits, 192-bits or 256-bits. Depending on the key size, the number of rounds is either 10, 12 or 14 respectively. For the work here, only the 128-bit key size is examined, however the attack is directly applicable to larger key sizes.

Algorithm III-A outlines a high-level description of the AES algorithm. First, the plaintext block p is copied into the state variable, which is a 4×4 matrix of bytes. Then, an initial *AddRoundKey* function simply XORs the initial key to the state. This is followed by nine identical round transformations consisting of the functions; *S-Box*, *ShiftRows*, *MixColumn*, and *AddRoundKey*. The tenth round skips the *MixColumn* operation to generate the ciphertext.

It has been shown that the non-linear *S-Box* operation in AES provides a suitable target when performing SCA [31] and this is the target value used here also, such that $y = \mathcal{F}\left(p \otimes s\right)$. As this is a bijective function, recovering y is equivalent to

```
1: procedure AES128( p , s )
2:     r ← p
3:     r ← AddRoundKey( r , s )
4:     for i in 1 to 10 do
5:         r ← S-Box( r )
6:         r ← ShiftRows( r )
7:         if i ≠ 10 then
8:             r ← MixColumn( r )
9:         end if
10:        r ← AddRoundKey( r , s_i )
11:    end for
12:    return r
13: end procedure
```

Fig. 1. Advanced Encryption Standard.

recovering the secret s hence all error rates are given for recovering y. As the aim of the work is to compare the error rates for secret key recovery when building templates on different devices using only a single attack trace, the class is assigned directly according to the intermediate value and no leakage model is used. When performing SCAs on AES, typically one would attack each byte individually hence a total of 16 attacks is required to recover the entire key (note the same set of traces can be for all 16 attacks). For the profiling attack under consideration, this gives a $|\mathcal{K}| = 256$ multi-class learning problem for each byte. In the experiments that follow, only the first byte of the output of the first round *S-Box* function is attacked, rather than the state as a whole.

B. Experimental Setup

To perform the analysis, 20 low-cost PIC smart-cards were used. These are low-power devices which should perform favourably in the experiments compared to ARM or AVR based microprocessors, or dedicated hardware platforms such as application-specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs). They were programmed to perform the initial *AddRoundKey* and *S-Box* operations for a single plaintext and key byte, with the same code used for all smart-cards. The smart-cards were driven at a clock frequency of 4 MHz, and the power traces were acquired using LeCroy WaveRunner 104Xi oscilloscope with a LeCroy AP034 differential probe measuring across a $10\,\Omega$ resistor placed in series with the V_{dd} supply pin of the smart-card. The sampling rate of the oscilloscope was set to $250\,\text{MSs}^{-1}$, and the internal analog bandwidth limiter of 25 MHz was used to reduce noise on the traces. 10 k power traces were recorded for *each* smart-card, with uniformly random plaintext and key bytes selected for each trace. No suitable trigger point for the oscilloscope was available to ensure the power traces were aligned, hence the communications line was used as a trigger leading to desynchronised signals.

C. Trace Pre-Processing

Before performing machine learning analysis on the power traces, a number of pre-processing steps must first be performed. The DC component of each trace is first removed by subtracting the mean of that trace. The traces are then filtered using a low-pass finite impulse response (FIR) filter with a 50-point Blackman window and a cut-off frequency of 6 MHz. Next, each of the 20 sets is *individually* aligned using cross-correlation. The mean of each set is then taken and the Euclidean distance between the means used to align the sets with other. It has been shown that the number of points n in a trace can be reduced to a single point per clock cycle without adversely affecting SCA [7]. As n is in the region of $\sim 25\,\text{k}$ for the $20 \times 10\,\text{k}$ traces under consideration, to reduce the computational requirements of the analysis the traces are reduced to just the maximum point per clock cycle. This reduces the length of each trace such that $n \approx 400$.

After compression of the power traces, there will still be many points that are unrelated to the processing of the target intermediate value hence some sort of feature selection is required. The are many proposed methods such as difference of means [16], Pearson's correlation, or transformations such as principal component analysis (PCA) or Fisher's linear discriminant [32]. In this work an analysis of variance method called normalised inter-class variance (NICV) is used as proposed in [33]. This selects the points of interest according to the ratio of the explained variance and the total variance as given in Equation 8. When selecting $n' < n$ features, the points which return the highest NICV values are selected.

$$\text{NICV} = \frac{\text{Var}\left[\mathbb{E}\left(x \mid y\right)\right]}{\text{Var}\left[x\right]} \tag{8}$$

D. Multi-Device Attacks

As an initial test, the feature containing the largest "leakage" is first calculated using the NICV value generated across the entire power trace set by treating all 20 devices as a single set. The box plot of each of the individual trace sets at that point in time is then shown in Figure 2. It can be seen that although the overall expected mean is $\approx 15.5\,\text{mV}$, there is a significant deviation between the sets in both the mean and distribution of the traces at that point, despite the relatively simple architecture of the PIC devices under consideration.

Experimental analysis using cross-validation on the individual data sets determined that the selection of 40 features allowed for the highest accuracy classification without encountering numerical difficulties in any of the sets. For each of the sets $1 \rightarrow 20$, $m = 9\,\text{k}$ between the sets traces were used to build templates for the *S-Box* output giving $\frac{9k}{256} \approx 35$ traces to estimate the template means $\hat{\mu}^{(i)}$, but all 9 k to estimate the pooled covariance matrix $\hat{\Sigma}$. These templates were then used to classify the remaining 1 k traces from that device, as well as 1 k traces from each of the other 19 devices. The split of the sets into 9 k training and 1 k testing traces was randomly selected each time. Normalisation of the sets was applied by taking the *z-scores* of the data as suggested in [25], however the method of applying it is performed differently. In [25] it is assumed a number of attack traces are available for key recovery hence the normalisation can be performed separately

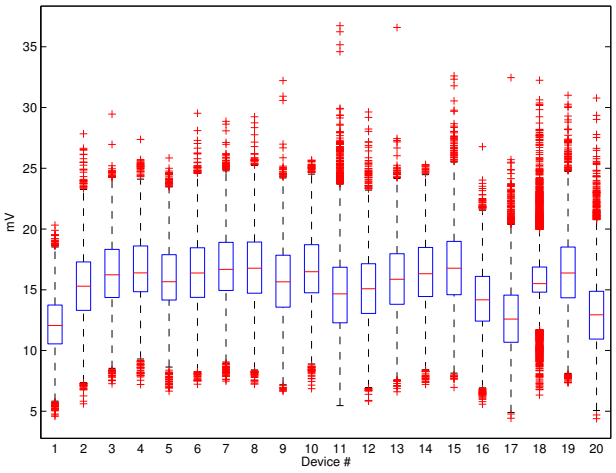

Fig. 2. Comparison of a single point in time across devices.

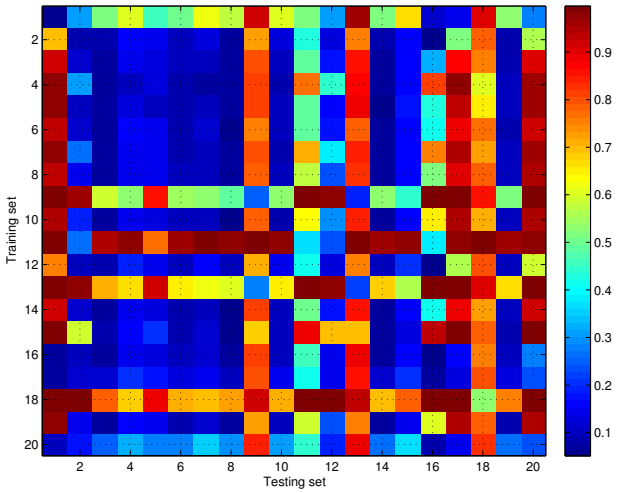

Fig. 3. Comparison of a single point in time across devices.

on the training and test traces. As we look to recover the key from a single trace, we cannot presume to separately estimate the mean and standard deviation of the test data. Hence the estimated parameters from the training set are used to normalise the testing sets each time. A similar principal applies for feature selection, once the index of the points of interest are calculated from a given training set, these are then used to select to points of interest from all the other testing sets as would be the case in a real-world scenario.

The error rate for each set, while using the templates generated from every other set is given in Figure 3. The top left to bottom right diagonal gives the error when the same device is used for both training and testing. This can be viewed as the baseline "best case" scenario for an adversary for this particular setup. It is clear from the image that classification is not equivalent between devices. For example, classifying devices $\{1, 9, 11, 13, 17, 18, 20\}$ generally returns a higher error rate regardless of what device is used (apart from the same device) to generate the templates, as can be seen by the redder colouring. On the contrary however, devices $\{2 - 8\}$ mostly return a low error rate regardless of the training device used as indicated by the blue.

A more general way to generate the templates is the use traces from many devices [23]. Figure 4 shows error rates where $m = 9\,\text{k}$ randomly selected traces from $d = 19$ devices are used to generate the templates, and used to classify $1\,\text{k}$ traces from the remaining device. Only $9\,\text{k}$ traces in total are randomly selected from the $19 \times 10\,\text{k}$ available in order to perform a fair comparison with the previous results by keeping the size of the training set m constant. For reference, the average error rate of generating the templates with difference devices, and the error rate of generating the templates with the same device are also given.

It can be seen in Figure 4, that in general, using the traces randomly selected from a number of devices gives considerably better classification than when only a single device is used. Of the 20 sets, only devices $\{9, 13, 18\}$ could

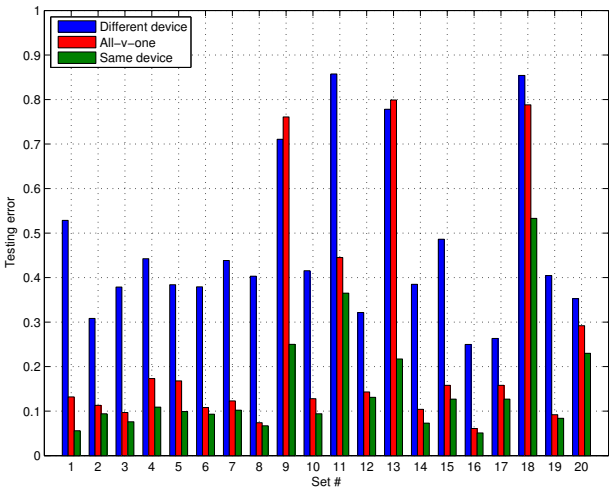

Fig. 4. Effect of building templates on multiple devices.

be viewed as performing poorly, while the majority of devices have error rates comparable to when the same device is used to build the templates.

IV. Conclusion

In this work an empirical analysis of one of the fundamental assumptions of a TA has been performed, namely that it is feasible to profile the power consumption on one device when attacking a different one. It has been shown that while an attack is still possible using only a single attack trace, the error rate does significantly increase when different devices are used, even on the relatively simple PIC devices used here, hence multiple devices are desirable for profiling. It must be noted that SCAs are by their nature implementation specific therefore, while the work here confirms the viability of TA from an adversarial viewpoint, the success or otherwise for different attack platforms cannot be inferred from these

results. Likewise, the optimal number of devices to use to build templates will be dependent on the underlying distribution of noise on the target platform. Further research into the real-world feasibility of TA on more advanced platforms, such as dedicated hardware circuits is required. Taking FPGAs for example, it would be interesting to examine what the effect of regenerating the circuit has on a TA, due to the non-deterministic nature of the synthesis tools leading to a slightly different circuit layout each time it is re-run.

REFERENCES

[1] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems," in *Advances in Cryptology — CRYPTO 1996*, ser. Lecture Notes in Computer Science, N. Koblitz, Ed., vol. 1109. Springer-Verlag, 1996, pp. 104–113.

[2] P. C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology — CRYPTO 1999*, ser. Lecture Notes in Computer Science, M. J. Wiener, Ed., vol. 1666. Springer-Verlag, 1999, pp. 388–397.

[3] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," in *Cryptographic Hardware and Embedded Systems — CHES 2001*, ser. Lecture Notes in Computer Science, Çetin Kaya Koç, D. Naccache, and C. Paar, Eds., vol. 2162. Springer-Verlag, 2001, pp. 251–261.

[4] J.-J. Quisquater and D. Samyde, "ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards," in *Smart Card Programming and Security — E-Smart 2001*, ser. Lecture Notes in Computer Science, I. Attali and T. P. Jensen, Eds., vol. 2140. Springer-Verlag, 2001, pp. 200–210.

[5] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *Cryptographic Hardware and Embedded Systems — CHES 2004*, ser. Lecture Notes in Computer Science, M. Joye and J.-J. Quisquater, Eds., vol. 3156. Springer-Verlag, 2004, pp. 16–29.

[6] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual Information Analysis," in *Cryptographic Hardware and Embedded Systems — CHES 2008*, ser. Lecture Notes in Computer Science, E. Oswald and P. Rohatgi, Eds., vol. 5154. Springer-Verlag, 2008, pp. 426–442.

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

[8] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani, "On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoqCode Hopping Scheme," in *Advances in Cryptology — CRYPTO 2008*, ser. Lecture Notes in Computer Science, D. Wagner, Ed., vol. 5157. Springer-Verlag, 2008, pp. 203–220.

[9] A. Moradi, M. Kasper, and C. Paar, "Black-Box Side-Channel Attacks Highlight the Importance of Countermeasures - An Analysis of the Xilinx Virtex-4 and Virtex-5 Bitstream Encryption Mechanism," in *Topics in Cryptology — CT-RSA 2012*, ser. Lecture Notes in Computer Science, O. Dunkelman, Ed., vol. 7178. Springer-Verlag, 2012, pp. 1–18.

[10] D. Oswald and C. Paar, "Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World," in *Cryptographic Hardware and Embedded Systems — CHES 2011*, ser. Lecture Notes in Computer Science, B. Preneel and T. Takagi, Eds., vol. 6917. Springer-Verlag, 2011, pp. 207–222.

[11] L. Goubin and J. Patarin, "DES and Differential Power Analysis (The "Duplication" Method)," in *Cryptographic Hardware and Embedded Systems — CHES 1999*, ser. Lecture Notes in Computer Science, Çetin Kaya Koç and C. Paar, Eds., vol. 1717. Springer-Verlag, 1999, pp. 158–172.

[12] M.-L. Akkar and C. Giraud, "An Implementation of DES and AES, Secure against Some Attacks," in *Cryptographic Hardware and Embedded Systems — CHES 2001*, ser. Lecture Notes in Computer Science, Çetin Kaya Koç, D. Naccache, and C. Paar, Eds., vol. 2162. Springer-Verlag, 2001, pp. 309–318.

[13] C. Clavier, J.-S. Coron, and N. Dabbous, "Differential Power Analysis in the Presence of Hardware Countermeasures," in *Cryptographic Hardware and Embedded Systems — CHES 2000*, ser. Lecture Notes in Computer Science, C. Paar and Çetin Kaya Koç, Eds., vol. 1965. Springer-Verlag, 2000, pp. 252–263.

[14] K. Tiri and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation," in *Design, Automation and Test in Europe Conference and Exposition — DATE 2004*. IEEE Computer Society, 2004, pp. 246–251.

[15] P. N. Fahn and P. K. Pearson, "IPA: A New Class of Power Attacks," in *Cryptographic Hardware and Embedded Systems — CHES 1999*, ser. Lecture Notes in Computer Science, Çetin Kaya Koç and C. Paar, Eds., vol. 1717. Springer-Verlag, 1999, pp. 173–186.

[16] S. Chari, J. R. Rao, and P. Rohatgi, "Template Attacks," in *Cryptographic Hardware and Embedded Systems — CHES 2002*, ser. Lecture Notes in Computer Science, B. S. Kaliski, Çetin Kaya Koç, and C. Paar, Eds., vol. 2523. Springer-Verlag, 2003, pp. 13–28.

[17] G. Hospodar, B. Gierlichs, E. D. Mulder, I. Verbauwhede, and J. Vandewalle, "Machine learning in side-channel analysis: a first study," *Journal of Cryptographic Engineering*, vol. 1, no. 4, pp. 293–302, 2011.

[18] L. Lerman, G. Bontempi, and O. Markowitch, "Side channel attack: an approach based on machine learning," in *Constructive Side-Channel Analysis and Secure Design — COSADE 2011*, W. Schindler and S. A. Huss, Eds., 2011, pp. 29–41.

[19] W. Schindler, K. Lemke, and C. Paar, "A Stochastic Model for Differential Side Channel Cryptanalysis," in *Cryptographic Hardware and Embedded Systems — CHES 2005*, ser. Lecture Notes in Computer Science, J. R. Rao and B. Sunar, Eds., vol. 3659. Springer-Verlag, 2005, pp. 30–46.

[20] D. Agrawal, J. R. Rao, P. Rohatgi, and K. Schramm, "Templates as Master Keys," in *Cryptographic Hardware and Embedded Systems — CHES 2005*, ser. Lecture Notes in Computer Science, J. R. Rao and B. Sunar, Eds., vol. 3659. Springer-Verlag, 2005, pp. 15–29.

[21] L. Lerman, S. F. Medeiros, N. Veshchikov, C. Meuter, G. Bontempi, and O. Markowitch, "Semi-Supervised Template Attack," in *Constructive Side-Channel Analysis and Secure Design — COSADE 2013*, ser. Lecture Notes in Computer Science, E. Prouff, Ed., vol. 7864. Springer-Verlag, 2013, pp. 184–199.

[22] N. Hanley, M. Tunstall, and W. P. Marnane, "Using Templates to Distinguish Multiplications from Squaring Operations," *International Journal of Information Security*, vol. 10, no. 4, pp. 255–266, 2011.

[23] M. Renauld, F.-X. Standaert, N. Veyrat-Charvillon, D. Kamel, and D. Flandre, "A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices," in *Advances in Cryptology — EUROCRYPT 2011*, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed., vol. 6632. Springer-Verlag, 2011, pp. 109–128.

[24] M. A. Elaabid and S. Guilley, "Portability of templates," *Journal of Cryptographic Engineering*, vol. 2, no. 1, pp. 63–74, 2012.

[25] D. P. Montminy, R. O. Baldwin, M. A. Temple, and E. D. Laspe, "Improving cross-device attacks using zero-mean unit-variance normalization," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 99–110, 2013.

[26] V. Lomné, E. Prouff, and T. Roche, "Behind the Scene of Side Channel Attacks," in *Advances in Cryptology — ASIACRYPT 2013*, ser. Lecture Notes in Computer Science, K. Sako and P. Sarkar, Eds., vol. 8269. Springer-Verlag, 2013, pp. 506–525.

[27] E. Oswald and S. Mangard, "Template Attacks on Masking — Resistance is Futile," in *Topics in Cryptology — CT-RSA 2007*, ser. Lecture Notes in Computer Science, M. Abe, Ed., vol. 4377. Springer-Verlag, 2007, pp. 243–256.

[28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer-Verlag, 2009.

[29] O. Choudary and M. G. Kuhn, "Efficient Template Attacks," Cryptology ePrint Archive, Report 2013/770, 2013, http://eprint.iacr.org/.

[30] NIST, "FIPS-197: Advanced Encryption Standard (AES)," National Institute of Standards and Technology publication, 2001, http://www.nist.gov/publication-portal.cfm.

[31] E. Prouff, "DPA Attacks and S-Boxes," in *Fast Software Encryption — FSE 2005*, ser. Lecture Notes in Computer Science, H. Gilbert and H. Handschuh, Eds., vol. 3557. Springer-Verlag, 2005, pp. 424–441.

[32] F.-X. Standaert and C. Archambeau, "Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages," in *Cryptographic Hardware and Embedded Systems — CHES 2008*, ser. Lecture Notes in Computer Science, E. Oswald and P. Rohatgi, Eds., vol. 5154. Springer-Verlag, 2008, pp. 411–425.

[33] S. Bhasin, J.-L. Danger, S. Guilley, and Z. Najm, "NICV: Normalized Inter-Class Variance for Detection of Side-Channel Leakage," Cryptology ePrint Archive, Report 2013/717, 2013, http://eprint.iacr.org/.