Template Attack Against AES in Counter Mode With Unknown Initial Counter

Marcial Tienteu
Morgan State University, E.C.E
Baltimore, Maryland - 21251
Email: matie1@morgan.edu

Edmund Smith
Morgan State University, E.C.E
Baltimore, Maryland - 21251
Email: edaho1@morgan.edu

Edgar Mateos Santillan
Riscure North America
Email: MateosSantillan@riscure.com

Kevin Kornegay
Morgan State University, E.C.E
Baltimore, Maryland - 21251
Email: kevin.kornegay@morgan.edu

Paige Harvey
Morgan State University, E.C.E
Baltimore, Maryland - 21251
Email: pahar9@morgan.edu

Otily Toutsop
Morgan State University, E.C.E
Baltimore, Maryland - 21251
Email: ottou1@morgan.edu

Tsion Yimer
Morgan State University, E.C.E
Baltimore, Maryland - 21251
Email: tsyim1@morgan.edu

Vinton Morris
Morgan State University, E.C.E
Baltimore, Maryland - 21251
Email: vinton.morris@morgan.edu

Ketchiozo Wandji
Morgan State University, E.C.E
Baltimore, Maryland - 21251
Email: ketchiozo.wandji@morgan.edu

2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC) | 979-8-3503-3286-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCWC57344.2023.10099238

***Abstract*—Despite long-contested viability, numerous applications still rely upon Advance Encryption Standard (AES) in Counter mode (AES-CTR). Research supports that the vulnerabilities associated with CTR from a mathematical perspective, mainly forgery attempts, stem from misusing the nonce. When paired with cryptographic algorithms, assuming no nonce misuse increases the complexity of unraveling CTR. This paper examines the pairing of CTR with AES-128 (AES-CTR). It includes (1) full key recovery for a software implementation of AES-CTR utilizing a template attack (TA) and (2) enhancing the TA analysis's point of interest (POI) using first-order analysis and known key to identify leaky samples.**

***Index Terms*—AES, Counter mode, CTR, EM SCA, POI, Sidechannel Analysis, Template Attack**

I. Introduction

Block cipher modes of operation provide confidentiality to a block cipher [1]. Some modes, such as cipher feedback (CFB) and counter (CTR), have the added advantage of converting a block cipher to a stream cipher. Since the introduction of block cipher modes of operation, many studies have addressed attacks on the algorithms [2], [3], [4] and their implementations in software and hardware. Techniques such as side-channel analysis (SCA) is well known for finding exploitation in the implementation, filling in the gaps left by analytical approaches.

SCA's first introduction was given in the work of [5], where a power analysis attack utilizing a difference of mean (DoM) distinguisher allows for a full key recovery attack.Following the work in [5], several other studies enhanced the prior result, [6] introduce the correlation coefficient as a more efficient statistical tool which ignores the noise compared to the DoM; the attack became known as correlation power analysis (CPA), [7] mutual information (MIA), and [8] TA. The attacks listed above work well against leakage at a single point, first-order leakage. However, AES-CTR and CTR mode is unique due to their nonce coupled with counter input into the cipher, which leads to the failure of standard first-order analysis techniques such as DoM, and CPA, when the nonce is unknown. To alleviate this facto, r [9] created a method that handles the unknown initialization vector (nonce) and an unknown key while showing the primary method for evaluating side-channel analysis and attacks of AES-CTR as well as other CTR-related implementations (e.g.,Galois counter mode and counter with cipher block chaining message authentication code). Other methods, such as deep learning and neural networks [10], [11], [12] have been tested, [13], but have yet to yield an efficiency compared to the [9] approach.

The current work shows that TA can produce comparable results with similar devices to [9] and [13]. However, region containing the POIs must be carefully selected. To find the POIs, a comparison algorithm searches numerous first-order analyses for a family of devices to determine where the key bytes are located. In practice, the nonce for cryptosystem based on AES-CTR will be unknown, creating challenges when performing first-order analysis without relying on [9]. Using a set of known nonce and plaintext from a family of embedded devices running the same cryptographic module makes it possible to ascertain POIs for a device with an unknown key. Although it is possible to use first-order analysis, such as differential power analysis(DPA) and CPA,

979-8-3503-3286-5/23/$31.00 ©2023 IEEE

to develop a set of POIs, a system with an unknown key is complex. The method provided by [9] can accomplish this task but is computationally intensive. Using TA with a set of POIs reduces the computational workload.

The remainder of this paper is organized as follows: Section II describes our contributions. Section III covers the background and literature review. Section IV describes the threat model. A description of the experimental configuration is given in sections V and VI. The implementation is discussed in section VII. Results are presented in Section VIII, and conclusions are presented in section IX.

II. Contribution of this Work

TA may require profiling a family of devices running the same algorithm with a known key(s) before performing an attack against a device from the same family running an unknown key. Since the samples in a trace may be large ($n > 1000$), there is a high computational complexity when processing the noise covariance matrix needed to execute an attack. In performing a TA on AES-CTR, this paper contains the following contributions:

- Implementing an algorithm that compares multiple first-order analysis results to determine the best POIs for a family of devices running the same cryptographic algorithm.
- Performing a full key recovery attack on a CoraZ7-07s device running AES-CTR.

III. Background

A. Advanced Encryption Standard (AES)

FIPS-197 [14] standardized the AES in 2001. AES is a 128-bit block cipher with a substitution permutation network. AES has three variants of master keys (K), 128, 192, and 256-bits. Concerning key sizes, the rounds are 10, 12, and 14, respectively. For AES-128, the first nine rounds break down into four transforms, Addroundkey, Subbytes, Shiftrows, and Mixcolumns. The tenth round omits Mixcolumns. AES 128-bit state and key state map to a four-by-four matrix. AES takes 16-byte input A and returns 16-byte output A'.The four transformations for AES operation are:

- AddRoundKey: for a round key K^r: $A'_{i,j} = A_{i,j} \oplus K_{i,j}$
- SubByte: substitution (S) permutation on a 256-byte on the output from AddRoundKey: $A'_{i,j} = S[A_{i,j}] = S[A_{i,j} \oplus K_{i,j}]$
- ShiftRows: circular shift on the last three rows of the 4-by-4 matrix: $A'_{i,j} = A_{i,j+i}$
- MixColumns: A 4-byte lookup table M is applied: $A'_{i,j} = \sum_{n=0}^{3} M_{n-i} A_{j,n}$

Several attacks have targeted AES, from an implementation perspective [9], [13], [15], [16], [17] in the name of key recovery. These attacks utilize power, temperature, radiation, timing, and sometimes even acoustic variations. They have been able to break AES and recover the secret key. However, several modes of operations, such as counter mode (CTR) and cipher block chain (CBC), are attached to AES to provide confidentiality.

B. AES Counter Mode (AES-CTR)

CTR is a NIST standard mode of operation for block ciphers [18]. In CTR, instead of encrypting the plaintext, the nonce plus counter is encrypted. The output from the encryption process is XORed with the plaintext block to generate the ciphertext. Let B denote the block cipher, K the key, C the initial counter, X_T represent the T^{th} block to be encrypted, and Y_T represent the T^{th} block to be ciphertext. Then the T^{th} block of the ciphertext can be written as seen in equations 1 and 2:

$$Y_T = X_T \oplus B(C + T, K) \tag{1}$$

Subsequently, the decryption process is as follows:

$$X_T = Y_T \oplus B(C + T, K) \tag{2}$$

C. Side-channel Analysis (SCA)

The beginning of modern SCA began with [19] in 1996. Their study focuses on key recovery using timing traces collected from a device while performing a cryptographic operation. Since then, several studies have expanded upon [19] work. Side-channel attacks break down into passive attacks (non-invasive) and active attacks (invasive). Passive attack channels include power measurement [20] of a device running cryptographic operation and reading of electromagnetic (EM) emanation [21]. Example of power analysis methods include simple power analysis, DPA [5], CPA [22],MIA [7], linear regression analysis [23], and TA [24], [25], [26], [8], which is a profiling-based attack.

There have been several countermeasures proposed to counter these side-channel attacks. Most countermeasures focus on either reducing the release of leaked information or removing any relationship between the leaked information and the secret. Some algorithms reduce the number of times a key [27] is used, which prevents adversaries from collecting a high number of traces, thereby deterring the adversary's advantage. A constant tug-of-war exist between adversaries and researchers. Side-channel analysis is an evolving field that exploit physical system weaknesses. Since it is challenging to balance performance and security, there will always be flaws in the physical system, leading to exploitable gaps for adversaries.

D. EM SCA

EM emanation is a form of radiation leakage that yields an exploitable side channel of physical systems. When a system running a cryptographic operation radiates high EM emanation, an adversary can exploit the leakage through the radiation and recover the secrets about the cryptosystem. Special shields [28] encase the system to reduce radiation leaks, thus, mitigating information leakage. There are other

available methods for reducing the leakage of information. The approach described in [28] suggests a low-level approach that requiring the designer to start by developing the integrated circuit (IC) with built in radiation prevention techniques. By routing the cipher at the lower levels of the integrated circuit and avoiding the top layer (metal layer), reducing the cryptographic algorithm's leakage level is possible.

E. Template Attack (TA)

TA apply concepts from signal detection, estimation theory, and information-theory [22]. TA is a two-phase process. However, it is treated as a three-phase in this work. The initial two phases of a template attack are the profiling (learning) phase and the exploitation (attack) phase. The most crucial phase in any practical template attack is considered to be the POI selection. The POI phase is not mandatory [8]; however, if POI is omitted, the TA could suffer from timing complexity issues due to the large sample size. Making the POI selection mandatory eliminates the timing complexity issue in this work. Furthermore, a suitable method for selecting points can reduce adversary's time to deploy an attack.

As described by [8], TA follow a multivariate Gaussian model. [25] simplifies the steps to achieve a practical TA. For the sake of convenience, we reiterate these steps. Trace classification requires building a template for each of the possible operations that reside in the trace. The templates then reflect the statistical characteristics of the trace. In other words, the distribution of the trace properties for all points.

1) Points of Interest (POI): POI or selected points serve as a way to reduce the computational complexity of the template by reducing the number of inputs to the noise covariance matrix. Researchers have proposed many techniques to select POIs [25], [24], [29]. Overall, they all lead to the same conclusion. Given a trace with an N sample, a set $n < N$ must be chosen to reduce the computational complexity directly related to the noise covariance matrix for a template attack. [8], [25] describe some properties the selected points should possess. The properties are as follows:

- The minimum distance between these points should be approximately a clock cycle since additional points in the same cycle do not provide further information.
- The minimum height of a selected point should be higher than the noise floor of the sum of differences traces.

Selected points without these properties lead to poor classification performance during the template generation phase. This process relies on CPA for selecting points with the highest peak and ensuring that per byte of the key, a point exist such that the minimum height is higher than the noise floor. CPA also ensures that our points do not fall into the same clock cycle.

2) Template Building: The mean vector consists of the mean for each operation taken from all traces in the set. Assuming m operations need to be distinguished and for those operations, several q traces captured named $t_1, ..., t_q$. If one of the m operations is considered, then the mean vector is calculated as shown in equation 3 :

$$\bar{t} = M = \frac{1}{q}\sum_{j=1}^{q} t_j \tag{3}$$

All traces are taken from the same operation. For all t_i traces of an operation, there exists a corresponding noise vector N_i such that $N_i = t_i - M$. The noise vectors form the basis of the noise covariance matrix; the second part is needed to build a template. The covariance matrix measures the linear dependence between two random variables. The covariance of two random variables, X and Y, is defined as shown in equation 4:

$$cov(X, Y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{4}$$

Our trace set contains more than two random variables. Therefore, a covariance matrix is needed for its description. The noise covariance matrix can be defined as shown in equation 5:

$$CM(u, v) = cov(N_u, N_v) \tag{5}$$

For all m operations, template tuple defined as: (M, CM), where M is the mean vector and CM is the covariance matrix.

3) Trace Classification: After building all possible templates for a device, it becomes possible to classify a captured trace t from the chosen attack device. Distinguishing the captured trace requires an application of the maximum likelihood principle to select a template that best fits the trace. The steps below show how to accomplish this process.

1) For a specific key value and its associated template, compute the noise N_i as if the device under attack is associated with that template. The mean vector M_i refers to the vector of means generated during the template-building phase, as shown in equation 6.

$$N_j = M_i - t \tag{6}$$

2) Assuming every point in the trace is drawn from a Gaussian probability distribution, it is possible to use the $n-$dimensional Gaussian probability distribution along with the maximum likelihood to select a template with the highest probability for observing the calculated noise covariance as shown in equation 7.

$$\rho(N_i) = \frac{1}{\sqrt{(2\pi)^n |CM_i|}} e^{-\frac{1}{2} N_i^T CM_i^{-1} N_i} \tag{7}$$

Equation 7 measures how well the template fits a given trace.

3) Finally, the newly generated template based on the unknown key is compared to the known key template generated in the template-building phase. The decision rule is the maximum likelihood, which minimizes the probability of selecting the wrong decision given a hypothetical pair (M, CM). The maximum likelihood decision rule is calculated as shown in equation 8.

$$\rho(t;(M,CM)_j) > \rho(t;(M,CM)_l) \quad \forall j \neq l \quad (8)$$

Where t represents a trace from the unknown set,$(M,CM)_j$ is hypothetical for the unknown key and $(M,CM)_l$ the hypothetical for the known key.

IV. Threat Model

The threat model is split into two parts, the first part encompasses the POIs phase, which relies on CPA. The second part covers a DPA-like TA relying on the DoM.

A. POI

Using CPA, the POIs are determined by the Hamming weight model in a first-order attack scenario. The Hamming weight model assumes the data leakage through the power side channel depends on the number of bits switching [22]. Thus, given an n-bit processor with binary data $D = \sum_{j=0}^{n-1} d_j$,$d_j = 0$ or 1. The hamming weight is the number of bits set in a data word. $H(D) = \sum_{j=0}^{n-1} d_j$ The intermediary attack model relies on the output of the encryption of the ciphertext C and the key K. Assuming some associated data D consisting of the ciphertext; the leakage is defined as L, in equation 9:

$$L(D,K) = HW(S^{-1}[C_{i,j} \oplus K_{i,j}]) \quad (9)$$

B. Template

Once all POIs are located, the next step is building templates. Naturally, templates are characterized using the multivariate normal distribution of the pair (M, CM), where M is the mean vector and CM is the covariance matrix. During this step, some modifications are made to the original attacks.

(1) This work assumes that the noise did not significantly affect the attack's performance. This assumption simplifies the template-building phase since the covariance matrix is replaced with the covariance identity matrix. The reduced multivariate normal distribution equation 7 is as shown in equation 10:

$$p(t;m) = \frac{1}{\sqrt{(2\pi)^n}} \exp -\frac{1}{2} N_i^T N_i \quad (10)$$

Furthermore, the natural log of the distribution is taken to eliminate the issue with small values during the normal distribution calculation, as shown in equation 11.

$$\ln p(t;m) = -\frac{1}{2}((\ln 2\pi)^n + N_i^T N_i) \quad (11)$$

(2) During the classification (exploitation) phase, the least-square (LSQ) test is applied. The only relevant term in the simplified multivariate normal distribution is the square of the difference between t and m. The decision rule for the reduced template simplifies to equation 12:

$$(t-m_{d_i,k_j})^T(t-m_{d_i,k_j}) < (t-m_{d_i,k_l})^T(t-m_{d_i,k_l}) \quad (12)$$

where $N_i = t - m_{d_i,k_j}$ for the known set of trace and $(t - m_{d_i,k_l})$ for the unknown set of trace. The reduced equations 11 and 12 are easier to work with since they remove the problem of a singular matrix and eliminate the timing cost that would exist if a covariance matrix is present. They should be adapted if the noise does not contribute significantly to the trace set under attack.

V. Experimental Setup

The device under test is the CoraZ7-07s [30], containing a Xilinx System-on-Chip. Our code executes on the ZYNQ hard processor, an ARM Cortex A9 single-core processor operating at 667 MHz.

A. EM Probe

The experiment utilizes an EM probe for trace collection. The probe is a high-precision probe using a tip of 0.5mm diameter. The tip includes an adjustable shield and direct coil, allowing the search to pick up EM fields with frequencies up to 6 GHz; the collected frequencies get converted to AC signals for the scope to process. In addition, the probe gets attached to an XYZ station which allows for scans of the chip to find the precise area of operation of the cryptographic algorithm when collecting traces. Figure 1 shows the setup.

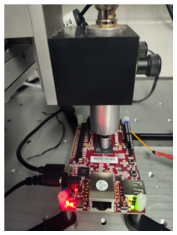

Fig. 1. EM Acquisition of traces from CoraZ7-07s using a EM probe mounted on an XYZ station

B. Oscilloscope

The oscilloscope is the Lecroy Waverunner 9404M Oscilloscope. It is a 4 GHz, 4 Channel, 40 GS/s oscilloscope with 64 Mpts/CH. The scope runs a driver that integrates with the collection computer for later processing.

C. Collection PC

For data collection and processing, a Dell Precision Tower 5820 runs an 8-core Xeon processor. Riscure inspector is used to conduct the attack utilizing the built-in modules for TA. Figure 2 shows a schematic of the experimental setup.

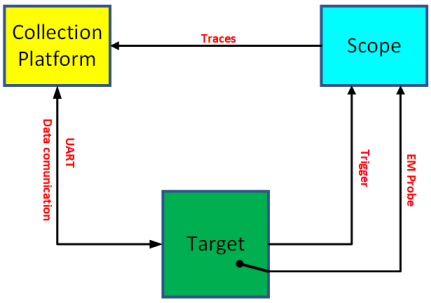

Fig. 2. SCA setup for trace collection and processing

VI. Experimentation

Our experiment included two phases trace collection and trace processing.

A. Trace Collection

EM trace collection requires a precise understanding of where the cryptographic operation is active on the chip. The typical way of determining areas that may house the cryptographic function is to divide the chip into a grid and scan each grid region using an EM probe; thus, a detailed map is developed of the area where the operation occurs by splitting the grid into tiny rectangles. The best arrangement usually is an n-by-n. Once the grid arrangement is finished, the next step is to run the cryptographic operation and collect a trace from each grid segment. After collecting traces per segment, it is best to apply spectral analysis to the set of traces to look at the spectral intensity. The spectral intensity allows for a visual perspective of high-activity areas on the chip. Assuming the only operation running on the chip is the cryptographic operation, then areas of high activity indicate where the operation is happening. Figure 3 shows an example of a grid using a 10-by-10 square showcasing the spectral intensity of AES-CTR on CoraZ7-07, the point with the highest intensity is shown in red. After selecting an

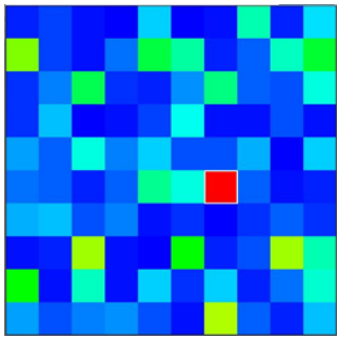

Fig. 3. Spectral intensity graph of the scanned region of the chip for the device under test

area based on the spectral intensity graph, the EM probe is moved to that area to collect some traces.

B. Trace Processing

The next step involves collecting traces and confirming the correctness of the inputs and outputs. This step is done by collecting the input and outputs from a set of encryption with a known key. An AES-CTR calculator can be used to verify the information, and the result is then compared to the output to ensure it is correct.

VII. Results

Finding the POIs proved to be difficult for this experiment. This work assumes the typical POIs methods such as CPA, Signal-to-Noise Ratios (SNR), Sum Of Squared pairwise T-difference (SoST), and Sum Of Squared Difference (SoSD) would be sufficient for generating successful POIs. However, those methods fell apart when attempting to recover the unknown key using the 10^{th} round AES-CTR. Several iterations were done in the recovery attempt, but they all produced unsatisfactory results. Finally, a first-order analysis on the trace set was launched with a known key, but the POIs from the first-order analysis did not match the points generated using CPA, SNR, SoST, and SoSD. This process was repeated for several traces with known keys from the same device family, and the results from the first-order analysis for those trace sets matched closely with each order compared with the POI techniques. Based on our findings, we decided to trust the first-order analysis results over the POI results.

A MATLAB algorithm was implemented to compare the results from all first-order analyses two at a time and on a byte-by-byte basis. The idea behind this approach is that devices from the same family running the same algorithm should have POIs within the same region for the same key byte. This approach and the method leading up to it are described in section VII-A. In section VII-B, a successful key recovery using templates built based on the method is shown in section VII-A

A. POI

In our approach, two sets of traces are gathered under identical conditions but with two distinct key sets for each set. A total of 1.5 million traces and 66,000 samples with the oscilloscope frequency set to 1 GHz were gathered for each trace set through the experimental phase of the research. The stages involved in the process are shown in Figure 4.

For algorithm 1, two correlation coefficient matrices are considered, $data_1$ and $data_2$, which are the results of the first-order analysis. The dimensions for the matrices are $T \times S \times N$, where T is traces, S is the number of samples, and N is the number of byte . Next, the absolute value of both data sets is taken to ensure the range is from 0 to 1. Next, the maximum index is collected for each byte, idx_1 and idx_2, and its corresponding peak value, S_1 and S_2, respectively, per trace. Then, sort the max peak values in descending order, keeping track of the indices for each

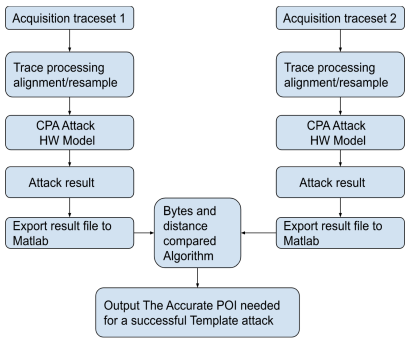

Fig. 4. Setup for comparing the first-order analysis results for two set traces from the same device family

maximum value. After sorting, the indexes are associated back to S_1 and S_2. Lastly, the top 10 maximum indexes for S_1 and S_2 are compared. Also, displaying the indexes and computing the difference if they are not matched. The difference computation is necessary because it determines how well samples related to the same byte from a family of devices overlap. If the samples are within a clock cycle based on the difference, then using any of those samples in the template-building phase will yield adequate results in the classification phase. However, if the indices' differences are too far apart (multiple cycles apart), the templates built using these samples will give poor results during key recovery.

Algorithm 1 Byte and Distance Compare

Input: $data_1$, $data_2$

All correlation coefficients should be in the range [0,1]
$[S_1, idx_1] \leftarrow max(abs(data1))$
$S_1 \leftarrow SORT(S_1)$
$[S_2, idx_2] \leftarrow max(abs(data2))$
$S_2 \leftarrow SORT(S_2)$
Keep track of the total number of traces (cols),
samples (rows), and bytes
$[T, S, N] \leftarrow SIZE(data1)$
Associate the index idx_1 and idx_2 back to
S_1 and S_2 after sorting. Then compare the top 10
max indices for both data sets on a byte-by-byte basis
for $i \leftarrow 0$ to $N-1$ **do**
    for $j \leftarrow 0$ to $(T-1)$ **do**
        if $idx_1[i,j] == idx_2[i,j]$ **then**
            print match and index value
        else
            $diff = abs(idx_1[i,j] - idx_2[i,j])$
            print i,j, S1, and S2 indices
        end if
    end for
end for

The Leakage process includes some of the most important aspects considered throughout the decision-making procedure before a CPA attack is launched on the CoraZ7 board.

After an in-depth analysis of the CTR block diagram, we settled on attacking the last AES block, round 10, to retrieve the key. The hamming weight model is used as the primary model.

As previously stated, two devices from the same family are used to collect two sets of traces with two distinct keys. Figure 5 shows the two sets collected from the devices. The

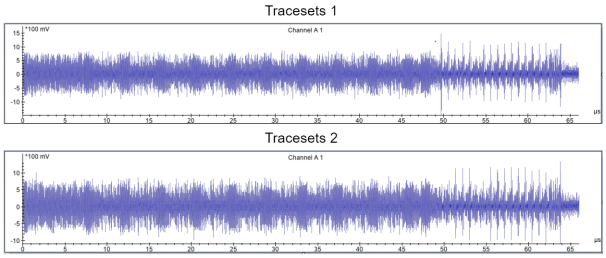

Fig. 5. Traces collected from similar devices using different keys

top image, called "trace sets 1," and the second, the down bottle image, called "trace sets 2," were collected under the same conditions and acquisition equipment. The only difference between the two sets of traces is the keys.

Because the device are identical and the key is the only difference, we assume a meaningful change in the trace sets could be found. The assumption was very quickly as incorrect. Due to the key difference, a slight change in the power level was expected but not in the timeshift. However, looking at the post-processing phase, a shift in time is noticed. This time shift is seen when at 50 µs position on trace sets 1 and 2. From these two sets of traces, there was a difference between the timeshift and the power level. Before diving into the attack, it is crucial to visually confirm that the trace sets showed the pattern matching the cryptography algorithm's operations. Figure 6 presents the trace sets' absolute value taken. After taking the absolute versus the

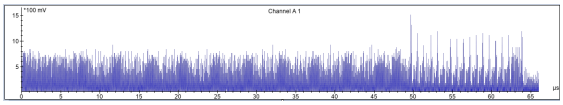

Fig. 6. The absolute of trace set 1

raw traces, it is much easier to identify partners in the traces since the absolute eliminates all minimum points from the original traces. Furthermore, during the inspection of the absolute traces set, the standard deviation is computed and shown in Figure 7.

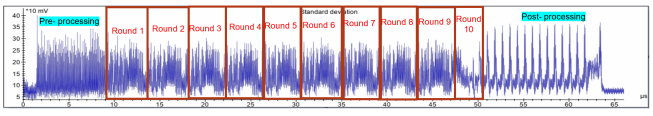

Fig. 7. Standard deviation for trace set 1

The standard deviation helps to visualize all the rounds of the cryptography operation from start to finish. For example the ten rounds of the AES-128 are easily counted; rounds 1 to 9 are similar, while round 10 is different. The trace set is correct because AES-128 in the nine first rounds contains the following operations, SubBytes, ShiftRows, MixColumns, and AddRoundKey. The tenth round of the algorithm is smaller since it does not contain MixColumns. The same process is repeated on the traces set 2.

Once both trace sets were confirmed to correct, the trace's static aligment and resampling were performed. The tenth round from each set was extracted to increase the processing performance. The new trace sets contain only round 10 of the operation and have only 6258 samples out of 66K from the original trace sets. Then a first-order analysis was conducted using correlation power analysis (CPA) on both trace sets. Without the loss of generality, the hamming weight attack model is chosen to attack the tenth round of the AES. Since AES-128 CTR is constant in the front (round 1) except for the last 4 bytes of the plaintext, an attack on round 1 using the plaintext would not yield success. Figure 8 showcases the successful results of the CPA attack on both sets of traces.

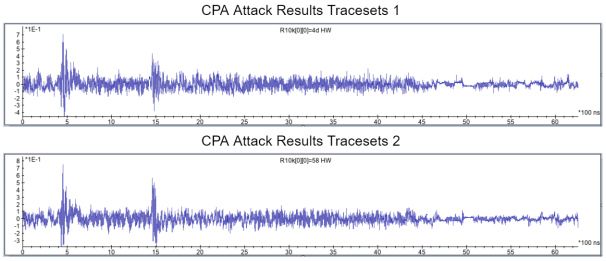

Fig. 8. Correlation power analysis result showing similar POIs for both sets of traces

The CPA results of trace sets one, and two show that at least one byte of the tenth round key is close to the same position on both trace sets. Indeed, this met our expectations of similar devices, the same operation, different keys, and the same attack, yielding similar results. These results then gave us confidence that, under the same circumstances, the TA would be a direct attack. Trace set 1 is used in the learning phase, and trace set 2 is for the matching set.

Knowing that the CPA attack was successful on both sets, the POI file was generated based on the CPA POI method. After generating the point of interest file, we created the template profile file and proceeded to the attack phase. Surprisingly, the attack failed. We began investigating the failure by inspecting the POIs graphs corresponding to the previous steps' CPA attack result. Figure 9 shows the POI using the CPA method. When comparing the CPA attack results in Figure 8 to the CPA base POIs in Figure 10, we quickly identified a high correlation at 0.5 microseconds, which did not exist in the CPA attack results in Figure 8. Next, we used other POI methods, such as the Sum Of

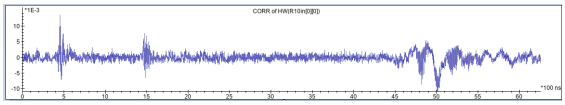

Fig. 9. Result of CPA for POIs with an extra peak at 5 microseconds overshadowing peaks at 0.5 and 1.5 microseconds

Squared pairwise T-differences (SOST) and the Signal-to-Noise Ratios (SNR). Then tried those additional methods, but the TA still failed. So we finally settled on examining the SOST and the SNR base POI graph of as shown in Figure 10.

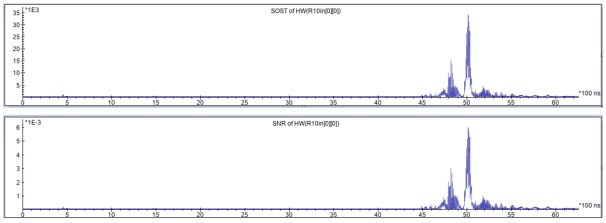

Fig. 10. Top Sum Of Squared pairwise T-differences (SOST) Bottom Signal-to-Noise Ratios (SNR)

We found that the high peak observed from the CPA-based POI is the only peak that the SOST and the SNR methods are capturing. When zooming in at around 0.5 microseconds on both the SOST and SNR graphs, the key byte located at that position is found with relatively low power. Figure 11 displays the lower power peak at 0.5 microseconds. Both of

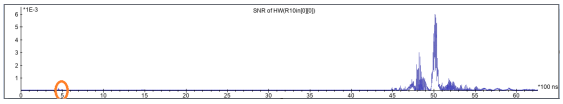

Fig. 11. Small peak at 0.5 microseconds is obscured by the peak at 5 microseconds

the abovementioned POI approaches helped to find the root cause of the issue. The peak at 5 microseconds masks the detection of the actual sample by the software.

We sorted through all the points in the CPA attack file to find the matching point between both datasets. Then imported the results into the Matlab code and searched for the point where the high peaks matched. The output shows the exact location and the distance between high peaks in both datasets. The comparison result is used to crop out the valuable parts from trace sets 1 and 2. Figure 12 shows the crop traces of sets 1 and 2.

The POI is generated from snipped trace set 1 using the CPA, SOST, and SNR POI methods. Figure 13 shows the new points of interest from the three methods used. After generating the POIs, all the byte positions on the graphs in Figure 8 are now visible and look very similar to the ones in Figure 13. Then the template profile with the CPA-based

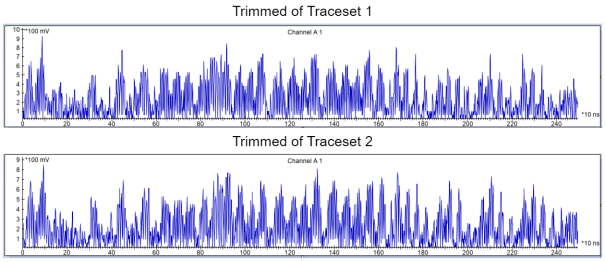

Fig. 12. Trimmed traces based on output from first-order comparison data

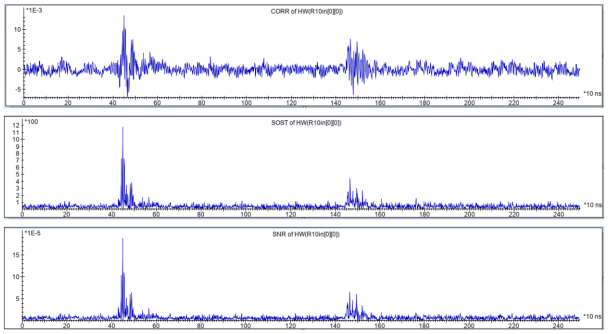

Fig. 13. After cropping the trace set, the result from the POI search matches the first-order attack results

POI mentioned initially is generated. Using the new template profile, we successfully retrieved all 16 bytes of keys on the known input trace set.

B. Template Attack

After settling on both trace sets above for the POI stage, a third trace set was collected. The third trace set has 500k traces, and this time the trace set contains an unknown key. We snipped the new trace set based on our previous results during the point of interest phase, keeping only the tenth round. Table I shows a list of POI selected from the POI phase for the template building.

TABLE I
POI FOR ATTACK

	Point of Interests									
	POI 1	POI 2	POI 3	POI 4	POI 5	POI 6	POI 7	POI 8	POI 9	POI 10
Byte 1	0	449	455	456	443	448	1466	490	483	444
Byte 2	512	1656	1657	511	505	1697	559	506	1696	519
Byte 3	1846	1886	1845	562	1879	563	1885	1878	1847	537
Byte 4	2035	621	622	627	2036	628	620	631	2068	2075
Byte 5	770	729	778	730	771	722	777	818	1500	1499
Byte 6	791	1689	1688	762	1755	1728	1696	790	1739	1780
Byte 7	1878	1969	1877	1879	1905	837	1904	1885	1906	1918
Byte 8	2158	2065	2066	2107	2134	2067	2117	2133	2068	2118
Byte 9	976	969	970	1018	977	1011	1010	1017	968	975
Byte 10	1031	1024	1771	1732	1030	1025	1023	1738	1022	1739
Byte 11	1928	1921	1080	1070	1087	1069	1054	1086	1088	1922
Byte 12	2150	2110	2111	1132	2117	2151	2129	1131	2149	2139
Byte 13	1653	1652	1654	1686	1216	1647	1693	1605	1651	1209
Byte 14	1843	1842	1882	1875	1825	1841	1833	1883	1834	1794
Byte 15	2031	2032	2030	2033	2023	2022	2014	2071	1971	2064
Byte 16	2218	2217	2219	2238	2233	2224	2173	2223	2230	2212

We successfully recovered the key for the tenth round using the built template against the new trace set. The POIs colored in green in Table I are the exact match of the new sample set in the template file. The columns in Table I are the 10 POIs for each key byte, and the rows represent the position for each key byte in the tenth round. Even though the last four rows in Table I did not precisely match the new sample set, it did not influence the full round key recovery. The original key is recovered by working backward from the tenth round using the AES key scheduler.

VIII. Conclusion

This work presents a TA on AES-128 CTR. We create a comparative algorithm to select the best point of interest from 2 sets of traces and show that the selected points are the suitable region to build the templates. We were able to apply the built template to a third set of traces and fully recover the key used in that set with 500k traces. The method presented in this work reduced the number of traces used to recover a key from an AES-128 CTR software implementation on a Coraz7 board.

References

[1] M. Dworkin, "Recommendation for Block Cipher Modes of Operation," *National Institute of Standards and Technology Special Publication 800-38A 2001 ED*, vol. X, no. December, pp. 1–23, 2005. [Online]. Available: http://csrc.nist.gov/publications/drafts/800-38g/sp800_38g_draft.pdf

[2] G. Leurent and F. Sibleyras, "The Missing Difference Problem, and Its Applications to Counter Mode Encryption," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10821 LNCS, pp. 745–770, 2018.

[3] G. Procter and C. Cid, "On Weak Keys and Forgery Attacks Against Polynomial-Based MAC Schemes," *Journal of Cryptology*, vol. 28, no. 4, pp. 769–795, 2015.

[4] T. Iwata, K. Ohashi, and K. Minematsu, "Breaking and repairing GCM security proofs," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7417 LNCS, pp. 31–49, 2012.

[5] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1666, pp. 388–397, 1999.

[6] E. Brier, C. Clavier, and F. Olivier, "Optimal statistical power analysis," 2003, francis.olivier@gemplus.com 12303 received 31 Jul 2003, last revised 8 Sep 2003. [Online]. Available: http://eprint.iacr.org/2003/152

[7] B. Gierlichs, L. Batina, and P. Tuyls, "Mutual Information Analysis–{A} Universal Differential Side-Channel Attack," *Cryptology ePrint Archive, Report 2007/198*, pp. 1–16, 2007.

[8] S. Chari, J. R. Rao, and P. Rohatgi, "Template Attacks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2523, pp. 13–28, 2003.

[9] J. Jaffe, "A first-order DPA attack against AES in counter mode with unknown initial counter," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4727 LNCS, pp. 1–13, 2007.

[10] L. Masure, C. Dumas, and E. Prouff, "A Comprehensive Study of Deep Learning for Side-Channel Analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, pp. 348–375, 2019.

[11] L. Wouters, V. Arribas, B. Gierlichs, and B. Preneel, "Revisiting a Methodology for Efficient CNN Architectures in Profiling Attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, pp. 147–168, 2020.

[12] S. Swaminathan, L. Chmielewski, G. Perin, and S. Picek, "Deep learning-based side-channel analysis against aes inner rounds," Cryptology ePrint Archive, Paper 2021/981, 2021, https://eprint.iacr.org/2021/981. [Online]. Available: https://eprint.iacr.org/2021/981

[13] L. De Meyer, "Recovering the CTR_DRBG state in 256 traces," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, pp. 37–65, 2019.

[14] J. Daemen and V. Rijmen, "The advanced encryption standard process," *Information Security and Cryptography*, pp. 1–8, 2020.

[15] J. Longo, E. De Mulder, D. Page, and M. Tunst All, "SoC it to EM: Electromagnetic side-channel attacks on a complex system-on-chip," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9293, pp. 620–640, 2015.

[16] H. Naghibijouybari, A. Neupane, Z. Qian, and N. Abu Ghazaleh, "Side Channel Attacks on GPUs," *IEEE Transactions on Dependable and Secure Computing*, vol. 5971, no. c, pp. 1–1, 2019.

[17] S. Mangard, N. Pramstaller, and E. Oswald, "Successfully attacking masked AES hardware implementations," *Lecture Notes in Computer Science*, vol. 3659, pp. 157–171, 2005.

[18] Barker, Elaine and Kelsey, J, "NIST SP 800-90A Revision 1 Recommendation for random number generation using deterministic random bit generators (revised)," *NIST Special publication*, no. Giugno, 2015.

[19] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology — CRYPTO '96*, N. Koblitz, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 104–113.

[20] H. Gamaarachchi and H. Ganegoda, "Power analysis based side channel attack," *arXiv*, 2018.

[21] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2162, pp. 251–261, 2001.

[22] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3156, pp. 16–29, 2004.

[23] S. Fu, Z. Wang, F. Wei, G. Xu, and A. Wang, "Linear Regression Side Channel Attack Applied on Constant {XOR}," *Cryptology ePrint Archive: Report 2017/1217*, 2017. [Online]. Available: https://eprint.iacr.org/2017/1217.pdf

[24] O. Choudary and M. G. Kuhn, "Efficient Template Attacks," *CARDIS 2013*, no. Section 4, pp. 253–270, 2014.

[25] C. Rechberger and E. Oswald, "Practical template attacks," *Lecture Notes in Computer Science*, vol. 3325, pp. 440–456, 2005.

[26] B. Liu and J. H. Kong, "Practical template attacks based on pooled covariance matrix," *Proceedings - 7th Asia-Pacific Conference on Environmental Electromagnetics, CEEM 2015*, no. 1, pp. 184–188, 2015.

[27] E. Barker and J. Kelsey, "NIST SP 800-90A Revision 1 Recommendation for random number generation using deterministic random bit generators (revised)," *NIST Special publication*, no. Giugno, 2015.

[28] D. Das, M. Nath, B. Chatterjee, S. Ghosh, and S. Sen, "STELLAR: A Generic em Side-Channel attack protection through ground-up root-cause analysis," *Proceedings of the 2019 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2019*, pp. 11–20, 2019.

[29] C. Archambeau, E. Peeters, F. X. Standaert, and J. J. Quisquater, "Template attacks in principal subspaces," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4249 LNCS, pp. 1–14, 2006.

[30] DILIGENT. (2015) Cora Z7 Reference Manual. [Online]. Available: https://reference.digilentinc.com/programmable-logic/cora-z7/reference-manual