A Systematic Evaluation of Wavelet-Based Attack Framework on Random Delay Countermeasures

Fan Zhang, *Member, IEEE*, Xiaofei Dong, Bolin Yang, Yajin Zhou, and Kui Ren, *Fellow, IEEE*

***Abstract*— Random delay countermeasure is a commonly used defense against side-channel attacks, which brings certain interference and disturbance to those calculation sequences in the time domain. Data alignment and frequency attack are considered as typical techniques to counteract the random delay countermeasure. However, these attacks have limitations from the perspectives of both efficiency and performance. In comparison, facing those delays, wavelet analysis is considered as a more efficient technique due to its detailed and comprehensive interpretation of a signal. This paper applies different wavelet techniques to three attack components: noise reduction, trace alignment and key extraction. For the first time, the unified wavelet-based attack framework against random delays is proposed where wavelet analysis is fully applied in the entire attack life cycle. In particular, a novel method of trace alignment at the wavelet level is proposed in this framework, which is based on wavelet pattern detection to synchronize the misaligned power traces. Most importantly, the overall wavelet-based attack framework is systematically evaluated over three random delay strategies, after the respective contribution of each component is investigated through a series of comparative experiments. Experimental results show that the performance of the wavelet-based attack framework is significantly improved compared to standard attack procedures and frequency ones, which can be regarded as a unified and effective solution to conquer random delay countermeasures.**

***Index Terms*— Random delay countermeasures, side-channel analysis, wavelet framework, performance evaluation.**

Manuscript received March 30, 2019; revised August 9, 2019; accepted September 7, 2019. Date of publication September 16, 2019; date of current version December 13, 2019. This work was supported in part by the Open Fund of State Key Laboratory of Cryptology under Grant MMKFKT201805, in part by the Zhejiang Key Research and Development Plan under Grant 2019C03133, in part by the Alibaba-Zhejiang University Joint Institute of Frontier Technologies, in part by the Research Institute of Cyberspace Governance in Zhejiang University, in part by the National Natural Science Foundation of China under Grant 61571063, and Grant 61772236, in part by the Young Elite Scientists Sponsorship Program by CAST under Grant 17-JCJQ-QT-045, and in part by the Major Scientific Research Project of Zhejiang Lab under Grant 2018FD0ZX01. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jean-Luc Danger. *(Corresponding author: Yajin Zhou.)*

F. Zhang is with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China, with the State Key Laboratory of Cryptology, Beijing 100878, China, with the School of Cyber Science and Technology, College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China, with the Alibaba Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310027, China, and also with the Zhejiang Lab, Hangzhou 310027, China (e-mail: fanzhang@zju.edu.cn).

X. Dong and B. Yang are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: xaiofei_dong@zju.edu.cn; yangbolin@zju.edu.cn).

Y. Zhou and K. Ren are with the School of Cyber Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: yajin_zhou@zju.edu.cn; kuiren@zju.edu.cn).

Digital Object Identifier 10.1109/TIFS.2019.2941774

I. Introduction

Side channel attack (SCA) tends to extract the secret keys using physical leakages such as power, electromagnetic emissions, time, and more. Among them, *power analysis* [1] is one of the most powerful means which exploits the power consumption leaked through cryptographic devices. The instantaneous power consumption is closely related to the executed operations and processed intermediate data, so the secret keys could be extracted by means of analyzing the power features. Differential power analysis (DPA) and correlation power analysis (CPA) [2] are two foremost and fundamental power analysis methods to recover the keys based on statistical methodology. To exploit the dependence of the leaked power characteristics and executed operations as well as the processed data, the target parts of those power measurements have to be aligned to the same position. In order to mitigate SCAs, different countermeasures are proposed to increase the difficulty for adversaries.

The concept of *random delay countermeasure* against SCA was proposed in [3]. By making power traces out of synchronization, random delays reduce the correlation of the consumed power and executed operations along with processed data. Compared with random masking, it improves the attack complexity as a type of hiding manner. The reason why random delay countermeasure is still acknowledged and widely used is that its implementation strategy is quite flexible and easy to adopt. In a nutshell, strategies to implement random delay countermeasures can be divided into three categories.

From the aspect of measurements, it is quite normal that the power traces are triggered and then collected in an oscilloscope controlled by a host computer. If the trigger signal arrives at a random interval, the starting moment of encryptions varies every time, which is called *random propagation delay*.

From the aspect of the algorithm itself, the delays are often realized through dummy operations such as ***NOP*** instructions, which is called *random delays insertion*. At first, a gate-level random delay insertion against DPA was proposed in [4] where the instantaneous power consumption and total charge quantity are randomized. Besides the hardware solution, Tunstall et al. proposed a different method of random delays in embedded software, which increases the desynchronization compared to those whose lengths are uniformly distributed with less time consumption [5]. As an improvement of Benoit and Tunstall's work, Coron et al. proposed effective countermeasures of random delays in CHES 2009 and 2010 [6], [7], which are also implemented at the software level.

Moreover, from the aspect of the device's clock, the *random clock jitters* could introduce random delays to power traces because the instructions being executed are pulsed by the clock frequency. Zafar et al. first introduced a pseudo-random clocking scheme to AES based on single inverter ring oscillators (SIROs) to resist DPA attacks [8]. Based on Zafar's work, Boey et al. proposed a method to insert dummy operations during idling cycles of random clock [9], which reduces the signal-to-noise ratio (SNR) significantly and increases the resistance capability.

A. Related Work

Traditional attacks on random delay countermeasures are divided into two categories. One is to launch an attack in the time domain after aligning the power traces. The most basic method—*static alignment* was first proposed by Mangard et al. and well acknowledged by SCA community [1]. It is easy to perform such an attack. However, it mainly works in the scenario of intrinsic jitter and small delays. Moreover, it cannot fully align the traces when complex random delays are induced or varying clock frequency is applied. After that, a more advanced aligning method—*elastic alignment* was proposed in [10], which matches different parts at different offsets and performs nonlinear resampling of the traces. The elastic alignment has a better performance in terms of alignment. However, its computational complexity is very high. In order to achieve higher efficiency, Muijrers et al. proposed the *rapid alignment method* in [11] whose computational complexity is significantly reduced as compared to elastic alignment. In addition, some other alignment methods based on waveform matching, pattern recognition, and hidden Markov models were proposed in [12], [13] and [14]. However, the operation of alignment itself still experiences a great deterioration in performance and efficiency.

The other category is to transform the power signals from the time domain to frequency domain, and to conduct the attack directly on the frequency spectrum. Since the shifts in time domain will only change the phase spectrum in frequency domain and the amplitude spectrum will not be influenced, differential power frequency analysis (DPFA) and correlation power frequency analysis (CPFA) based on power spectral density were first put forward in [15] and [16]. However, Lu et al. showed that the frequency attack could not succeed when the maximum window size of Discrete Fourier Transform ($\mathbb{DFT}$) is smaller than the length of delays [17]. So it is very important to choose an appropriate window size for $\mathbb{DFT}$ in such attacks.

Over the past few years, wavelet related techniques have already been adopted to SCA in practice. For example, it was investigated that wavelet coefficients in different levels contribute to DPA differently [18]. It could improve the performance of DPA if those detrimental coefficients are removed. Whereas the most applications of wavelet analysis in SCA is to denoise the power signals. For example, Liu et al. and Ai et al. respectively proposed a method to reduce the noise of power traces in order to improve the signal-to-noise ratio [19], [20]. In addition, wavelet analysis was applied to denoise the misaligned signals in [21]. However, a minimization algorithm rather than wavelet ones was employed to align them, and a classic DPA was performed to recover the keys at the end. Furthermore, the idea of recovering the keys based on wavelet coefficients was proposed in [22] for the first time. However, it only discussed the situations without countermeasures. Recently, the machine learning technique has been applied to SCA based on wavelet transform. For instance, the power traces were preprocessed using wavelet transform and then used to train the probabilistic neural network (PNN) [23]. Besides, wavelet analysis and support vector machine (SVM) algorithm were combined and wavelet SVM was used to recover the keys of unmasked or masked AES implementations [24]. So the wavelet analysis becomes more and more important in SCA, making the proposal of a unified wavelet-based attack framework in random delay scenarios naturally valuable.

B. Contribution

Our contributions can be summarized as below:

- We propose a unified wavelet-based attack framework (WAF) against random delay countermeasures, including three components: wavelet trace denoising (WTD), wavelet trace alignment (WTA) and wavelet key extraction (WKE). Due to the lack of WTA, there is no such an all-in-one framework that has even been proposed, to the best of our knowledge. In this framework, we particularly propose a novel alignment method using wavelet decomposition, which can detect and remove the random delays, efficiently improving the attack against random delay countermeasures.
- We evaluate each component of this framework individually and systematically, and compare it with traditional methods in confrontation of three types of random delay countermeasures. As shown in experiments, each component has its own contribution to improve the attack, respectively, which is evaluated in a quantitative manner.
- Through physical experiments on a microcontroller and FPGA, we evaluate the performance and efficiency of the overall wavelet framework. Our experimental analysis proves that the wavelet-based attack framework results in better attacks in comparison to those classic methods and frequency-based ones, when the wavelet technique is applied throughout the entire framework.

C. Organization

This paper is organized as follows: Section II gives the necessary background of wavelet analysis. Section III introduces three strategies of random delay countermeasures and explains the attack difficulty of traditional methods. Section IV describes the whole wavelet-based attack framework—WAF, including our proposed method of alignment—WTA. Section V demonstrates our systematic evaluation of the wavelet framework. Section VI concludes the paper and lists our future work.

II. Background of Wavelet Analysis

Wavelet analysis is a more advanced tool than Fourier analysis that only represents a signal in frequency domain with a constant resolution. Instead, wavelet analysis can provide both time and frequency analysis with variant resolution to a signal. It describes the similarity of a time domain signal $f(t)$ using a wavelet basis function ψ with two parameters: scaling s and shift l. Eq.(1) shows an example of wavelet basis function (or wavelet function in short). Eq.(2) lists the wavelet transform ($\mathbb{WT}$) which is integrated over the product of target function $f(t)$ and wavelet basis function ψ. Different wavelet basis functions can be classified to different families according to their properties and characteristics, such as Daubechies (DbN), Mexican Hat (Mexh), Morlet (Morl), Meyer (Meyr), Symmlets (SymN) and Coiflets (CoifN) [25]. The "N" in DbN, SymN and CoifN indicates the order of wavelet basis functions. For example, the Db1 (also called as Haar wavelet) has the order of one.

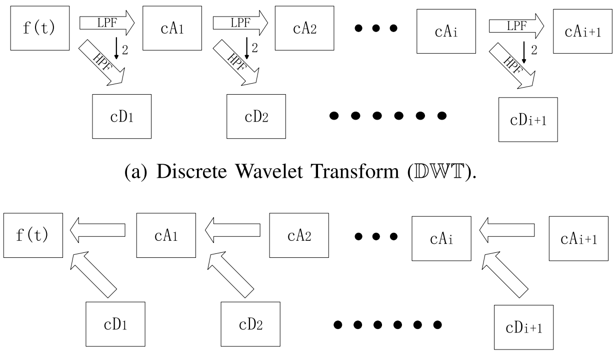
$$\psi_{l,s}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-l}{s}) \tag{1}$$

$$\mathbb{WT}(s,l) = \frac{1}{\sqrt{s}}\int_{-\infty}^{+\infty} f(t) * \psi(\frac{t-l}{s})dt \tag{2}$$

The selection of wavelet basis function in wavelet decomposition is very crucial to the performance and results of signal processing. There are two key factors that should be considered when choosing different mother wavelets. One is those properties of wavelet families, such as compact support, orthogonality, biorthogonality, symmetry and vanishing moment. Among them, the compact support and the vanishing moment are the most important ones to be taken into account [26]. The other is the similarity between the wavelet basis function and the original signal to be analyzed [27]. Namely, the wavelet basis function is more suitable when it is more similar to the original signal.

There are two types of wavelet transforms: Continuous Wavelet Transform ($\mathbb{CWT}$) and Discrete Wavelet Transform ($\mathbb{DWT}$). Since the power traces are discrete samples collected by digital oscilloscopes, $\mathbb{DWT}$ is mainly used as the tool to process power traces and it is also called as wavelet decomposition. It should be noted that in $\mathbb{DWT}$, the scaling s and shift l of $\mathbb{CWT}$ are discrete, but the time t of $f(t)$ and $\psi(t)$ is still continuous.

As Fig. 1(a) shows, the wavelet decomposition of a signal consists of two processes: filtering and down-sampling. The signal $f(t)$ is initially filtered by a low pass filter (LPF) and a high pass filter (HPF). Then in order to remove the redundant information, each filtered signal is down-sampled by two. Therefore, the time signal is transformed into two parts: the detail wavelet coefficients (cD_i) and the approximation wavelet coefficients (cA_i), where i is the decomposition level. Note that cA_i can be further decomposed at the $(i+1)$ level, denoted as cD_{i+1} and cA_{i+1}. If a three-level decomposition is applied to a signal, cD_1, cD_2, cD_3, cA_3 are obtained, which can represent the whole information of this original signal. The approximation wavelet coefficients cA_i generally represent patterns and pivotal information of the signal, and they are critical to SCA. While, cD_i generally represent the noise and irrelevant information.



(a) Discrete Wavelet Transform ($\mathbb{DWT}$).

(b) Inverse Discrete Wavelet Transform ($\mathbb{IDWT}$).

Fig. 1. The schematic description and illustration of $\mathbb{DWT}$ and $\mathbb{IDWT}$.

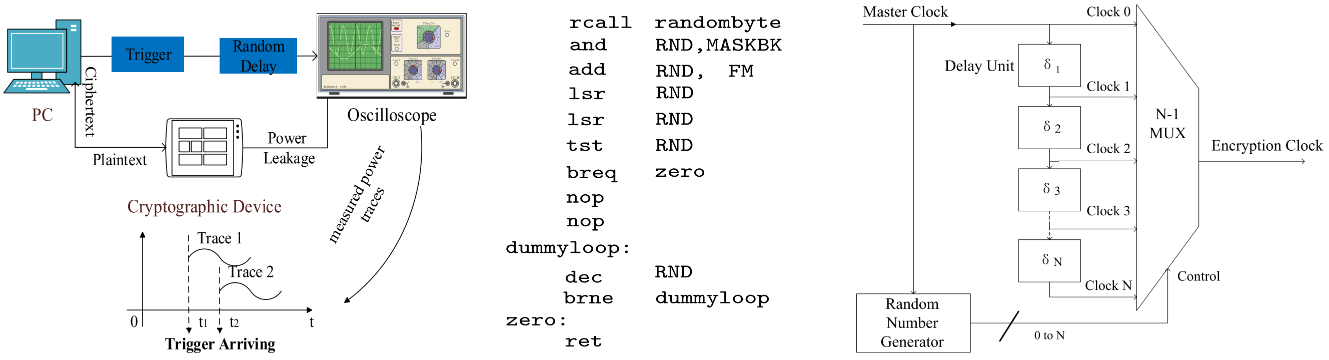
Conversely, wavelet coefficients cA_i and cD_i can be reconstructed to time domain signals A_i and D_i using Inverse Discrete Wavelet Transform ($\mathbb{IDWT}$), indicated by Fig. 1(b). For example, cD_1, cD_2, cD_3, cA_3 can be reconstructed to D_1, D_2, D_3, A_3, respectively. And the time domain signal $f(t)$ is reconstructed as $f(t) = D_1 + D_2 + D_3 + A_3$.

III. Random Delay Countermeasures

Different strategies of random delays will influence the performance of countermeasures against SCA. Therefore, it is important to select appropriate strategies to ensure the effectiveness of defense. In this section, three types of strategies are applied including random propagation delays of trigger signals, multiple random delay insertions to encryption operations and random jitters of clock frequency. The high-level description of implementation principles is shown in Fig. 2. Even though the methods of implementations are different, the objective of these countermeasures is essentially all about how to desynchronize the collected power traces so as to increase the attack difficulty.

A. Strategies of Random Delay Countermeasures

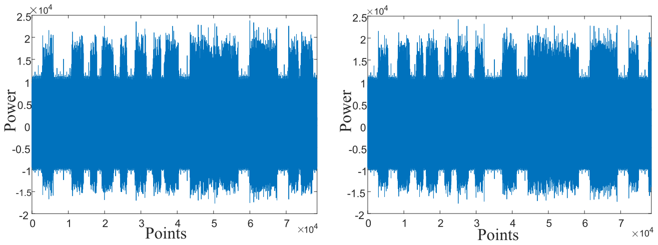
1) Strategy 1: Random Propagation Delay: The encryption is started by a trigger signal generated from a host computer. The trace is measured only when the oscilloscope receives this trigger signal. Therefore a random delay is added to the trigger signal to make the time of propagation vary randomly. For each encryption, the starting time is randomly changed every time, so the collected power traces are misaligned, which is equivalent to a random delay inserted into the beginning of encryption. Fig. 2(a) is the illustration of random propagation delay. Trace 1 and 2 are two schematic traces that are asynchronous due to the difference of the arriving time of trigger signals. The propagation delays are t_1 and t_2, respectively, which makes the positions of encryption operations different from each other when $t_1 \neq t_2$. This kind of countermeasure is liable to put into practice even if there is no knowledge about the details of the cryptographic algorithm. The adversary has to align the power traces first. Therefore, this countermeasure is selected as a target of our attack framework to be proposed.



```
      rcall  randombyte
      and    RND,MASKBK
      add    RND,  FM
      lsr    RND
      lsr    RND
      tst    RND
      breq   zero
      nop
      nop
dummyloop:
      dec    RND
      brne   dummyloop
zero:
      ret
```

(a) Random propagation delay. (b) Random delays insertion. (c) Random clock jitters.

Fig. 2. High-level illustrations and sketches of mechanism for three strategies of random delay countermeasures used for targets of our proposed framework.



(a) Trace 1 of original signals. (b) Trace 2 of original signals.

Fig. 3. Two illustrative traces with random delays insertion.

2) *Strategy 2: Random Delays Insertion:* In CHES 2009, a new method of constructing and inserting random delays called Float Mean (FM) was proposed in [6] which is more secure and lightweight as claimed. With the same mean, it can generate a much greater variance. However, it was pointed out in [7] that the chosen parameters of Float Mean were inappropriate and thus the Improved Float Mean scheme was proposed, which is also adopted in our experiments. Random delays are generated according to those assembly codes given by [7], which is shown in Fig. 2(b) to illustrate the core idea of the implementation of random delays. Considering the trade-off between the performance and overhead merely for the purpose of illustration, 4 random delays are inserted into the 16 S-Box lookups in the first round of AES cipher, which is usually a typical focus of SCA on AES. Fig. 3 shows two illustrative power traces inserted with multiple random delays. It is observed that 16 lookups are separated randomly due to those delays and the operations in these two encryptions are not synchronous.

3) *Strategy 3: Random Clock Jitters:* Generally, random clock is implemented in hardware circuits such as FPGA due to its flexibility of clock systems while the implementation in the embedded devices is relatively difficult because of their immutable clock frequency. Therefore, the third strategy—random clock jitters, which is a kind of random delay in essence, is implemented in an FPGA. Fig. 2(c) interprets the core idea of random clock which is based on a multiplexer and multiple delay units, where Clock 0 to N are $N + 1$ possible input clocks with 0 to N delay cycles. A master clock goes through N random delay modules and then one of the input clocks is selected as the encryption clock controlled by a random number generator (RNG). If the random number is N, the clock will be postponed N clock cycles. The AES encryption implemented in FPGA requires ten clock cycles because there are ten iterations of round and only one round is executed within one clock cycle. Hence ten random clocks are totally imported to complete an AES encryption and each round of AES is executed with different clock frequency. As a result, the execution time of the ten rounds are randomly changed and the collected power traces are asynchronous in the time domain. Usually, preprocessing of alignment and frequency attack are also employed to react to the random clock countermeasure. So this strategy of random delay is also chosen as a target of our proposed wavelet-based attack framework.

TABLE I
COSTS OF COUNTERMEASURES IN SOFTWARE

Countermeasures	Code Size(KB)	Running Time(ms)
No countermeasure	18.2	117.3
Strategy 1	18.6	324.1
Strategy 2	20.8	274.2

B. Implementation Costs of Random Delay Countermeasures

During the implementations of random delay countermeasures, the resistance against SCA is not free and comes with the additional costs. Here the overheads to implement above three strategies are discussed. Table I and Table II respectively show the costs of implementations in software and hardware. As a baseline, the AES implementation without any countermeasure is also listed.

Strategy 1 and Strategy 2 are implemented in a microprocessor at the software level. The second column in Table I lists the total size of codes (in KB) written to the microprocessor. The third column is the average time (in ms) of each execution of AES which includes the time of communication with the host PC. Since the random delays are inserted, the code size is larger and the running time is increased to 2∼3 times. However, those encryptions are still working properly.

TABLE II
COSTS OF COUNTERMEASURE IN HARDWARE

Countermeasures	#. of Slice Registers	#. of Slice LUTs
No countermeasure	748 (2.6%)	2206 (7.7%)
Strategy 3	746 (2.6%)	2280 (7.9%)

Strategy 3 is implemented in an FPGA. The second column in Table II lists the number of the slices used as Flip-Flops after synthesis. The third column shows the number of look-up tables in the FPGA used as RAM or logics. Note that the FPGA in use has 28800 sliced registers and LUTs in total. These implementations (with/without Strategy 3) just use a small portion of the available hardware resource, roughly around 2% of total registers and 7% of LUTs.

C. The Difficulty of Attacking on Random Delays

It is not easy to launch a direct DPA or CPA attack on those power traces when the three aforementioned strategies are applied as countermeasures. This is because the instantaneous power consumption of the same moment is not caused by the same encryption operation and it is hard to analyze the correlation. Even though in theory, random delay countermeasures cannot prevent the attacks completely. In practice, it can cause great difficulties in key extraction. More specifically, it requires more coverage of points of interest, more pattern recognition of target operations, and more alignments to compensate the side-effects of delays.

Trace alignment and frequency attack are normally considered as an efficient response to random delay countermeasures. Unfortunately, there are some obvious disadvantages when using them.

As for standard alignments methods such as static or elastic alignment, it is difficult to align all the traces completely if the random delays are complex. So the correlation between power dissipation and sensitive data may be low and the key recovery is still difficult. In terms of multiple delays, multiple alignments have to be conducted, each of which has to go through all traces. In addition, the process of alignment itself is very time-consuming, so the attack efficiency is quite low.

As for frequency attack, DPFA could not succeed when the window size of $\mathbb{DFT}$ is smaller than the length of delays as claimed in [17]. Moreover, frequency analysis completely discards the time-domain information which can actually be well exploited in the traditional side-channel analysis. For instance, in DPFA, adversaries cannot know the precise moment when the target operation is executed. As declared in [1], whether frequency attack works well or not essentially depends on the spectral characteristics of the leakage and random delays.

Due to the listed difficulties of those mainstream analysis methods on random delay countermeasures, it is interesting and worth to explore a new type of analysis, which could enhance the attack performance and efficiency especially towards those random delay strategies.

IV. Wavelet-Based Attack Framework Against Random Delay Countermeasures

This section elaborates our unified wavelet-based attack framework with three components in the scenario of random delay countermeasure. Meanwhile, the standard attack procedures and the frequency-based attack procedures are also briefly introduced to compare with the proposed framework.

A. Overview of Wavelet-Based Attack Framework

There are two existing attack procedures against random delay countermeasures: standard attack procedure (SAP) [1] and frequency-based attack procedure (FAP) [16]. A typical SAP consists of the following steps: standard traces denoising by a low pass filter (STD), standard traces alignment via static or elastic alignment (STA) and standard keys extraction in the time domain (SKE). Meanwhile, FAP is often applied to break the protection of random delays, which includes the noise reduction using a low pass filter and the key recovery using DPA or CPA in the frequency domain (FKE).

Wavelet analysis has been applied to SCA in different aspects. However, when facing random delay countermeasures, there is no such a uniform solution that is merely based on wavelet technique. The main reason is that how to use wavelet to align power traces has not been explored. In this paper, the problem is carefully solved, which is to be shown in Section IV-C. Therefore, we are able to propose an attack based on a unified wavelet framework *for the first time*, to the best of our knowledge.

The wavelet-based attack framework (WAF) includes *wavelet trace denoising* (WTD), *wavelet trace alignment* (WTA), and *wavelet key extraction* (WKE), which correspond to those three steps in SAP. Later in Section V, the performance and efficiency of the proposed WAF against those three strategies of random delay countermeasures are evaluated and compared with SAP and FAP. The wavelet-based attack framework is elaborated as following.

Step 1 **Wavelet trace denoising** (WTD): Denoising power traces based on wavelet analysis. Different from traditional denoising with low pass filter, WTD can not only remove those high frequency noise and glitches influenced by measurement setup and environment, but also eliminate massive information which are unrelated to encryption operations.

Step 2 **Wavelet trace alignment** (WTA): Aligning power traces based on wavelet analysis. Through selecting appropriate wavelet basis functions and decomposition levels, and applying a carefully chosen threshold, random delays can be first distinguished from encryptions based on pattern recognition using wavelet decomposition and then removed. Thereby, the aligned wavelet coefficients can be exploited directly.

Step 3 **Wavelet key extraction** (WKE): Recovering the secret keys in wavelet domain. Extracting the keys with DPA or CPA is implemented in wavelet domain instead of time domain when applying the wavelet framework. There is no need to reconstruct the power traces since

Algorithm 1 The Algorithm of the Whole Wavelet-Based Attack Framework (WAF), Compared With SAP and FAP

	WAF: Wavelet-based Attack Framework	SAP: Standard Attack Procedures	FAP: Frequency Attack Procedures
1	$\mathcal{S} = \mathcal{S}_{ep}(t) + \mathcal{S}_{dp}(t) + \mathcal{S}_{np}(t)$	$\mathcal{S} = \mathcal{S}_{ep}(t) + \mathcal{S}_{dp}(t) + \mathcal{S}_{np}(t)$	$\mathcal{S} = \mathcal{S}_{ep}(t) + \mathcal{S}_{dp}(t) + \mathcal{S}_{np}(t)$
2	$\mathbb{DWT}_{\mathtt{WTD}}(n_{\mathtt{WTD}}, l_{\mathtt{WTD}}) : \mathcal{S} \longrightarrow \mathcal{W}$		
3	$\mathcal{W} = \mathcal{W}_{ep}(w) + \mathcal{W}_{dp}(w) + \mathcal{W}_{np}(w)$		
4	$\mathsf{WTD}(\lambda_{\mathtt{WTD}}) : \mathcal{W} \longrightarrow \mathcal{W}^1$	$\mathsf{STD} : \mathcal{S} \longrightarrow \mathcal{S}^1$	$\mathsf{STD} : \mathcal{S} \longrightarrow \mathcal{S}^1$
5	$\mathbb{IDWT}_{\mathtt{WTD}}(n_{\mathtt{WTD}}, l_{\mathtt{WTD}}) : \mathcal{W}^1 \longrightarrow \mathcal{S}^1$		
6	$\mathcal{S}^1 = \mathcal{S}^1_{ep}(t) + \mathcal{S}^1_{dp}(t)$	$\mathcal{S}^1 = \mathcal{S}_{ep}(t) + \mathcal{S}_{dp}(t)$	$\mathcal{S}^1 = \mathcal{S}_{ep}(t) + \mathcal{S}_{dp}(t)$
7	$\mathbb{DWT}_{\mathtt{WTA}}(n_{\mathtt{WTA}}, l_{\mathtt{WTA}}) : \mathcal{S}^1 \longrightarrow \mathcal{W}^2$		$\mathbb{DFT} : \mathcal{S}^1 \longrightarrow \mathcal{F}$
8	$\mathcal{W}^2 = \mathcal{W}^2_{ep}(w) + \mathcal{W}^2_{dp}(w)$		$\mathcal{F} = \mathcal{F}_{ep}(f) + \mathcal{F}_{dp}(f)$
9	$\mathsf{WTA}(\lambda_{\mathtt{WTA}}) : \mathcal{W}^2 \longrightarrow \mathcal{W}^{2*}$	$\mathsf{STA} : \mathcal{S}^1 \longrightarrow \mathcal{S}^2$	
10	$\mathcal{W}^{2*} = \mathcal{W}^2_{ep}(w)$	$\mathcal{S}^2 = \mathcal{S}_{ep}(t)$	
11	~~$\mathbb{IDWT}_{\mathtt{WTA}}(n_{\mathtt{WTA}}, l_{\mathtt{WTA}}) : \mathcal{W}^{2*} \longrightarrow \mathcal{S}^2$~~		
12	$\mathsf{WKE} : \mathcal{W}^{2*} \longrightarrow \mathcal{K}$	$\mathsf{SKE} : \mathcal{S}^2 \longrightarrow \mathcal{K}$	$\mathsf{FKE} : \mathcal{F} \longrightarrow \mathcal{K}$

TABLE III
NOTATIONS OF WAF, SAP AND FAP

Notation	Description
WTD	Step 1 of WAF: the wavelet trace denoising
WTA	Step 2 of WAF: the wavelet trace alignment
WKE	Step 3 of WAF: the wavelet key extraction
$\mathbb{DWT}_{\mathtt{WTD}}, \mathbb{DWT}_{\mathtt{WTA}}$	Discrete Wavelet Transform in WTD and WTA
$\mathbb{IDWT}_{\mathtt{WTD}}, \mathbb{IDWT}_{\mathtt{WTA}}$	Inverse $\mathbb{DWT}$ in WTD and WTA
$\mathbb{DFT}$	Discrete Fourier Transform
$\mathcal{S}$	Original signal with countermeasures
$\mathcal{W}$	Wavelet-domain signal of $\mathcal{S}$
$\mathcal{S}^1$	Denoised time-domain signal
$\mathcal{W}^1$	Denoised wavelet-domain signal after WTD
$\mathcal{S}^2$	Aligned time-domain signal after WTA or STA
$\mathcal{W}^2$	Wavelet-domain signal of $\mathcal{S}^1$ via $\mathbb{DWT}_{\mathtt{WTA}}$
$\mathcal{W}^{2*}$	Aligned wavelet-domain signal
$\mathcal{S}_{ep}, \mathcal{S}_{dp}, \mathcal{S}_{np}$	Encryption, delay and noise part of $\mathcal{S}$
$\mathcal{W}_{ep}, \mathcal{W}_{dp}, \mathcal{W}_{np}$	Encryption, delay and noise part of $\mathcal{W}$
$n_{\mathtt{WTD}}, n_{\mathtt{WTA}}$	Order of wavelet basis function in WTD,WTA
$l_{\mathtt{WTD}}, l_{\mathtt{WTA}}$	Decomposition levels of $\mathbb{DWT}$ in WTD,WTA
$\lambda_{\mathtt{WTD}}$	Threshold to remove the noise part in WTD
$\lambda_{\mathtt{WTA}}$	Threshold to remove the delay part in WTA
$\mathcal{F}$	Frequency-domain signal of $\mathcal{S}$ through $\mathbb{DFT}$
$\mathcal{K}$	Secret key

the wavelet coefficients obtained from Step 2 can be directly used for WKE.

The details of the proposed WAF are depicted in the first column of Algorithm 1 at a high level, where the notations are summarized in Table III.

The original signal $\mathcal{S}$ contains three parts in the time domain: the encryption part $\mathcal{S}_{ep}$, the random delay part $\mathcal{S}_{dp}$ and the noise part $\mathcal{S}_{np}$. In Step 1 (i.e., WTD), $\mathcal{S}$ is transformed into the signal $\mathcal{W}$ in wavelet domain through $\mathbb{DWT}_{\mathtt{WTD}}$ with two pivotal parameters—$n_{\mathtt{WTD}}$ and $l_{\mathtt{WTD}}$, which stand for the order of the wavelet basis function and the wavelet decomposition level. A threshold $\lambda_{\mathtt{WTD}}$ is applied to eliminate the noise part $\mathcal{W}_{np}$ in wavelet domain, and to obtain the denoised signal $\mathcal{W}^1$ after WTD. Then $\mathcal{W}^1$ can be reconstructed to the denoised signal $\mathcal{S}^1$ in the time domain through $\mathbb{IDWT}$. In Step 2 (i.e., WTA), the denoised signal $\mathcal{S}^1$ is transformed into $\mathcal{W}^2$ after $\mathbb{DWT}_{\mathtt{WTA}}$. Similar to $n_{\mathtt{WTD}}$ and $l_{\mathtt{WTD}}$, $n_{\mathtt{WTA}}$ and $l_{\mathtt{WTA}}$ denote the order of wavelet basis function and the decomposition level used in WTA. Next, another threshold $\lambda_{\mathtt{WTA}}$ is applied to remove the random delay part $\mathcal{W}^2_{dp}$ and get the aligned signal $\mathcal{W}^{2*}$ in wavelet domain. Finally, in Step 3 (i.e., WKE), the secret key $\mathcal{K}$ can be recovered from the aligned wavelet signal $\mathcal{W}^{2*}$ directly.

As a comparison, SAP, listed in the second column of Algorithm 1, recovers the keys from the aligned time-domain signal $\mathcal{S}^2$. Besides, FAP is also introduced in the last column of Algorithm 1, which does not require the trace alignment compared with SAP, and FKE is just executed based on frequency-domain signal $\mathcal{F}$.

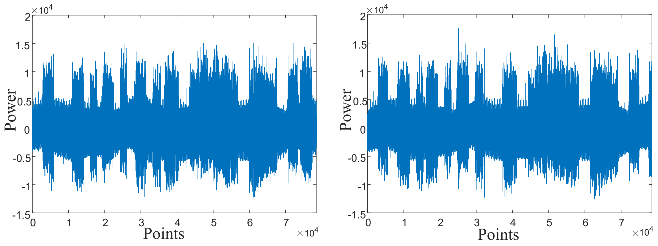
Different from SKE and FKE, there is no need for WKE to reconstruct the time-domain signal $\mathcal{S}^2$ from $\mathcal{W}^{2*}$ via $\mathbb{IDWT}$ for a second time. Therefore, Line 11 is crossed out in the proposed WAF as shown in Algorithm 1.

B. Wavelet-Based Trace Denoising

Noise in the collected power traces will hide those sensitive leakages and reduce SNR. Hence it is very important to remove the noise as much as possible to improve the performance of DPA or CPA. Wavelet analysis is a powerful tool to partition various parts of a signal (the noise and the key-dependent part). As described in Section II, a signal $\mathcal{S}$ with noise can be decomposed to series of wavelet coefficients (cA_i and cD_i) through $\mathbb{DWT}$. Parts of cD_i, which cause negative effects to SCA, should be identified and removed.

The first problem to be solved in WTD is the selection of a suitable wavelet family along with $n_{\mathtt{WTD}}$ and $l_{\mathtt{WTD}}$. In fact, the wavelet technique used for denoising has already been investigated and verified in [18]–[22], [28]. Therefore the parameter selections in WTD can be referred to previous works. First of all, the Daubechies (DbN) is selected as the wavelet family, because it is one of the most commonly used mother wavelets in denoising. Among the DbN family, Db4 is chosen as the specific wavelet basis function of $\mathbb{DWT}_{\mathtt{WTD}}$, which is adopted as empirical experience from [28]. That means $n_{\mathtt{WTD}} = 4$ in Algorithm 1. In addition, since only those coefficients less than 6 levels are required to carry out DPA successfully [18], $l_{\mathtt{WTD}}$ in Algorithm 1 is set as 6 empirically.

After $\mathbb{DWT}$, the subsequent work is to find out those coefficients caused by noise and then remove them. Due to



(a) Trace 1 of denoised signals. (b) Trace 2 of denoised signals.

Fig. 4. Two denoised power traces after WTD.

the fact that the coefficients of normal signals are distinctly larger than those of noise, the most well-known method to filter the noise in the wavelet domain is based on a *threshold*. The main idea is that: each detail coefficient is compared to a threshold $\lambda_{\mathtt{WTD}}$; if it is less than $\lambda_{\mathtt{WTD}}$, the coefficient is considered as noise-related and should be set to zero. Here the threshold level of denoising is determined according to the most common method proposed by Donoho [29]. Therefore $\lambda_{\mathtt{WTD}}$ is resolved by Donoho's formula shown in Eq.(3) [30]. δ is the variance of noise which is calculated from the median of the absolute value of detail coefficients cD_i, and len is the length of those cD_i.

$$\lambda_{\mathtt{WTD}} = \delta\sqrt{2\log(len)}, \quad \delta = \frac{median|cD_i|}{0.6745} \tag{3}$$

In fact, a threshold function $\mathtt{TF}$ is usually defined on the threshold $\lambda_{\mathtt{WTD}}$ and usually of two types. One is the hard threshold function $\mathtt{TF}_h$ shown in Eq.(4), where the coefficients less than $\lambda_{\mathtt{WTD}}$ are set to zero and the rest are kept constant. The other is the soft threshold function $\mathtt{TF}_s$ whose coefficients less than $\lambda_{\mathtt{WTD}}$ are also set to zero. But the rest are recomputed to improve the accuracy as shown in Eq.(5).

$$\mathtt{TF}_h(c) = \begin{cases} c & if\ |c| > \lambda_{\mathtt{WTD}} \\ 0 & if\ |c| < \lambda_{\mathtt{WTD}} \end{cases} \tag{4}$$

$$\mathtt{TF}_s(c) = \begin{cases} c\ (1 - \dfrac{\lambda_{\mathtt{WTD}}}{|c|}) & if\ |c| > \lambda_{\mathtt{WTD}} \\ 0 & if\ |c| < \lambda_{\mathtt{WTD}} \end{cases} \tag{5}$$

In this way, the original signal $\mathcal{S}$ is decomposed to 6 levels of wavelet coefficients $\mathcal{W}$, resulting in $cD_1, cD_2, \ldots, cD_6$ and cA_6. Then all the detail coefficients are filtered via the threshold function $\mathtt{TF}_h$ or $\mathtt{TF}_s$. A new wavelet-domain signal $\mathcal{W}^1$ is generated, where the noise part has been removed. However, $\mathcal{W}^1$ can not be directly fed to Step 2 as the parameters of wavelet decomposition in WTD are different from those in WTA. So, $\mathcal{W}^1$ needs to be transformed back to a time-domain signal $\mathcal{S}^1$ through $\mathbb{IDWT}$.

This step is very important in the whole WAF. Unlike the general filters which only cut down those high frequency noises, WTD can also remove extra information that is unrelated to power analysis attack. To clarify this, Fig. 4 shows two denoised power traces after Step 1, where the original signals are protected by Strategy 2. Compared with those in Fig. 3, the traces in Fig. 4 consist of fewer glitches and noise in the time domain.

TABLE IV
Properties of Six Wavelet Families

Properties	Wavelet families					
	Meyr	Mexh	Morl	DbN	CoifN	SymN
Compact support	no	no	no	yes	yes	yes
Orthogonality	yes	no	no	yes	yes	yes
Biorthogonality	yes	no	no	yes	yes	yes
Symmetry	yes	yes	yes	no	near	near
Vanishing moment	no	no	no	N	2N-1	N

C. Wavelet-Based Traces Alignment

Wavelet analysis can identify various characteristics of signals based on different wavelet decomposition levels and wavelet basis functions. In this paper, three strategies of random delay countermeasures stay on a fact. That is, there exist non-negligible differences in amplitude when random delays are introduced to encryptions. For example, Strategy 2 is adopted from papers in CHES [6], [7], and the direct measurement and observation on their schemes reveal such a difference. Thereby those amplitude features are mainly targeted in wavelet-based alignment. A threshold in WTA, denoted as $\lambda_{\mathtt{WTA}}$, is sought to distinguish the features of delays from encryptions, and to remove random delays. Note that such fact is reproduced from the original CHES paper [6], [7] under the same experimental settings. Since the major focus of our work is on how to use wavelet technique to recognize those delays and align power traces, no hiding techniques are further utilized to minimize such difference in [6], [7], which can be studied as a more challenging work in the future.

1) Selection of Wavelet Basis Function: In WTD, which wavelet basis function is selected is determined according to the reference that is ready verified in previous works. Unlike WTD, there is no existing work in WTA that can be referred as empirical experience. Therefore, detailed mathematical and theoretical proof is provided here, showing a concrete and reasonable logic to select appropriate wavelet basis function for WTA and to pursue a better performance of wavelet-based alignment. Here six commonly used wavelet families are considered as candidates of wavelet decomposition in WTA: Meyr, Mexh, Morl, DbN, CoifN and SymN.

The first factor to be balanced when selecting appropriate wavelet basis function is the properties which directly determine the results of wavelet decomposition and performance of WTA. Table IV lists those dominant properties, including compact support, orthogonality, biorthogonality, symmetry and vanishing moment. Among them, the vanishing moment (VM) is an important metric, which can depict the smoothness and concentration of the wavelet function in both time and frequency domain. In general, the value of VM is preferred to be large, because a larger value of VM indicates a less number of coefficients for the inner product [31]. In Table IV, the first three columns in the first and last rows have the value of "no". Since there are no compact support and vanishing moment in the families of Meyr, Mexh and Morl, these three wavelet families are excluded from further consideration. Then, in order to keep a larger VM, we only consider a part of basis functions in the remaining families, i.e., Db7, Db8, Db9, Db10, Sym7, Sym8, Sym9, Sym10 and Coif5.

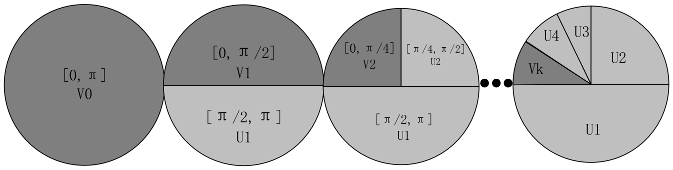


Fig. 5. Multi-resolutions analysis with different wavelet decomposition levels.

The selection of wavelet basis function is further enhanced according to the similarity between the wavelet basis functions and signals to be decomposed. The Mean-Square Error (MSE) is a sound metric that reflects the degree of similarity between the reference and target to be estimated [32]. The equation to compute MSE is presented in Eq.(6), where n is the number of samples, y_i and $\hat{y}_i$ are the samples of reference and target signals, respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{6}$$

Therefore, the denoised signal $\mathcal{S}^1$ is decomposed using those remaining nine wavelet basis functions in Table IV. And then they are used to reconstruct the signal $\hat{y}(t)$ based on the decomposition coefficients. Finally, the MSE between $\hat{y}(t)$ and reference signal $\mathcal{S}^1$ is computed. To make sure the robustness, this process is repeated ten times with different reference signals $\mathcal{S}^1$, and the results are shown in Table V. The MSE accurately depicts the similarity of two signals. A lower MSE corresponds to a higher similarity. Thus, the final selection of wavelet basis function can be determined by the MSE. In Table V, the MSE based on Db9 leads to the smallest value in all ten tests. Therefore, the Db9 is considered as the best selection to carry out WTA and it is selected as the wavelet basis function in the wavelet decomposition. Accordingly, $n_{\mathtt{WTA}}$ in Algorithm 1, the order of wavelet basis function in WTA, is determined as 9.

2) Selection of Wavelet Decomposition Levels: In order to differentiate random delays and encryptions as clearly as possible, the decomposition levels should also be appropriately selected in addition to the wavelet basis function.

The decomposition level is related to the resolution level, which indicates the signals in different frequency bands [33]. Supposing the frequency space of original signal is defined as V_0 over the interval $[0, \pi]$. After a decomposition, V_0 is divided into two subintervals: a low frequency subspace V_1 over $[0, \frac{\pi}{2}]$ and a high one U_1 over $[\frac{\pi}{2}, \pi]$. Then, V_1 continues to be partitioned into two halves: V_2 and U_2. Finally, a series of $V_k, U_k, \ldots, U_2, U_1$ are obtained if k decomposition levels are applied, as shown in Fig. 5. As a result, a signal after multiple wavelet decomposition with different levels will generate wavelet coefficients at different resolution levels, which makes multi-resolutions analysis possible. That is, with more decomposition levels, the frequency-domain resolutions will increase and time-domain resolutions will decrease conversely. And the number of points in decomposed signals will be reduced. Therefore, a suitable level selection matters a lot to the performance of wavelet-based alignment.

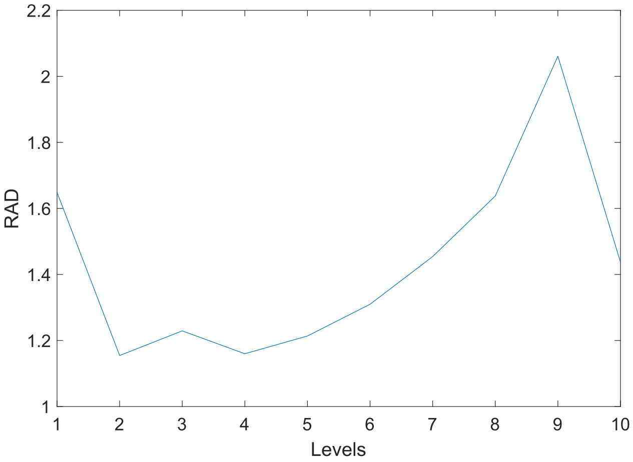


Fig. 6. The RAD values based on different decomposition levels from 1 to 10.

We first determine the optimal level from a mathematical and theoretical perspective. According to the wavelet theory, the approximation coefficients mainly contain the low-frequency part of original signal such as amplitude characteristics while the detail coefficients contain those high-frequency part such as glitches and disturbances. Note that both approximation and detail coefficients will contain the signal as well as the delay. However, the ratio of signal to delay will depend on wavelet transform and decomposition levels. In practice, since the adversary is not clear about which part is the delay, he has to do multiple wavelet decompositions which will suppress the delay in the trace of approximation coefficients and only keep the signal-related part. That's the way how the approximation coefficients get mapped to the encryption part eventually. Therefore, the contrast Ratio of Approximation to Detail coefficients (RAD) is defined to depict the degree of divergence between the random delays and encryptions. Eq.(7) shows how to compute RAD with normalization, where cA_i and cD_i are the approximation and detail coefficients at the i-th level. A larger value of RAD indicates a more significant contrast of encryptions over delays. So the newly introduced RAD can be considered as a metric to select the decomposition level.

$$RAD = \frac{\sum\left|\frac{cA_i - mean(cA_i)}{max(cA_i) - min(cA_i)}\right|}{\sum\left|\frac{cD_i - mean(cD_i)}{max(cD_i) - min(cD_i)}\right|} \tag{7}$$

Fig.6 shows the RAD for 10 levels where Db9 is served as the basis function. In Fig.6, the value of RAD reaches to the maximum when the level of decomposition increases to 9. However, it starts to decrease when the level goes beyond 9. Therefore 9 levels are considered to be the best selection in WTA, and the level of wavelet decomposition—$l_{\mathtt{WTA}}$ is determined as 9.

The effectiveness of newly-defined RAD can be interpreted and verified through practical experiments. Fig. 7 shows the traces of approximation coefficients which corresponds to 10 decomposition levels. We can see that the contrast of random delays to encryptions is not very obvious when the decomposition level is less than 9, where some high frequency

TABLE V

MSE COMPARISONS OF CANDIDATE WAVELET BASIS FUNCTIONS (UNIT: $*10^{-8}$)

Signals	MSE								
	Db7	Db8	Db9	Db10	Sym7	Sym8	Sym9	Sym10	Coif5
1	4.3732	4.0631	3.9577	4.5001	6.2840	7.7625	9.9742	9.0496	9.3176
2	4.2782	3.9088	3.7492	4.3677	6.0574	7.4081	9.5961	8.8433	8.9394
3	4.3111	3.8495	3.8088	4.4596	6.0015	7.5912	9.6762	8.7236	8.9387
4	4.1983	3.8197	3.6764	4.3123	5.9197	7.2193	9.3064	8.7311	8.8372
5	4.5324	4.0935	4.0094	4.6381	6.4460	8.0370	10.229	9.1714	9.8888
6	4.4608	4.0320	3.8781	4.5084	6.2216	7.6838	9.9294	9.2847	9.4678
7	4.5302	4.0397	3.8676	4.4968	6.2800	7.8773	9.9424	9.1963	9.5097
8	4.3782	3.9884	3.9190	4.4701	6.2471	7.8298	10.132	8.9735	9.3786
9	4.5368	4.0655	4.0069	4.4658	6.1007	7.9021	9.6740	9.1644	9.4963
10	4.4505	4.0491	3.9169	4.6188	6.1935	7.8672	9.5917	8.8355	9.7209

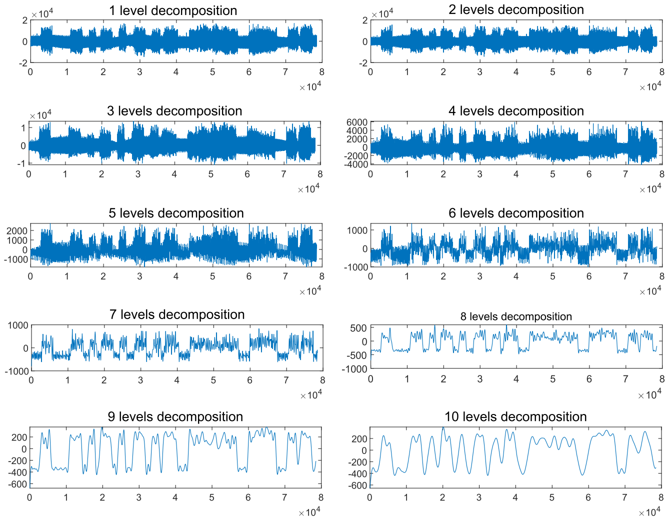


Fig. 7. The approximation coefficients after wavelet decomposition with Db9 wavelet basis under different decomposition levels from 1 to 10.

information, such as those glitches in the first eight subfigures, still exists in the trace of approximation coefficients. As the number of decomposition levels increases, the high frequency information is reduced, and the curve of approximation coefficients is smoothed. Meanwhile, the amplitude characteristics of signal are getting much clearer. However, in the last subfigure with 10 levels of decompositions, the curve is over-smoothed and the power change of encryption parts cannot be recognized. Through the observation in Fig. 7, the 9th level is verified as a turning point, which is consistent to the optimal value depicted in Fig.6. With these double verifications, $l_{\mathtt{WTA}} = 9$ is considered as the best choice for the decomposition level.

3) Selection of Threshold and Traces Alignment: In order to map the random delay and encryption part, a method based on threshold is applied, just like the method in WTD. When the denoised trace $\mathcal{S}^1$ is decomposed through $\mathbb{DWT}_{\mathtt{WTA}}$, the construct signal after Step 1 ($\mathcal{W}^2$) still consists of the wavelet coefficients for the encryption and delay part ($\mathcal{W}^2_{ep}$ and $\mathcal{W}^2_{dp}$). In WTA, the minimum coefficients for encryption part and maximum coefficient for delay part can be calculated, which are denoted as e_{min} and d_{max}, respectively. Due to the difference of amplitude features, $\mathcal{W}^2_{dp}$ is distinctively smaller than $\mathcal{W}^2_{ep}$, so the maximum value of $\mathcal{W}^2_{dp}$ (d_{max}) is still smaller than the minimum value of $\mathcal{W}^2_{ep}$ (e_{min}). In this sense, those coefficients larger than e_{min} are actually associated with the encryption part, while those lower than d_{max} are with the delay part. It can be determined that the differentiating threshold $\lambda_{\mathtt{WTA}}$ is within the range of $[d_{max}, e_{min}]$. This process is described in Line 1 to 5 of Algorithm 2.

Then in order to decide the exact threshold $\lambda_{\mathtt{WTA}}$, all the possible values are tested. The process of trace alignment is carried out with regard to different candidate thresholds which are increased with a certain step value. Then a mathematical concept—correlation coefficient (Corr) is introduced to evaluate the performance of alignment and thus determine the best choice.

The method to align the traces based on $\lambda_{\mathtt{WTA}}$ mainly depends on the fact that those coefficients larger than the threshold correspond to encryption operations while those less than the threshold map to random delays. Therefore, for a specific coefficient $\mathcal{W}^2(w)$, if $\mathcal{W}^2(w) < \lambda_{\mathtt{WTA}}$, it is considered as the delay part and then discarded; otherwise it is reserved. The processed wavelet coefficients are denoted as $\mathcal{W}^{2*}$, which actually corresponds to the encryption part in wavelet domain with neither noise nor random delays.

To evaluate the effectiveness of alignment using a certain threshold, the $Corr$ between two aligned traces is calculated. If the encryptions are well aligned, the value of $Corr$ should be larger, indicating that the random delays are removed to some extent. In the contrary, the $Corr$ value will be relatively smaller if the traces are not fully aligned, namely, this value is not an appropriate choice for the threshold. Finally, the best threshold can be determined when it produces the maximum $Corr$ value in trace alignment. This process is shown in Line 6 to 12 of Algorithm 2, where $\mathcal{W}^{2*}(w_1)$ and $\mathcal{W}^{2*}(w_2)$ are two aligned traces in wavelet domain after removing the random delays, and $corrcoef$ is a function to calculate the correlation coefficient.

Fig. 8 shows how the values of different correlation coefficients correspond to different thresholds. The step is 0.1, and the stepped value of threshold increases from the maximum coefficient of encryptions (d_{max}) to the minimum coefficient of delays (e_{min}). Note that in Fig. 8, the stepped value is represented as the offset to the base d_{max}. There is a distinct peak when the offset is 29.8. And the corresponding value of $Corr$ is up to 0.9476. Therefore, the base with an offset of 29.8 is considered the best selection of the threshold to

Algorithm 2 Find the Threshold $\lambda_{\mathtt{WTA}}$ and Align the Traces in Wavelet Domain

```
Input  : Power traces: $\mathcal{S}^1 = \mathcal{S}^1_{ep}(t) + \mathcal{S}^1_{dp}(t)$
Output: Threshold $\lambda_{\mathtt{WTA}}$
$\mathbb{DWT}(\mathcal{S}^1) \longrightarrow \mathcal{W}^2(w), w \in [0, length(\mathcal{S}^1)/2^8]$;
$\mathcal{W}^2(w) = \mathcal{W}^2_{ep}(w) + \mathcal{W}^2_{dp}(w)$;
$max\{\mathcal{W}^2_{dp}(w)\} \longrightarrow d_{max}$;
$min\{\mathcal{W}^2_{ep}(w)\} \longrightarrow e_{min}$;
Threshold: $\lambda_{\mathtt{WTA}} \in [d_{max}, e_{min}]$;
for $\lambda_{\mathtt{WTA}} = d_{max} : step : e_{min}$ do
    if $\mathcal{W}^2(w) > \lambda_{\mathtt{WTA}}$ then
        $\mathcal{W}^{2*} = \mathcal{W}^2(w)$;
    end
    $Corr = corrcoef(\mathcal{W}^{2*}(w_1), \mathcal{W}^{2*}(w_2))$
end
The best threshold: $\lambda_{\mathtt{WTA}}$ with maximum $Corr$
```

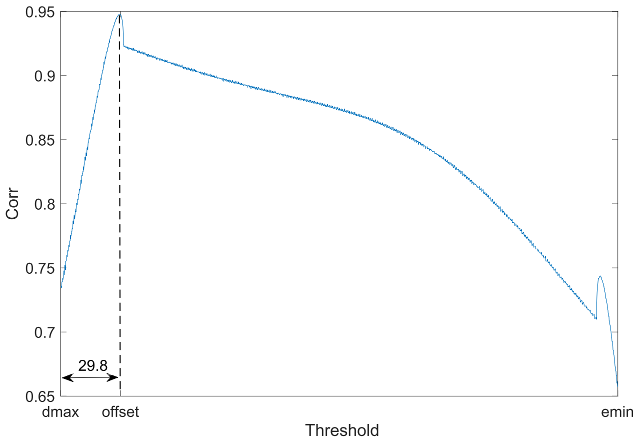
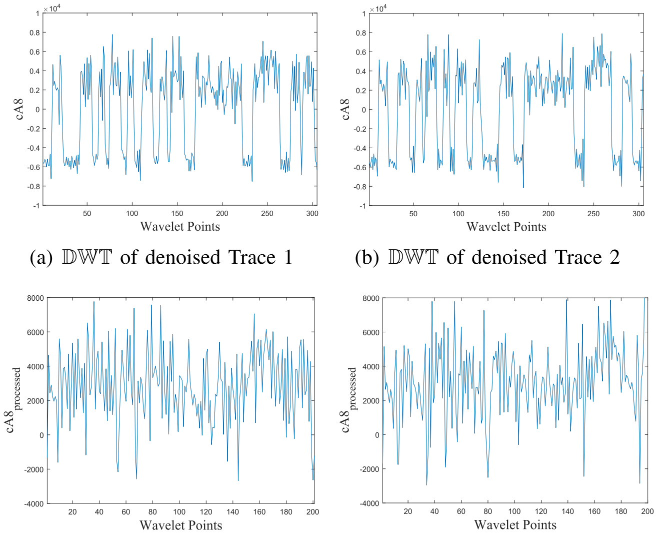


Fig. 8. The correlation coefficient (Corr) varies with the threshold $\lambda_{\mathtt{WTA}}$ from d_{max} to e_{min}. (step = 0.1)

remove random delays and align the traces. Furthermore, the $Corr$ value with 0.9476 is sufficient and reasonable to consider that the traces in wavelet domain is fully aligned and the WTA achieves good performance.

The functionality of WTA can be validated through experiments. Taking the power traces obtained from Strategy 2 as an example, Fig. 9 shows the comparisons between the output of Algorithm 2 with those in Fig. 4.

Fig. 9(a) and 9(b) show two traces that are the results of wavelet decomposition. That is, $\mathcal{S}^1$ is processed by the wavelet decomposition of WTA, where the wavelet basis function is Db9 (i.e., $n_{\mathtt{WTA}} = 9$) and the level decomposition is 9 (i.e., $l_{\mathtt{WTA}} = 9$). In both Fig. 9(a) and 9(b), the encryption and random delay part can be clearly identified. More specifically, higher coefficient values around $0.4 \sim 0.6 \times 10^4$ represent encryption operations while lower ones around -0.6×10^4 imply random delays. Fig. 9(c) and 9(d) depict two traces in wavelet domain output by Algorithm 2, where all the random delays are removed using the differentiating threshold $\lambda_{\mathtt{WTA}} = d_{max}+29.8$. These two wavelet-domain signals, represented by $\mathcal{W}^{2*}$, are obviously quite synchronized which can be further used in key extraction.



(a) $\mathbb{DWT}$ of denoised Trace 1 (b) $\mathbb{DWT}$ of denoised Trace 2

(c) WTA of denoised Trace 1 (d) WTA of denoised Trace 2

Fig. 9. Unaligned traces $\mathcal{W}^2(w)$ in wavelet domain with random delays (above) and aligned traces $\mathcal{W}^{2*}$ in wavelet domain after WTA (below).

With the realization of WTA, it is feasible to build the complete wavelet-based attack framework—WAF now. This is because only aligned wavelet coefficients are required in the subsequent Step 3 when extracting keys.

D. Wavelet-Based Key Extraction

In this step, $\mathcal{W}^{2*}$, the output of WTA, is directly used, where the signals are already aligned in wavelet domain and the useful encryption parts are well-preserved.

Algorithm 3 describes the practical procedures of key extraction WKE against three strategies in wavelet domain. It should be noted that WKE is quite different in terms of strategies and platforms of implementations.

The power model of leakages used in key recovery is depended on the physical characteristics of chip where the algorithm is running. In embedded software implementation, the number of logic ones, namely hamming weight power model ($\mathbb{HW}$), can depict the power consumption. However, on most of hardware platforms, the power consumption is related to the number of logic transitions, that is hamming distance model ($\mathbb{HD}$). Therefore the $\mathbb{HW}$ power model should be applied to Strategy 1 and 2 while $\mathbb{HD}$ model ought to be applied to Strategy 3.

Besides the power model, when Strategy 1 and 2 are considered, the attack target is on the SBox output in the first round of AES. The hypothetical intermediate value $V_{i,j}$ is calculated as the SBox output for each guessed key value $\mathcal{K}_g$ and plaintext pt, where i is the index of power traces (the total number of power traces is N_t) and j is the value of possible keys. Then the hypothetical power consumption value $H_{i,j}$ is calculated from $V_{i,j}$ based on $\mathbb{HW}$ power model.

In contrary, when Strategy 3, a hardware countermeasure, is considered, the target is on the SBox input in the last round.

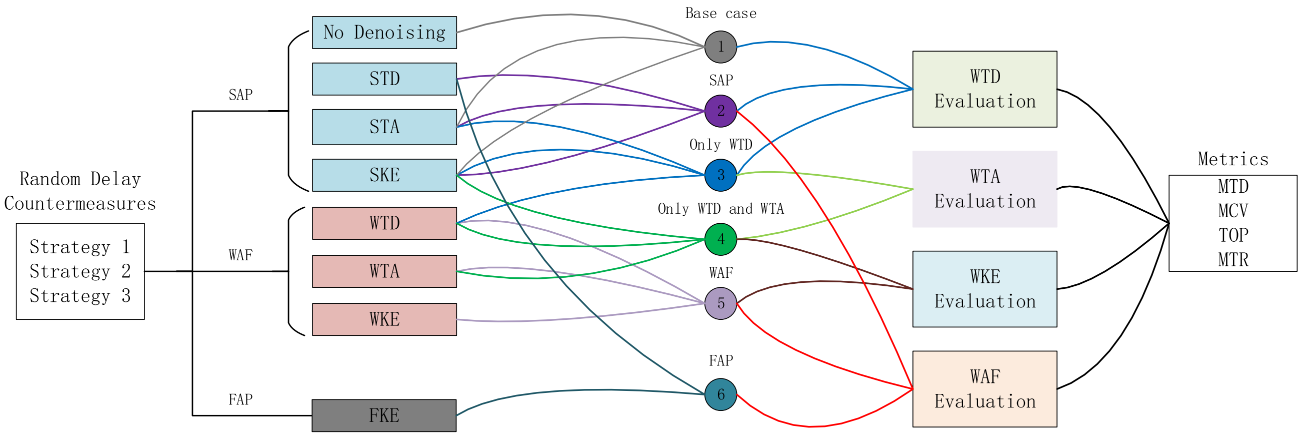


Fig. 10. The methodology to evaluate the WAF with ① ∼ ⑥ 6 comparative experiments.

$V_{i,j}$ is derived from the inversive SBox^{-1} input from $\mathcal{K}_g$ and ciphertext ct. Then $H_{i,j}$ is mapped from $V_{i,j}$ with the $\mathbb{HD}$ power model [34].

Finally, for both software and hardware implementations, the correlation $\rho_{j,w}$ between $H_{i,j}$ and $\mathcal{W}^{2*}$ is computed, where w is the index of samples in $\mathcal{W}^{2*}$. Eventually, the key guess $\mathcal{K}_g$ with the largest correlation coefficient value can be considered as the correct key byte $\mathcal{K}$.

Algorithm 3 Wavelet Key Extraction

```
for $\mathcal{K}_g = 0$ To 255 do
    // Strategy 1,2
    $V_{i,j} = \mathsf{SBOX}(pt \oplus \mathcal{K}_g), i = 1..N_t, j = 1..256;$
    // Strategy 3
    Or
    $V_{i,j} = \mathsf{SBOX}^{-1}(ct \oplus \mathcal{K}_g), i = 1..N_t, j = 1..256;$
end
foreach $V_{i,j}$ do
    // Strategy 1,2
    $H_{i,j} = \mathbb{HW}(V_{i,j}), i = 1..N_t, j = 1..256;$
    // Strategy 3
    Or $H_{i,j} = \mathbb{HD}(V_{i,j}, ct), i = 1..N_t, j = 1..256;$
end
foreach $\mathcal{W}^{2*}$ do
    $\rho_{j,w} = corr(H_{i,j}, \mathcal{W}^{2*}(w)), j = 1..256, w =$
    $1..length(\mathcal{W}^{2*});$
end
```

V. Evaluation of Proposed Attack Framework

In this section, series of comparative experiments are conducted in software and hardware platforms, respectively. These experiments are well-designed as in an internal and external manner to evaluate the proposed framework. Some of them are intended to assess the individual attacking power of each step/component. Meanwhile others are designed to compare WAF with those counterparts, i.e., the standard attack procedure SAP and the frequency-based attack procedure FAP. In the evaluation process, four metrics are carefully selected to reveal the performance and efficiency of WAF systematically.

TABLE VI
Experiment Settings

Countermeasures	Platform	Number of Traces	Sampling Rate
Strategy 1	SASEBO-W	4096	100 MHz
Strategy 2	SASEBO-W	10000	100 MHz
Strategy 3	SASEBO-GII	16364	1 GHz

A. Setup and Metrics

To measure the power leakage of cryptographic devices, two side-channel attack standard evaluation boards are served as our main experiment platforms. For Strategy 1 and 2, SASEBO-W is utilized to implement the cryptographic algorithm and countermeasures at the software level. Details about SASEBO-W can be referred to [35]. On SASEBO-W, AES-128 implemented in C is written to the ATMega163 microcontroller inside a smart card. For Strategy 3 with random clock, SASEBO-GII is chosen to implement the AES algorithm and countermeasures at the hardware level. Since SASEBO-GII is a hardware platform based on FPGA, AES-128 with random clock is implemented in Verilog. In addition, an oscilloscope (Agilent DSO-X3034T) is used to measure the power leakages whose bandwidth is 350MHz and the maximum sampling frequency is 5GSa/s (Samples per second). Here 4096 and 10000 power traces are collected at a sampling rate of 100MHz respectively when Strategy 1 and 2 are applied, and 16364 power traces are measured with a sampling rate of 1GHz when Strategy 3 is used. Three steps of WAF are fully implemented in MATLAB2018a, including wavelet-based traces denoising WTD, wavelet-based traces alignment WTA and wavelet-based key extraction WKE. Table VI lists our experiment settings.

In order to illustrate the performance and efficiency of WAF, four standard metrics, which are widely acknowledged in SCA community [34], [36], [37], are used to evaluate practical attack results. Among them, the first two MTD and MCV are used to appraise the performance in terms of accuracy, and the rest two TOP and MTR are used for evaluating the efficiency in terms of processing time.

TABLE VII
EVALUATIONS OF EACH COMPONENT OF WAF: WTD, WTA AND WKE

		Denoising Component			Alignment Component			Recovering Component	
Strategies	Metrics	No denoising	STD	WTD	Static alignment	Elastic alignment	WTA	SKE	WKE
Strategy 1	MTD	2307	1655	**825**	825	517	**132**	132	**264**
	MCV	0.096	0.121	**0.207**	0.207	0.291	**0.661**	0.661	**0.370**
	TOP	0	14.52	**18.74**	1795.95	3863.25	**18.04**	18.04	**13.86**
	MTR	76.79	78.65	**77.05**	77.05	79.52	**344.80**	344.80	**18.89**
Strategy 2	MTD	7873	6865	**3997**	3997	2712	**262**	262	**463**
	MCV	0.063	0.073	**0.094**	0.094	0.109	**0.377**	0.378	**0.299**
	TOP	0	65.81	**211.43**	22562.98	28873.25	**386.85**	386.85	**322.66**
	MTR	2811.31	2803.74	**2809.20**	2809.20	5374.16	**5242.74**	5242.74	**103.48**
Strategy 3	MTD	13610	12340	**10450**	10450	9128	**3993**	3993	**4979**
	MCV	0.047	0.052	**0.057**	0.057	0.066	**0.106**	0.106	**0.091**
	TOP	0	32.13	**73.73**	1976.29	3123.52	**133.25**	133.25	**116.86**
	MTR	233.17	240.16	**244.78**	244.78	257.29	**408.64**	408.64	**16.84**

i MTD: *Minimum Traces to Disclose the correct key byte* [36]. Since massive measurements take time and the corresponding traces storage, it is always desirable to extract the secret key with fewer traces. The attack that uses the smaller number of traces will be considered as a more powerful one.

ii MCV: *Maximum Correlation Value of the correct key byte* [34]. This metric calculates the correlation between the key guesses and the actual key value, which reflects the accuracy of attack results. To some extent, the attack method is better if its MCV is larger.

iii TOP: *Time Of Preprocessing of power traces* [37]. Before the key recovery, the pre-processing of traces (including WTD and WTA) will take some time. The attack with less processing time is always preferred. If its preprocessing takes too much workload and time, this attack can be considered as inefficient.

iv MTR: *Minimum Time to Recover a key byte correctly* [37]. The time for key recovery is an important metric in SCA. The less time it takes in key recovery, the more efficient the attack is.

B. Experiment Design Principle

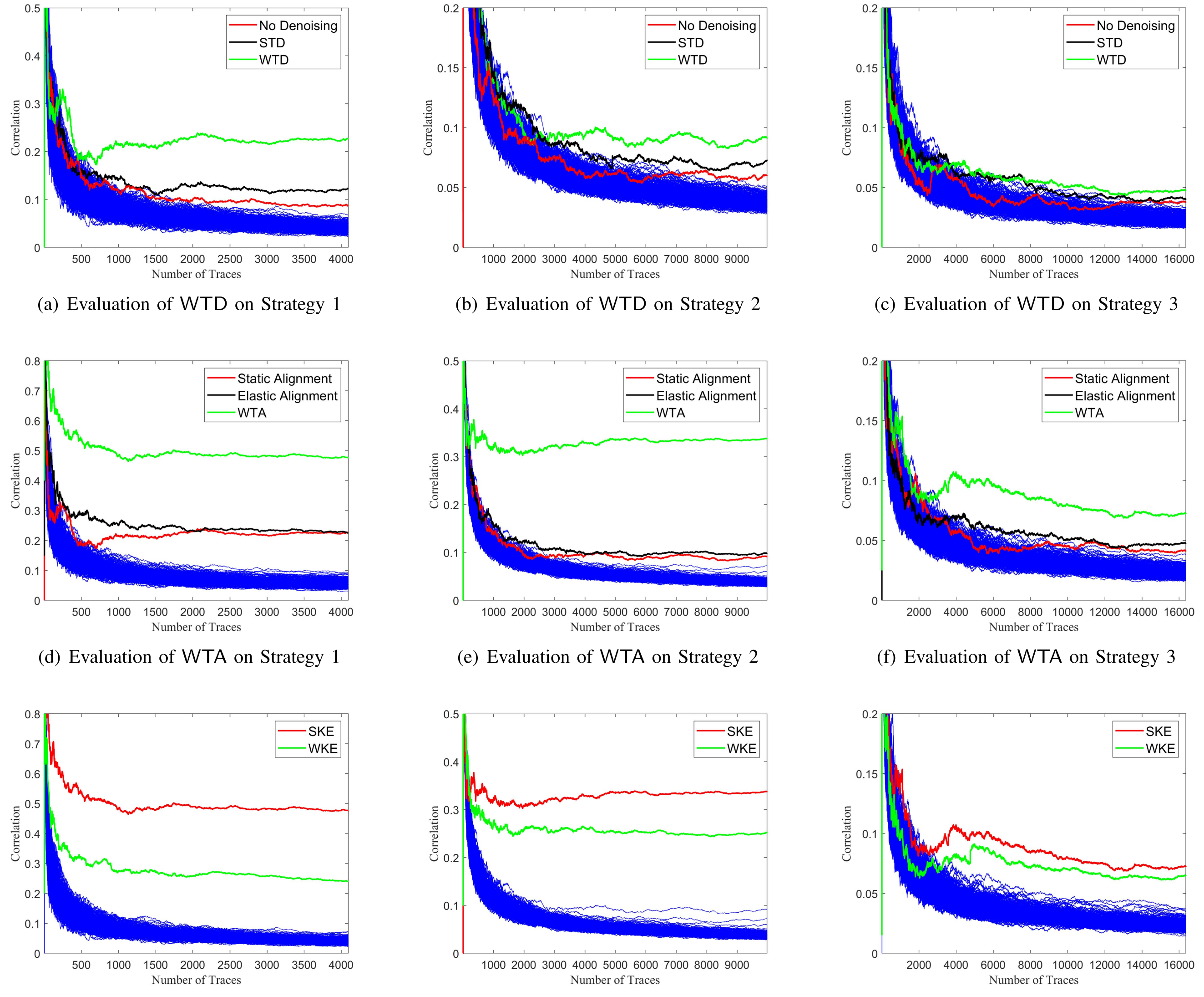
In order to evaluate WAF in a systematic way, a series of experiments are designed as shown in Fig. 10, which takes three random delay countermeasures and four evaluation metrics into consideration. The design principle of those experiments can be described as following: the whole framework (WAF) can be viewed as a system with three degrees of freedom in corresponding to the three steps/components. First, several experiments are designed internally for the WAF framework. Any of WTD, WTA and WKE is individually viewed as a single degree of freedom; each time, only one of them is sent for estimation and compared with those counterparts (i.e., STD and STA) while other two degrees of freedom are fixed. Evaluations through these experiments are detailed in Section V-C, V-D and V-E, respectively. Second, more experiments are designed externally out of the framework in order to compare WAF as a whole against SAP and FAP. The evaluations are elaborated in Section V-F.

To conduct the evaluation internally and externally, Experiment ① to ⑥ are carefully designed to fulfill this principle, as shown in Fig. 10. Experiment ① is a base case where the power traces are not processed with denoising. Experiment ② is a SAP method which relies on STD to denoise the power traces. In order to evaluate the denoise component, Experiment ③ is designed to utilize WTD only and the other two components are the same with Experiment ① and ②. For further evaluation, Experiment ④ is designed to align the power traces with WTA, while other two components are the same with Experiment ③. Experiment ⑤ is a complete WAF framework where only WKE is different from Experiment ④, which can be equivalently used to assess the key extraction component. In addition, Experiment ⑥ is an attack case in frequency domain—FAP, using the FKE to recover keys after STD.

Meanwhile, Fig. 10 also depicts our logic of experiment design through a connecting network diagram. The lines from components/steps (color blocks on the left, e.g., WTD, WTA, WKE) to Experiments (circles at the center, e.g., ③ and ④) show the component structure of each experiment. For example, Experiment ④ is made up of those components connected by green curves (WTD, WTA, and SKE). In addition, the lines from Experiments to the Evaluation parts (color blocks on the right, e.g., WAF Evaluation) indicate that: the evaluation has to be conducted by analyzing the results from those experiments that are connected with it. For instance, the WAF Evaluation is accomplished by Experiment ②, ⑤ and ⑥ which are connected through red curves. Finally, the black lines to the Metrics part show that all the evaluations are metered with those four criterions.

C. Evaluation of Wavelet-Based Trace Denoising

To measure WTD individually, three experiments—①, ② and ③ are selected. Only the denoising component is different, while STA and SKE are applied in both Step 2 and 3.



(a) Evaluation of WTD on Strategy 1

(b) Evaluation of WTD on Strategy 2

(c) Evaluation of WTD on Strategy 3

(d) Evaluation of WTA on Strategy 1

(e) Evaluation of WTA on Strategy 2

(f) Evaluation of WTA on Strategy 3

(g) Evaluation of WKE on Strategy 1

(h) Evaluation of WKE on Strategy 2

(i) Evaluation of WKE on Strategy 3

Fig. 11. The correlation curves of each inner component in framework WAF. (left: strategy 1 (random propagation delay); center: strategy 2 (random delay insertion); right: strategy 3 (random clock jitters)).

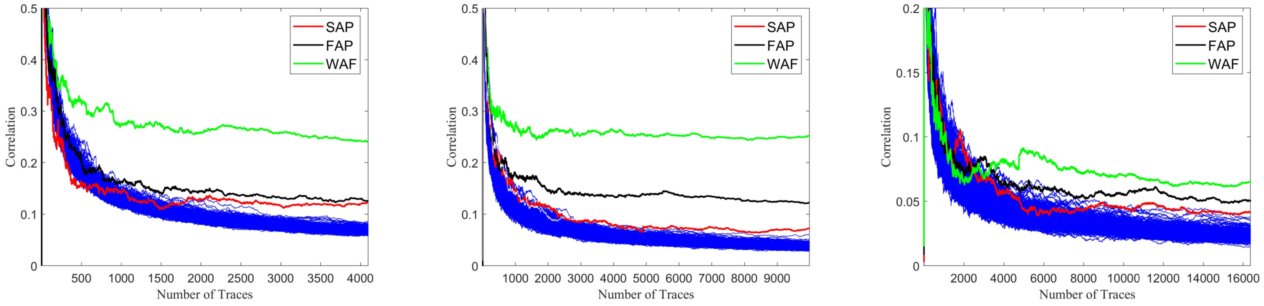
The Denoising Component column of Table VII lists the results of three types of denoising methods against different countermeasures. Compared to the base case (Experiment ①), standard trace denoising (Experiment ②) only reduces MTD from 2307 to 1655 and increases MCV from 0.096 to 0.121 when using Strategy 1. The improvement from STD is quite limited. This is because there still remain many signals that are unrelated to SCA and cannot be removed by low pass filters, not like common high frequency noises. In contrast, wavelet-based trace denoising produces a better result in terms of MTD and MCV due to its thorough analysis on signals. For example, in both Strategy 1 and 2, WTD reduces MTD by around 50% compared to STD. And it can also double the correlation MCV (from 0.121 to 0.207) in Strategy 1. Moreover, the cost of WTD over STD is not that much in terms of the time metric TOP. The processing time is merely increased from 14.52/65.81/32.13 to 18.74/211.43/73.73. To this point, it is better to use wavelet-based technology in denoising for both software and hardware platforms.

To show its advantages clearly, Fig. 11(a) to Fig. 11(c) display the correlation curves for the base case, Experiment ② and ③ (marked in red, black and green respectively) when three countermeasure schemes are applied. Among all the key guesses where those incorrect ones are marked in blue, attacks using WTD can use the least number of traces to reveal the correct key with a maximal value of correlation coefficient, which is more obvious for Strategy 1 and 2.

D. Evaluation of Wavelet-Based Trace Alignment

The output traces denoised by WTD can be used in further alignment. The performance of WTA is examined through two experiments–③ (including static and elastic alignment) and ④. In this evaluation, WTD and SKE are applied in Step 1 and 3, leaving the alignment component as the single degree of freedom.

The results of these experiments are shown in the Alignment Component column of Table VII. Taking Strategy 2 as



(a) Random propagation delay

(b) Random delays insertion

(c) Random clock jitters

Fig. 12. The correlation curves of attack based on standard attack procedures SAP (red), frequency-based framework FAP (black) and wavelet-based attack framework WAF (green).

an example. With regard to MTD, WTA only requires 132 traces to recover the first key byte, which is far less than those 3997/2712 traces when static/elastic alignment are used. As for the MCV, WTA can raise the correlation coefficient from 0.094/0.109 to 0.377 (about 4/3.4 times), showing that STA can not fully align all traces and could be further enhanced by wavelet-based techniques.

The most impressive advantage of WTA is its speed. It only takes 386.85 seconds during the processing in comparison to the six hours for static/elastic alignments. The speedup for Strategy 2 is about 60 times. Similar conclusions can also be made for Strategy 1 and 3.

In short summary, our proposed wavelet-based traces alignment (WTA) brings a significant improvement to SCA against random delay countermeasures with a quite low cost of processing time. Fig. 11(d) to Fig. 11(f) illustrate the effects of WTA facing with those three strategies. The correlation curves for static alignment, elastic alignment and WTA are marked in red, black and green, respectively. The green curve corresponding to WTA clearly overwhelms the other two cases regarding the small number of traces that are required and the large correlation values. The results are also consistent with the data in Table VII.

E. Evaluation of Wavelet-Based Key Extraction

After all the power traces are aligned through WTA, two experiments—④ and ⑤ are selected to show the effects of wavelet-based key extraction WKE. Note that WTD and WTA are fixed in Step 1 and 2, leaving the component of key extraction as the single degree of freedom.

As shown in the Recovering Component column of Table VII, it requires 264/463/4979 traces for WKE as compared to 132/262/3993 for SKE, where the increased number of traces as the additional cost is not that much. Honestly speaking, correlation values for WKE are actually smaller as compared to SKE, which may not be as good as expected. However, its MCV is 0.370/0.299/0.091 in all three cases, which is already sufficient enough for practical key recovery.

Most importantly, the time that is required to extract a key byte, i.e., MTR, is only about 18.89/103.48/16.84 seconds in all three strategies, which is a significant reduction compared to SKE (344.80/5242.74/408.64 seconds). Besides, since there is no need to reconstruct signals in time domain, the preprocessing time (TOP) is also reduced a lot. In a word, even though WKE performs a little worse in terms of MTD and MCV, it shows great advantages in the wavelet domain regarding to the effectiveness and processing time.

TABLE VIII
EVALUATION OF PROPOSED FRAMEWORK WAF

Strategies	Metrics	SAP	**WAF**	FAP
Strategy 1	MTD	1655	**264**	1205
	MCV	0.121	**0.370**	0.164
	TOP	2296.17	**13.86**	5.53
	MTR	78.65	**18.89**	434.36
Strategy 2	MTD	6865	**463**	993
	MCV	0.073	**0.299**	0.181
	TOP	22562.98	**322.66**	241.93
	MTR	2809.20	**103.48**	6020.45
Strategy 3	MTD	12340	**7651**	8013
	MCV	0.052	**0.082**	0.061
	TOP	2008.42	**116.86**	19.90
	MTR	240.16	**16.84**	214.53

Similarly, Fig. 11(g) to Fig. 11(i) depict the performance of WKE when three countermeasures are applied. The correlation curves for Experiment ④ and ⑤ are marked in red and green, respectively. All the red curves are at the top, showing that SKE is doing well. However, note that the cost for reconstructing signals back to time domain is not included in the comparisons. In practice, the MTD and MCV of WKE are already good enough to fully recover a key byte. Taking all the pros and cons into consideration, it is still suggested to recover keys in wavelet domain based on WKE after traces alignment, especially for the unified wavelet-based framework.

F. Evaluation of the Whole Wavelet-Based Attack Framework

Section V-C, V-D and V-E have evaluated the components of WAF individually and internally, whereas, this section compares the whole framework with other commonly-used methods from an external point of view. To this end, three experiments are selected for the comparison: the proposed framework WAF (Experiment ⑤), the standard attack procedures SAP (Experiment ②) and the frequency-based attack procedures FAP (Experiment ⑥). And the results are listed in Table VIII.

Overall, WAF presents an outstanding performance and effectiveness in almost all evaluation criterions in Table VIII. The number of power traces that is required to conduct a successful attack is only 264/463/7651, which is reduced by 84%/93.2%/38% of that for SAP and 77.8%/53.4%/4.5% for FAP, respectively. Besides, the correlation value reaches to 0.370/0.299/0.082, which is increased by 205.8%/309.6%/57.7% of that for SAP and 125.6%/65.2%/34.4% for FAP, respectively. In addition, the total time of preprocessing and keys recovery ($\mathtt{TOP}+\mathtt{MTR}$) is only 32.75/426.14/133.70 seconds, which is reduced quite a lot as compared with other two methods.

Fig. 12 makes a much better presentation of the attack performance on three different strategies. The correlation curves of Experiment ②, ⑤ and ⑥ are marked in red, green and black respectively. It can be seen that SAP works not that well among them because the normal denoising and alignment processes actually bring very limited effects, which have been analyzed in Section V-C and V-D. Moreover, WAF performs better than FAP since the correlation is reduced in frequency domain.

Considering all the evaluation results, the wavelet-based attack framework (WAF) is a better choice when encountering random delay countermeasures.

VI. Conclusion and Future Work

In this paper, a unified wavelet-based attack framework (WAF) is proposed, including three components: wavelet trace denoising (WTD), wavelet trace alignment (WTA) and wavelet-based keys extraction (WKE). Even though the isolated idea of denoising and recovering in wavelet domain has been proposed before our work, the all-in-one solution against random delay countermeasures as a complete wavelet framework has not been realized. This is mainly due to the lack of the wavelet-based alignment which should have connected the existing WTD and WKE. In particular, we propose a novel method of aligning with high performance and effectiveness in the wavelet domain for this framework.

For the sake of completeness, the entire framework and its individual component are systematically evaluated through a series of comparative experiments. As shown from those experimental results, attacks based on the whole wavelet framework—WAF are significantly improved in terms of performance and efficiency. Compared to standard attack procedures (SAP) and frequency-based attack procedures (FAP), wavelet-based attack framework on various random delay countermeasures can recover the key bytes with a short processing time, a small number of traces and a much larger correlation coefficient value.

Future work will focus on unifying two wavelet decomposition parameters—the order of wavelet basis function ($n_{\mathtt{WTD}}$, $n_{\mathtt{WTA}}$) and the wavelet decomposition levels ($l_{\mathtt{WTD}}$, $l_{\mathtt{WTA}}$) in both Step 1 and 2. In this paper, due to the inconsistency of these two parameters in WTD and WTA, the reconstruction to time-domain signals through $\mathbb{IDWT}$ has to be done after WTD and wavelet decomposition via $\mathbb{DWT}$ must be performed again in WTA. This will undoubtedly have an impact on the efficiency of attack. So when $n_{\mathtt{WTD}}$ and $l_{\mathtt{WTD}}$ are unified with $n_{\mathtt{WTA}}$ and $l_{\mathtt{WTA}}$, the wavelet decomposition needs to be done only once and the entire analysis can be finished in wavelet domain, which will possibly bring some top-up improvement to our proposed wavelet-based framework.

References

[1] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Berlin, Germany: Springer, 2008.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Annu. Int. Cryptol. Conf.* New York, NY, USA: Springer, 1999, pp. 388–397.

[3] C. Clavier, J.-S. Coron, and N. Dabbous, "Differential power analysis in the presence of hardware countermeasures," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2000, pp. 252–263.

[4] M. Bucci, R. Luzzi, M. Guglielmo, and A. Trifiletti, "A countermeasure against differential power analysis based on random delay insertion," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 4, May 2005, pp. 3547–3550.

[5] M. Tunstall and O. Benoit, "Efficient use of random delays in embedded software," in *Proc. IFIP Int. Workshop Inf. Security Theory Practices*. Berlin, Germany: Springer, 2007, pp. 27–38.

[6] J.-S. Coron and I. Kizhvatov, "An efficient method for random delay generation in embedded software," in *Cryptographic Hardware and Embedded Systems—CHES* Berlin, Germany: Springer, 2009, pp. 156–170.

[7] J.-S. Coron and I. Kizhvatov, "Analysis and improvement of the random delay countermeasure of CHES 2009," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2010, pp. 95–109.

[8] Y. Zafar, J. Park, and D. Har, "Random clocking induced DPA attack immunity in FPGAs," in *Proc. IEEE Int. Conf. Ind. Technol.*, Mar. 2010, pp. 1068–1070.

[9] K. H. Boey, Y. Lu, M. O'Neill, and R. Woods, "Random clock against differential power analysis," in *Proc. IEEE Asia Pacific Conf. Circuits Syst.*, Dec. 2010, pp. 756–759.

[10] J. G. van Woudenberg, M. F. Witteman, and B. Bakker, "Improving differential power analysis by elastic alignment," in *Proc. Cryptographers' Track RSA Conf.* Berlin, Germany: Springer, 2011, pp. 104–119.

[11] R. A. Muijrers, J. G. van Woudenberg, and L. Batina, "RAM: Rapid alignment method," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.* Berlin, Germany: Springer, 2011, pp. 266–282.

[12] S. Nagashima, N. Homma, Y. Imai, T. Aoki, and A. Satoh, "DPA using phase-based waveform matching against random-delay countermeasure," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2007, pp. 1807–1810.

[13] D. Strobel and C. Paar, "An efficient method for eliminating random delays in power traces of embedded software," in *Proc. Int. Conf. Inf. Secur. Cryptol.*, 2011, pp. 48–60.

[14] F. Durvaux, M. Renauld, F.-X. Standaert, L. Van Oldeneel tot Oldenzeel, and N. Veyrat-Charvillon, "Efficient removal of random delays from embedded software implementations using hidden Markov models," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.* Berlin, Germany: Springer, 2012, pp. 123–140.

[15] C. H. Gebotys, C. C. Tiu, and X. Chen, "A countermeasure for EM attack of a wireless PDA," in *Proc. Int. Conf. Inf. Technol., Coding Comput. (ITCC)-Volume II*, vol. 1, Apr. 2005, pp. 544–549.

[16] O. Schimmel, P. Duplys, E. Boehl, J. Hayek, R. Bosch, and W. Rosenstiel, "Correlation power analysis in frequency domain," in *Proc. COSADE 1st Int. Workshop Constructive SideChannel Anal. Secure Design*, Feb. 2010, pp. 1–3.

[17] Y. Lu, K. Boey, M. O'Neill, J. V. McCanny, and A. Satoh, "Is the differential frequency-based attack effective against random delay insertion," in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 2009, pp. 051–056.

[18] H. Patel and R. Baldwin, "Differential power analysis using wavelet decomposition," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2012, pp. 1–5.

[19] W. Liu, L. Wu, X. Zhang, and A. Wang, "Wavelet-based noise reduction in power analysis attack," in *Proc. 10th Int. Conf. Comput. Intell. Secur.*, Nov. 2014, pp. 405–409.

[20] J. Ai, Z. Wang, X. Zhou, and C. Ou, "Improved wavelet transform for noise reduction in power analysis attacks," in *Proc. IEEE Int. Conf. Signal Image Process. (ICSIP)*, Aug. 2016, pp. 602–606.

[21] X. Charvet and H. Pelletier, "Improving the DPA attack using wavelet transform," in *Proc. NIST Phys. Secur. Test. Workshop*, vol. 46, Sep. 2005, pp. 1–15.

[22] N. Debande, Y. Souissi, M. A. El Aabid, S. Guilley, and J.-L. Danger, "Wavelet transform based pre-processing for side channel analysis," in *Proc. 45th Annu. IEEE/ACM Int. Symp. Microarchitecture Workshops*, Dec. 2012, pp. 32–38.

[23] P. Saravanan and P. Kalpana, "A novel approach to attack smartcards using machine learning method," *J. Sci. Ind. Res.*, vol. 76, no. 2, pp. 95–99, Feb. 2017.

[24] S. Hou, Y. Zhou, H. Liu, and N. Zhu, "Wavelet support vector machine algorithm in power analysis attacks," *Radioengineering*, vol. 26, no. 3, pp. 890–902, Sep. 2017.

[25] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, *Wavelet Toolbox User's Guide*, vol. 15. MathWorks Inc., Natick, MA, USA, 1996, pp. 30–35.

[26] N. Ahuja, S. Lertrattanapanich, and N. K. Bose, "Properties determining choice of mother wavelet," *IEE Proc. Vis., Image Signal Process.*, vol. 152, no. 5, pp. 659–664, Oct. 2005.

[27] S. Li, Y. Ji, and G. Liu, "Optimal wavelet basis selection of wavelet shrinkage for ECG de-noising," in *Proc. Int. Conf. Manage. Service Sci.*, Sep. 2009, pp. 1–4.

[28] N. Deng and C.-S. Jiang, "Selection of optimal wavelet basis for signal denoising," in *Proc. 9th Int. Conf. Fuzzy Syst. Knowl. Discovery*, May 2012, pp. 1939–1943.

[29] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[30] S. Lin and X. Huang, "Advanced research on computer education, simulation and modeling," in *Proc. Int. Conf., (CESM)*, Wuhan, China, vol. 175, 2011, pp. 18–19.

[31] S. J. Gortler, P. Schröder, M. F. Cohen, and P. Hanrahan, "Wavelet radiosity," in *Proc. 20th Annu. Conf. Comput. Graph. Interact. Techn.*, 1993, pp. 221–230.

[32] M. Srivastava, C. L. Anderson, and J. H. Freed, "A new wavelet denoising method for selecting decomposition levels and noise thresholds," *IEEE Access*, vol. 4, pp. 3862–3877, 2016.

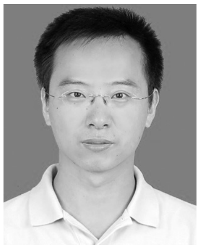
[33] R. Rao, "Wavelet transforms," *Encyclopedia Imag. Sci. Technol.*, Jan. 2002. doi: 10.1002/0471443395.img112.

[34] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2004, pp. 16–29.

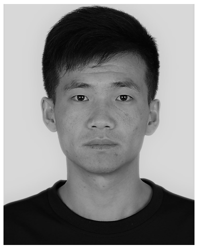
[35] *Side-Channel Attack Standard Evaluation Board (SASEBO-W)*. Accessed: Feb. 18, 2019. [Online]. Available: http://satoh.cs.uec.ac.jp/SASEBO/en/board/sasebo-w.html

[36] D. D. Hwang *et al.*, "AES-based security coprocessor IC in 0.18-*muhboxm* CMOS with resistance to differential power analysis side-channel attacks," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 781–792, Apr. 2006.

[37] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2009, pp. 443–461.



Fan Zhang was born in 1978. He received the Ph.D. degree from the Department of Computer Science and Engineering, University of Connecticut, USA, in 2012. He is currently an Associate Professor with the School of Cyber Science and Technology, College of Computer Science and Technology, Zhejiang University. He is also affiliated with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include system security, hardware security, cryptography, computer architecture, and sensor networks.



Xiaofei Dong was born in 1994. He received the bachelor's degree from the College of Electronics and Information Engineering, Qingdao University of Technology, in 2017. He is currently pursuing the master's degree with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include side-channel analysis, microprocessors and hardware security, fault injection, and fault analysis.

Bolin Yang was born in 1996. He received the bachelor's degree from the College of Information Science and Electronic Engineering, Zhejiang University, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include hardware security, cryptography, and fault injection and attack.

Yajin Zhou received the Ph.D. degree from North Carolina State University. He was a Senior Security Researcher with Qihoo 360. He is currently a ZJU100 Young Professor with the College of Computer Science and Technology, Institute of Cyberspace Research, Zhejiang University, China. He has published over 30 papers with 5700+ citations. His current focus is on identifying real-world threats and building practical solutions, in the context of software security of embedded systems (or IoT devices). He is also interested in emerging threats, including the security of smart contracts.

Kui Ren (A'07–M'07–SM'11–F'16) received the Ph.D. degree from Worcester Polytechnic Institute, Worcester, MA, USA. He is currently a Professor of computer science and technology and the Director of the Institute of Cyberspace Research with Zhejiang University, Hangzhou, China. His current research interests span cloud and outsourcing security, wireless and wearable system security, and artificial intelligence security. He is a Distinguished Scientist of the ACM. He was a recipient of the NSF CAREER Award in 2011 and the IEEE CISTC Technical Recognition Award 2017.