Jin and Zhou *Cybersecurity (2023)6:15*
https://doi.org/10.1186/s42400-023-00149-w

Cybersecurity

RESEARCH

Open Access

Enhancing non-profiled side-channel attacks by time-frequency analysis

Chengbin Jin[1,2] and Yongbin Zhou[1,3*]

Abstract

Side-channel analysis (SCA) has become an increasing important method to assess the physical security of cryptographic systems. In the process of SCA, the number of attack data directly determines the performance of SCA. With sufficient attack data, the adversary can achieve a successful SCA. However, in reality, the cryptographic device may be protected with some countermeasures to limit the number of encryptions using the same key. In this case, the adversary cannot use casual numbers of data to perform SCA. The performance of SCA will be severely dropped if the attack traces are insufficient. In this paper, we introduce wavelet scatter transform (WST) and short-time fourier transform (STFT) to non-profiled side-channel analysis domains, to improve the performance of side-channel attacks in the context of insufficient data. We design a practical framework to provide suitable parameters for WST/STFT-based SCA. Using the proposed method, the WST/STFT-based SCA method can significantly enhance the performance and robustness of non-profiled SCA. The practical attacks against four public datasets show that the proposed method is able to achieve more robust performance. Compared with the original correlation power analysis (CPA), the number of attack data can be reduced by 50–95%.

Keywords Correlation power analysis, Side-channel analysis, Proposed attack framework, Wavelet scatter transform, Short-time fourier transform

Introduction

Side-Channel Analysis has become a serious threat to cryptographic hardware units since the groundbreaking work by Kocher (1996). SCA can break mathematically sound cryptographic algorithms by utilizing time (Kocher 1996), power consumption (Goubin and Patarin 1999) and electromagnetic radiation (EM) (Gandolfi et al. 2001) and other physical side-channel leakages only. According to different assumptions about adversaries' attack capability, current SCA methods generally have two categories:

- *Profiled attacks*. Typical examples include Template Attack (TA) (Chari et al. 2003), Stochastic Attack (SA) (Schindler et al. 2005), Machine-Learning based Profiled Attacks (Lerman et al. 2015) and Deep-Learning based Profiled Attacks (Maghrebi et al. 2016; Cagli et al. 2017).
- *Non-profiled attacks*. Typical examples include Differential Power Analysis (DPA) (Goubin and Patarin 1999), Correlation Power Analysis (CPA) (Brier et al. 2004), Mutual Information Analysis (MIA) (Gierlichs et al. 2008) and recent Non-profiled Deep-Learning based Side-Channel Attack (Timon 2019).

In profiled attacks scenario, the adversary is allowed to have full access to a cloned device, where the cryptographic implementation is the same as the targeted

*Correspondence:
Yongbin Zhou
zhouyongbin@njust.edu.cn
[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

cryptographic implementation. Profiled attacks usually consist of two stages: (1) Profile stage; (2) Attack stage. In the profile stage, the adversary can collect a lot of data from the cloned device, and construct online templates to profile the leakage characteristics of each possible intermediate value, using prior knowledge about the cryptographic implementation, secret key, collected traces and corresponding plaintext/ciphertext. In the attack stage, the adversary collects a few physical traces from the analysis target, and then utilizes the constructed templates to extract the secret key from the analysis target. Compared with non-profiled attacks, profiled attacks allow adversaries to extract the secret key with much fewer traces. However, in realistic, the adversary usually is unable to have full access to cloned devices. In this case, profiled attacks cannot efficiently work.

Unlike profiled attacks, non-profiled attacks do not require full access to cloned devices. The adversary can search the whole hypothesis key value space and calculate corresponding intermediate value. The adversary can utilize some leakage models, such as hamming weight (HW) leakage model and hamming distance (HD) leakage model, to calculate the information leakage value of intermediate value, then adopt some mathematical metrics to calculate the linear relationship (e.g. mutual information, Pearson correlation coefficient) between the information leakage value and physical traces, to directly recover the secret key. Usually, the adversary selects the hypothesis key with the maximum metrics as the secret key.

In the past two decades, non-profiled attacks have emerged as an increasing important method for physical security evaluations. Liu et al. (2015) and Jin et al. (2022) adopted CPA to break commercial 3G/4G Universal Subscriber Identity Module (USIM) cards. They showed that the sensitive parameters of USIM cards can be fully extracted within 100,000 traces. In USENIX Security Symposium 2019, Batina et al. (2019) applied DPA to extract weight and bias parameters of Multilayer Perceptron (MLP) model and Convolutional Neural Networks (CNN) model. They showed that the adversary can efficiently reverse-engineer the machine-learning models if the target is not protected with some side-channel countermeasures (e.g. masking (Akkar and Giraud 2001), shuffling (Veyrat-Charvillon et al. 2012) and random delay (Coron and Kizhvatov 2010)). Besides, International Organization for Standardization (ISO)/ International Electro technical Commission (IEC) 19790-2012 International Standard (ISO/IEC-17825 2016) and American Federal Information Processing Standards (FIPS) 140-3 Standard (FIPS_140-3 2020) also adopt DPA and CPA to assess the physical security of crypto products. However, these works mainly focus on an idealized scenario that the adversary can use casual numbers of attack data. In an ideal scenario, the implementation details of the cryptographic devices is public to the adversary. The adversary can design a corresponding analysis method according to the characteristic of cryptographic implementation, and use sufficient traces to break the target. This kind of attack strategy is rational in an ideal scenario. However, when it applies to some specific applications or commercial crypto products, this kind of attack strategy can not efficiently work due to time and countermeasure constraints. In reality, the secret key, source code, and implementation details belong to the proprietary intellectual property of hardware vendors and are usually kept secret to the public. Some cryptographic devices or applications even adopt some countermeasures to limit the adversary's attack capability. For instance, National Institute of Standards and Technology (NIST) Counter Deterministic Random Byte Generator (CTR_DRBG) specification (Barker and Kelsey 2015) limits the number of times the same key used in Advanced Encryption Standard Counter Mode (AES-CTR) encryption to 4096. In this case, using a lot of attack data to perform CPA becomes impossible. Adversaries need to extract the secret key of CTR_DRBG within 4096 traces. Besides, some newest crypto products also adopt some specific protections to render adversaries' attack capability. For example, Zynq Ultracale+ (ZU+) Encryption Engine employs a key rolling scheme and Rivest Shamir Adleman (RSA) authentication to resist side-channel attacks (Hettwer et al. 2021). Similar to NIST CTR_DRBG specification, ZU+ utilizes key rolling scheme in AES-CTR encryption. ZU+ Encryption Engine only operates on specific data which is authenticated by RSA authentication. In this case, using sufficient traces to perform SCA becomes impossible. Consequently, the performance of SCA will be severely dropped.

To enhance original SCA methods, some researchers have considered applying certain data-augmentation techniques, such as Synthetic Minority Oversampling Technique (SMOTE) (Picek et al. 2019), adding gaussian noise (Kim et al. 2019), to increase the size of original dataset. They show that these kinds of methods (Picek et al. 2019; Kim et al. 2019) can efficiently enhance profiled SCA in the case of analyzing public datasets. However, data-augmentation techniques are only limited to profiled attacks scenario. In addition to enlarging the number of attack data, some researchers considered applying some preprocessing techniques to improve the quality of collected signals. For instance, Pozo et al. and Bruneau et al. applied Singular Spectrum Analysis (SSA) (Merino Del Pozo and Standaert 2015) and Principal Component Analysis (PCA) (Bruneau et al. 2015) to preprocess the original traces. They showed that SSA and

PCA can enhance CPA in the case of analyzing unprotected software-based implementations of Advanced Encryption Standard (AES). With suitable parameters selection, PCA and SSA can reduce the number of attack data by 20% at least. Unlike the researches (Merino Del Pozo and Standaert 2015; Bruneau et al. 2015), Yang et al. (2020) and Wu and Picek (2020) adopted denoise-autoencoder to preprocess the physical traces. They showed that deep-learning based preprocessing methods have superior performances than traditional preprocessing methods (Merino Del Pozo and Standaert 2015; Bruneau et al. 2015) in the case of analyzing hiding countermeasures. However, this kind of method (Yang et al. 2020; Wu and Picek 2020) is limited to profiled attacks scenario. The adversary needs to have prior knowledge about the secret key, plaintext/ciphertext when training denoise-autoencoders. This kind of method (Yang et al. 2020; Wu and Picek 2020) requires a lot of training data and is not practical in non-profiled attack scenario. In addition to applying time-domain based preprocessing methods to preprocess the physical traces, some researches considered applying frequency-based preprocessing method, such as wavelet transform (WT) (Debande et al. 2012; Udvarhelyi et al. 2021; Destouet et al. 2021), fast fourier transform (FFT) (Zhang et al. 2020) to enhance SCA. In MICRO 2012, Debande et al. (2012) adopted wavelet transform to enhance CPA in the case of analyzing DPA Contest V4. They showed that wavelet transform can reduce the number of attack data by 30% when the parameters are properly selected. Based on the research (Debande et al. 2012), Udvarhelyi et al. (2021) and Destouet et al. (2021) applied wavelet transform to enhance profiled attacks in the case of analyzing masked implementation of AES and commercial crypto products. In Design Automation Conference (DAC) 2020, Zhang et al. (2020) applied FFT to enhance cross-device attacks in the case of analyzing heterogeneous devices. Compared with data-augmentation techniques (Picek et al. 2019; Kim et al. 2019), frequency-based methods are applicable for non-profiled attacks and profiled attacks. Compared with typical time-domain based preprocessing methods (Merino Del Pozo and Standaert 2015; Bruneau et al. 2015), frequency-based preprocessing methods do not have specific requirements for input data-dimension. Frequency-based preprocessing method can be applied to analyze arbitrarily distributed dataset. In general, frequency-based preprocessing method theoretically has more appealing technique potential in the case of enhancing non-profiled attacks. However, current related works (Gebotys et al. 2005; Belgarric et al. 2014) are mostly limited to an ideal scenario that the adversary is assumed to know the suitable parameters, and they are limited to unprotected implementations and specific platforms. They do not deeply investigate whether frequency-based preprocessing methods are applicable for other more complex cryptographic implementations. In addition, previous works (Udvarhelyi et al. 2021; Destouet et al. 2021; Gebotys et al. 2005; Belgarric et al. 2014) require high-expert degree. They do not consider how to select suitable parameters for frequency-based parameters in non-profiled attacks scenario. The adversary needs to select frequency components empirically to enhance original SCA methods. The effect of parameters setting on the performance of frequency-based SCA has not been studied in depth. In practice, the parameter values, such as the standard deviation value, the size of the Gaussian window and the frequency component (used in the STFT-based SCA scenario), play important roles in the scenario of frequency-based SCA. Frequency-based SCA can significantly reduce the side-channel distinguisher's requirement for the scale of attack data with suitable parameters setting. However, if the parameters are not properly selected, it may even reduce the performance of the original SCA. Hence, designing a practical framework for proper selection of parameters is the most paramount thing for improving the performance of frequency-based SCA.

Aiming to address the limitation of current researches (Udvarhelyi et al. 2021; Destouet et al. 2021; Gebotys et al. 2005; Belgarric et al. 2014), we propose a practical framework to provide suitable parameters for frequency-based SCA. Specifically, we apply the concept of grid-search method to search the suitable parameters for frequency-based SCA, and design three evaluation metrics to evaluate the quality of extracted frequency components. The framework updates the parameters of frequency-based preprocessing methods iteratively according to the feedbacks from designed evaluation metrics. Unlike traditional grid-search methods, our framework can obtain the suitable parameters settings in non-profiled attack scenarios. As a result, our method can efficiently enhance original CPA methods in the case of analyzing multiple unprotected/protected implementations of AES. Compared with previous works (Udvarhelyi et al. 2021; Destouet et al. 2021; Gebotys et al. 2005; Belgarric et al. 2014), our work is more generic and does not require any expert-knowledge dependence degree. To summarize, the contributions of our work mainly include following:

- Introduce Wavelet Scatter Transform (WST) (andén and Mallat 2013) to non-profiled SCA domain, to efficiently improve the performance of CPA attacks. This is the first work that applies WST to enhance non-profiled attacks in the context of analyzing different AES implementations (e.g. software/hardware-

based implementation of AES, which is unprotected or protected with masking or random delay countermeasures).

- Propose a practical framework to select suitable parameters for WST-based CPA and STFT-based CPA. With our proposed framework, the adversary can obtain suitable parameters without any expert-knowledge dependence. The performance of original CPA can be significantly enhanced with our proposed method.
- Evaluate the extendability and applicability of our attack framework, we present practical attacks on four public datasets, including DPA Contest V4 (DPA_Contest_v4 2014), AES_HD (AES_HD 2018), AES_RD (AES_RD 2017) and ASCAD (2018) datasets. The result of our experiment shows that our method can reduce the number of attack traces by 50–95% in comparison with original CPA attacks, which achieves more robust performance of attacks.
- Carry out a systematic empirical research about the effectiveness and applicability of STFT-based CPA and WST-based CPA. The performance of WST-based CPA and STFT-based CPA is evaluated under different parameter values in a fine-grain manner. According to the analysis results, we provide empirical suggestions about parameter selections for non-profiled attacks scenario.

The graphic summary of our work is given in Fig. 1.

The paper is organized as 7 main sections: "Introduction" Section gives an introduction of the paper. "Preliminary" Section gives a brief background about some frequency-based preprocessing methods. "A practical framework for frequency-based CPA attack" Section proposes the practical framework for frequency-based CPA attack. "Experiment results" Section presents experiment analysis and practical attacks on four public datasets. "A fine-grain analysis on parameter settings for frequency-based CPA attacks" Section presents a fine-grain analysis on parameter settings for Frequency-based CPA attack. According to the experiment results (Sects. "Experiment results" and "A fine-grain analysis on parameter settings for frequency-based CPA attacks") and "Discussions" Section discusses related works, the advantages/disadvantages of attack framework and future works. The paper is concluded "Conclusions" Section

The abbreviations used in the paper are listed in the section of Abbreviations.

Preliminary

A brief background about the three typical time-frequency transformations - DFT, STFT and WST that are used for SCA is presented in this paper. We illustrate the advantages/disadvantages of these three time-frequency transformations used in SCA and then point out the importance of suitable parameters selection for time-frequency transformations. In the following, the collected

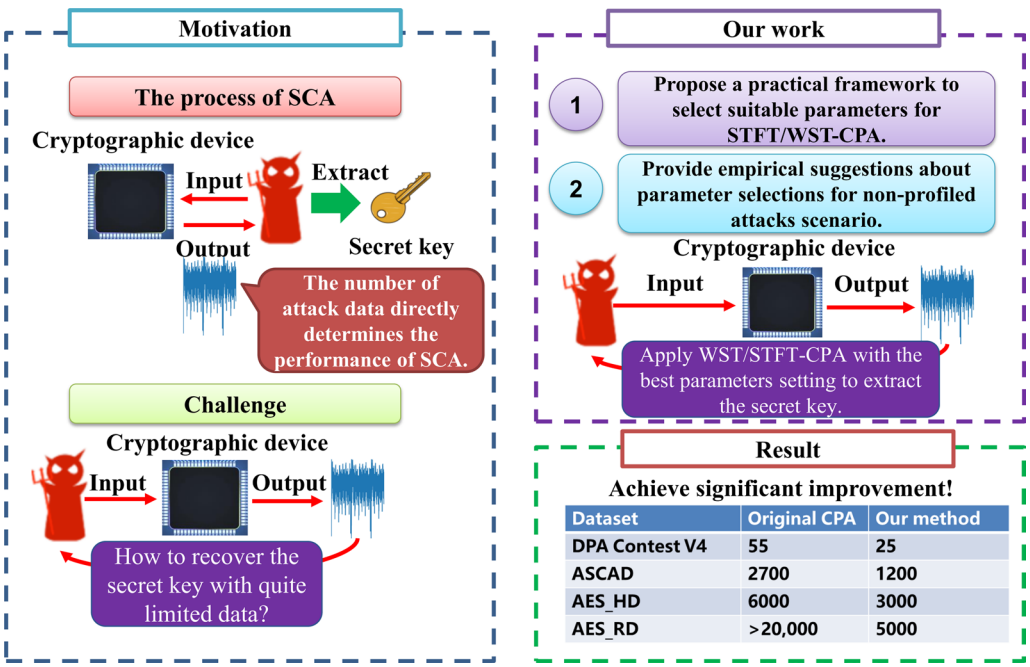

Fig. 1 Graphic summary of our work

power traces are denoted by the vector $\times 1 \in \mathbb{R}^{d1}$, where d1 represents the number of sampling points.

Discrete fourier transform (DFT)

In signal preprocessing, DFT is most popularly used to transform the signal from time domain to frequency domain. DFT can be regarded as the specific projection on periodic signals $\{e^{2ik_1\pi/d_1}\}_{0\le k_1\le d_1-1}$, which is contrary to the analysis done on the Dirac basis. In the scenario of discrete Fourier transformation, the original signal x_1 is considered to consist of periodic signals with infinitesimally small frequency bandwidth. Equation 1 and 2 depict the representations of DFT and inverse DFT respectively. In Eqs. 1 and 2, $\widehat{x_1}$ denotes DFT of x_1, k_1 and p_1 denote the index of time and frequency respectively.

$$\widehat{x_1}(k_1) = (x_1|e^{2ik_1\pi/d_1}) = \sum_{p_1} x_1(p_1)e^{-2ik_1p_1\pi/d_1} \tag{1}$$

$$x_1(p_1) = \frac{1}{d_1}\sum_{k_1}(x_1|e^{2ik_1\pi/d_1})e^{2ik_1p_1\pi/d_1} = \frac{1}{d_1}\sum_{k_1}\widehat{x_1}(k_1)e^{2ip_1k_1\pi/d_1} \tag{2}$$

In practice, the adversary can adopt Fast Fourier Transform (FFT) algorithm to further optimize DFT. Fast Fourier Transform has shown powerful technical potential in the scenario of SCA. For instance, Zhang et al. (2020) applied FFT to enhance the performance of cross homogeneous/heterogeneous device attack. They showed that with the FFT preprocessing method, the adversary can extract the secret key of heterogeneous devices within 1000 traces. However, it is not always the case that FFT can significantly enhance origianl SCA method. For instance, we find that FFT-based CPA has poorer performance than original CPA in the context of analyzing DPA Contest V4 and AES_HD datasets. The Fourier transform has inherent limitations in dealing with non-stationary signals. It can only obtain the components of which frequencies are contained in a signal as a whole, but it can not capture the moment when each component appears. Hence, if the collected traces are non-stationary signals, directly applying FFT may make the performance of CPA attacks even worse. To address the limitation of FFT, we introduce short-time Fourier transform to non-profiled attacks domain.

Short-time fourier transform (STFT)

The central idea of STFT is adding a specific sliding-window function (e.g. Gaussian window function) on the temporal sampling points, and then performing Fourier transformation on the signal inside the window to extract a spectrogram of original signals. Currently, STFT is one of the most popular preprocessing method in the context of analyzing non-stationary signals, which can be applied to obtain the frequency and phase of local time-varying signals. The representation of discrete short-time Fourier transformation can be denoted by Eq. 3. In Eq. 3, $x_1[d_1]$ denotes the temporal signals, $w_1[d_1 - m_1]$ denotes the selected slide-window and m_1 denotes a variable value. In this paper, we select the gaussian window function to perform STFT.

$$STFT[x_1(d_1)](m_1, w_1) = \sum_{m_1=-\infty}^{\infty} x_1[m_1] * w_1[d_1 - m_1]e^{-jw_1m_1} \tag{3}$$

With a smaller-sized window, the adversary can obtain a finer division of time-domain based signals and better time-domain resolution. However, the frequency domain resolution will become worse if the selected window function is a smaller one. In practice, the parameters of STFT, such as the size of the window function w_1, the standard deviation value *std*, and the frequency component f play important roles in frequency-based SCA attacks. The adversary needs to select the above parameters carefully to enhance STFT-based CPA attacks.

Wavelet scatter transform

In the context of analyzing time-varying non-stationary signals, small windows are considered to be suitable for high frequencies while large windows are considered to be suitable for low frequencies (Allen 1977). In practice, the size of window used in STFT kept fixed, and the width is unchangeable during time-frequency transformation. Hence, STFT cannot fully meet the requirements for extracting the frequency components in the context of analyzing unsteady changeable signals. To better extract various frequency components from unsteady changeable signals, Wavelet Transform (WT) (Debande et al. 2012) $\{\psi_{u_1,s_1}\}_{u_1,s_1}$ adopts finite-length decaying wavelet basis $\psi_{u_1,s_1}(t_1) = \frac{1}{\sqrt{s_1}}\psi(\frac{t_1-u_1}{s_1})$ to preprocess the signals, where t_1 denotes the sampling point in the time domain, ψ denotes the mother wavelet, s_1 represents dilation coefficients and u_1 represents translation. The notation of wavelet transformation can be formulated as the following equation:

$$(\widetilde{x_1}|\psi_{u_1,s_1}) = \int \widetilde{x_1}(t_1)\frac{1}{\sqrt{s_1}}\psi^*\left(\frac{t_1-u_1}{s_1}\right)dt_1 = \widetilde{x_1} * \overline{\psi_{s_1}}(u_1) \tag{4}$$

$\widetilde{x_1}$ denotes the original signal, $*$ represents the convolutional operator and x_1^* represents x_1's complex conjugate ($\overline{x_1}(t_1) = x_1^*(-t_1)$).

In the process of WT, dilation coefficient $s_1 = 2^{-j}(j \in \mathbb{N})$ is varied. Given the mother wavelet ψ and corresponding center frequency f_0, the j-th dilated

version of f_0 can be formalized as $\frac{f_0}{2^j}$. In the process of STFT, the origianl signals are concentrated into a fix area - time-frequency boxes $\alpha(t,f)$, expressed as $\alpha(t,f) = \gamma_f(t)\gamma_t(f)$. γ_t and γ_f represent constant temporal support and frequency bandwidth for the window w_1. In WT, the bandwidth γ_f is inversely proportional to the temporal support γ_t when variable parameter s is changing. Unlike STFT, the shape of area $\alpha(t,f)$ keeps variable across the time-frequency domain. Compared with STFT, WT is stable to small deformations but does not have translation invariance, whereas STFT is unstable to small deformation but is robust to translation invariance. To enable WT stable to translation invariant, Mallat et al. (andén and Mallat 2013) proposed wavelet scattering transform (WST). The notation of WST can be formalized as:

$$W_1[\lambda]x_1(u_1) = x_1 * \overline{\psi_\lambda} = \int x_1(t_1)\frac{1}{\sqrt{\lambda}}\psi^*\left(\frac{u_1 - t_1}{\lambda}\right) dt_1 (x_1 \in \mathbf{L}^2(R), \psi \in \mathbf{L}^2(R)) \tag{5}$$

where t_1 denotes the sampling point in the time domain, u_1 represents translation, x_1 denotes the original signal, $*$ and ψ denote convolutional operator and the mother wavelet respectively. The wavelet ψ_λ is composed of scale parameters λ that are applied to the non-linear operation $|.|$ and averaged on the time-domain of 2^{j1} signals with $A_{j1}x_1 = x_1 * \phi_{2^{j1}}$. Given the path $p1 = (\lambda_1, ..., \lambda_m)$ with $\lambda_i > 2^{-j1}$, the windowed scattering transform S_{j1} of the time-domain signals x_1 can be formalized as:

$$\begin{aligned} S_{j1}[p]x &= ||||x_1 * \psi_{\lambda_1}| * \psi_{\lambda_2}| ... * \psi_{\lambda_m}| * \phi_{2^{j1}} \\ &= |W[\lambda_m]...W[\lambda_2]|W[\lambda_1]x_1||| * \phi_{2^{j1}} \\ &= A_{j1}|W[\lambda_m]...W[\lambda_2]|W[\lambda_1]x_1||| \\ &= A_{j1}U[\lambda_m]...U[\lambda_2]U[\lambda_1]x_1 \end{aligned} \tag{6}$$

where $U[\lambda]x_1 = |W[\lambda]x_1| = |x_1 * \psi_\lambda|$. In practice, S_{j1} is calculated on the path subset $\Omega_{j1,m}$, where m denotes the maximum length of paths $p \in \Omega_{j1,m}$ and λ satisfies $\lambda > 2^{-j1}$ (andén and Mallat 2013). In the scenario of WST, the wavelet transform only captures specific frequency components that are superior than 2^{-j1}, and the rest frequency components are captured by $\phi_{2^{j1}}$. In the python software-based implementation of WST (Andreux et al. 2020), the wavelets are used on dyadic scales $2^{-j}(0 \le j < J)$ or on intermediate scales $2^{\frac{-j}{Q}}(0 \le j < JQ)$, in which Q denotes the amount of wavelet by an octave. In practice, the WST is composed of three parameters, such as the scale $2^J(J \ge 1, J \in N)$ of signals for averaging, the octave $Q(Q \ge 1, Q \in N)$ and the number of levels of the scattering transform $m \in [1, 2]$. Previous works showed that WST can provide stability over time-translation invariant (andén and Mallat 2013) and can achieve a satifactory improvement in the case of profiled attack against jitter-protected implementations of AES (Destouet et al. 2021) when the parameters are properly selected. However, these works mainly focus on an idealized scenario that adversaries can fully control cloned devices. They can empirically select the parameters in the profile stage by exploiting sensitive information about the cryptographic implementation, whereas in non-profiled attacks scenario, it is difficult to empirically select suitable parameters as the adversary has no detailed information about the analysis target prior to the attack. Besides, the parameter settings vary according to different cryptographic implementations. In this case, the adversary needs to consider designing a practical framework to select suitable parameters for WST. In this paper, we focus on providing suitable parameters for $\{j, q, f\}$ $(m = 2)$ to enhance the performance of WST-based CPA attacks, where $j \in J$, $q \in Q$ and f represents the extracted frequency component.

A practical framework for frequency-based CPA attack

Our method

CPA is currently the most popular non-profiled side-channel analysis method. Focusing on the CPA performance optimization, we design a practical framework to provide suitable parameters for WST/STFT-based non-profiled CPA. The general measurement setup for WST/STFT-based non-profiled CPA can be illustrated in Fig. 2. In this paper, we aim to solve two challenges where previous works (Udvarhelyi et al. 2021; Destouet et al. 2021; Gebotys et al. 2005; Belgarric et al. 2014) do not investigate in depth:

- How to select suitable parameters for WST/STFT-CPA attacks? From "Short-time fourier transform (STFT)" and "Wavelet scatter transform" sections, we can learn that the parameter $\{j, q, f\}$ and $\{w, std, f\}$ directly determine the performance of WST-CPA and STFT-CPA attacks. With suitable parameters setting, WST/STFT-CPA can significantly enhance the performance of original CPA attacks. In practice, the value of suitable parameters setting varies according to discrete cryptographic implementations.
- How to evaluate the quality of extracted frequency components? In non-profiled attack scenario, the secret key, the intermediate value and implementation details kept secret prior to the attack. To find the best parameters setting, the adversary needs to select or design suitable and reliable metrics to evaluate the quality of extracted frequency components. The met-

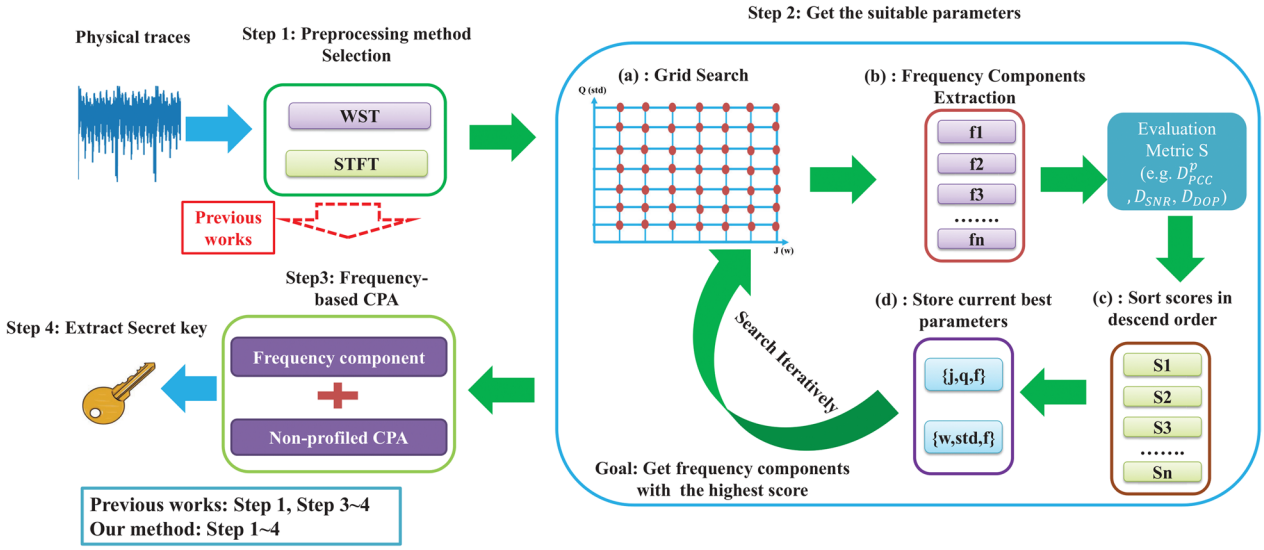

Fig. 2 The overall framework for selecting parameters for STFT/WST-based CPA attacks

rics should be closely related to SCA metric, such as Success Rate (SR), Guess Entropy (GE) or minimum number of attack data n_{min} (Standaert et al. 2009).

In an ideal scenario, the adversary is assumed to have sufficient attack data and know detailed information about cryptographic implementations, such as secret key, noise level and the characteristics of collected signals. In this case, the adversary can directly obtain suitable frequency components using prior knowledge about the cryptographic implementations. Hence, precious works do not consider how to select suitable parameters for WST/STFT-CPA attacks and evaluate the quality of extracted frequency components in non-profiled attacks scenario. In reality, the cryptographic implementations details are usually kept secret to the public. Designers may even adopt countermeasures, such as key rolling schemes, to limit the adversary's attack ability. In this case, direct extraction of the suitable parameters becomes impossible. Designing a practical framework to properly select suitable parameters becomes the most paramount thing in the case of enhancing WST/STFT-CPA attacks with insufficient data. To address this issue, we introduce the concept of grid-search method from deep-learning (DL) domain to non-profiled SCA domain, to select suitable parameters for WST/STFT-CPA attacks. Grid search (Pontes et al. 2016) method is one of the most popular hyperparameters tuning method in machine learning domain. It can efficiently work when the parameter categories and attack data are not quite large. In DL domain, grid search is applied to search the suitable hyperparameters, such as learning rate and network architecture, for neural network models. The adversary updates the hyperparameters with fixed sizes according to the feedbacks from accuracy or loss value. Unlike traditional DL methods, we apply grid-search method to search suitable parameters for WST/STFT. Figure 3 provides an example for grid-search used in WST/STFT. The overall process of grid method used for parameters selection can be divided into four steps:

(1) Design the evaluation metric D, evaluate the quality of original data D(T) and assign temp = D(T).

(2) Search every possible parameters setting {j, q} or {w, std} for WST/STFT. Preprocess the physical

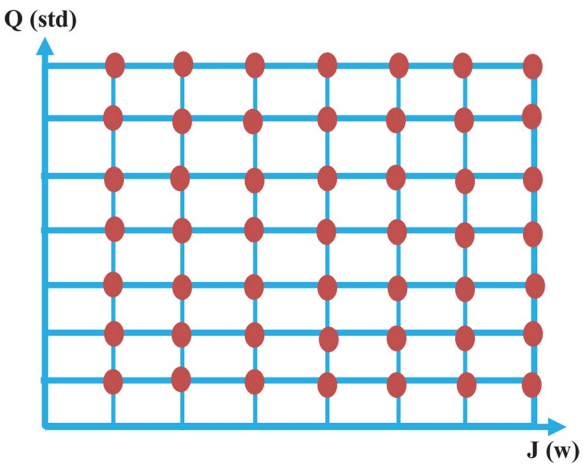

Fig. 3 An example for grid-search used in WST/STFT

traces T with the selected parameters setting $\{j1, q1\}$ ($\{w1, std1\}$), and then extract corresponding frequency components $\{f_1, f_2, f_3, ..., f_n\}$.

(3) Evaluate the quality of extracted frequency components with the designed metric D:

$$\begin{aligned} S_1 &= D(f_1) \\ S_2 &= D(f_2) \\ S_3 &= D(f_3) \\ &\ldots \\ S_n &= D(f_n) \end{aligned} \tag{7}$$

Sort the evaluation scores in descending order. Compare temp and S_1. If $S_1 >$ temp, set temp $= S_1$ and store the parameters setting $\{j1, q1, f1\}$ ($\{w1, std1, f1\}$) with the highest scores.

(4) Execute step (2) and step (3) iteratively. Finally get the parameters setting $\{j1, q1, f1\}$ ($\{w1, std1, f1\}$) with the highest scores.

To evaluate the quality of extracted frequency components f, we adopt Pearson Correlation Coefficient (PCC), Signal-to-Noise Ratio (SNR) and Absolute-Differences-Of-PCC (DOP) as main evaluation metrics to perform grid-search method. The notation of PCC $D^p_{PCC}(f)$ can be formalized as:

$$D^p_{PCC}(f) = \max_{k^* \in K} |\rho(f, L^p_{k^*})| \tag{8}$$

where $L^p_{k^*}$ represents the hypothesis power assumption, p denotes ciphertext or plaintext, k^* denotes the hypothesis key and ρ represents PCC value. The parameter $L^p_{k^*}$ can be further formalized as: $L^p_{k^*} = h(F_1(k^*, p_1), F_2(k^*, p_2)..F_n(k^*, p_n))$, where F represents sensitive cryptographic operation (e.g. AES SubBytes operation) and h denotes the selected leakage model. Let m1 and m2 denote the minimum number of attack data (n_{min}) to perform a successful CPA for frequency components f_1 and f_2 respectively. We have

$$m1 < m2 \tag{9}$$

when

$$D^p_{PCC}(f_1) > D^p_{PCC}(f_2) \tag{10}$$

According to the theorem in Mangard et al. (2007), there exists theoretical linear relationship between n_{min} and ρ: $n_{min} = \frac{28}{\rho^2}$. If it satisfies Eq. 10, we can infer that

$$\frac{1}{D^p_{PCC}(f_1)} < \frac{1}{D^p_{PCC}(f_2)} \tag{11}$$

$$\frac{28}{(D^p_{PCC}(f_1))^2} < \frac{28}{(D^p_{PCC}(f_2))^2} \tag{12}$$

When it satisfies Eq. 12, we can directly infer that $m1 < m2$. As a result, using the f_1 frequency component, the adversary can extract the secret key with fewer traces.

Like PCC evaluation metric, we have

$$m1 < m2 \tag{13}$$

when

$$D_{SNR}(f_1) > D_{SNR}(f_2) \tag{14}$$

The notation of SNR evaluation metric $D_{SNR}(f)$ can be formalized as:

$$D_{SNR}(f) = SNR(f) \tag{15}$$

In SCA domain, the sampling point of physical traces L_{total} can be formalized as $L_{total} = L_{exp} + L_{noise}$, where L_{exp} denotes exploitable physical leakages and L_{noise} represents noise components. The relationship between SNR and ρ satisfies (Mangard et al. 2007): $\rho(h, L_{total}) = \rho(h, L_{exp} + L_{noise}) = \frac{\rho(h, L_{exp})}{\sqrt{1 + \frac{1}{SNR}}}$, where h denotes the hypothesis leakage value. WST/STFT methods can be regarded as a special noise reduction method to reduce L_{noise}. SNR will increase if L_{noise} becomes smaller. In theory, a higher SNR leads to a higher ρ value. If it satisfies Eq. 14, we can infer that

$$1 + \frac{1}{D_{SNR}(f_1)} < 1 + \frac{1}{D_{SNR}(f_2)} \tag{16}$$

$$\frac{1}{\sqrt{1 + \frac{1}{D_{SNR}(f_1)}}} > \frac{1}{\sqrt{1 + \frac{1}{D_{SNR}(f_2)}}} \tag{17}$$

$$\frac{\rho(h, L_{exp})}{\sqrt{1 + \frac{1}{D_{SNR}(f_1)}}} > \frac{\rho(h, L_{exp})}{\sqrt{1 + \frac{1}{D_{SNR}(f_2)}}} \tag{18}$$

When it satisfies Eq. 18, we can infer that $\rho_{f_1} > \rho_{f_2}$. According to the equation $n_{min} = \frac{28}{\rho^2}$, we can infer that $m1 < m2$. Hence, the adversary can also adopt $D_{SNR}(f)$ to directly measure the quality of extracted frequency components.

In addition to adopting $D^p_{PCC}(f)$ and $D_{SNR}(f)$ evaluation metrics, we also consider DOP $D_{DOP}(f)$ as an alternative evaluate metric to assess the quality of extracted frequency components. The notation of $D_{DOP}(f)$ can be formalized as

$$D_{DOP}(f) = \left| \frac{\rho_{K1}(h, f) - \rho_{K2}(h, f)}{\rho_{K2}(h, f)} \right| \tag{19}$$

where $\rho_{K1}(h, f)$ and $\rho_{K2}(h, f)$ represent the first and second maximum PCC value respectively. We have

$$m1 < m2 \tag{20}$$

when

$$D_{DOP}(f_1) > D_{DOP}(f_2) \tag{21}$$

In the process of CPA attacks, the PCC differences between the PCC of the correct key (with the maximum PCC) and other hypothesis key will become larger when the number of collected data is increasing or the quality of the collected data is significantly improved. Hence, given the same number of attack data, the frequency component f_1 is considered to lead a better SCA performance than f_2, when it satisfies Eq. 21.

Alg. 1 and Alg. 2 summarize the process of selecting suitable parameters with $D^{p}_{PCC}(f)$ metric for WST/STFT-based CPA attacks respectively. $WST_{j,q}(T)$ denotes the processed traces with WST method, while $STFT_{w,std}(T)$ represents the preprocessed traces with STFT method. The adversary utilizes $D^{p}_{PCC}(f)$ to calculate the score of original traces $S_{ori} = D^{p}_{PCC}(T)$, and then assign $temp = S_{ori}$. The adversary searches the whole parameters setting with grid-search method, and calculates D^{p}_{PCC} of each extracted frequency component $D^{p}_{PCC}(f^*)$. If $D^{p}_{PCC}(f^*)$ satisfies $D^{p}_{PCC}(f^*) > temp$, assign $D^{p}_{PCC}(f^*)$ to the variable parameter temp. The adversary performs the process iteratively to obtain the best parameters setting. The processes of selecting suitable parameters for WST/STFT-based CPA with SNR and DOP are also similar to Alg. 1 and Alg. 2. The adversary just needs to replace the evaluation metric D^{p}_{PCC} with D_{SNR} or D_{DOP} respectively. Hence, the detailed processes of selecting suitable parameters for WST/STFT-based CPA attacks with other evaluation metrics are not given here. The adversary utilizes the designed evaluation metrics to search the suitable parameters settings iteratively, and then performs CPA on the processed physical traces with the best-selected parameters setting.

Algorithm 1 Selecting suitable parameters for WST-based CPA with the evaluation metric PCC (WST-PCC-based CPA)

Input: trace T, plaintext p, finite set J1 and Q1
Output: J,Q,f

$S_{ori}=D^{p}_{PCC}(T)$
$temp = S_{ori}$
$J = 1, Q = 1, f = 0$
while $j \in J1$ **do**
  while $q \in Q1$ **do**
    $F = \{f_1, f_2, f_3, ...f_n\} = WST_{j,q}(T)$
    while $f^* \in F$ **do**
      $S_{f^*}=D^{p}_{PCC}(f^*)$
      if $S_{f^*} > temp$ **then**
        $J = j, Q = q, f = f^*$
        $temp = S_{f^*}$
      end if
    end while
  end while
end while

Algorithm 2 Selecting suitable parameters for STFT-based CPA with the evaluation metric PCC (STFT-PCC-based CPA)

Input: trace T, plaintext p, finite set W and STD
Output: f,w,std

S_{ori}=D^{p}_{PCC}(T)
temp = S_{ori}
f = 0,w = 0,std = 0
while w1 ∈ W **do**
  while std1 ∈ STD **do**
    F = {$f_1, f_2, f_3, ...f_n$} = $STFT_{w1,std1}$(T)
    while f* ∈ F **do**
      S_{f*}=D^{p}_{PCC}(f*)
      if S_{f*} > temp **then**
        f = f*,w = w1,std = std1
        temp = S_{f*}
      end if
    end while
  end while
end while

Further optimization. Although grid-search provides a straightforward way to achieve suitable hyperparameters selection in non-profiled attacks scenario, it requires a lot of time when the number of attack data or parameters is large. The adversary needs to search all parameter settings to obtain suitable parameters setting, which inevitably brings additional time overhead. To further optimize grid-search-based SCA, we propose to apply halving-grid search method to accelerate the attack. The overall process of halving-grid search method for WST/STFT-CPA can be divided into six steps:

(1) Measure the distribution of original dataset T by calculating the distribution of single-byte plaintext value. Select a smaller size dataset T_1 from original dataset T, where the distribution of T_1 is nearly the same as T.
(2) Apply grid-search method to search the whole parameters setting {j, q} or {w, std}, and then calculate corresponding scores S with designed evaluation metrics (e.g. D^{p}_{PCC}, D_{SNR} and D_{DOP}).
(3) Sort the evaluation values in descending order, and eliminate the last half of the parameters {j, q} or {w, std} according to the sorted values S.
(4) The adversary selects new subset T2 from the remaining dataset T2 = T − T1, where T2 is twice as large as T1. The adversary searches the rest parameters setting {j, q} or {w, std} to calculate corresponding evaluation values. Sort the evaluation values in descending order, and eliminate the last half of the remaining parameters {j, q} or {w, std} according to the sorted values S.
(5) Repeat step (4) iteratively until the remaining dataset is not enough or there is only 1 group of parameters left.
(6) Apply grid search to obtain the best-performance parameters.

How to apply grid-search or halving-grid search method in non-profiled attacks scenario? Compared with original grid-search method, halving-grid search method can efficiently reduce time-overhead in the case of searching parameters. In this paper, we find that applying halving-grid search method can achieve nearly the same performance as grid-search method, when the initial selected data is set to around one-third number of original dataset. This kind of method certainly can be applicable for other similar preprocessing methods in the case of enhancing non-profiled SCA attacks. However, halving-grid search method does not always have absolute superior performance than grid-search method in the case of performing non-profiled SCA attacks. Halving-grid search method has inherent limitations when the number of attack number is quite small. Evaluators may eliminate the suitable parameters wrongly if the number of selected data is extremely small. As a suggestion, we recommend applying grid-search method to perform non-profiled SCA attacks when the number of attack data is insufficient. When the number of attack data or parameters is large, we suggest applying halving-grid search method to perform non-profiled SCA. In this paper, we aim to enhance CPA in the case of insufficient attack data. Hence, we adopt grid-search method as the main method to perform the attack.

Parameters setting

To efficiently enhance the performance of CPA attacks, it is vital to select suitable parameters for WST/STFT. We refer to the operation mode of grid-search method to select the best-performance parameters according to feedbacks from the designed evaluation metrics (Eqs. 8, 15 and 19). The finite set $J1$ and $Q1$ used for WST-based CPA are designed as follows: $J1 \in [1, 8]$ and $Q1 \in [2, 6]$. In the scenario of STFT-based CPA attacks, the finite set W and STD are designed as follows: $W \in \{0.01L, 0.02L, 0.04L, 0.08L, 0.1L\}$ and $STD \in \{0.25, 0.5, 1, 2, 4, 8, 16\}$, where L denotes the length of sampling points. In this section, FFT-based CPA is regarded as a special type of STFT-based CPA attack.

To make our work reproducible, we utilize opened-source framework to implement the following preprocessing method. The FFT preprocessing method is implemented through Numpy.FFT.FFT (Numpy 2022) function while the STFT preprocessing method is implemented through Scipy.signal.STFT (Scipy 2022) function. We adopt Kymatio.numpy.Scattering1D function from Andreux et al. (2020) to implement WST-based CPA attacks. Besides, we use the analysis of the variance as an alternative method (Bubberman et al. 2020) to measure the SNR of the physical traces, as the intermediate value kept secret to the adversary in non-profiled attacks scenario.

Experiment results

To access the effectiveness of our proposed attack framework (Figs. 2 and 3), we present practical attacks on four public datasets, including DPA Contest V4 (DPA_Contest_v4 2014), AES_HD (AES_HD 2018), AES_RD (AES_RD 2017) and ASCAD (2018) datasets. The practical attack results show that with the proposed attack framework, the WST/STFT-based CPA attack achieves more robust performance. Compared with the original CPA method, the number of attack traces can be reduced by 50–95%.

Public datasets

Four various public datasets covering main types of SCA scenarios are adopted in our experiment. The first one is a software-based unprotected implementation of AES, which represents an ideal scenario that the noise level is quite low and adversaries can use limited data to successfully extract the secret key. The second one is also a unprotected implementation of AES but with high-level noises. As a consequence, the adversary needs to collect a lot of data to break the device. The third dataset adopts random delay countermeasure that is a typical hiding countermeasure and has been widely used in various commercial crypto products (e.g. commercial contactless/contact smart cards (Kim et al. 2012)). Finally, the last dataset adopts first-order boolean masking that is currently the most popular side-channel countermeasure in SCA community nowadays. Detailed information about the public datasets are as follows:

(1) DPA Contest V4 dataset (DPA_Contest_v4 2014). It measures EM leakages of first-order boolean masked implementation of AES (Nassar et al. 2012). In this paper, the mask value is assumed to be known prior to non-profiled attacks, turning the protected implementation to the unprotected one. The notation of the intermediate value is formalized as follows:

$$Y(K^*) = S(P_i \oplus K^*) \oplus \underbrace{M}_{known-mask} \tag{22}$$

where K^* denotes the secret AES key, Y represents the targeted intermediate value, P_i represents the i-th byte of plaintext and S represents AES SubBytes operation. The maximum of measured SNR is up to 5.8577. We target the first byte of Y and select the hamming weight (HW) leakage model to perform SCA.

(2) AES_HD (AES_HD 2018). AES_HD dataset provides EM measurements of paralled implementation of AES. The AES-128 is hardware-based implemented on the Xilinx Virtex-5 FPGA. AES_HD does not adopt side-channel countermeasures. The maximum of measured SNR is up to 0.0096. The notation of the intermediate value is formalized as follows (Kim et al. 2019):

$$Y(K^*) = \underbrace{S^{-1}(C_a \oplus K^*)}_{previous-register-value} \oplus \underbrace{C_b}_{ciphertext-byte} \tag{23}$$

where K^* denotes the secret AES key, S^{-1} represents inverse AES SubBytes operation, C_a denotes a-th byte of ciphertext and C_b denotes b-th byte of ciphertext. The relationship between a and b can be extracted through the inverse AES ShiftRows operation. Like previous works (Picek et al. 2019; Kim et al. 2019), we select $a = 12$ resulting in $b = 8$ to perform the attack as it is one of the easiest intermediate value byte to recover. In the context of analyzing AES_HD dataset, we select hamming distance (HD) as the main leakage model to present non-profiled attacks as HD is suitable in the scenario of analyzing paralleled implementations.

(3) AES_RD dataset (AES_RD 2017). AES_RD provides power measurements of protected software-based implementation of AES (Coron and Kizhvatov

2010). The random delay countermeasure is implemented on an 8-bit AVR platform. The notation of the intermediate value is formalized as follows:

$$Y(K^*) = S(P_i \oplus K^*) \tag{24}$$

where K^* denotes the secret AES key, Y represents the targeted intermediate value, P_i represents the i-th byte of plaintext and S represents AES SubBytes operation. The maximum of measured SNR is up to 0.0556. We target the first byte of Y and select the HW leakage model to perform the SCA.

(4) ASCAD dataset (ASCAD 2018). ASCAD dataset adopts first-order boolean masking (Benadjila et al. 2020) to resist side-channel attack. The ATmega8515 microcontroller (8-bit AVR) provides the platform for running the masked AES algorithm, and corresponding measurements are made by using EM leakages. The notation of the intermediate value is formalized as follows:

$$Y(K^*) = S(P_i \oplus K^*) \oplus Mask_{out} \tag{25}$$

where K^* denotes the secret AES key, Y represents the targeted intermediate value, $Mask_{out}$ represents the output mask value, P_i represents the i-th byte of plaintext and S represents AES SubBytes operation. The maximum of measured SNR is up to 0.8. In this paper, we target the third byte of Y and perform 2nd-order CPA attacks (Rivain et al. 2009) with the HW model.

Practical attacks on public datasets

Based on the designed "Parameters setting" section, we apply the proposed framework in ("A practical framework for frequency-based CPA attack" section, Figs. 2 and 3) to enhance original CPA attacks. In this paper, we mainly plot the performance of WST/STFT-(PCC,SNR)-based CPA attacks, as WST/STFT-DOP-based CPA has nearly the same performance as WST/STFT-PCC-based CPA attacks. We adopt success rate (SR) (Standaert et al. 2009) as the main SCA evaluation metric to systematically compare the performance of WST/STFT-based CPA and the original CPA method in the context of analyzing DPA-Contest V4, AES_HD, AES_RD and ASCAD datasets. The practical attacks are repeated 100 times on average to calculate the value of SR.

In the scenario of analyzing DPA Contest V4 dataset, we select 500 traces to perform non-profiled attacks. Figures 4 and 5 depict the performance of WST/STFT-based CPA attacks on DPA Contest V4 dataset respectively. The proposed attack framework achieves more robust performance. With the best-selected parameters, WST/STFT-based CPA enables adversaries to extract the secret key within 25 traces, while original CPA methods require 55 traces at least to achieve a successful non-profiled CPA attack. Besides, we find that FFT-based CPA does not always have the amazing performance in the context of SCA. Sometimes, it might make the performance of original CPA attacks even worse, as shown in Fig. 5.

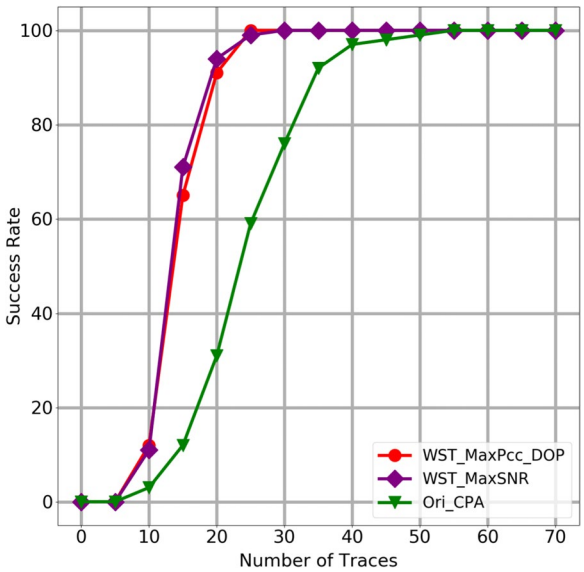

Fig. 4 The performance of WST-based CPA and original CPA attack on DPA Contest V4 dataset

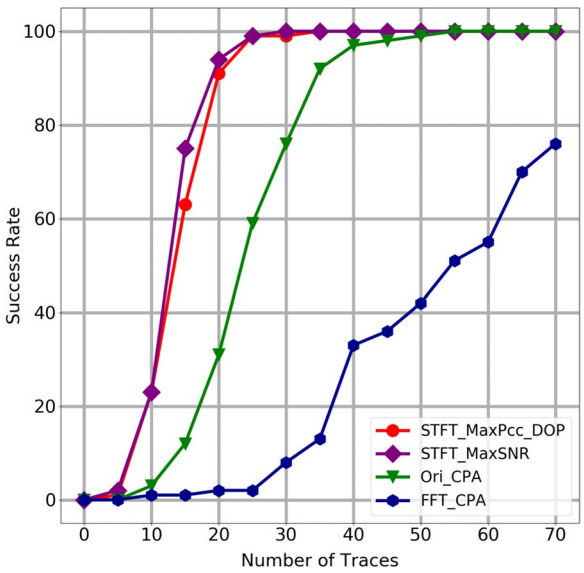

Fig. 5 The performance of STFT-based CPA and original CPA attack on DPA Contest V4 dataset

In the context of analyzing AES_HD dataset, we select 9000 traces to perform non-profiled attacks. Figures 6 and 7 plot the performance of WST/STFT-based CPA attacks on AES_HD dataset respectively. As expected, the proposed method can also efficiently enhance non-profiled attacks in the case of analyzing hardware-based cryptographic implementation. Using the proposed attack framework (Alg. 1 and Alg. 2), WST-based CPA attacks can reduce the number of attack data from 6000 to 3000 while STFT-based CPA attacks reduce the number of attack data from 6000 to 4000. Compared with STFT-based CPA attacks, WST-based CPA attacks have relative superior performance in the case of analyzing AES_HD dataset. Besides, DOP/PCC-WST/STFT-based CPA leads to a better performance than SNR-WST/STFT-based CPA, as shown in Figs. 6 and 7. Although SNR-WST-based CPA is also able to efficiently improve the performance of CPA attacks, DOP/PCC-WST/STFT-based CPA allows extracting the secret key within fewer traces. Similar to Fig. 5, FFT-based CPA has poorer performance in the context of analyzing AES_HD dataset, as shown in Fig. 7.

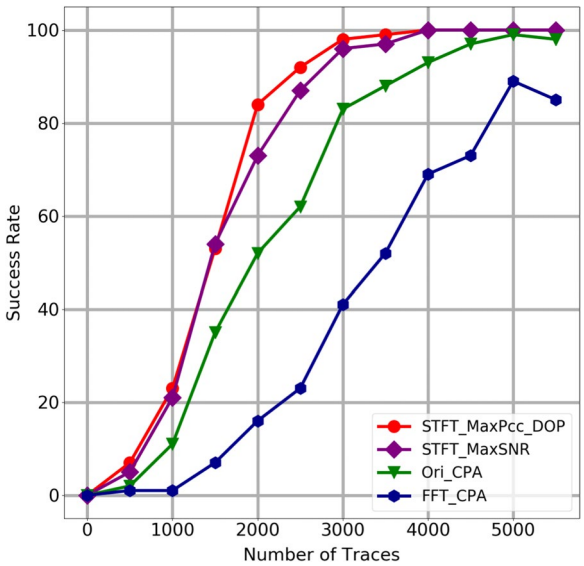

Fig. 7 The performance of STFT-based CPA and original CPA attack on AES_HD dataset

In the process of attacking ASCAD dataset, we select 5000 traces to perform non-profiled attacks. The original data is preprocessed with window compress preprocessing method, reducing the dimension of the original sampling point from 700 to 70. After data dimension, we apply WST and STFT to preprocess the processed traces and then perform 2nd-order CPA attacks (Rivain et al. 2009) subsequently. Figures 8 and 9 plot the performance of WST/STFT-based CPA attacks on ASCAD dataset respectively. As expected, our proposed attack framework can efficiently work in the context of analyzing ASCAD dataset and FFT-based CPA still has the poorest performance. With the proposed attack framework (Alg. 1 and Alg. 2), WST/STFT-based CPA attack can

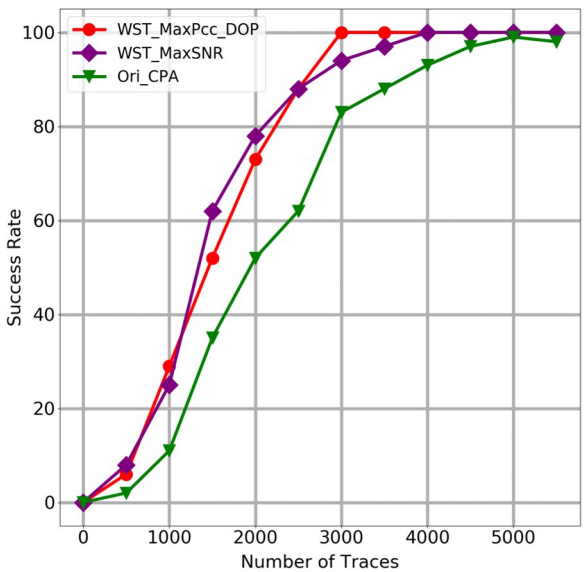

Fig. 6 The performance of WST-based CPA and original CPA attack on AES_HD dataset

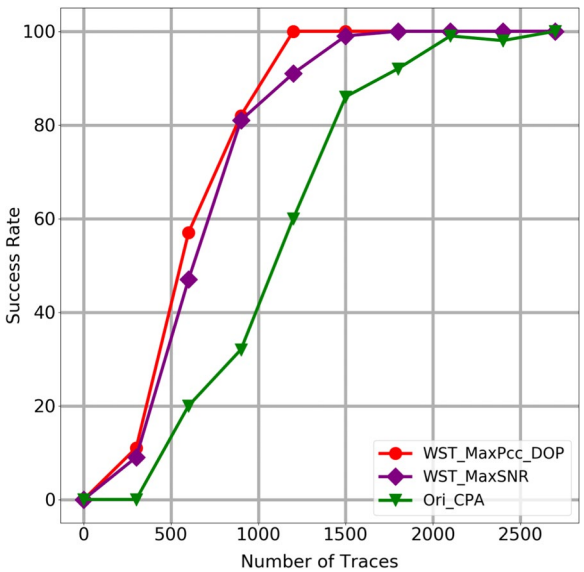

Fig. 8 The performance of WST-based CPA and original CPA attack on ASCAD dataset

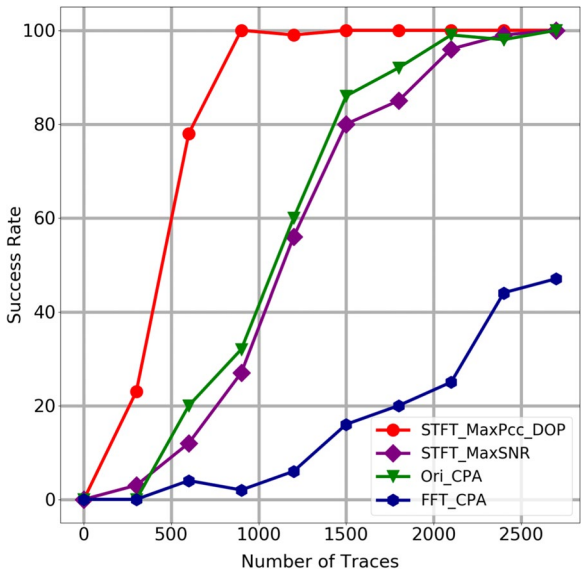

Fig. 9 The performance of STFT-based CPA and original CPA attack on ASCAD dataset

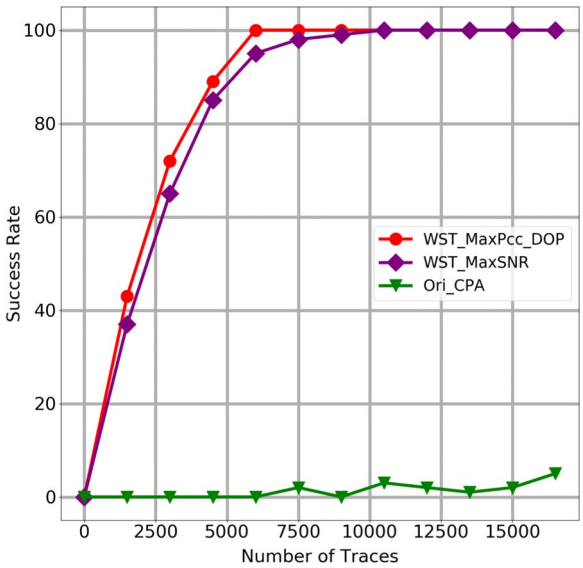

Fig. 10 The performance of WST-based CPA and original CPA attack on AES_RD dataset

successfully extract the AES key within 1200 traces while original methods require 2700 traces at least to achieve successful 2nd-order CPA attacks. From Figs. 8 and 9, we can learn that PCC/DOP evaluation metrics have better performance than SNR evaluation metric in the scenario of WST/STFT-based CPA attacks. In the context of STFT-based-CPA attacks, SNR-STFT-based CPA makes the performance of original method even worse.

In the scenario of analyzing AES_RD dataset, we select 20,000 traces to perform non-profiled attacks. Unlike analyzing three previous datasets, FFT-based CPA attacks achieve the best performance in the context of attacking AES_RD dataset, as shown in Figs. 10 and 11. The adversary can successfully extract the secret key within 3000 traces by FFT-based CPA attacks while WST-based CPA attacks require 5500 traces to achieve successful non-profiled CPA attacks. Compared with original CPA attacks, WST/FFT-based CPA attacks achieve more robust performance. The number of attack data can be reduced by 95% at least. Compared with SNR evaluation metric, PCC/DOP evaluation metrics achieve better performance in the scenario of WST-based CPA attacks. Besides, we find that STFT-based CPA attacks do not efficiently enhance CPA attacks in the case of analyzing AES_RD dataset. Origianl CPA methods cannot achieve 20% success rate even though the number of attack data is increased to 18,000.

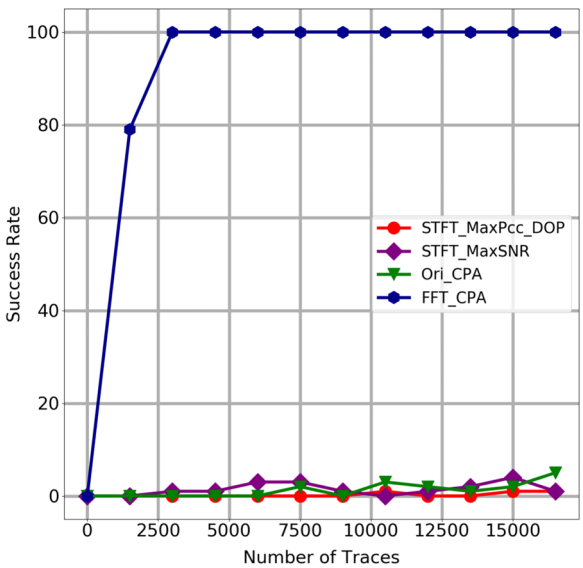

Fig. 11 The performance of STFT-based CPA and original CPA attack on AES_RD dataset

Comparing the proposed method with other popular preprocess methods

From "Practical attacks on public datasets" section it can be inferred that our method is able to effectively enhance original CPA attacks. However, it is uncertain whether our proposed method has superior performance than other popular preprocess methods (Bruneau et al. 2015;

Destouet et al. 2021; Yang et al. 2017; Riscure 2021) in the case of analyzing public datasets. Hence, we conduct a comparative experiment to investigate the performance of these preprocess methods (Bruneau et al. 2015; Destouet et al. 2021; Yang et al. 2017; Riscure 2021) in the scenario of analyzing public datasets. In this section, we select Principal Component Analysis (PCA) (Bruneau et al. 2015), Non-negative Matrix Factorization (NMF) (Yang et al. 2017), Ensemble method with WST (Destouet et al. 2021), Lowpass-filter and Moving Average (Implemented by Riscure (2021)) as main methods to perform the attack. These methods are currently the most popular preprocess methods, and they can be easily reproduced by scikit-learn (Fabian Pedregosa et al. 2020) and kymatio (Andreux et al. 2020) library. Unlike the research (Destouet et al. 2021), we apply the central idea of Ensemble method with WST to non-profiled scenarios. We aim to investigate whether their proposed method can efficiently work in non-profiled attacks scenario. The motivation of this study is certainly not to deform or replicate previous studies. Instead, our goal is to provide some practical insight into the selection of preprocessing methods to enhance the performance of non-profiled attacks.

During the experiment, the number of components used in PCA/NMF is set to 10–40 and we select the parameter that leads to best SCA performance to perform the attack. To fairly compare Ensemble method with WST (Destouet et al. 2021) and our proposed method, Ensemble method with WST uses the same best-selected parameters ("Practical attacks on public datasets" secton). In this section, WST-CPA adopts PCC as the main evaluation metric. Figure 12 plots SR results of our method and other preprocess methods. As expected, our method is more generic and effective than other preprocess method. From Fig. 12, it can be learned that directly apply dimension reduction techniques (Bruneau et al. 2015; Yang et al. 2017) might make the performance of original CPA method even worse. Although previous works (Bruneau et al. 2015; Yang et al. 2017) discover that using PCA/NMF can enhance the performance of CPA in the case of analyzing cryptographic implementations, it does not have amazing performance as Bruneau et al. (2015); Yang et al. (2017) say in the context of analyzing four public datasets. Researchers need to conduct more investigations to further optimize NMF/PCA-CPA attacks. Besides, we find that Ensemble method with WST (Destouet et al. 2021) cannot efficiently enhance the performance of non-profiled attacks. It makes the performance of original CPA attacks worse in the context of analyzing DPA Contest V4, AES_HD and ASCAD datasets. In general, our method is more generic and effective in the scenario of enhancing the performance of CPA attacks.

Summary of the attack framework

To assess extendability and applicability of our method, we present practical attacks on four different cryptographic implementations. With suitable parameters, WST-based CPA and STFT(FFT)-based CPA attacks achieve more robust performance. Compared with original CPA attacks, the attack method can reduce the number of attack data by 50–95% which allows adversaries to extract the secret key within much fewer data.

Through the above comparative experiments, we can learn that WST-based CPA attacks have superior performance than STFT-based CPA attacks in term of stability. With suitable parameters, WST-based CPA attacks can effectively enhance the performance of non-profiled attack in the case of analyzing four datasets while STFT-based CPA attacks do not efficiently work in the context of analyzing AES_RD dataset. Although FFT-based CPA attack achieves the best performance when analyzing AES_RD dataset, it makes the performance of original attack methods worse in the scenario of analyzing the rest three datasets. We speculate the main reason is that the critical information in side-channel traces is contained in transient patterns, of which corresponding signals are non-stationary. As a consequence, the critical information in SCA's traces is not well captured by FFT method. The adversary needs to introduce STFT method to address the limitation of FFT while WST can efficiently work in the scenario of analyzing non-stationary signals. Besides, we find that DOP/PCC evaluation metrics have superior performance than SNR evaluation metric in the context of proposed attack framework. DOP/PCC evaluation metrics allow adversaries to extract the secret key with fewer traces in the scenario of analyzing AES_RD and AES_HD datasets. As a suggestion, we recommend selecting DOP/PCC as the primary evaluation metrics when applying the proposed attack framework.

Countermeasures. Through the practical experiment results, it can be learned that the proposed method can achieve significant improvements in the case of analyzing masking and random delay countermeasures. Designers need to consider the threats of our proposed method when implementing their cryptographic designs, especially designing random delay countermeasures. Current random delay countermeasures are mostly applied to resist time-domain based SCA. They can misalign the sampling points in the time domain to increase the difficulty to perform a successful time-domain based SCA. However, they do not ensure the sampling points that are transformed in the frequency domain are also misaligned.

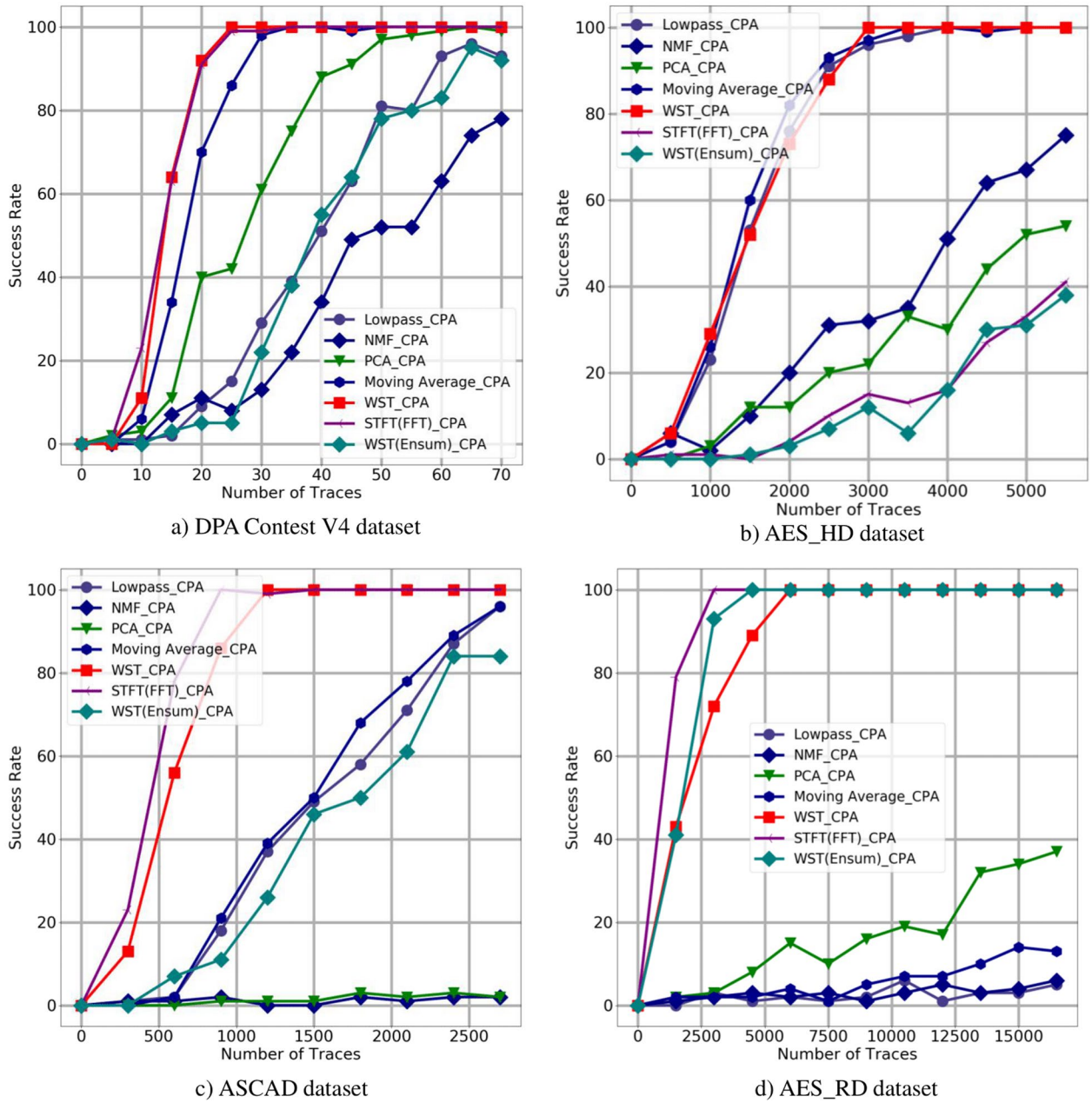

Fig. 12 The performance of our method and other popular preprocess methods

Figure 10 and Fig. 11 plot the performance of frequency-based CPA in the case of analyzing random delay countermeasures. From Figs. 10 and 11, we can learn that random delay cannot efficiently resist frequency-based CPA attacks. The adversary can successfully recover the secret key with quite limited power traces by WST/FFT-CPA attacks. WST/FFT-CPA attacks do not even require additional align techniques to preprocess the traces. Designers need to additionally consider how to misalign the sampling points in the frequency domain when designing random delay countermeasures. In addition, designers also need to consider the threats of our proposed method when designing key-rolling scheme. Given N power traces, the adversary is unable to extract the secret key by time-domain based SCA method. However, the adversary may successfully extract the secret key

within N power traces by our proposed method. Moreover, it is advisable to adopt multiple countermeasures to resist our proposed method. Through the practical experiment results, it can be inferred that using single side-channel countermeasure (e.g. masking or random delay) cannot effectively resist WST/STFT-based CPA attacks. The adversary can successfully break masking or random delay countermeasure within 3000 traces by the proposed method. The designer needs to introduce additional countermeasures or more complex countermeasures to enhance the physical security level of crypto produces. For example, designers can combine shuffling and masking countermeasures to resist our proposed method. Although shuffling cannot resist SCA when the attack data is huge, it can randomize casual independent operations and efficiently increase the number of attack data to perform a successful SCA. The protection is considered effective when the cost of successful SCA is unaffordable for the adversary.

A fine-grain analysis on parameter settings for frequency-based CPA attacks

Based on the experiment results ("Experiment results" section), we conduct a systematic empirical study to investigate the effectiveness of STFT-based CPA and WST-based CPA in non-profiled attacks scenario. The performance of WST/STFT-based CPA is evaluated under different parameter values in a fine-grain manner. According to the analysis result, we provide empirical suggestions for parameter selections in non-profiled attacks scenario. In this section, we mainly focus on PCC-WST/STFT-based CPA attacks.

A fine-grain analysis on parameter settings for WST-based CPA attacks

To assess the performance of WST-CPA attacks on DPA Contest V4, AES_HD, ASCAD and AES_RD datasets, we select the best-performance parameter-settings, and then systematically compare their performance with various $\{J, Q\}$ on four public datasets.

Figure 13 and Fig. 14 plot success rate results of WST-baed CPA with various $\{J, Q\}$ parameters on DPA Contest V4 dataset. From Figs. 13 and 14, we can learn that the parameter J plays a more important role in the scenario of improving WST-based CPA attacks. With a suitable parameter J, the adversary can efficiently improve the performance of WST-based CPA attacks even though the parameter Q is not properly selected. In the context of analyzing DPA Contest V4 dataset or similar implementations, the adversary can achieve a more efficient non-profiled attack when the variable parameters $\{J, Q\}$ satisfy: $J = 1$ and $Q \in \{3, 4, 5, 6\}$.

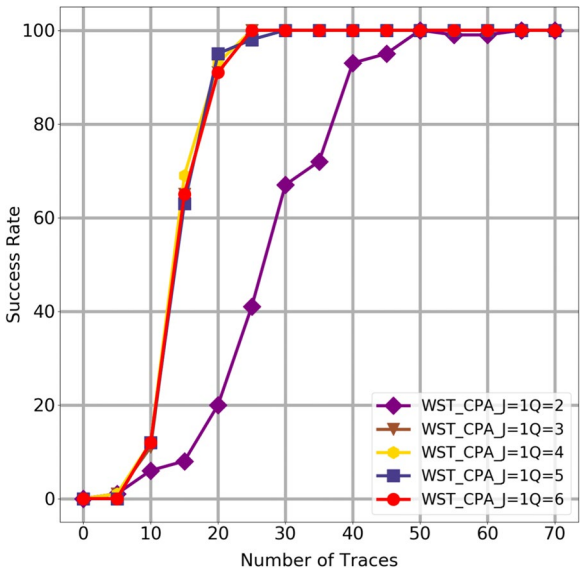

Fig. 13 The performance of WST-based CPA with different Q on DPA Contest V4 dataset

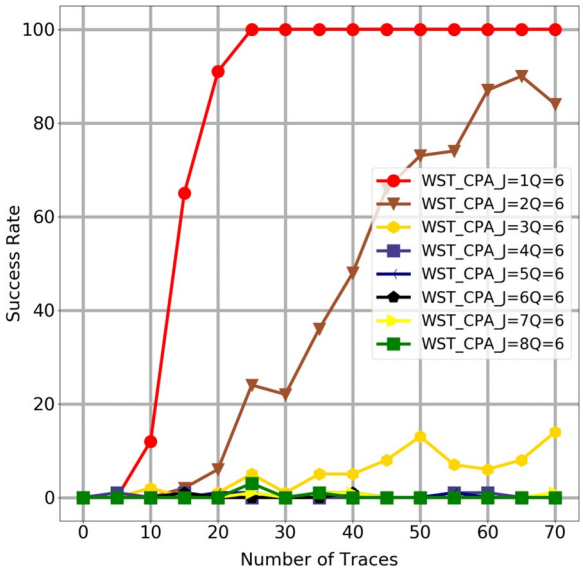

Fig. 14 The performance of WST-based CPA with different J on DPA Contest V4 dataset

In the context of analyzing AES_HD dataset, the performances of WST-based CPA with various $\{J, Q\}$

are very analogous to WST-based CPA against DPA Contest V4 dataset. The parameter J also plays a more important role in the scenario of improving WST-based CPA attacks, as depicted in Figs. 15 and 16. The AES key of AES_HD dataset is successfully recovered within 4000 traces when the parameter J is set to 2. We recommend setting parameters $\{J, Q\}$ to $\{2, 6\}$ when analyzing AES_HD dataset or similar implementations for significant improvement of the performance of original CPA attacks.

To analyze ASCAD dataset, the original traces are preprocessed with window compression method. The number of sampling points is reduced to 70. To make WST work, we modify the value range of parameter J ($J \in \{1, 2, 3\}$). Figures 17 and 18 plot the performance of WST-based CPA with various $\{J, Q\}$ against ASCAD dataset. As expected, the adversary can efficiently enhance non-profiled attacks when the parameter J is

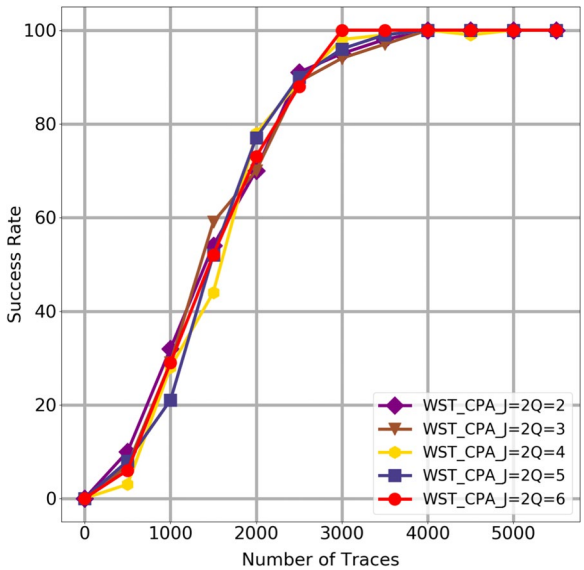

Fig. 15 The performance of WST-based CPA with different Q on AES_HD dataset

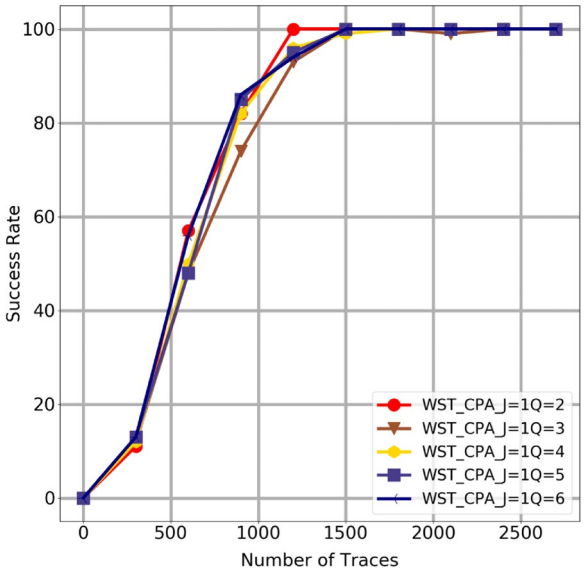

Fig. 17 The performance of WST-based CPA with different Q on ASCAD dataset

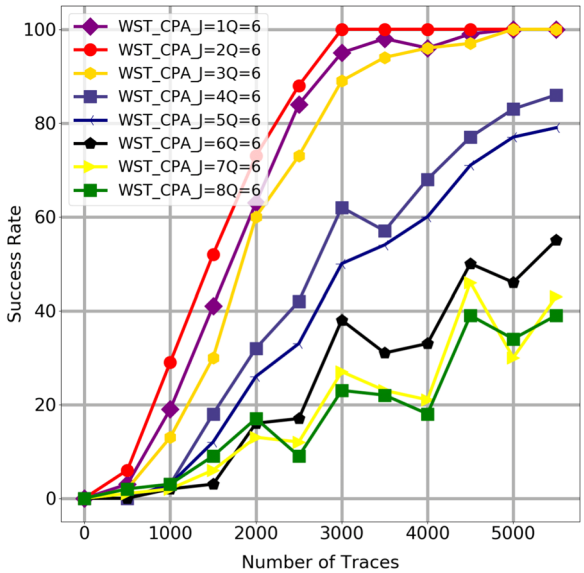

Fig. 16 The performance of WST-based CPA with different J on AES_HD dataset

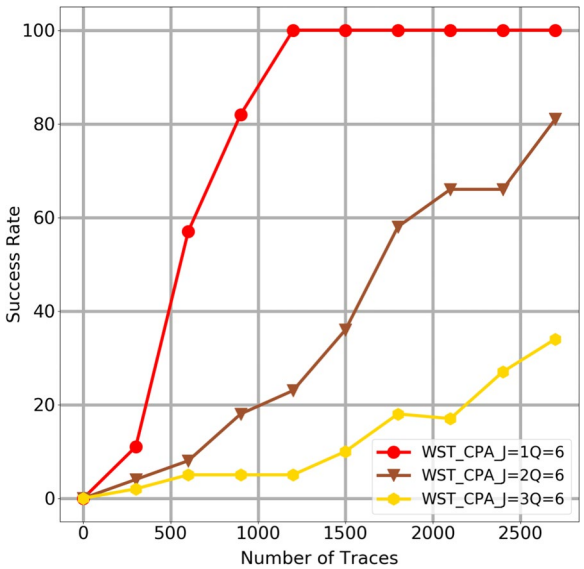

Fig. 18 The performance of WST-based CPA with different J on ASCAD dataset

properly selected. The adversary can achieve the best performance when the parameters $\{J, Q\}$ satisfy: $J = 1$ and $Q \in \{2, 3\}$.

In the context of analyzing AES_RD dataset, the adversary can efficiently break random delay countermeasures when the parameters $\{J, Q\}$ satisfy: $J \in [4, 8]$ and $Q \in [5, 8]$ as shown in Figs. 19 and 20. The proposed framework can achieve a more robust attack performance when the adversary adopts larger parameters $\{J, Q\}$.

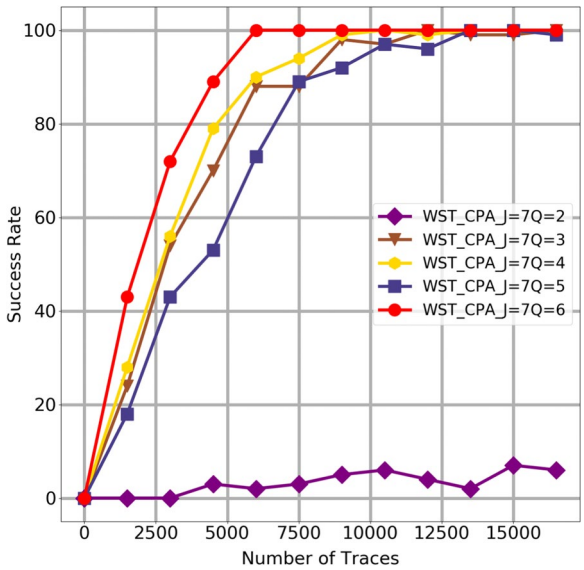

Fig. 19 The performance of WST-based CPA with different Q on AES_RD dataset

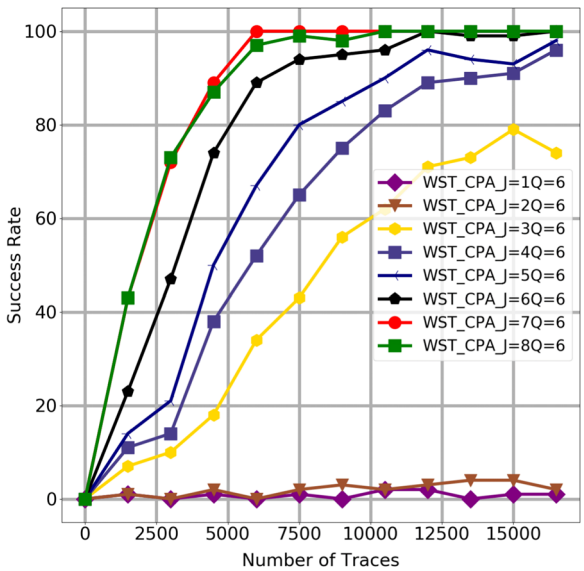

Fig. 20 The performance of WST-based CPA with different J on AES_RD dataset

A fine-grain analysis on parameter settings for STFT-based CPA attacks

To assess the performance of STFT-CPA attacks on DPA Contest V4, AES_HD and ASCAD, we select the best-performance frequency-components, and then systematically compare their performance with various $\{std, window\}$ on three public datasets, where *std* denotes the size of standard deviation and *window* represents the size of gaussian window used in STFT.

Figures 21 and 22 plot success rate results of STFT-based CPA with various parameters $\{std, window\}$ on

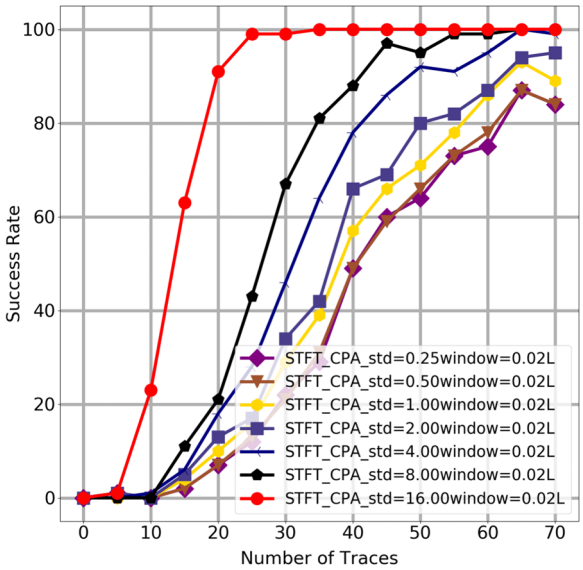

Fig. 21 The performance of STFT-based CPA with the same *window* on DPA Contest V4 dataset

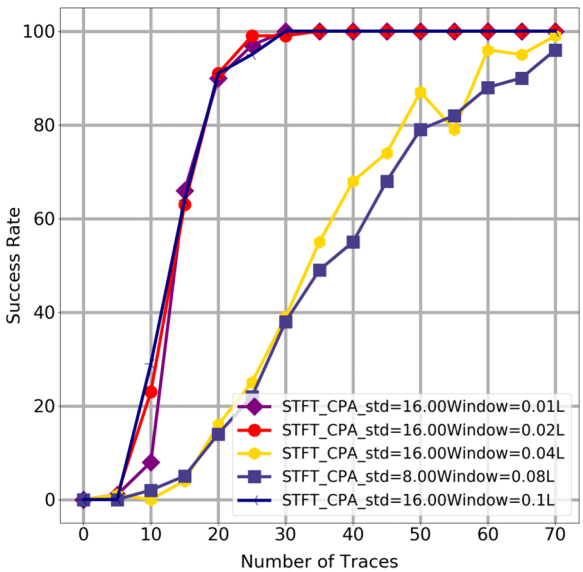

Fig. 22 The performance of various *window*-sized STFT-based CPA on DPA Contest V4 dataset

DPA Contest V4 dataset. From Figs. 21 and 22, we can learn that the size of *window* and *std* play important roles in the scenario of STFT-based CPA attacks. With a smaller-size *window* and a larger-size *std*, the performance of original CPA attacks can be significantly improved. The proposed attack framework achieves a satisfactory improvement when the parameters $\{std, window\}$ satisfy: $std \in \{8, 16\}$ and $window \in \{0.01L, 0.02L\}$.

Similar to the analysis of DPA Contest V4 dataset, the adversary can efficiently improve the performance of CPA attacks on AES_HD dataset with a smaller-size *window* and a larger-size *std*, as shown in Figs. 23 and 24. The adversary can achieve the best performance when the parameters $\{window, std\}$ satisfy: $std = 16$ and $window \in \{0.01L, 0.02L\}$. Similar to the analysis of DPA Contest V4 and AES_HD datasets, using a smaller-size *window* allows adversaries to extract the secret key

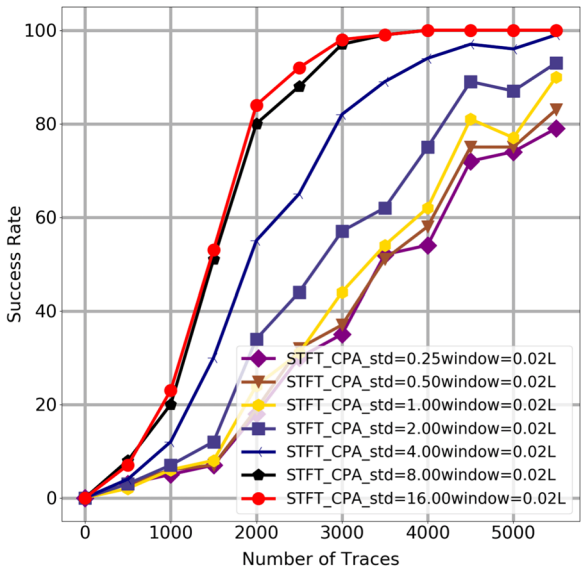

Fig. 23 The performance of STFT-based CPA with the same *window* on AES_HD dataset

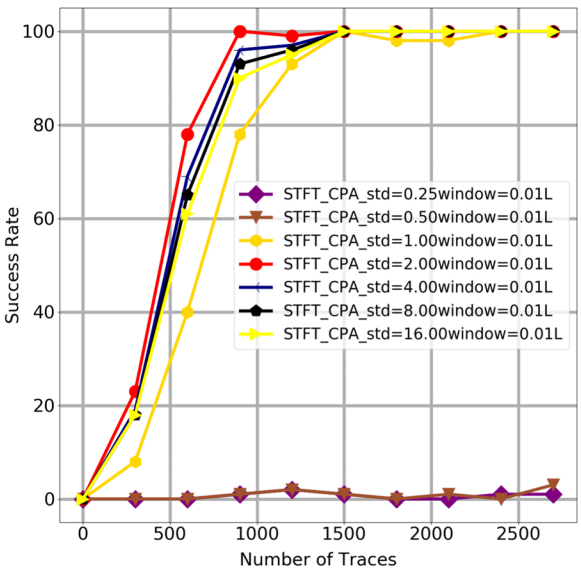

Fig. 25 The performance of STFT-based CPA with the same *window* on ASCAD dataset

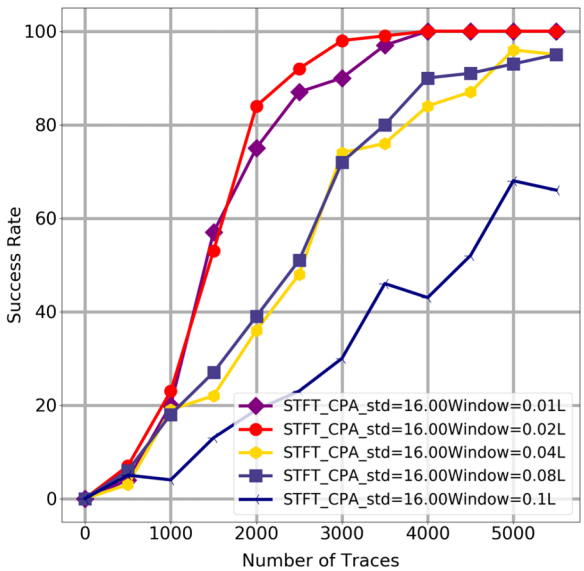

Fig. 24 The performance of various *window*-sized STFT-based CPA on AES_HD dataset

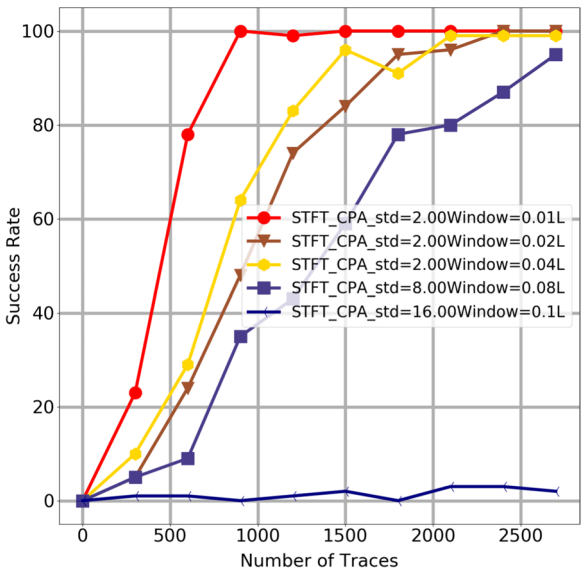

Fig. 26 The performance of various *window*-sized-STFT based CPA on ASCAD dataset

of ASCAD dataset with fewer traces (See Figs. 25 and 26). Our method significantly enhances CPA attacks on ASCAD dataset when the $\{std, window\}$ are designed as follows: $std \in \{2, 4, 8, 16\}$ and $window = 0.01L$. Through these three comparative experiment results, we can conclude that STFT-based CPA attacks tend to achieve a more robust non-profiled attacks when the length of *window* becomes smaller. To enhance the performance of non-profiled attacks, we recommend setting the parameter *window* as follows: $window \in \{0.01L, 0.02L\}$ when performing STFT-based CPA attacks.

Conclusions

We present a systematic research about the impact of $\{J, Q\}$ and $\{std, window\}$ on the performance of WST/STFT-based CPA attacks in non-profiled attacks scenario. Through the practical experiments, we obtain following important and interesting findings:

- In the case of WST-based CPA attacks, the parameter J plays a more important role in enhancing CPA attacks. Using a smaller-size parameter $J (J \in \{1, 2\})$, the performance of original CPA attacks can be significantly enhanced with the proposed method ("A practical framework for frequency-based CPA attack" section) when the target does not adopt random delay countermeasures. When analyzing random delay countermeasures, adversaries can adopt larger-size parameter-settings $\{J, Q\}(J \in \{7, 8\}, Q \in \{5, 6\})$ to efficiently break random-delay countermeasures.
- In the scenario of STFT-based CPA attacks, we find that the smaller-size parameter *window* allows adversaries to achieve a successful CPA attack with fewer traces. The proposed attack framework ("A practical framework for frequency-based CPA attack" section) achieves a satisfactory performance-improvement when the size of window is set to $\{0.01L, 0.02L\}$.

Discussions

Related works

Through former experiment results, it can be inferred that using our proposed attack framework and grid research method, WST/STFT-based CPA attacks can significantly enhance the performance of original CPA attacks with suitable parameters. Currently, there are two categories of related works in side-channel attacks domain: (1) Applying preprocessing technique to improve the performance of SCA; (2) Applying hyperparameters-search method for Deep Learning based Side-Channel Analysis (DL-SCA).

Applying preprocessing technique to improve the performance of SCA. Preprocessing the physical signals is the first important step in the case of improving the quality of collected data. The adversary can efficiently extract the secret key with quite limited traces if the quality of collected data is significantly improved. In theory, the preprocessing method is not limited to specific platform or cryptographic implementations. It can be applied to any kind of cryptographic implementation. Many researchers have considered applying preprocessing method to improve the performance of SCA. For instance, Bruneau et al. (2015) applied PCA in processing the original traces. They showed that PCA can efficiently enhance non-profiled attacks if the principal components are properly selected. Merino Del Pozo and Standaert (2015) adopted Singular Spectrum Analysis method to improve the quality of the collected signals. They showed that their proposed method can improve the SNR by 250% in the context of analyzing software-based unprotected/masked implementations of AES. However, these kinds of methods require expert-knowledge to some extents. The adversary needs to carefully select the components to enhance the quality of the collected traces. Besides, they did not deeply investigate whether their method can efficiently work in the case of analyzing random-delay countermeasures. To further optimize the performance of CPA attacks, Maghrebi and Prouff (2018) designed a practical Independent-Component Analysis (ICA) based framework to enhance the performance of CPA attacks. Compared with the previous work (Merino Del Pozo and Standaert 2015), their method allows reducing the number of data from 6000 to 2000 in the case of analyzing software-based unprotected implementation of AES. Compared with previous works (Merino Del Pozo and Standaert 2015; Bruneau et al. 2015), their method does not require dedicated parameters selection. However, their method requires adversaries to collect two traces for each hypothesis intermediate value at least. Unlike their method, our method does not have this kind of limitation. In addition to improving the SNR of the collected data, some researchers considered applying data-augmentation techniques (e.g. SMOTE (Picek et al. 2019), adding gaussian noise (Kim et al. 2019)) to further optimize DL-SCA. They showed that the performance of SCA can be significantly enhanced by adding synthetic data to the original collected data in profile stages. However, these methods (Picek et al. 2019; Kim et al. 2019) are only limited to profiled attacks scenario. Compared with their works, our method theoretically can be also applied to profiled attacks scenario. Currently, there exist some similar works that consider applying frequency-based CPA method to enhance SCA.

Typical examples include applying DFT (Zhang et al. 2020; Gebotys et al. 2005), STFT (Belgarric et al. 2014) and Wavelet transform (Debande et al. 2012; Udvarhelyi et al. 2021; Destouet et al. 2021) to improve the performance of CPA attacks. The authors (Zhang et al. 2020; Gebotys et al. 2005) showed that FFT has appealing technical potential in the case of improving the performance of SCA. However, in this paper, we discover that FFT-based CPA does not have amazing performance as original works say (Zhang et al. 2020; Gebotys et al. 2005). It makes the performance of original CPA even worse when attacking DPA Contest V4, AES_HD and ASCAD datasets. In general, FFT-based preprocessing method is not very mature when analyzing different cryptographic implementations. Different from previous works (Debande et al. 2012; Udvarhelyi et al. 2021; Destouet et al. 2021; Belgarric et al. 2014), our WST-based CPA method does not require any expert-knowledge dependency. An adversary, who has no prior knowledge about the WST/STFT and cryptographic implementations, can obtain suitable parameters for WST/STFT with the proposed framework. Compared with original CPA attacks, our method allows reducing the number of traces by 50–95% in the case of attacking different kinds of cryptographic implementations ("Experiment results" section). Our approach provides a convenient and effective solution to enhance non-profiled CPA attacks when the collected data is insufficient, which certainly deserves more in-depth researches.

Applying hyperparameters-search method for DL-SCA. In DL-SCA, selecting suitable hyperparameters, such as loss function, neural parameters and network architecture, is vital for constructing a robust profiled model. With suitable hyperparameters and sufficient training data, the adversary can successfully extract the secret key with quite limited data. In recent years, some researchers have introduced hyperparameters-search methods to SCA domain to enhance the performance of DL-SCA. For instance, Perin and Picek (2021) adopted grid search method to select the optimizers for DL-SCA. Wu et al. (2020) applied random search with Bayesian optimizations to design neural network architecture for DL-SCA. On the basis of the research (Wu et al. 2020), Rijsdijk et al. (2021) adopted reinforcement learning to achieve hyperparameters tuning for DL-SCA. These works adopted success rate, guess entropy, loss value and the size of neural network model as main evaluation metrics to select suitable parameters iteratively. Unlike these researches, we adopt grid search as the main method and apply it in non-profiled attacks scenario. We adopt PCC, SNR and DOP as main evaluation metrics to select hyperparameters for WST/STFT-CPA attacks.

Discussions and future directions

Advantages of the attack method we presented. With our proposed attack framework, WST/STFT-based CPA can achieve significant performance improvements when the attack data is insufficient. The practical attack results ("Experiment results" section) prove that our work provides a convenient and effective approach to enhance non-profiled attacks when the collected data is insufficient. The presented method is applicable to other symmetric ciphers, such as Midori (Banik et al. 2015), GIFT (Banik et al. 2017) and Pyjmask (Goudarzi et al. 2020). In practice, the adversary can firstly apply our method to select suitable parameters for WST/STFT-based CPA when analyzing the first-byte secret key. Then he can directly apply STFT/WST-based CPA with suitable parameters to extract other secret key bytes to accelerate non-profiled attacks. In addition, he can also directly use our recommended parameters setting ("A fine-grain analysis on parameter settings for frequency-based CPA attacks" section) to enhance WST/STFT-CPA when the cryptographic implementation or platform is similar to the analyzed dataset ("Public datasets" section).

Future prospects. We plan to investigate the performance of our proposed method in the case of analyzing protected implementations of lightweight block ciphers in our future research. We want to explore whether the proposed method can efficiently work in the case of analyzing other important block ciphers. In addition, it would be also valuable to design new random delay countermeasures to efficiently resist WST/STFT-based CPA attacks. The experiment results ("Experiment results" section) show that our method can efficiently break random delay countermeasures without any alignment. Designers need to ensure the sampling points in the frequency-domain are also misaligned when implementing random delay countermeasures. Designing a reliable random delay countermeasure to resist time-domain based SCA and our proposed method is also an interesting and meaningful task for future works.

Limitations of the attack method. Though former practical experiment results show that our approach has great technical potential in enhancing non-profiled CPA attacks, the method has some limitations that need improvements in the future:

- *Bring extra time-overhead.* The attack framework requires extra preprocessing steps. The adversary needs to iteratively search the suitable parameters, which certainly brings additional time-overhead. Compared with our proposed method, original CPA does not require hyperparameters selection and can directly extract the secret key. Hence, our method

currently is not practicable in realistic when the collected traces are sufficient. Optimization of parameter search process will be the focus of future works.

- *Require searching parameters manually*. In this work, we adopt grid-search as main parameters-searching methods to search parameters for WST/STFT-based CPA. We need to manually define the ranges for finite sets, such as J and Q used in WST-based CPA attacks, prior to the attack. In DL-SCA, the adversary has considered applying reinforcement learning to achieve intelligent parameters tuning (Rijsdijk et al. 2021), which provides a living case for our research. However, in this paper, we do not consider how to achieve intelligent parameters tuning for WST/STFT-based CPA attacks.
- *The theoretical interpretability issue of the frequency domain analysis method itself is not effectively solved*. Through the practical experiment results ("Experiment results" and "A fine-grain analysis on parameter settings for frequency-based CPA attacks" sections), it can be inferred that WST/STFT-CPA achieves different improvement levels according to different cryptographic implementations. Theoretically, the occurrence of significant technical effects has correlation with some physical characteristics of the analyzed information leakages. However, we do not conduct in-depth researches on the mechanism cognition and principle demonstration of the attack method from the theoretical level. We leave those challenging tasks for future works.

Future improvements. To achieve intelligent parameters tuning for WST/STFT-based CPA attacks, evaluators can consider applying a more advanced parameters-searching method (e.g. reinforcement learning) to WST/STFT-based CPA attacks. Evaluators can combine random searching and Bayesian Optimization (He et al. 2021) to further optimize the parameter search process. In addition, it would be valuable to give corresponding feasible and powerful explanations from some perspectives of signal characteristics. Moreover, designing more reliable evaluation metrics is helpful for enhancing the practicality of our proposed method.

Future directions. In addition to the above limitations, there are several challenging and interesting works left uncompleted in our research:

- *Applying the attack framework to analyze machine-learning algorithms*. In this paper, we focus on AES block-cipher. The physical security of other important algorithms such as machine-learning (ML) algorithms are not deeply investigated. Recently, physical evaluations of machine learning algorithms have become a hot topic in SCA community. Many interesting studies have been conducted to investigate the physical security of ML models, such as extracting the IEEE-754 floating points of CNN/MLP (Batina et al. 2019) models, extracting the integer weight parameters of Binarized Neural Networks (BNN) (Yli-Mayry et al. 2021) and Bonsai (Jap et al. 2020). It would be interesting to study whether the proposed method is suitable for further enhancing CPA in the case of analyzing IEEE-754 floating points or quantized ML models.
- *Extending the proposed method to high security-level commercial crypto products*. In our work, we conduct practical attacks on four datasets to assess the effectiveness of our approach. We do not consider whether our approach can be applied to analyze high security-level commercial crypto products. In recent years, some hardware vendors have considered designing countermeasures for commercial crypto products to resist various physical attacks. For instance, Xilinx Zynq Ultracale+ (ZU+) Encryption Engine adopts proprietary countermeasures to resist SCA (Hettwer et al. 2021). In this case, adversaries need to extract the secret key within limited data. The previous work (Hettwer et al. 2021) shows that ZU+ platform can successfully resist original non-profiled attacks. However, if adversaries adopt our attack framework or more advanced parameter tuning method (Rijsdijk et al. 2021), the secret key might be successfully recovered with limited traces. Hence, it would be meaningful to investigate if our method is applicable for analyzing this kind of high security-level commercial crypto products.
- *Building a generic framework to scientifically compare various preprocess methods*. This paper mainly compares the performance of our method with some popular preprocess methods, such as PCA (Bruneau et al. 2015), NMF (Yang et al. 2017), Ensemble method with WST (Destouet et al. 2021), Lowpass-filter and Moving Average (Provided by Riscure (2021)). The other complex preprocess methods, such as wavelet transform (WT), Kalman filter (KF) and Singular Spectrum Analysis (SSA) are not considered, as they are mainly heuristic methods and usually require dedicated parameters selection. Empirically select parameters may make the performance of those preprocess methods (WT, KF and SSA) unstable. Sometimes it might make the performance of original attacks even worse. Hence, it is necessary to build a generic framework to comprehensively and scientifically compare the performance of various preprocess methods. The practical attack results show that our framework can provide suitable

parameters selection for WST/STFT-CPA attacks. It would be interesting to investigate whether our framework ("A practical framework for frequency-based CPA attack" section) is able to enhance other complex preprocess methods.

The above interesting tasks and unexploitable areas are left for future works.

Conclusions

We propose a practical framework to provide suitable parameters for WST-based SCA and STFT-based SCA. With the suitable parameters, the performance of WST/STFT-based CPA can be significantly improved. The performance of the designed attack framework is assessed by practical experiments on four public datasets, including DPA Contest V4, AES_HD, AES_RD and ASCAD datasets. Compared with original non-profiled attacks, the proposed method can reduce the number of data by 50–95%. In general, the proposed attack framework provides a straightforward and effective solution to enhance CPA in the case of insufficient data, which certainly deserves more-in depth researches.

Abbreviations

AES	Advanced encryption standard
AES-CTR	Advanced encryption standard counter mode
BNN Binarized	Neural Networks
CNN	Convolutional Neural Networks
CPA	Correlation power analysis
CTR_DRBG	Counter deterministic random byte generator
DAC	Design automation conference
DFT	Discrete fourier transform
DL	Deep learning
DL-SCA	Deep learning based side-channel analysis
DOP	Absolute-differences-of-pearson correlation coefficient
DPA	Differential power analysis
EM	Electromagnetic radiation
FFT	Fast fourier transform
FIPS	Federal information processing standards
GE	Guess entropy
HD	Hamming distance
HW	Hamming weight
ICA	Independent-component analysis
IEC	International electro technical commission
ISO	International organization for standardization
KF	Kalman filter
MIA	Mutual information analysis
ML	Machine learning
MLP	Multilayer perceptron
NIST	National institute of standards and technology
NMF	Non-negative matrix factorization
PCA	Principal component analysis
PCC	Pearson correlation coefficient
RSA	Rivest Shamir Adleman
SA	Stochastic attack
SCA	Side-channel analysis
SMOTE	Synthetic minority oversampling technique
SNR	Signal-to-noise ratio
SR	Success rate
SSA	Singular spectrum analysis
STFT	Short-time fourier transform
TA	Template attack
USIM	Universal subscriber identity module
WST	Wavelet scatter transform
WT	Wavelet transform
ZU+	Zynq ultracale+

Acknowledgements
Not applicable.

Author contributions
CJ completed the main work of the paper and drafted the manuscript. YZ participated in problem discussions and improvements of the manuscript. All authors read and approved the final manuscript.

Author's information
Chengbin Jin (Email: jinchengbin@iie.ac.cn) is with Institute of Information Engineering, Chinese Academy of Sciences, and School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China.
Yongbin Zhou (Email: zhouyongbin@njust.edu.cn) is the corresponding author of this paper. He is currently a Full-time Professor with the School of Cyber Science and Engineering, Nanjing University of Science and Technology. He is also an Adjunct Professor with Institute of Information Engineering, Chinese Academy of Sciences. His main research interests include theories and technologies of network and information security.

Funding
This work is supported in part by National Key R&D Program of China (No. 2022YFB3103800), National Natural Science Foundation of China (No. U1936209, No.62002353, No.62202231 and No.62202230), China Postdoctoral Science Foundation (No.2021M701726), Jiangsu Funding Program for Excellent Postdoctoral Talent (No.2022ZB270) and Yunnan Provincial Major Science and Technology Special Plan Projects (No.202103AA080015).

Availability of data and materials
Not applicable.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 18 October 2022 Accepted: 23 February 2023
Published online: 01 August 2023

References

AES_HD (2018) The AES_HD database - Unprotected hardware-based implementation of AES. https://github.com/AESHD/AES_HD_Dataset
AES_RD (2017) The AES_RD database - Trace sets with random delays. https://github.com/ikizhvatov/randomdelays-traces
Akkar M-L, Giraud C (2001) An implementation of des and aes, secure against some attacks. In: Koç ÇK, Naccache D, Paar C (eds) Cryptographic hardware and embedded systems—CHES 2001. Springer, Berlin, Heidelberg, pp 309–318
Allen J (1977) Short term spectral analysis, synthesis, and modification by discrete fourier transform. IEEE Trans Acoust Speech Signal Process 25(3):235–238. https://doi.org/10.1109/TASSP.1977.1162950
Andén J, Mallat S (2013) Deep scattering spectrum. IEEE Trans Signal Process. https://doi.org/10.1109/TSP.2014.2326991
Andreux M, Angles T, Exarchakis G, Leonarduzzi R, Rochette G, Thiry L, Zarka J, Mallat S, Andén J, Belilovsky E, Bruna J, Lostanlen V, Chaudhary M, Hirn MJ, Oyallon E, Zhang S, Cella C, Eickenberg M (2020) Kymatio: Scattering transforms in python. J Mach Learn Res 21:60–1606
ASCAD (2018) The ASCAD database - First-order boolean masked AES implementation on an ATMEGA8515. https://github.com/ANSSI-FR/ASCAD

Banik S, Bogdanov A, Isobe T, Shibutani K, Hiwatari H, Akishita T, Regazzoni F (2015) Midori: A block cipher for low energy. In: Iwata T, Cheon JH (eds) Advances in cryptology—ASIACRYPT 2015. Springer, Berlin, Heidelberg, pp 411–436

Banik S, Pandey SK, Peyrin T, Sasaki Y, Sim SM, Todo Y (2017) Gift: A small present. In: Fischer W, Homma N (eds) Cryptographic hardware and embedded systems—CHES 2017. Springer, Cham, pp 321–345

Barker E, Kelsey J (2015) Recommendations for random number generation using deterministic random bit generators. NIST SP 800-90A Rev. 1. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf

Batina L, Bhasin S, Jap D, Picek S (2019) CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel. In: 28th USENIX Security Symposium (USENIX Security 19). USENIX Association, Santa Clara, pp 515–532

Belgarric P, Bhasin S, Bruneau N, Danger J-L, Debande N, Guilley S, Heuser A, Najm Z, Rioul O (2014) Time-frequency analysis for second-order attacks. In: Francillon A, Rohatgi P (eds) Smart card research and advanced applications. Springer, Cham, pp 108–122

Benadjila R, Prouff E, Strullu R, Cagli E, Dumas C (2020) Deep learning for side-channel analysis and introduction to ASCAD database. J Cryptogr Eng 10(2):163–188. https://doi.org/10.1007/s13389-019-00220-8

Brier E, Clavier C, Olivier F (2004) Correlation power analysis with a leakage model. In: Joye M, Quisquater J-J (eds) Cryptographic hardware and embedded systems—CHES 2004. Springer, Berlin, Heidelberg, pp 16–29

Bruneau N, Guilley S, Heuser A, Marion D, Rioul O (2015) Less is more. In: Güneysu T, Handschuh H (eds) Cryptographic hardware and embedded systems—CHES 2015. Springer, Berlin, Heidelberg, pp 22–41

Bubberman W, Karayalcin S, Meester M, Braakman O, Picek S (2020) Side-channel Analysis Toolbox. https://github.com/AISyLab/side-channel-analysis-toolbox/blob/master/sca/analysis/snr.py

Cagli E, Dumas C, Prouff E (2017) Convolutional neural networks with data augmentation against jitter-based countermeasures. In: Fischer W, Homma N (eds) Cryptographic hardware and embedded systems—CHES 2017. Springer, Cham, pp 45–68

Chari S, Rao JR, Rohatgi P (2003) Template attacks. In: Kaliski BS, Koç çK, Paar C, (eds) Cryptographic Hardware and Embedded Systems—CHES 2002. Springer, Berlin, pp 13–28

Coron J-S, Kizhvatov I (2010) Analysis and improvement of the random delay countermeasure of ches 2009. In: Mangard S, Standaert F-X (eds) Cryptographic hardware and embedded systems, CHES 2010. Springer, Berlin, Heidelberg, pp 95–109

Debande N, Souissi Y, Aabid M, Guilley S, Danger J-L (2012) Wavelet transform based pre-processing for side channel analysis, pp 32–38. https://doi.org/10.1109/MICROW.2012.15

Destouet G, Dumas C, Frassati A, Perrier V (2021) Wavelet scattering transform and ensemble methods for side-channel analysis. In: Bertoni GM, Regazzoni F (eds) Constructive side-channel analysis and secure design. Springer, Cham, pp 71–89

DPA_Contest_v4 (2014) TELECOM ParisTech SEN research group. DPA Contest (4th edition). http://www.DPAcontest.org/v4/

Fabian Pedregosa AG, Gael Varoquaux, Michel V (2020) HalvingGridSearchCV. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.HalvingGridSearchCV.html

FIPS_140-3 (2020) FIPS Publication 140-3. The National Institute of Standards and Technology. https://csrc.nist.gov/publications/detail/fips/140/3/final

Gandolfi K, Mourtel C, Olivier F (2001) Electromagnetic analysis: concrete results. In: Koç ÇK, Naccache D, Paar C (eds) Cryptographic hardware and embedded systems—CHES 2001. Springer, Berlin, Heidelberg, pp 251–261

Gebotys CH, Ho S, Tiu CC (2005) Em analysis of rijndael and ecc on a wireless java-based pda. In: Rao JR, Sunar B (eds) Cryptographic hardware and embedded systems—CHES 2005. Springer, Berlin, Heidelberg, pp 250–264

Gierlichs B, Batina L, Tuyls P, Preneel B (2008) Mutual information analysis. In: Oswald E, Rohatgi P (eds) Cryptographic hardware and embedded systems—CHES 2008. Springer, Berlin, Heidelberg, pp 426–442

Goubin L, Patarin J (1999) Des and differential power analysis the duplication method. In: Koç ÇK, Paar C (eds) Cryptographic hardware and embedded systems. Springer, Berlin, pp 158–172

Goudarzi D, Jean J, Kölbl S, Peyrin T, Rivain M, Sasaki Y, Sim SM (2020) Pyjamask: Block cipher and authenticated encryption with highly efficient masked implementation. IACR Trans Symmetric Cryptol 2020(S1):31–59. https://doi.org/10.13154/tosc.v2020.iS1.31-59

He X, Zhao K, Chu X (2021) Automl: A survey of the state-of-the-art. Knowl Based Syst 212:106622. https://doi.org/10.1016/j.knosys.2020.106622

Hettwer B, Leger S, Fennes D, Gehrer S, Güneysu T (2021) Side-channel analysis of the xilinx zynq ultrascale+ encryption engine. IACR Trans Cryptogr Hardw Embed Syst 1:279–304. https://doi.org/10.46586/tches.v2021.i1.279-304

ISO/IEC-17825 (2016) Testing methods for the mitigation of non-invasive attack classes against cryptographic modules. ISO/IEC 17825-2016. International Organization for Standardization. https://www.iso.org/standard/60612.html

Jap D, Yli-Mäyry V, Ito A, Ueno R, Bhasin S, Homma N (2020) Practical side-channel based model extraction attack on tree-based machine learning algorithm. In: Zhou J, Conti M, Ahmed CM, Au MH, Batina L, Li Z, Lin J, Losiouk E, Luo B, Majumdar S, Meng W, Ochoa M, Picek S, Portokalidis G, Wang C, Zhang K (eds) Applied cryptography and network security workshops. Springer, Cham, pp 93–105

Jin C, Zhou Y, Qiu X, Feng Q, Zhang Q (2022) Breaking real-world COTS USIM cards with unknown side-channel countermeasures. Comput Secur 113:102531. https://doi.org/10.1016/j.cose.2021.102531

Kim TH, Kim C, Park I (2012) Side channel analysis attacks using AM demodulation on commercial smart cards with SEED. J Syst Softw 85(12):2899–2908. https://doi.org/10.1016/j.jss.2012.06.063

Kim J, Picek S, Heuser A, Bhasin S, Hanjalic A (2019) Make some noise unleashing the power of convolutional neural networks for profiled side-channel analysis. IACR Trans Cryptogr Hardw Embed Syst 3:148–179. https://doi.org/10.13154/tches.v2019.i3.148-179

Kocher PC (1996) Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: Koblitz N (ed) Advances in cryptology—CRYPTO '96. Springer, Berlin, Heidelberg, pp 104–113

Lerman L, Poussier R, Bontempi G, Markowitch O, Standaert F-X (2015) Template attacks vs. machine learning revisited and the curse of dimensionality in side-channel analysis. In: Mangard S, Poschmann AY (eds) Constructive side-channel analysis and secure design. Springer, Cham, pp 20–33

Liu J, Yu Y, Standaert F-X, Guo Z, Gu D, Sun W, Ge Y, Xie X (2015) Small tweaks do not help: differential power analysis of milenage implementations in 3g/4g usim cards. In: Pernul G, Ryan P, Weippl E (eds) Computer security–ESORICS 2015. Springer, Cham, pp 468–480

Maghrebi H, Prouff E (2018) On the use of independent component analysis to denoise side-channel measurements. In: Fan J, Gierlichs B (eds) Constructive side-channel analysis and secure design. Springer, Cham, pp 61–81

Maghrebi H, Portigliatti T, Prouff E (2016) Breaking cryptographic implementations using deep learning techniques. In: Carlet C, Hasan MA, Saraswat V (eds) Security, privacy, and applied cryptography engineering. Springer, Cham, pp 3–26

Mangard S, Oswald E, Popp T (2007). Power analysis attacks: revealing the secrets of smart cards. https://doi.org/10.1007/978-0-387-38162-6

Merino Del Pozo S, Standaert F-X (2015) Blind source separation from single measurements using singular spectrum analysis. In: Güneysu T, Handschuh H (eds) Cryptographic hardware and embedded systems— CHES 2015. Springer, Berlin, Heidelberg, pp 42–59

Nassar M, Souissi Y, Guilley S, Danger J-L (2012) Rsm: a small and fast countermeasure for aes, secure against 1st and 2nd-order zero-offset scas. https://doi.org/10.1109/DATE.2012.6176671

Numpy (2022) numpy 1.22.4-The fundamental package for array computing with Python. https://pypi.org/project/numpy/

Perin G, Picek S (2021) On the influence of optimizers in deep learning-based side-channel analysis. In: Dunkelman O, Jacobson MJ Jr, O'Flynn C (eds) Selected areas in cryptography. Springer, Cham, pp 615–636

Picek S, Heuser A, Jovic A, Bhasin S, Regazzoni F (2019) The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. IACR Trans Cryptogr Hardw Embed Syst 1:209–237. https://doi.org/10.13154/tches.v2019.i1.209-237

Pontes FJ, Amorim GF, Balestrassi PP, Paiva AP, Ferreira JR (2016) Design of experiments and focused grid search for neural network parameter optimization. Neurocomputing 186:22–34. https://doi.org/10.1016/j.neucom.2015.12.061

Rijsdijk J, Wu L, Perin G, Picek S (2021) Reinforcement learning for hyperparameter tuning in deep learning-based side-channel analysis. IACR Trans

Cryptogr Hardw Embed Syst 2021(3):677–707. https://doi.org/10.46586/tches.v2021.i3.677-707
Riscure (2021) Inspector side channel analysis. https://getquote.riscure.com/en/inspector-side-channel-analysis.html
Rivain M, Prouff E, Doget J (2009) Higher-order masking and shuffling for software implementations of block ciphers. In: Clavier C, Gaj K (eds) Cryptographic hardware and embedded systems—CHES 2009. Springer, Berlin, Heidelberg, pp 171–188
Schindler W, Lemke K, Paar C (2005) A stochastic model for differential side channel cryptanalysis. In: Rao JR, Sunar B (eds) Cryptographic hardware and embedded systems–CHES 2005. Springer, Berlin, pp 30–46
Scipy (2022) scipy 1.8.1-SciPy: Scientific Library for Python. https://pypi.org/project/scipy/
Standaert F-X, Malkin TG, Yung M (2009) A unified framework for the analysis of side-channel key recovery attacks. In: Joux A (ed) Advances in cryptology - EUROCRYPT 2009. Springer, Berlin, Heidelberg, pp 443–461
Timon B (2019) Non-profiled deep learning-based side-channel attacks with sensitivity analysis. IACR Trans Cryptogr Hardw Embed Syst 2019(2):107–131. https://doi.org/10.13154/tches.v2019.i2.107-131
Udvarhelyi B, van Wassenhove A, Bronchain O, Standaert F-X (2021) On the security of off-the-shelf microcontrollers: hardware is not enough. In: Liardet P-Y, Mentens N (eds) Smart card research and advanced applications. Springer, Cham, pp 103–118
Veyrat-Charvillon N, Medwed M, Kerckhof S, Standaert F-X (2012) Shuffling against side-channel attacks: a comprehensive study with cautionary note. In: Wang X, Sako K (eds) Advances in cryptology - ASIACRYPT 2012. Springer, Berlin, Heidelberg, pp 740–757
Wu L, Picek S (2020) Remove some noise: on pre-processing of side-channel measurements with autoencoders. IACR Trans Cryptogr Hardw Embed Syst 4:389–415. https://doi.org/10.13154/tches.v2020.i4.389-415
Wu L, Perin G, Picek S (2020) I choose you: automated hyperparameter tuning for deep learning-based side-channel analysis. IACR Cryptol ePrint Arch
Yang W, Zhou Y, Cao Y, Zhang H, Zhang Q, Wang H (2017) Multi-channel fusion attacks. IEEE Trans Inf Forensics Secur 12(8):1757–1771. https://doi.org/10.1109/TIFS.2017.2672521
Yang G, Li H, Ming J, Zhou Y (2020) Cdae: towards empowering denoising in side-channel analysis. In: Zhou J, Luo X, Shen Q, Xu Z (eds) Information and communications security. Springer, Cham, pp 269–286
Yli-Mayry V, Ito A, Homma N, Bhasin S, Jap D (2021) Extraction of binarized neural network architecture and secret parameters using side-channel information, pp. 1–5. https://doi.org/10.1109/ISCAS51556.2021.9401626
Zhang F, Shao B, Xu G, Yang B, Yang Z, Qin Z, Ren K (2020) From homogeneous to heterogeneous: Leveraging deep learning based power analysis across devices. pp 1–6. https://doi.org/10.1109/DAC18072.2020.9218693

Publisher's Note