Abstract—Side-channel analysis, originally used in cryptanalysis is growing in use cases, both offensive and defensive. Wavelet analysis is a commonly employed time-frequency analysis technique used across disciplines, with a variety of purposes, and has shown increasing prevalence within side-channel literature.

This paper explores wavelet selection and analysis parameters for use in side-channel analysis, particularly power side-channel-based instruction disassembly and classification. Experiments are conducted on an ATmega328P microcontroller and a subset of the AVR instruction set. Classification performance is evaluated with a time-series convolutional neural network (CNN) at clock-cycle fidelity. This work demonstrates that wavelet selection and employment parameters have meaningful impact on analysis outcomes. Practitioners should make informed decisions and consider optimizing these factors similarly to machine learning architecture and hyperparameters. We conclude that the gaus1 wavelet with scales 1-21 and grayscale colormap provided the best balance of classification performance, time, and memory efficiency in our application.

Keywords—side-channel analysis, wavelet transform, scalogram, disassembly, classification

I. INTRODUCTION

Side-channel analysis gained significant attention and adoption for use in cryptanalysis after Kocher’s seminal work [1]-[2] in the late 1990s. Since then, additional side-channels such as: electromagnetic [3], thermal [4], optical [5], methods such as template [6], and use cases such as detecting malware [7] and hardware trojans [8]-[9], reverse engineering [10], and disassembly [11]-[15] have been increasingly explored.

Wavelet analysis has been leveraged for myriad purposes, such as signal filtering [16]-[18], data compression [18]-[20], and feature analysis or extraction. As a feature analysis and extraction technique its vast usage examples include electrocardiogram [21], seismic data [22], 3D spatial reconstruction [23], and side-channel analysis [14], [18], just to name several. Prior work exists towards establishing wavelet selection methods, but these are largely domain and application specific [23]-[26].

As above establishes, there exist numerous uses for both side-channel analysis, and wavelet analysis, including multiple purposes for the use of wavelet analysis in support of side-channel analysis. However, the authors of this work are not aware of prior work that deliberately explored and described wavelet selection methods or employment considerations for side-channel analysis. Further, existing side-channel literature leveraging wavelet analysis methods are largely silent on specific implementations, which can lead to issues with reproducibility. Additionally, as some factors include trade-offs, this is important to explore in order to bring techniques out of the research laboratory and into engineering practice.

Contributions of this work include but are not necessarily limited to: first known exploration of wavelet selection and employment considerations for side-channel analysis, quantified by classification performance; establishing side-channel measurements as a function of the ideal signal and associated noise, measurement uncertainty, and variance (6); assessment of a wavelet selection technique from non-side-channel literature for use in side-channel disassembly, with comparison to actual results.

This paper is organized with background of time-frequency analysis and side-channel analysis in section II; an overview of the evaluation approach, including experimental setup, wavelets explored, and wavelet selection considerations in section III; experimental results in section IV; and conclusions in section V.

II. BACKGROUND

The following subsections provide a high-level overview and introductory context for the time-frequency
analysis and side-channel analysis techniques employed within this paper.

A. Time-Frequency Analysis

Several techniques exist within the area of time-frequency analysis which explores input signals in both the time and frequency domain concurrently to garner additional insights than inspecting each domain separately. Two common techniques are the Short-Time Fourier Transform (STFT) and wavelet transforms. Wavelet transform analysis offers advantages over the more traditional STFT as discussed in the following subsections.

1) Fourier Transform

Whereas the Fourier series approximates periodic functions through sums of harmonically-related sine and cosine functions [27], the Fourier transform provides a similar capability that can be applied to aperiodic functions [27]. For our purposes, this enables the analysis of time series signals through their approximated frequency components. However, it does not provide any temporal localization of those components. A standard representation of the Fourier transform [28] is shown in (1). Fig. 1 provides examples of the Fast Fourier Transform (FFT) (a.2, b.2) of side-channel input signals (a.1, b.1).

$$\mathcal{F}(f)(\omega) = \frac{1}{\sqrt{2\pi}} \int dt e^{-i\omega t} f(t) \tag{1}$$

2) Short-Time Fourier Transform

The STFT computes the Fourier transform in windowed-fashion, $g$, along the length of an input signal or function $f$. This provides some degree of temporal localization of spectral components, such that the STFT, or $(T_{\text{win}}^m(f)(\omega, t), \text{“can be interpreted loosely as the ‘content’ of $f$ near time $t$ and near frequency $\omega$ [28].”}$ The discrete version of the STFT is shown in (2), with $t$ assigned interval $n_t$, and $\omega$ assigned interval $m_\omega$. While this provides a combined temporal-frequency view, due to inherent uncertainty the resolution is limited by window-size and involves tradeoffs between domains. With narrow windows, temporal resolution is increased while frequency resolution decreases; whereas with broad windows the opposite is true.

Fig. 1 provides examples of the STFT depicted as spectrograms (a.3, b.3) of side-channel input signals (a.1, b.1). The STFT limitations, are observable in both (a.3) and (b.3) through blocky-representations.

$$T_{\text{win}}^m(f) = \int ds f(s) g(s - n_t) e^{-i m_\omega s} \tag{2}$$

3) Wavelet Transform

Wavelet transforms overcome the STFT resolution tradeoff by executing multiresolution rather than fixed analysis. Like the STFT, wavelet transforms can be applied with continuous or discrete variants. However, instead of calculating the Fourier transform of fixed windows, a wavelet function is variably scaled and

![Time-frequency analysis examples](image-url)
translated over the input signal producing coefficients which are combined as the multiresolution time-frequency output. This enables superior ability of the wavelet transform over the STFT “to ‘zoom in’ on very short-lived high frequency phenomena” such as transients or singularities [28]. Equations (3) and (4) describe the continuous wavelet transform function [28].

Fig. 1 provides examples of the continuous wavelet transform depicted as scalograms (a.4, b.4) of side-channel input signals (a.1, b.1). The y-axis was converted to show pseudo-frequency instead of scale for more ready comparison to the STFTs.

\( (T^\text{wav})(a,b) = |a|^{1/2} \int dt f(t) \psi \left( \frac{t-b}{a} \right), \)

with \( \psi \) satisfying:

\( \int dt \psi(t) = 0. \)  \( (4) \)

A baseline wavelet function, of which many exist, is often referred to as a “mother wavelet”, and is represented by \( \psi \). Specific instances of these functions with scale \( a \) and translation factor \( b \), are represented by \( \psi^{a,b} \), and referred to as “wavelets.” [28] Frequency and scale have an inverse relationship in wavelet transforms, with small scales corresponding to high frequencies and large scales corresponding to low frequencies. Additionally, the scale and frequency correspondence is imprecise, so approximations, or pseudo-frequencies are referenced instead by [29]:

\[ F_a = \frac{F_c}{a \times \Delta}. \]

The pseudo-frequency, in Hz, at a given scale \( a \), is \( F_a \), while \( F_c \) is the wavelet’s center frequency in Hz, and \( \Delta \) represents the sampling period in seconds.

B. Side-Channel Analysis

Side-channels, generically, are unintended sources of information. For computing systems this usually refers to sensitive information about the system and its internal operations that can be gleaned through monitoring and analysis of one or more of the system’s side-channels. Numerous side-channels exist, with power [2] and electromagnetic emissions [3] being especially common.

Typically, side-channel analysis involves extracting cryptographic keys or other private information, and some level of access to the system is assumed: direct for power or proximity (usually near field) for electromagnetic emissions. For power side-channels, simple or differential power analysis [2], template [6], or other analysis means would then be employed to extract the desired information from the captured side-channel trace data.

In this work, the use case of disassembly through power side-channels is explored. This follows closely to other analysis methods where large amounts of side-channel data are collected and analyzed ahead of time, but instead of looking for a specific correlation (e.g., the correct cryptographic key in the haystack of trace data), all trace information is intended for correlation to a corresponding system state and function, namely the set and sequence of executed processor instructions. This information can then support a variety of end purposes, such as reverse engineering or integrity monitoring of the device under test (DUT).

Noise and uncertainty are facts of life within side-channel measurement and analysis. These factors are introduced by a variety of sources such as noisy power input, electromagnetic noise, uncertainty in measurements of even calibrated equipment, as well as variability in the device(s) under measurement. A generic representation of these factors, introduced in this work, is shown as:

\[ T_{\text{meas}} = S + (N_1 + \ldots N_n) + (M_1 + \ldots M_m) + (V_1 + \ldots V_v) \]  \( (6) \)

where \( T_{\text{meas}} \) is the measured side-channel trace, \( S \) is the ideal signal, \( N \) represents individual noise sources, \( M \) represents individual measurement uncertainties, and \( V \) represents the variability in the device(s) under measurement, to include non-noise induced environmental factors. Accounting for factors in (6) can be key in achieving the objectives for a given side-channel use case effectively and efficiently. This can also support a more robust solution, as the variability between devices under measurement and within a given device over its operational life are important considerations.

Examples of \( T_{\text{meas}} \) are shown in Fig. 1 (a.1, b.1), with the impact of \( N \) being particularly apparent as serrated features along the main signal pattern in the clock-cycle trace of (b.1), which largely disappear when many traces of the same signal are averaged.
establish general impacts and considerations towards the central purpose of wavelet employment. The authors would point readers towards the large body of work in either area: classification and side-channel disassembly, for additional reading on the state of the art in each.

A. Experimental Setup and Baseline Analysis

An Elegoo Uno R3 served as the DUT, with an ATmega328P microcontroller down-clocked to 1 MHz. A digital sampling oscilloscope with a sampling rate of 500 MSa/s and bandwidth of 200 MHz was used to capture the power side-channel traces from voltage drop across a 1 kΩ shunt-resistor on the ground-side.

The DUT executed a program loop with each instruction from the included subset (listed in Table I) paired with an instance of every other subset instruction. The exception being the rjmp instruction was only used to restart the looped instruction sequence. The program loop consisted of 191 instructions over 287 total clock-cycles of execution.

Five separate side-channel trace files were collected at a memory depth of 14 Mpts. This provides for 485 complete program loops, and 9,215 instances of each instruction (except rjmp), for analysis.

Analysis was performed in Python 3 notebooks with the PyWavelets [31], NumPy [32], and SciPy [33] packages for time-frequency analysis and TensorFlow [34] with Keras [35] for classification. A time-series CNN, with wavelet coefficients of each clock-cycle window transformed into scalograms was used for classification, with a 70/30 train/test split of data. A semi-optimized set of scalogram, model architecture, and hyper-parameters was selected as the baseline for wavelet transform assessment that provided a balance between classification performance and time complexity, but also helped avoid overfitting. Specific optimizations are considered outside the scope of this paper.

Since the use-case explored within this work is side-channel disassembly, the evaluation metrics employed are instruction classification accuracy and computational time. Scalogram and CNN memory size, relative to color-depth, is also considered. Conventional cryptanalytic side-channel metrics, such as success rate, guessing entropy, and key rank are not relevant here.

B. Wavelets Examined

Within the PyWavelets package, wavelets are grouped into 14 families, each containing one or more mother wavelets. In total, there are more than 120 built-in mother wavelets, as well as the ability to introduce custom mother wavelets. Mother wavelets are further categorized as discrete or continuous, by conditions such as orthogonality, symmetry, real or complex, and by attributes such as lower and upper bounds, bandwidth frequency, and center frequency.

This work explores the continuous and real mother wavelets listed in Table II and depicted in Fig 2. Additionally, of the wavelets explored, none were orthogonal or bi-orthogonal, so that condition was excluded from the table. Furthermore, as only the complex wavelet families Complex Morlet, Shannon, and Frequency B-Spline include a bandwidth parameter, it is also excluded from Table II. The upper and lower bounds of the included mother wavelets are shown as the x-axis bounds in Fig. 2.

The mother wavelets in the Gaussian family, \( gaus_{N_{ord}} \), represent derivatives of the Gaussian function [36]:

\[
\text{Gaus}(x) = \frac{1}{\sigma(\sqrt{2\pi})} e^{-(x - \mu)^2 / (2\sigma^2)},
\]

where \( N_{ord} \) represents the derivative order from 1-8, \( \mu \) the mean, and \( \sigma^2 \) the variance.

Fig. 3 shows the scale to pseudo-frequency correspondence, described in (5), for the mother wavelets explored in this work. As several mother wavelets share \( F_c \), as listed in Table II, these have been combined in Fig. 3. As the oscilloscope sampling rate, and thus \( \Delta \) remained constant, the impact of \( F_c \) on pseudo-frequency \( F_a \) across

| Instruction | Clock-Length | Type  |
|-------------|--------------|-------|
| sbi         | 2            | bit / IO |
| nop         | 1            | control |
| add         | 1            | arithmetic |
| sub         | 1            | arithmetic |
| cbi         | 2            | bit / IO |
| push        | 2            | transfer |
| pop         | 2            | transfer |
| mul         | 2            | arithmetic |
| eor         | 1            | logic |
| movw        | 1            | transfer |
| rjmp        | 2            | program flow |

| Wavelet Family | Mother Wavelet | Symmetry   | Center Frequency |
|----------------|----------------|------------|-------------------|
| Gaussian       | gaus1          | anti-sym.  | 0.2 Hz            |
|                | gaus2          | symmetric  | 0.3 Hz            |
|                | gaus3          | anti-sym.  | 0.4 Hz            |
|                | gaus4          | symmetric  | 0.5 Hz            |
|                | gaus5          | anti-sym.  | 0.5 Hz            |
|                | gaus6          | symmetric  | 0.6 Hz            |
|                | gaus7          | anti-sym.  | 0.6 Hz            |
|                | gaus8          | symmetric  | 0.6 Hz            |
| Mexican Hat    | mexh           | symmetric  | 0.25 Hz           |
| Morlet         | morl           | symmetric  | 0.8125 Hz         |

©2021 The Johns Hopkins University Applied Physics Laboratory LLC.
scales becomes apparent. For instance, the morl wavelet provides coarser resolution, spanning approximately 200 MHz between scales 1-2, while gaus1 spans approximately 50 MHz in the same range. However, with regard to bandwidth morl’s frequency range extends up to around 400 MHz while gaus1 tops-out around 100 MHz. Depending upon application and input signal, a higher bandwidth, such as with morl, or a finer resolution as with gaus1, may be desired.

C. Wavelet Selection Overview

Since the application of varied wavelet functions to the same input will produce different results, selecting a suitable, or preferably optimal, mother wavelet for analysis is an important step. Some methods have leveraged wavelet function properties such as symmetry, orthogonality, and vanishing moments. Others explored the wavelet function in conjunction with the input data through methods such as visual similarity, cross-correlation, mean squared error, joint entropy measures, and more [23]-[26].

This work explored the use of similarity analysis through cross-correlation in the time-domain between ψ and a sample raw trace data file.

IV. EXPERIMENTAL RESULTS

A. Wavelet Accuracy

Each mother wavelet from Table II was assessed for impact on resultant classification accuracy. Five program loops from each trace file were analyzed. Wavelet coefficients for each scale from 1-50 were calculated. Fifteen trials of 100 training epochs for each mother wavelet were conducted, with test accuracy results shown in Fig. 4.

This experiment was conducted again with 20 trials at 125 training epochs and similar results were obtained: although it’s relatively close, gaus1 and mexh consistently achieved highest test accuracy, while morl was lowest in all cases.

B. Wavelet Time

1) Coefficient Calculation Time

Although model training time is a primary factor in most efforts, with tens of thousands of clock-cycle samples, the time to calculate wavelet coefficients for each cannot be discounted. This is also an important consideration for integrity monitoring of a DUT to achieve near-real-time throughput. As part of the above wavelet accuracy experiments, wavelet coefficient calculation and time across all samples for each mother wavelet was also collected. Coefficients only need to be calculated once in preparation for CNN training, so this
aspect was not repeated for each trial. Fig. 5 shows the coefficient calculation time for each mother wavelet, with gaus variants achieving similar times while mexh and morl were slightly longer in duration.

A dedicated experiment was then conducted, exploring the impact of scale as well as choice of wavelet on coefficient calculation time. In this experiment, the time to calculate wavelet coefficients for 10,000 clock-cycle window samples per wavelet at various scales between 1-1,000 was measured, and repeated for 100 trials. Fig. 6 shows the average time across trials for each mother wavelet at each scale. Three additional mother wavelets, cmor, fbsp, and shan were also included to demonstrate the significant impact both wavelet selection and scale can have in coefficient calculation times. In all cases there exists a linear relationship between scale and coefficient calculation time. Again, mexh and morl show similar times, slightly above (in general) that of gaus variants. However, at low scale ranges, mexh and morl appear slightly faster than gaus variants. Additionally, some separation is also shown between gaus variants, whereby higher orders account for longer durations. These factors can be observed more clearly in the inset of Fig. 6.

2) Training Time

The CNN training time (with a fixed number of epochs and no early-stopping) for each wavelet was also collected and assessed. No meaningful connection was established. Additional work would be required, to include early-stopping assessments, to identify if and how wavelet selection impacts CNN training time.

C. Scale Accuracy

Most experiments in this work applied a fixed scale range covering a broad or focused range of interest relative to included pseudo-frequencies. In general, lower scales (higher frequencies) have proven more information dense and resulted in higher classification performance. This experiment explores the impact of scale on test set classification accuracy using sliding scale windows, conducted in two parts: scales 1 to 596 were explored with a window-width of 100, and sliding the window in increments of 5; scales 500 to 1000 were explored with a window-width of 100, and sliding the window in increments of 10. In both parts, significant overlap of windows was used to approximately localize performance changes. The morl wavelet, 40 epochs, and 10 trials were executed. The results are shown in Fig. 7.

As expected, the lower scale range (from 1 to around 140) offered the best performance for the morl wavelet. Classification continues to decrease as scales increase until plateauing and subsequently decreasing again. It is also worth noting that variance increases at higher scales.

D. Colormaps

Wavelet transform coefficients are calculated and returned as a matrix \( C \) with dimensions \( s \times t \), with \( s \) representing the number of individual scales calculated, and \( t \) the number of time samples. \( C \) can then be used to generate the input to the CNN, typically by applying a
color-mapping whereby the wavelet transform coefficients are translated, for example, to Red Green Blue (RGB) values in accordance with a specified colormap, and dimensionality of the matrix now including three color-channels (in the RGB case).\(^1\)

1) Classification Accuracy

The set of 176 colormaps from matplotlib [37] was used to assess the impact of colormap selection on resultant classification performance. This experiment was conducted for exploratory purposes to establish relative performance, and is an exception to the previously specified program loop and semi-optimized analysis set. It was performed on an earlier program loop consisting of the same instructions in Table I, but with different quantities and sequencing of each. Additionally, due to the amount of computation: less input data and a scaled-back CNN was used. Two trials were conducted using morl with wavelet coefficients from scale 1-251 and 100 epochs, for each colormap. The average classification accuracy of the test set across trials, grouped by colormap class is shown in Table III.

At the class level, there exists little difference between the top four classes, unless increased optimization is sought. The class outlier is Qualitative, which not only produced significantly lower performance but also higher variation within its membership. The top-2 by average accuracy was achieved with cubehelix and twilight\(\_r\) colormaps of the Miscellaneous and Cyclic classes.

However, the top-100 were all within about 1% of the maximum average accuracy achieved.

2) Training Time

Although the CNN training time (with a fixed number of epochs and no early-stopping) for each colormap was also collected and assessed, a meaningful connection was not established. Instead a general increase in training time was observed over the course of the evaluation. Additional work would be required, to include early-stopping assessments, to identify if and how choice of colormap impacts CNN training time.

E. Wavelet Selection Example Results

As mentioned in section III.C., a time-domain cross-correlation technique was explored for mother wavelet selection. Initial results included gaus3 with the highest cross-correlation, and morl with the lowest. Further, many of the Guassian variants achieved similar but slightly lower cross-correlation to gaus3, while mexh’s cross-correlation was also near the top. The real outlier was morl, which resulted in significantly lower average and maximum cross-correlation values as shown in Table IV. Although these results do not completely match the primary wavelet analysis findings in this work, they appear fairly consistent, especially with regard to morl as a non-ideal choice.

\(^1\) CWT and/or scalogram subsampling, resizing, and other preparatory data munging techniques are vital but beyond the scope of descriptions in this paper.

| Class        | Number in Class | Class Avg. Accuracy | Class Std. Deviation |
|--------------|-----------------|---------------------|----------------------|
| Cyclic       | 6               | 92.1 %              | 0.42                 |
| Diverging    | 28              | 92.1 %              | 0.30                 |
| Misc.        | 32              | 91.7 %              | 0.96                 |
| Sequential   | 86              | 91.5 %              | 0.60                 |
| Qualitative  | 24              | 76.9 %              | 3.16                 |

| Mother Wavelet | Mean | Max  |
|----------------|------|------|
| gaus1          | 0.168| 0.801|
| gaus2          | 0.200| 0.839|
| gaus3          | 0.216| 0.946|
| gaus4          | 0.217| 0.900|
| gaus5          | 0.207| 0.929|
| gaus6          | 0.190| 0.822|
| gaus7          | 0.168| 0.807|
| gaus8          | 0.145| 0.706|
| mexh           | 0.202| 0.861|
| morl           | 0.015| 0.109|

Fig. 7: Windowed scale vs. accuracy over 10 trials for morl mother wavelet, with decreasing performance approximately after scale window 40-140.
V. CONCLUSIONS

Wavelet selection impacts both classification performance and efficient coefficient calculation time, as indicated by Figs. 4-6. Scales should be deliberately selected for a given wavelet and input signal under analysis based upon classification performance objectives and constraints to select the minimum quantity and lowest range of scales that achieve this result. As shown for morl, higher scales result in decreased classification performance (Fig. 7), and increased calculation time (Fig. 6). Correlation appears to be an average indicator of classification performance for wavelet selection, while selection based upon experimental results is more favorable.

The choice of colormap, when Qualitative is excluded, is not a significant factor in classification performance. Therefore, 1-channel grayscale was subsequently adopted. This provided sufficient classification performance, with lower memory requirements, and faster CNN training times as compared to 3-channel scalograms.

The gaus1 mother wavelet was found to provide the best overall balance between classification performance and efficient coefficient calculation. Combined with 1-channel grayscale colormap to scalogram, a singular scale of 1 has provided greater than 99% classification accuracy in testing. Considering (6), a scale range of 1 to 22 may provide more robust results in practice.

This work demonstrates that wavelet selection and employment parameters have meaningful impact on analysis outcomes. Researchers should describe this information in their work to support reproducibility. Practitioners can use this information to make informed decisions and should consider optimizing these factors similarly to machine learning architecture and hyperparameters. Although these findings are only claimed for our application, it is believed they’ll apply to other methodologies and use-cases, but additional work from the community is required to achieve generalized results.

REFERENCES

[1] P. Kocher. (1996). “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems.” Presented at the 16th Annual International Cryptology Conference on Advances in Cryptology. [Online]. Available: https://paulkocher.com/doc/TimingAttacks.pdf

[2] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis.” Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, Springer-Verlag, 1999, pp. 388-397.

[3] D. Agrawal, B. Archambeault, J.R. Rao, and P. Rohatgi. “The EM Side—Channel(s).” In: Kalisky B.S., Koç K., Paar C. (eds) Cryptographic Hardware and Embedded Systems - CHES 2002. CHES 2002. Lecture Notes in Computer Science, vol 2523. Springer, 2003, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36400-5_4

[4] M. Hutter, and J.M. Schmidt, “The temperature side channel and heating fault attacks.” Proceedings of the 12th International Conference, CARDIS 2013, Springer International, 2014, pp. 217-235.

[5] J. Ferrigno, and M. Hlaváč. (2008). “When AES blinks: introducing optical side channel.” IET Information Security 2(3): 94-98.

[6] S. Chari, J.R. Rao, and P. Rohatgi. "Template Attacks." In: Kalisky B.S., Koç K., Paar C. (eds) Cryptographic Hardware and Embedded Systems - CHES 2002. CHES 2002. Lecture Notes in Computer Science, vol 2523. Springer, 2003, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36400-5_3

[7] S.S. Clark, B. Ransford, A. Rahmati, S. Guineau, J. Sorber, K. Fu, and W. Xu. “WattsUpDoc: Power side channels to nonintrusively discover untargeted malware on embedded medical devices.” Proceedings of the 2013 USENIX conference on Safety, Security, Privacy and Interoperability of Health Information Technologies. Washington, DC, USENIX Association: 9-9.

[8] J. Aarestad, D. Acharyya, R. Rad, and J. Plusquellic. “Detecting trojans through leakage current analysis using multiple supply pad I_{DDQ}s.” IEEE Transactions on Information Forensics and Security 5(4), 2010, pp. 893-904.

[9] D. Du, S. Narasimhan, R.S. Chakraborty, and S. Bhunia. “Self-referencing: a scalable side-channel approach for hardware trojan detection.” In: Mangard S., Standaert FX. (eds) Cryptographic Hardware and Embedded Systems, CHES 2010. CHES 2010. Lecture Notes in Computer Science, vol 6225. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15031-9_12

[10] M.L. Wilt, M.M. Baker and S.J. Papadakis. “Toward an RF side-channel reverse engineering tool.” 2020 IEEE Physical Assurance and Inspection of Electronics (PAINE), 2020. doi: 10.1109/PAINE49178.2020.937729.

[11] T. Eisenbarth, C. Paar, and B. Wegehnkel. “Building a side channel based disassembler.” In: Gavriloa M.L., Tan C.J.K., Moreno E.D. (eds) Transactions on Computational Science X. Lecture Notes in Computer Science, vol 6340. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-17449-5_4

[12] D. Strobel, F. Bache, D. Oswald, F. Schellenberger, and C. Paar. “SCANDALee: a side-channel-based disassembler using local
electromagnetic emanations.” Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, 2015, pp. 139-144, doi: 10.7873/DATE.2015.0639.

[13] D. Krishnankutty, R. Robucci, N. Banerjee and C. Patel. “Fiscal: Firmware identification using side-channel power analysis.” 2017 IEEE 35th VLSI Test Symposium (VTS), 2017, doi: 10.1109/VTS.2017.798948.

[14] J. Park, X. Xu, Y. Jin, D. Forte, and M. Tehranipoor. “Power-based Side-Channel Instruction-level Disassembler.” 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), 2018, doi: 10.1109/DAC.2018.8465848.

[15] V. Cristiani, M. Lecomte, and T. Hiscock. (2020). “A Bit-Level Approach to Side Channel Based Disassembling.” In: Belaïd S., Güneysu T. (eds) Smart Card Research and Advanced Applications. CARDIS 2019. Lecture Notes in Computer Science, vol 11833. Springer, Cham. https://doi.org/10.1007/978-3-030-42068-0_9

[16] C. Torrence, and G.P. Compo. “A Practical Guide to Wavelet Analysis.” Bulletin of the American Meteorological Society 79:1, 1998, pp. 61-78.

[17] X. Charvet, and H. Pelletier. “Improving the DPA attack using wavelet transform,” September 26-29 2005, Honolulu, Hawaii, USA, in NIST Physical Security Testing Workshop.

[18] N. Debande, Y. Souissi, M.A. El Aabid, S. Guillely, and J. Danger. “Wavelet Transform Based Pre-processing for Side Channel Analysis.” In Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture Workshops (MICROW ‘12). IEEE Computer Society, USA, pp. 32–38.

[19] JPEG 2000 image coding system: Core Coding System, ISO/IEC IS 15444-1 ITU-T 800, 2019.

[20] S. Lawson, and J. Zhu. “Image compression using wavelets and JPEG2000: a tutorial.” Electronics & Communication Engineering Journal, June 2002, pp. 112-121.

[21] M.J. Burke, and M. Nasor. (2004). “Wavelet based analysis and characterization of the ECG signal.” Journal of Medical Engineering & Technology, 28:2, pp. 47-55, DOI: 10.1080/0309190031000121532.

[22] A. Chakraborty, and D. Okaya. (1995). “Frequency-time decomposition of seismic data using wavelet-based methods.” GEOPHYSICS 60:6, pp. 1906-1916. https://doi.org/10.1190/1.1443922

[23] L. Xu, Y. Dan, and Q. Wang. “Comparisons between real and complex Gauss wavelet transform methods of three-dimensional shape reconstruction,” Proceedings of SPIE Applied Optics and Photonics AOPC 2015: Optical Test, Measurement, and Equipment, (8 October 2015), doi: 10.1117/12.2199800

[24] W.K. Ngui, M.S. Leong, L.M. Hee, and A.M. Abdelrhman. “Wavelet analysis: mother wavelet selection methods.” Applied Mechanics and Materials Vol. 393, Trans Tech Publications, 2013, pp. 953-958. doi:10.4028/www.scientific.net/AMM.393.953

[25] A. Singh, S. Maiti, and R.K. Tiwari. “Selection of optimum wavelet in CWT analysis of geophysical downhole data.” Journal of Indian Geophysical Union, March 2017, 21:2, pp. 153-166.

[26] A. Saikia, N.M. Kakoty, and S.M. Hazarika. “Wavelet selection for EMG based grasp recognition through CWT.” In: Abraham A., Lloret Mauri J., Buflord J.F., Suzuki J., Thampi S.M. (eds) Advances in Computing and Communications, ACC 2011. Communications in Computer and Information Science, vol 191. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-22714-1_13

[27] J.W. Nilsson. Electric Circuits, Addison-Wesley Publishing Company, Inc., 1983, pp. 640-642, 688-689.

[28] I. Daubechies. “The What, Why, and How of Wavelets,” in Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, 1992. doi:10.1137/1.9781611970104.ch1

[29] “Frequently Asked Questions,” from Advanced Concepts (Wavelet Toolbox), 2002. [Online]. Available: http://www.ece.northwestern.edu/local-apps/matlabhelp/toolbox/wavelet/ch06_a26.html

[30] AVR® Instruction Set Manual, DS40002198A, Microchip Technology Inc., 2020. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/AVR-Instruction-Set-Manual-DS40002198A.pdf.

[31] G.R. Lee, R. Gommers, F. Wasilewski, K. Wohlfahrt, A. O’Leary. “PyWavelets: A Python package for wavelet analysis,” 2019. Journal of Open Source Software, 4(36), 1237, https://doi.org/10.21105/joss.01237.

[32] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, et al. “Array programming with NumPy,” 2020. Nature, 585, 357–362, DOI:10.1038/s41586-020-2649-2.

[33] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” 2020. Nature Methods, 17(3), 261-272.

[34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, et al. "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[35] F. Chollet and others. “Keras,” 2015. Software available from https://keras.io.

[36] MathWorld Team. “Gaussian Function,” from Wolfram MathWorld, 2021. [Online]. Available: https://mathworld.wolfram.com/GaussianFunction.html.

[37] J. Hunter, D. Dale, E. Firing, M. Droettboom, and the Matplotlib development team. “Choosing colormaps in Matplotlib,” from Matplotlib, 2021. [Online]. Available: https://matplotlib.org/stable/tutorials/colors/colormaps.html.

©2021 The Johns Hopkins University Applied Physics Laboratory LLC.