Less Traces Are All It Takes: Efficient Side-Channel Analysis on AES

Zhiyuan Xiao, Chen Wang, *Member, IEEE*, Jian Shen, *Member, IEEE*, Q. M. Jonathan Wu, *Senior Member, IEEE*, and Debiao He, *Member, IEEE*

***Abstract*—In cryptography, side-channel analysis (SCA) is a technique used to recover cryptographic keys by examining the physical leakages that occur during the operation of cryptographic devices. Recent advancements in deep learning (DL) have greatly enhanced the extraction of crucial information from intricate leakage patterns. A considerable amount of research is dedicated to studying the SubByte (SB) operations of the advanced encryption standard (AES). This is because the SB process, which generates numerous transitions between 0s and 1s during encryption, results in significant energy leakage. However, traditional analysis models primarily focus on the initial round of SB operations in AES, which are less effective on mobile terminals where it is difficult to collect enough signals. These models often neglect additional operations and subsequent rounds, thus providing limited insights from small datasets. Consequently, this limitation has a direct impact on the accuracy and efficiency of key recovery. Our study uses ρ-test analysis to show that significant leakage occurs not only during the S-box operation but also during the AddRoundKey (AR) phase of AES. To address these challenges, we propose a new SCA method, that is, optimized for small sample sizes. This method includes a new comprehensive round trace labeling algorithm, which simultaneously analyzes the SB and AR stages of each AES round. Additionally, we introduce the peak precise localization algorithm to accurately identify the points of energy leakage during each encryption round. Our experiments, conducted with power and electromagnetic (EM) datasets from the STM32F303 microcontroller, demonstrate that our method can reliably recover keys with as few as 20 traces. These results highlight the enhanced capability of our method in handling the complexities of small sample datasets in cryptographic analysis.**

***Index Terms*—Advanced encryption standard (AES), deep learning-based side-channel analysis (DLSCA), joint side-channel analysis (SCA), small sample.**

Received 21 July 2024; revised 22 October 2024; accepted 5 December 2024. Date of publication 16 December 2024; date of current version 21 May 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB2703700; in part by the National Natural Science Foundation of China under Grant U21A20465, Grant 62302457, and Grant 62325209; in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ24F020008; in part by the Program for Leading Innovative Research Team of Zhejiang Province under Grant 2023R01001; in part by the Fundamental Research Funds of Zhejiang Sci-tech University under Grant 23222092-Y and Grant 22222266-Y; in part by the "Pioneer" and "Leading Goose" Research and Development Program of Zhejiang under Grant 2023C01119; and in part by the Research Funds for the Central Universities under Grant 2042023KF0203. This article was recommended by Associate Editor Y. Lyu. *(Corresponding authors: Chen Wang; Jian Shen.)*

Zhiyuan Xiao, Chen Wang, and Jian Shen are with the School of Information Science and Engineering (School of Cyber Science and Technology), the Zhejiang Provincial Key Laboratory of Digital Fashion and Data Governance, and the Zhejiang Provincial International Cooperation Base for Science and Technology on Cloud Computing Security and Data Aggregation, Zhejiang Sci-tech University, Hangzhou 310018, China (e-mail: anderson__xiao@163.com; wangchen@zstu.edu.cn; s_shenjian@126.com).

Q. M. Jonathan Wu is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: jwu@uwindsor.ca).

Debiao He is with the State Key Laboratory of Software Engineering, Computer School, Wuhan University, Wuhan 430072, China (e-mail: hedebiao@163.com).

Digital Object Identifier 10.1109/TCAD.2024.3518414

I. Introduction

THE CONCEPT of side-channel analysis (SCA) was originally introduced by Paul Kocher in his influential paper on timing analysis [1], where he demonstrated that the varying runtimes of cryptographic algorithms can reveal critical key information. Kocher also introduced the technique of power analysis [2], which exploits the fact that circuits consume different amounts of power depending on the input data. Electromagnetic (EM) analysis, as discussed in [3], is a noncontact analysis method and an important tool in SCA. Numerous hardware and software implementations of key cryptographic algorithms, including the well-known advanced encryption standard (AES) [4], have been successfully compromised by SCA techniques [5], [6], [7].

In recent years, deep learning (DL) techniques [8] have achieved significant advancements in image classification [9] and natural language processing [10]. Due to their capability of extracting features from raw data, DL models can assist attackers in identifying correlations between physical measurements and the internal states of the processed algorithms. As a result, an increasing number of SCA researchers are utilizing DL techniques in SCA research [11], [12], [13].

Martinasek and Zeman [14] introduced the innovative deep learning-based SCA (DLSCA) method for analyzing the power consumption of smart cards. This method utilizes a three-layer multilayer perceptron network to break the AES-128 key implemented on smart cards [15]. Since its introduction, DLSCA has been applied to various hardware implementations of cryptographic algorithms [16]. Sayakara et al. [17] proposed an EM emission-based DLSCA method for accurately detecting cryptographic algorithms in microcontrollers. Additionally, Hu et al. [18] suggested a multiloss regularized denoising auto-encoder framework to reduce noise in SCA and thereby decrease the number of traces needed for key recovery. These studies highlight the effectiveness of DL techniques in the SCA community.

Traditional analysis models, which primarily focus on the initial round of SubByte (SB) operations in AES, are not

1937-4151 © 2024 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.

as effective in dynamic environments, such as mobile self-organizing networks. These models often overlook additional operations and subsequent rounds, limiting the insights gained from small datasets and impeding the accuracy and efficiency of key recovery. Furthermore, decrypting keys from these limited datasets proves to be challenging when only a few samples can be collected from highly mobile devices. As a consequence, the accuracy and effectiveness of key recovery methods suffer, as they are unable to adapt adequately to the restricted availability of data.

Various research efforts have been undertaken to tackle the challenge of limited data availability and enhance the accuracy and efficiency of SCA. Fu et al. [19] introduced a multibyte power analysis approach; however, they did not integrate it with a comprehensive analysis of 10 rounds. Hu et al. [20] proposed a cross-subkey training method customized for small sample scenarios, yet this method also did not include a 10-round analysis. Kong et al. [21] applied hybrid supervised neural networks to enhance power trace analysis in small sample settings, but the approach was not extended to cover all 16 bytes. Additionally, Cai et al. [22] employed a variational autoencoder to improve performance in small sample analyses, but similarly did not extend the methodology to cover all 16 bytes. Moreover, Huang et al. [23] focused on improving the security of satellite-terrestrial networks in downlink nonorthogonal multiple access (NOMA) systems through precoding techniques. Muhammad et al. [24] provided an in-depth analysis of various security threats faced by smartphones, including APT attacks, sensor-based attacks, and side-channel attacks. These studies highlight the recurring theme of not fully exploiting the potential of extending analysis across all operational rounds and byte capacities.

While previous research has examined SCA techniques for small-sample scenarios, these studies often fail to thoroughly address all potential points of information leakage in the AES encryption process. This includes the leakage of information in both the SB and AddRoundKey (AR) phases of each round, as well as the patterns of leakage among individual bytes. This lack of comprehensive coverage can hinder the effectiveness of SCA techniques in accurately identifying and exploiting the complete range of vulnerabilities inherent in the AES encryption process.

To address the limitations of partial analysis in small sample datasets, this study presents a new joint multibyte, multioperation analysis method. This approach effectively incorporates the SB and AR operations in each round of AES encryption, while also considering the combined leakage patterns of all 16 bytes in each round. By doing this, our method resolves the inaccuracies resulting from reliance on a single leakage source and fully utilizes the inherent characteristics of the AES algorithm's structure. As a result, it significantly improves the accuracy and efficiency of SCA, providing attackers with a reliable tool to analyze extensive leakage information and recover keys.

The contributions of this article are as follows.

1) *A joint SCA structure based on multibyte and multiple operations is proposed.* In this article, we present a joint SCA structure (Fig. 1) that considers the two most vulnerable operations, SB and AR, in AES. It simultaneously analyzes 16 bytes in each round, increasing the sample size for SCA with less traces collected. This method addresses the accuracy and efficiency issues caused by limited sample sizes, allowing for the utilization of abundant leaked information from the traces.

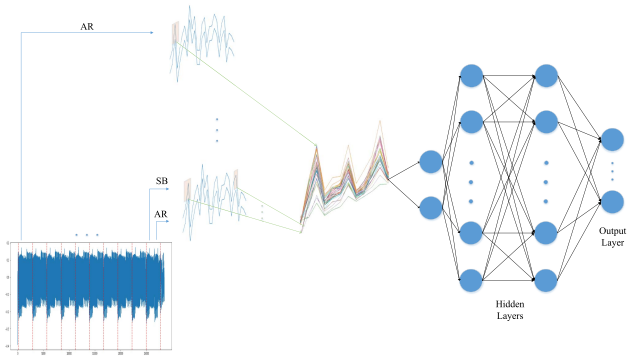


Fig. 1. Joint SCA structure.

2) *We present the comprehensive round trace labeling algorithm (CRTLA) to enhance cryptographic data analysis.* The CRTLA aligns the waveform peaks of the SB and AR stages, greatly increasing the number of usable samples.
3) *We present the peak precise localization algorithm (PPLA) to enhance cryptographic data analysis.* The PPLA generates detailed labels for each encryption round by accurately identifying the plaintext and key, allowing for a more thorough and precise analysis.
4) *Only 15 traces are required to recover the key.* Experimental results obtained from the power consumption and EM datasets of the STM32F303 demonstrate that our method can effectively retrieve the key using only 15 traces. This highlights its remarkable ability to address concerns related to small sample sizes.

The remainder of this article is structured as follows. Section II provides an introduction to several preliminary concepts, such as an overview of AES, the modeling of class SCA, and the relevant evaluation metrics. Section III elaborates on the primary ideas behind our proposed joint SCA method. Section IV discusses the experimental results and their analysis. Finally, Section V summarizes the main contributions and outlines future research directions.

II. Preliminaries

In this section, we introduce AES, DLSCA, and the evaluation metric for SCA.

A. Advanced Encryption Standard

Understanding the flow of operations within the AES encryption process is crucial for identifying potential vulnerabilities and points of attack in cryptographic implementations.

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.

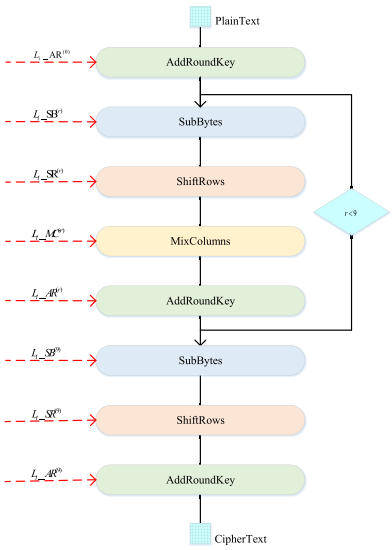


Fig. 2. AES flowchart illustrates the process of ten rounds of AES encryption, with each round named. In the ith process of encryption, during the first round (the round counter $r = 0$), the AR is denoted as $L_i_AR^{\{0\}}$.

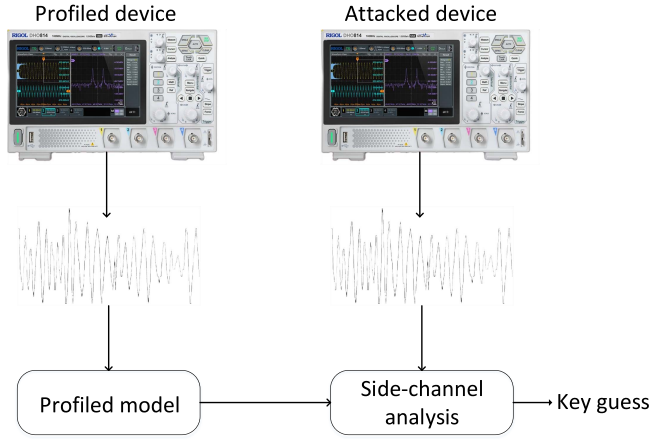


Fig. 3. Schematic of DLSCA.

As shown in Fig. 2, AES-128 encryption is structured as a ten-round process, with four different operations, named AR, SB, ShiftRows (SRs), and MixColumns (MCs), respectively. The initial round (round 0) consists solely of the AR operation, which combines the plaintext with the key. Subsequent rounds—up to the penultimate one—comprise a sequence of four operations: 1) SB; 2) SR; 3) MC; and 4) AR. These operations progressively transform the plaintext into an increasingly scrambled intermediate state. The final round omits the MC step, including only the SB, SR, and AR operations, resulting in the ciphertext.

Previous studies in SCA, as mentioned in [25] and [26], have generally identified the output of the S-Box in the SB operation as a primary point of attack due to its nonlinear properties, which are inherently difficult to predict and therefore a source of information leakage. Nevertheless, AES has several stages where key-dependent intermediate values, such as those from SR, MC, and AR, are equally susceptible to leakage. According to [27], a strategic combination of leaked information from these stages—particularly the interaction between SB and AR—can result in a more effective recovery of the encryption key.

By analyzing the sequential flow of operations, we gain insight into the progression of encryption and can better identify where sensitive information may be compromised. Such analysis is crucial in enhancing encryption against potential SCA attacks by identifying which stages require additional safeguards or enhanced security measures.

B. DLSCA

Fig. 3 outlines the two-phase process of DLSCA applied to cryptographic devices.

In the first phase, called profiling, the model learns the specific power consumption patterns—these are the "traces" captured by an oscilloscope, as shown on the left side of the image labeled "Profiled device." Each trace corresponds to known variables of the encryption process. A DL model is trained on these traces to identify how the device leaks information during operations like AR or SB.

The second phase is the attack phase, depicted on the right side of the image labeled "Attacked device." In this phase, the trained model from the profiling phase is utilized to analyze new, previously unseen traces collected from a targeted device. The objective is to predict the encryption key by examining the power consumption patterns during device operation. The model applies its learned parameters to these new traces and generates a probability distribution of potential keys.

The model calculates the probabilities of different keys as shown in the equation

$$p = f(L; \theta^*) \tag{1}$$

where θ^* represents the model parameters learned during the training phase and $p[i]$ denotes the predicted probability given the sensitive variable $Z = i$. The model is then used to classify new or unfamiliar traces. The key guess with the highest probability is selected as the most likely correct key.

DLSCA improves the precision and efficiency of SCA by using DL techniques to interpret noisy data and remain effective in complex attack scenarios. The application of advanced algorithms and loss functions, such as cross-entropy, as well as optimization methods like RMSprop, facilitates the handling of large datasets and helps reduce prediction errors.

The flow shown in Fig. 3 illustrates the entire process from profiling on the left, through DL analysis, to key prediction on the right. This process begins with the gathering of power traces from the profiled device, proceeds through the DL analysis where the model utilizes its learned parameters on the traces, and concludes with the attacked device, where the actual SCA occurs and the key guess is generated.

This DLSCA methodology represents a significant breakthrough in cryptanalysis, highlighting the importance of strong security measures in the design and testing of cryptographic devices. By successfully capturing and leveraging the subtle deviations in power traces, DLSCA presents a complex challenge to the security of existing cryptographic methods.

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.

C. Evaluation Metrics

The guessing entropy (GE) metric in SCA quantifies the effectiveness of an attack by measuring the average number of guesses required to discover the correct key. GE is calculated as the expected value of the rank of the correct key k among all possible key guesses g, as shown in the formula

$$GE = E(\text{rank}_{k^*}(g)) \tag{2}$$

where $\text{rank}_{k^*}(g)$ denotes the rank position of the correct key k^* within all the key guess vectors g and E stands for the expectation operator. A lower-GE value means the correct key is found with fewer guesses, indicating a more effective attack.

In addition to GE, partial GE (PGE) is useful for evaluating model performance, especially when dealing with noisy data. PGE measures the average rank of a subkey within all possible subkey predictions. By using a trained model, the probabilities for each key are calculated for every trace, forming probability matrices P_i. In high-signal-to-noise ratio (SNR) scenarios, fewer traces are required to find the correct key rank, making PGE a valuable metric.

The model calculates a score δ_k for each potential key using the predicted probabilities

$$\delta_k = \sum \left(\ln_{p_i}[K]\right) \tag{3}$$

where p_i is the predicted probability vector for the ith trace and $f(l_i; \theta^*)$ represents the model's predicted probabilities for the ith trace. This metric is crucial for evaluating the attack's success rate, particularly when noise complicates the data.

III. Main Idea

A. Theoretical Foundation

In the section on theoretical foundation, we explore the principles of leakage analysis and cross-byte analysis. Leakage analysis investigates the reasons why the SB and AR stages of the AES encryption process are particularly susceptible to leakage, taking into account both physical and mathematical viewpoints. Likewise, cross-byte analysis is examined from these two perspectives, elucidating how and why cross-byte attacks can be successfully carried out by exploiting these vulnerabilities.

1) Theoretical Leakage Analysis: During the execution of AES encryption software, the main cause of energy dissipation is the phenomenon of bit transitions within CMOS cells. This effect is governed by a correlation where the amount of data processed directly affects the level of power consumption, as mentioned in [25], [26]. While the nonlinear S-Box outputs are often the focus of cryptanalysis in software-based AES attacks, vulnerabilities go beyond the S-Box. Other stages in AES that handle key-related intermediate values also pose significant risks for cryptographic breaches.

Structured as a cipher with 10 rounds, AES-128 operates through a sequence of four key operations per cycle: 1) SB; 2) SR; 3) MC; and 4) AR. Importantly, the MC operation is not present in the final cycle.

From a physical standpoint, CMOS logic gates experience energy exchange when transitioning between binary states, resulting in variations in power consumption. The SB operation, well-known for its nonlinear properties, involves multiple bit transformations. The various operations within the S-Box increase the frequency of bit transitions, thereby magnifying the potential for energy leakage.

Although the AR operation is a linear transformation, its unique XOR operation with the AR in each cycle initiates changes in the bit states. In situations where there are mismatches between data and key bits at specific points, these discrepancies can lead to bit conversions, thereby creating additional pathways for energy dissipation.

In theory, the SB and AR processes are considered the main sources of potential information leakage within the AES ciphering algorithm. Building on this idea, a novel SCA structure can be carefully developed to track the energy dissipation specifically during these operations. By thoroughly examining the leakage at these crucial points, valuable insights into the encryption key can be obtained.

The functions describing these leakages are defined as follows.

The first function, $L_i_SB^r$, explains the leakage from the SB operations process

$$L_i_SB^{(r)} = \text{HW}\left(\text{SBOX}(K^{(r-1)} \bigoplus P^{(i)})\right) \tag{4}$$

where i indexes the encryption rounds, r represents the specific round in the AES procedure, HW$(\cdot)$ denotes the Hamming weight function, SBOX$(\cdot)$ signifies the SB operation, $K^{(r-1)}$ is the key from the previous round, $P^{(i)}$ denotes the plaintext input for round i, and $\bigoplus$ indicates the bitwise XOR operation.

The second function, $L_i_AR^r$, relates to the AR operation

$$L_i_AR^{(r)} = \text{HW}\left(K^{(r)} \bigoplus P^{(i)}\right). \tag{5}$$

Similarly, the variables maintain their defined meanings but are contextualized within the framework of the AR stage.

In AES, the encryption begins with an initial AR operation, followed by combinations of SB and AR operations in each round, amounting to 21 operations in total. Hence, a comprehensive leakage function, $L_{\text{total}}^{(r)}$, is formulated as

$$L_{\text{total}}^{(r)} = \sum_{i=0}^{n} \left(L_i_SB^{(r)} + L_i_AR^{(r)}\right) \tag{6}$$

where n represents the total number of rounds, encompassing the entire spectrum of encryption processes for analyzing leakage phenomena.

The detailed depiction of AES operations in Fig. 2 goes beyond mere pseudocode, providing a careful illustration of the operational schema.

2) Cross-Byte SCA: In the AES encryption process, every byte processed in the first round poses a possible vulnerability for an attacker. Therefore, it is essential to go beyond focusing solely on the initial byte and consider examining all processed bytes. This method, referred to as cross-byte SCA, entails the attacker analyzing not only one byte but multiple bytes or even the entire encryption process.

To establish a mathematical framework, one can employ a linear matrix method akin to multivariate linear regression. The

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.

relationship between the leakages and the cryptographic elements (the key and the plaintext) can be succinctly expressed through a matrix equation

$$Y = X\beta \tag{7}$$

where β represents an array of estimated scaling factors, Y denotes the vector of measured leakages, and X represents a design matrix. Each column of X encapsulates the Hamming weight observed after S-box transformations across different bytes, illustrated as follows:

$$\begin{bmatrix} L_1 \\ \vdots \\ L_N \end{bmatrix} = \begin{bmatrix} \mathrm{HW}(S[K_1 \oplus P_1]) & \cdots & \mathrm{HW}(S[K_N \oplus P_N]) \\ \vdots & \ddots & \vdots \\ \mathrm{HW}(S[K_1 \oplus P_1]) & \cdots & \mathrm{HW}(S[K_N \oplus P_N]) \end{bmatrix} \beta. \tag{8}$$

By using different mathematical formulas, we have demonstrated the theoretical feasibility of cross-byte SCA on AES encryption. This theoretical basis highlights the possibility of utilizing advanced analytical techniques to uncover vulnerabilities in cryptographic security.

To ensure that cross-byte SCA is practical and effective, it is crucial to have accurate power measurement techniques and advanced mathematical models. These models need to be able to identify and interpret the subtle variations that are part of the encryption process, extracting information related to the encryption key from the surrounding noise. Additionally, it is important to be skilled in handling large datasets. This expertise allows for thorough analyses of every byte in each encryption round, requiring an understanding of the nonlinear transformations found in S-box outputs and the interaction between bytes throughout the encryption cycle.

B. Introduction to Joint SCA

Here, the novel joint SCA structure is introduced, which is illustrated in Fig. 1. Both SB and AR operations in AES are considered in this structure. 16 bytes in each round are analyzed simultaneously to increase the sample size for SCA. This structure also includes two novel algorithms: 1) the CRTLA and 2) PPLA.

1) Comprehensive Round Trace Labeling Algorithm: In this subsection, we introduce a new algorithm called CRTLA, which aims to thoroughly investigate energy leakage in each round of the AES algorithm. Traditional SCA techniques typically focus on analyzing the initial round of encryption operations, particularly by examining the relationship between the plaintext and the first AR operation to identify intervals of interest for energy leakage. However, these methods overlook potential energy leakage beyond the first round of the AES encryption algorithm. To address this limitation, we have developed and implemented an algorithm capable of computing the round key for each round of the AES encryption process and determining the encryption result for every round. As a result, this algorithm generates the necessary labels for analysis, allowing the identification of intervals of interest for specific operations within each round.

As shown in Algorithm 1, the first step involves key expansion, which is crucial in AES-128 encryption. The key expansion function generates a series of round keys for each encryption round based on the initial 128-bit key. Initially, the 128-bit key is divided into four 32-bit words, which are then placed at the beginning of the round key array.

Algorithm 1 CRTLA

Input: *PT*, *KEY*, *C*
Output: Label[ ]

```
R_state = [ ];
kw = KE(key);
state = AR(PT, kw[0]);
Label[0] = S(state⊕kw[0]);
for i from 1 to 10 do
    state = SB(state);
    state = SR(state);
    state = MC(state);
    state = AR(state, kw[i]);
    R_state.append(state);
    Label[i] = S(state⊕kw[i]);
end for
state = SB(state);
state = SR(state);
state = AR(state, kw[10]);
R_state.append(state);
Label[10] = S(state⊕kw[10]);
return Label
```

Subsequent key generation involves various operations, such as RotWord, SubWord, and applying Rcon, to link each newly generated round key with its predecessor. This process produces a sequence of keys for the 10 encryption rounds in AES.

The key to the algorithm lies in calculating the labels produced at each encryption round. This calculation depends on the active states of the hardware processing units, which directly correspond to the values of the processed data. Specifically, when data is transferred from memory to the data bus during S-box replacement, there is a noticeable increase in energy consumption.

Therefore, it is reasonable to designate the output values generated during processing as labels for analysis. In AES-128, the S-boxes can generate 256 distinct output values, and each of these unique outputs is considered as a separate case for analysis.

In each encryption round, the key operations include SB, SR, MC, and AR. Together, these operations form the basis of AES encryption. The implemented encryption function captures the state from the initial AR operation and continues until after the final round of encryption is completed. This approach allows for a detailed analysis of energy leakage in each round, making it possible to examine the encryption process in great detail.

For each encryption operation round, we use a label obtained through S-box substitution after performing the XOR operation between the key for that round and the plaintext of the corresponding round. For example, R_{state} mentioned in the algorithm's initial line is a list used to store the state of each round. The label for the first round is calculated in the fourth

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2 Algorithm of Peak Localization

Input: ρ-test, *KEY*, *C*
Output: Trace[ ] Label[ ]

```
1: index[21][16] = [ ];
2: for round from 1 to 21 do
3:    for byte from 1 to 16 do
4:       start, end = windows(round, byte, ρ-test)
5:       index[round][byte]=-np.log10(ρ-test_data[round]
6:       [byte][start:end]);
7:    end for
8: end for
9: Trace = get_traces(traces, index)
10: Label = get_label(labels, index)
```

line as Label[0] $= S(\text{state} \bigoplus \text{key}[0])$. Then, the loops from the fifth to the twelfth lines calculate the labels for the second to the tenth rounds, where Label[i] $= S(\text{state} \bigoplus \text{key}[i])$.

Finally, the label for the last round is computed in the seventeenth line as Label[10] $= S(\text{state} \bigoplus \text{key}[10])$. This approach allows for targeted analysis of each round, improving the accuracy and efficiency of energy leakage detection and ultimately enhancing the effectiveness of key recovery.

We also employed a ρ-test analysis method [28], making use of the generated labels and data obtained during the encryption process (e.g., power consumption traces). This method helps to identify critical intervals of the SB and AR steps during AES encryption.

The ρ-test involves a statistical approach to determine whether there is a significant correlation between a specific operation and the collected data. By utilizing this method, researchers can accurately identify the steps within each round of the AES encryption algorithm that have the most significant impact on the final encryption result.

Combining the algorithm description with the aforementioned methodologies, our algorithm offers improved energy leakage labels for SCA by carefully calculating state changes in each round and the resulting state after AR operations. With these accurate labels, we gain a deeper understanding of and localize energy leakage within each encryption step of AES, thus providing new insights for identifying potential security vulnerabilities and enhancing the algorithm's security.

The function describing these leakages is defined as follows:

$$S_{\text{state}} = \text{SBOX}\Big[p[i] \bigoplus k[i]\Big] \tag{9}$$

where i represents the number of rounds encrypted by AES, HW$(\cdot)$ is the Hamming weight function, $S(\cdot)$ denotes the S-box operation, k represents the key, p signifies the plaintext, and $\bigoplus$ denotes the XOR operation.

2) Peak Precise Localization Algorithm: As shown in Algorithm 2, in our investigation of SCA for AES encryption, we have developed an algorithm called PPLA that aims to identify the peaks of energy leakage in each encryption round. The foundation of this alignment algorithm is based on the encryption labels for each round, which incorporate statistical tests to pinpoint the moment when energy leakage is most pronounced for each operation phase. By conducting a ρ-test on the encrypted data from each round, we specifically focus on extreme points during SB or AR operations that highlight the most significant energy leaks in the power consumption trace. These extreme points indicate critical moments in AES encryption operations.

To identify these extreme points, we created a specialized code flow. This code carefully examines each byte of the encrypted data in order, using a negative logarithmic transformation to highlight the more notable values in the ρ-test. Afterward, it precisely locates the most significant extreme points within a specified time window. Once found, these critical extreme points are reliably documented and stored in an array with dimensions "[21,16]," which represents 16 bytes from each of the 21 encryption rounds.

After extracting the indices of these extreme points, we introduced a visualization step to further validate that these indices indeed correspond to peaks in energy leakage. By utilizing the "plt.figure" and "plt.plot" functions from the matplotlib library, we generated a graph depicting the results of the ρ-test, with each byte in each round of encryption represented by a line. The negative logarithmic value of each line highlights regions of high-statistical significance. Within the graph, we labeled the locations of each predicted extreme point with red dashed lines to indicate the peak locations. Additionally, we plotted intervals around these points to ensure that the selected range is reasonable and encompasses all significant leaks.

This visualization serves not only to validate the accuracy of our extreme point selection but also to improve the precision of using these extreme points for data interception and label assignment in subsequent steps. Afterwards, we used the *get_traces* function to extract data segments centered on these extreme points from the overall power consumption trace data. This function navigates through the set of traces and captures fixed-length segments around each point based on the extreme point index, thus creating a new collection of trace data segments that will be used as input to the machine learning model.

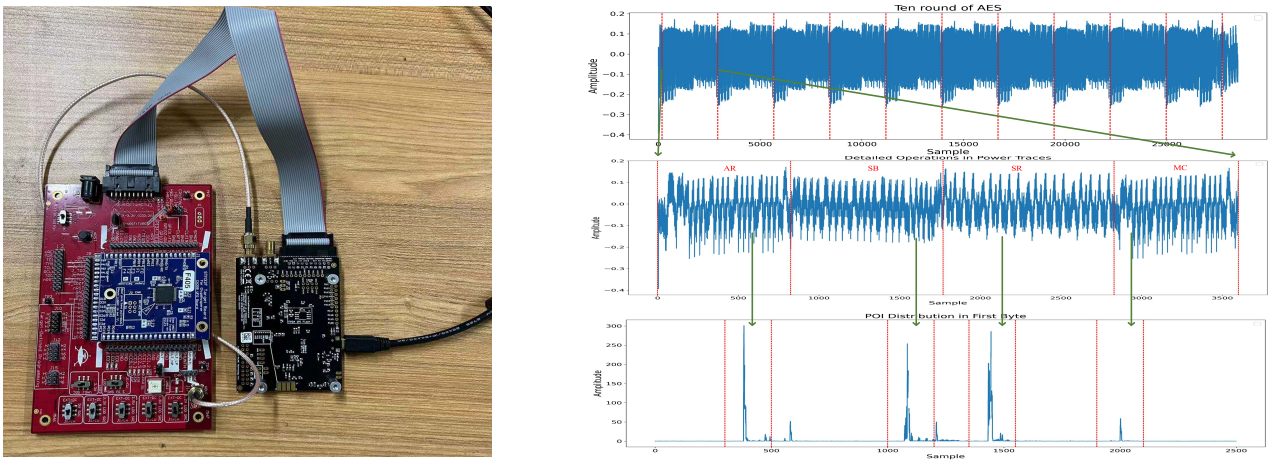
The *get_labels* function, however, works together with the *get_traces* function to create the corresponding set of labels. This ensures that each data fragment is aligned with a specific label, creating a well-structured dataset, that is, ready for DL model training.

Finally, we employ the state-of-the-art convolutional neural network model (CNN best) [29] to train the expanded trace and label dataset. This extensive procedure significantly improves our accuracy and efficiency in analyzing the security of AES encryption algorithms, thereby revealing the energy leakage phenomenon inherent in the encryption process.

IV. Experimental Results and Analysis

In this section, we first outline our experimental conditions. Next, we evaluate the proposed joint SCA model using traces obtained from the CW308T-STM32F3 board. Furthermore, we validate the effectiveness of our model on both the power consumption dataset and the EM dataset of the STM32F303. For these experiments, all of our models are constructed

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 4. (a) Device utilized for collecting power traces. (b) First row shows the collected power traces and the red line splits the ten rounds of AES. The second row shows The details of the four operations in power traces. The third row shows the POI for the four operations of the first byte in power traces.

and trained using the DL frameworks Keras-GPU 2.3.1 and tensorflow-GPU 2.1.0. The primary hardware configurations of the computers used include an Intel Core i7-9750H CPU @2.60 GHz and an NVIDIA GeForce GTX 4090 6-GB GPU, which are employed for all numerical computations and model training tasks throughout the experiment.

A. Assumptions

Applying CNN-best networks to SCA necessitates the fulfillment of the following three requirements.

1) Equipment Requirements: The attacker must possess equipment capable of deploying its encryption program and a capture device capable of recording power traces as well as EM traces during the encryption process.

2) Choose Plaintext and Key: The attacker must have the capability to select any plaintext and key on a device and execute the encryption process repeatedly as needed.

3) Data Acquisition: During the data acquisition phase, it should be possible to simultaneously acquire power traces and EM traces of devices engaged in the encryption process. This implies that the oscilloscope used for data acquisition must have at least three channels to capture data (voltage, power consumption, and EM) concurrently.

B. Experimental Results

Here, we show the experimental results from two aspects: 1) the detected power traces and 2) the EM traces via STM32F303.

1) Power Traces via STM32F303: The dataset was obtained using a Chipwhisperer device that was set up with a sampling frequency of 40MHz. This device is specifically designed to measure power leakage from a chip. The experimental setup involved using a CW308T-STM32F303 target board, which housed an Arm Cortex M4 chip. This chip executed the TinyAES-128 encryption algorithm in Electronic CodeBook (ECB) mode. The capture setup is depicted in Fig. 4(a).

The complete dataset consists of 24 power traces derived from fixed keys and randomly generated plaintexts obtained from a computer. Each of these traces captures 30 000 sampling points and covers the entire AES-128 encryption process, including both SB and AR operations. Fig. 4(b), first row, illustrates the power traces across all 10 rounds of encryption, with the main focus of the experiment being the analysis of power leakage during the SB and AR operations in each round. Except for round 0, each round consists of both SB and AR operations, totaling 21 distinct operations. By examining these power traces, the STM32_POWER handmade dataset was curated, resulting in 8064 traces (24 traces per operation multiplied by 21 operations multiplied by 16 bytes). Out of these, 5000 traces were used for analysis, and 3000 for the attack phase.

After acquiring the data, the CRTLA was used to thoroughly investigate power leakage in each round of the AES algorithm. Fig. 4(b), second row, shows the power leakage during the initial round of AES encryption operations. The x-axis represents the time series observed at different points in the encryption process, while the y-axis represents the magnitude of energy leakage. In this figure, distinct phases of the encryption operation are indicated by red dashed lines, and the 16 waveforms within each phase correspond to the 16-byte data blocks of AES. By analyzing the variation in these waveforms, we can identify the characteristics of each phase, such as SB and AR. Fig. 4(b), third row, highlights the leakage of the first byte of the first round, revealing that the SB and AR portions exhibit the most significant leakage. This row shows the distribution intervals of the different operations of the first byte in the first round of AES encryption after ρ-test analysis of the power traces. The x-axis represents the time series, while the y-axis represents the statistical values of the ρ-test. In this figure, the red dashed line marks the different operation intervals. It can be observed that several distinct peaks appear in the various operational intervals. These peaks indicate changes in statistical significance, which are usually related to power consumption during cryptographic operations. The highest peaks are seen in the first interval, indicating that the SB operation has the strongest statistical significance, followed closely by peaks generated by the AR operation.

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.

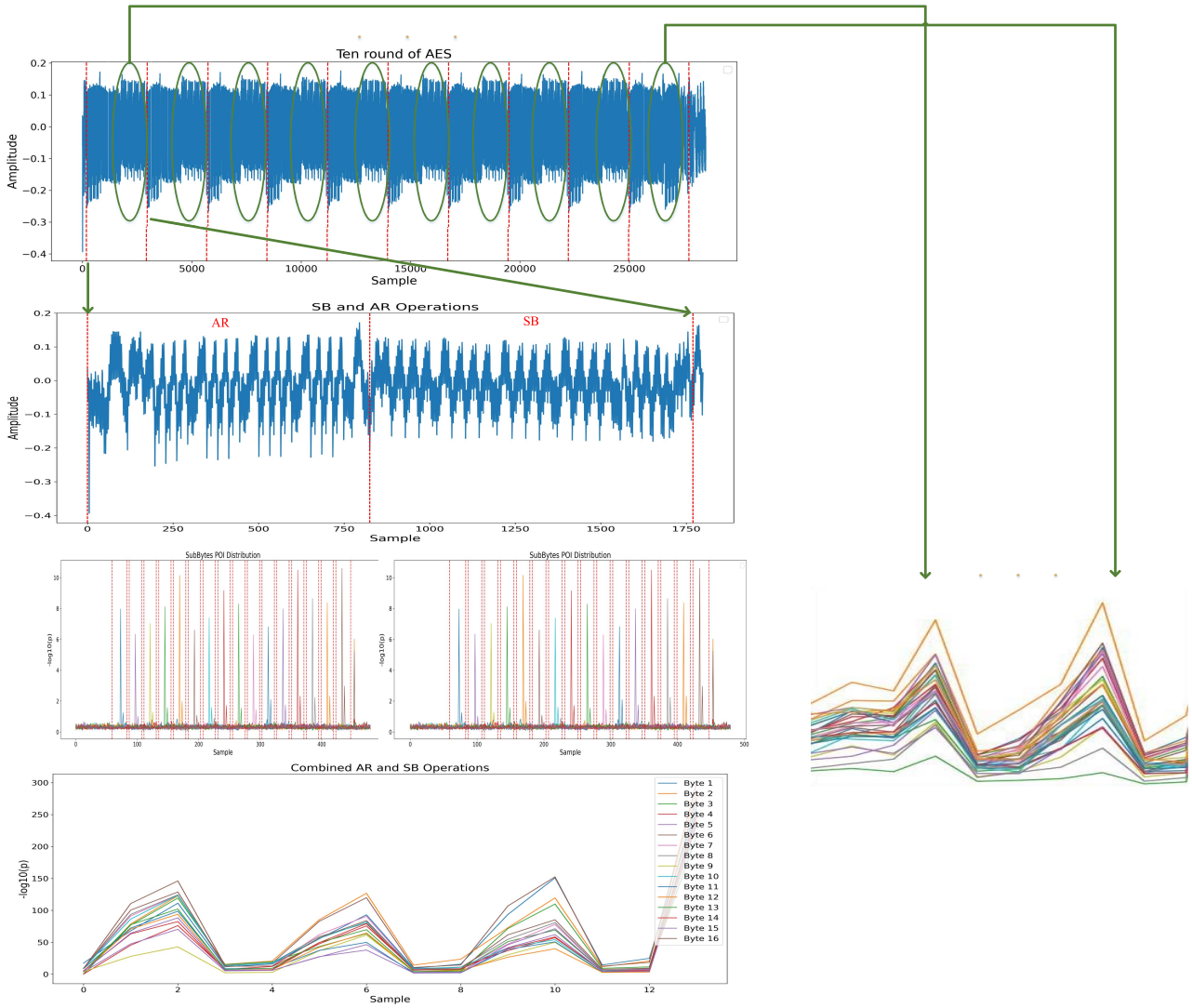


Fig. 5. First row shows the collected power traces and the red line separates the ten rounds of AES. The green circles highlight the SB and AR operations within each of the ten rounds of AES. The second row shows the details of the SB and AR operations in the first round, with red numbers representing each byte. The third row shows the distribution of the POI after performing the ρ-test on the AR and SB operations of the first round, with red numbers indicating the POIs for each byte. The fourth row shows the effect of combining the AR and SB operations of the first round. The right-hand figure shows the effect of combining the AR and SB operations for each of the ten rounds of AES.

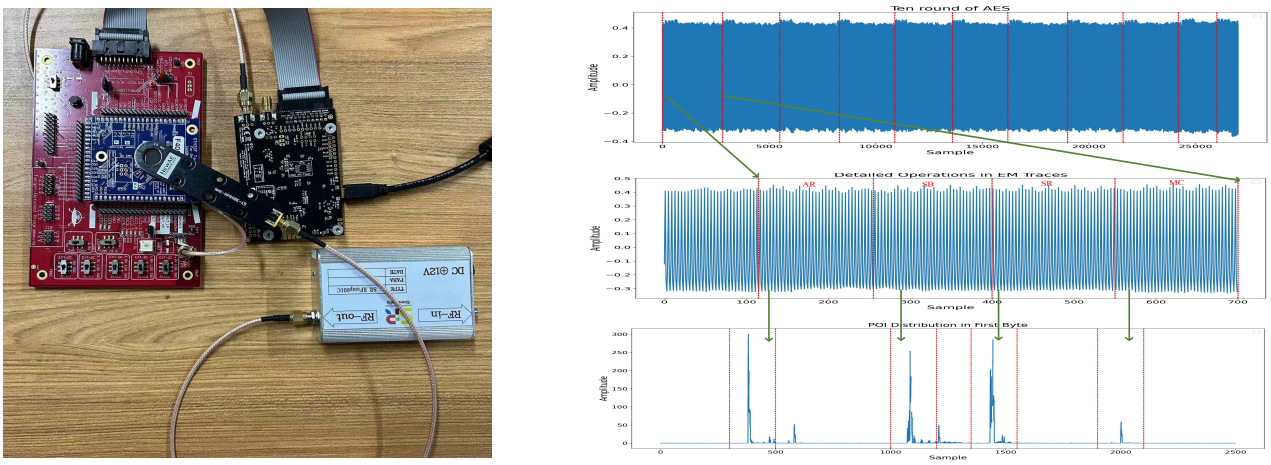
TABLE I
ATTACK ACCURACY RATES FOR SINGLE BYTES IN POWER TRACES

POI	First Byte	Forth Byte	Seventh Byte	Tenth Byte
AR	47%	7.8%	6.15%	6.75%
SB	50%	51.8%	50%	48%
SR	0.8%	0.7%	2.35%	2%
MC	0.85	0.8%	0.83%	0.86%
AR+SB	69%	62.5%	66%	64%

Through the analysis using the CRTLA, significant differences were found in the power attack accuracy rates for different operations (AR, SB, SR, MC, and AR + SB) for each byte. Table I shows the attack accuracy rates for the 1st, 4th, 7th, and 10th bytes under different operations. It can be seen that the combination of AR and SB operations yields the best results, with the highest-attack accuracy rates. Specifically, the attack accuracy rate for the 1st byte with the combined AR and SB operations is 69%, while for the 4th, 7th, and 10th bytes, the rates are 62.5%, 66%, and 64%, respectively. In contrast, the attack accuracy rates for the individual AR and SB operations are lower, and the rates for the SR and MC operations are even more negligible.

After confirming that the combination of AR and SB operations yields the best results, we used PPLA to identify the extreme points of energy leakage in each round of encryption. Initially, the PPLA is utilized to determine the index of the extreme point for each byte in the 21 rounds and store it in an index array. Subsequently, these indexes are verified to ensure that they accurately represent the actual extreme points. Fig. 5, third row, presents the results of the ρ-test analysis for 16 bytes from the first round of AES encryption. The x-axis represents the byte position, while the y-axis indicates the value of the ρ-test, reflecting the statistical significance of the leakage. The red dashed lines mark the leakage range

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 6. (a) Device utilized for collecting EM traces. (b) First row shows the collected EM traces and the red line splits the ten rounds of AES. The second row shows The details of the four operations in EM traces. The third row shows the POI for the four operations of the first byte in EM traces.

of each byte, and the peaks within each interval indicate the power leakage of the corresponding byte during the encryption process. According to the predictions of the PPLA, the correct leakage peaks should fall within these red intervals. The various colored peaks in the figure represent the energy leakage of each byte at specific moments. Fig. 5, forth rows, presents how the points of interest (POI) for the 16 bytes are aligned. This step involves aligning the POI for each byte so that their leakage intervals can be compared and analyzed on the same time axis. Fig. 5 (right) presents the alignment of 10 rounds of AES (each round containing two operations: 1) SB and 2) AR and includes the intervals for the 16 bytes.

Finally, the *get_labels* and *get_traces* functions were used to generate the corresponding dataset, which was then inputted into the CNN-best for training. By expanding the dataset to 1008 traces using only about three traces, we successfully recovered the key with a rank value of 0 and an accuracy of approximately 0.5%. This outcome not only validates the effectiveness of our joint training approach but also demonstrates that even with a limited number of traces, our method can efficiently achieve key recovery. However, it is important to note that using only three traces is typically insufficient to construct a neural network model, let alone recover keys. This emphasizes the efficiency of our method for SCA in scenarios with small sample sizes.

However, although three power traces are sufficient to support our key recovery method, we further investigated the impact of varying the number of traces on the key recovery performance. As shown by the experimental results, Fig. 7(b) shows that with the addition of one trace, the key can be reliably recovered, as the model can extract more useful information from the increased number of samples, improving the stability and accuracy of the recovery. Conversely, Fig. 7(c) shows that when the number of traces is reduced by one, the model's performance significantly degrades, and key recovery becomes impossible, indicating that the sample size is insufficient to support effective training.

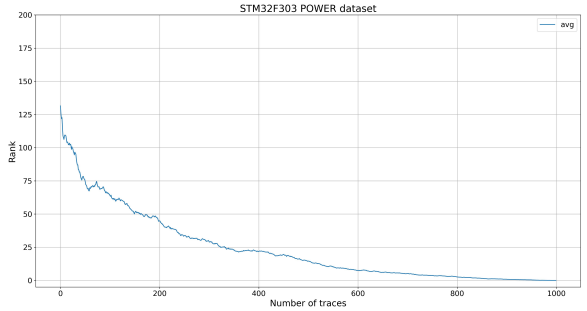
2) EM Traces via STM32F303: To obtain the second dataset, we used a Chipwhisperer device along with a CW505 planar H-Field probe. This setup was calibrated to a sampling frequency of 40MHz, specifically designed for recording the EM emissions of the chip. The data collection process was carried out using a CW308T-STM32F303 target board that housed an Arm Cortex M4 chip. This chip was programmed to execute the TinyAES-128 encryption algorithm in ECB mode. Fig. 6(a) visually represents the configuration of the capture device.

The dataset includes the entire AES-128 encryption process, with each of the 24 traces containing 30 000 sampling points. Fig. 6(b), first row, shows the EM traces from the 10 rounds of encryption. Fig. 6(b), second row, shows the details of four operations, AR, SB, SR and MC, where the EM-induced energy variations appear more subtle and intricate to discern. Each encryption round was carefully examined, with a particular focus on the SB and AR operations, except for round 0, which only involves the AR operation. In total, these operations consist of 21 different types of cryptographic manipulations. The STM32_EM handmade dataset was created by analyzing these EM traces, resulting in 8064 traces (24 traces per operation multiplied by 21 operations multiplied by 16 bytes). Out of these, 5000 traces were used for analysis, and 3000 for the attack phase.

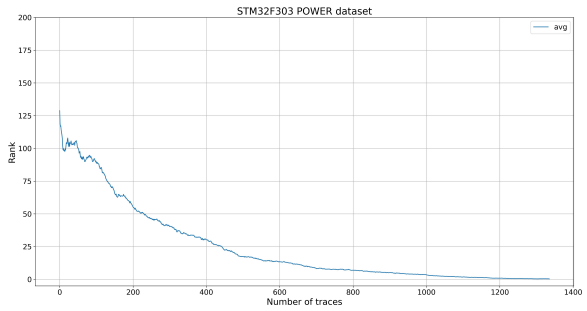
To acquire a thorough comprehension of the complexities of EM leakage, we developed and implemented two specially optimized algorithms known as the CRTLA and the PPLA. These algorithms enabled accurate identification of the points of EM leakage for each encryption round. Fig. 6(b), second row, demonstrates the EM leakage of the four operations during the initial round of AES encryption. By analyzing the subtle alterations in the waveforms, we can distinguish the features of each stage, such as SB and AR.

Through the analysis using the CRTLA, significant differences were found in the power attack accuracy rates for different operations (AR, SB, SR, MC, and AR+SB) for each byte. Table II shows the attack accuracy rates for the 1st, 4th,

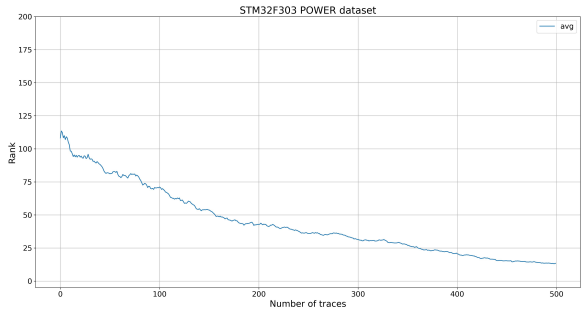
Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.



(a)

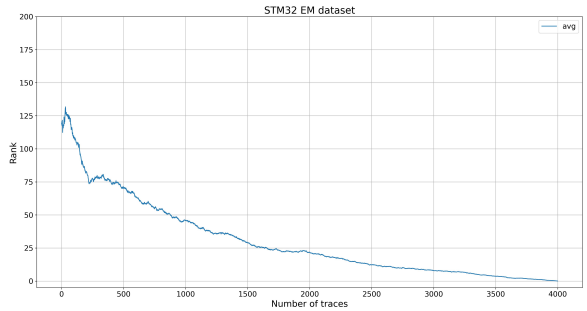


(b)

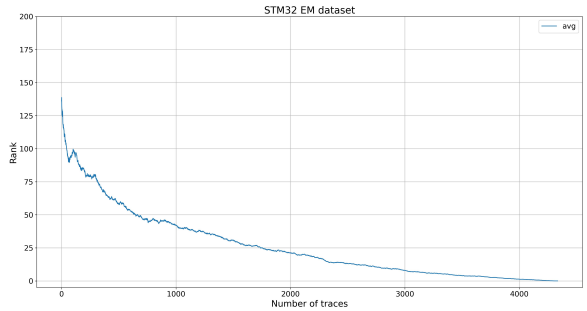


(c)

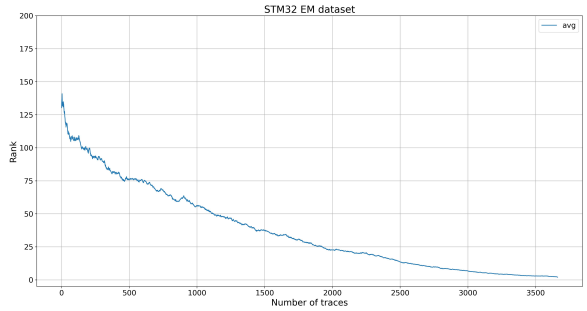
Fig. 7. (a) Rank for key recovery using three power traces. (b) Rank for key recovery using four power traces. (c) Shows the rank for unsuccessful key recovery using two power traces.



(a)



(b)



(c)

Fig. 8. (a) Shows the rank for key recovery using 15 EM traces. (b) Shows the rank for key recovery using 16 EM traces. (c) Shows the rank for unsuccessful key recovery using 14 EM traces.

TABLE II
ATTACK ACCURACY RATES FOR SINGLE BYTES IN EM TRACES

POI	First Byte	Forth Byte	Seventh Byte	Tenth Byte
AR	2.14%	7.8%	6.15%	6.75%
SB	3.08%	4.28%	3.36%	3%
SR	1.35%	1.20%	1.31%	1.28%
MC	1.14%	0.89%	1.11%	1.20%
AR+SB	4.35%	5.21%	3.89%	4.23%

7th, and 10th bytes under different operations. It can be seen that the combination of AR and SB operations yields the best results, with the highest-attack accuracy rates. Specifically, the attack accuracy rate for the 1st byte with the combined AR and SB operations is 69%, while for the 4th, 7th, and 10th bytes, the rates are 62.5%, 66%, and 64%, respectively. In contrast, the attack accuracy rates for the individual AR and SB operations are lower, and the rates for the SR and MC operations are even more negligible.

After confirming that the combination of AR and SB operations can achieve the best results, we used the PPLA to identify the extreme points of energy leakage in each round of encryption. After confirming that the combination of AR and SB operations can achieve the best results, we used the PPLA to identify the extreme points of energy leakage in each round of encryption. The procedure for EM traces is similar to Fig. 5. Initially, the ρ-test analysis is performed for the 16 bytes from the first round of AES encryption. Next, the POI for the 16 bytes are aligned. Finally, the AR and SB operation intervals for each of the ten rounds are aligned.

Using the labels and extreme point indexes generated from CRTLA and PPLA, we created the dataset using the *get_labels* and *get_traces* functions. These datasets were then inputted into an advanced convolutional neural network model for training. Through precise training and model optimization, we successfully retrieved the key using only approximately 3998 traces. Fig. 8(a) displays the rank of EM traces. The final experimental results, shown in the graphs below, highlight the efficiency of our EM–SCA approach and the strong generalization capability of our model. This demonstrates

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.

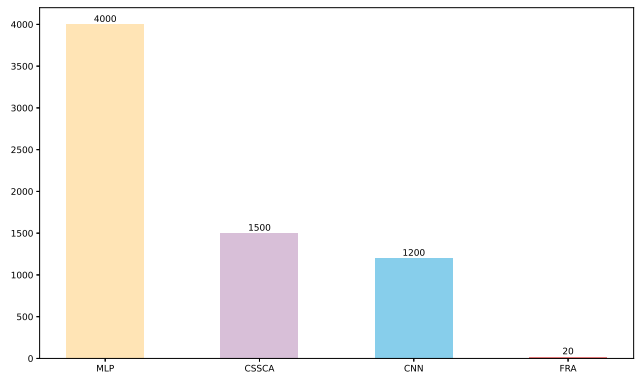


Fig. 9. Comparison results with other methods in terms of rank.

that even with a limited number of traces, our method can efficiently achieve key recovery.

Similarly, we conducted research on the impact of varying the number of EM traces on key recovery performance. The experimental results show that Fig. 8(b) demonstrates that with the addition of one EM trace, the key can be reliably recovered, as the model can extract useful information from the increased number of EM trace samples, improving the stability and accuracy of the recovery. Conversely, Fig. 8(c) shows that when the number of EM traces is reduced by one, the model's performance significantly degrades, and key recovery becomes impossible, indicating that the sample size of EM traces is insufficient to support effective training.

3) Comparison: The comprehensive AES analysis conducted in this study demonstrates a advantage in terms of trace utilization efficiency. With only 20 traces needed for modeling, this method can effectively recover the key within a range of 15 to 20 traces. In contrast, as shown in Fig. 9, other common analysis methods, such as cross-subkey SCA (CSSCA) [20], [30], [31], require a larger number of traces. CSSCA mandates up to 1500 traces for key recovery. Despite incorporating DL techniques to handle the intricacies of side-channel data, it still has the highest-trace requirement. For instance, the method described in [30] requires 4000 traces to successfully recover the key, highlighting its adaptability to complex side-channel environments but also indicating a high demand for traces. Although the method in [31] requires only 1200 traces, it still exceeds the trace count needed for full-round analysis.

The benefits of full-round analysis go beyond the considerable reduction in trace requirements. This substantial decrease not only reduces the cost of data acquisition but also speeds up the analysis process. Furthermore, by thoroughly covering every stage of the encryption process, this method provides a detailed analysis of leakage, greatly improving the accuracy of key recovery. Full-round analysis is not limited by specific hardware or leakage models, demonstrating strong adaptability and practicality, making it suitable for various hardware and environmental setups. The wide-ranging applicability and adaptability of full-round analysis make it a highly promising tool in the field of SCA, effectively addressing security challenges in complex scenarios.

TABLE III
NOTATIONS IN THIS ARTICLE

Symbol	Description
$SBOX$	The substitution box (S-box) used in AES encryption
ρ-test	A statistical test used to evaluate the correlation between side-channel traces and internal states
KEY	The main AES encryption key
kw	The round keys derived from the key expansion process
AR	AddRoundKey, an AES operation that XORs the state with a round key
SB	SubBytes, an AES operation that substitutes bytes using the S-box
$CRTLA$	Comprehensive Round Trace Labeling Algorithm
$PPLA$	Peak Precise Localization Algorithm
EM	Electromagnetic traces collected during side-channel analysis
GE	Guessing Entropy, a metric used to evaluate side-channel analysis
HW	Hamming Weight, a metric representing the number of 1-bits in a byte

V. CONCLUSION

In this article, a novel joint-SCA structure is proposed, with two algorithms, called CRTLA and PPLA, which greatly improve the depth and accuracy of SCA in AES encryption. These algorithms enable us to identify and analyze energy leakages throughout the AES encryption process more efficiently than traditional methods that mainly concentrate on the first round. CRTLA enables a thorough exploration of energy leakages across all encryption rounds, providing specific labels for targeted analysis. This broadens our understanding of vulnerabilities in AES operations and emphasizes the importance of analyzing subsequent rounds beyond the initial encryption. On the other hand, PPLA identifies key moments of high-energy leakage, which enables precise investigations into these vulnerabilities. Through thorough statistical tests and visualization techniques, we have developed a dependable method for effectively capturing and interpreting side-channel data. Going forward, our goal is to enhance the accuracy and efficiency of these algorithms. Future work will concentrate on adapting our methods to newer encryption technologies and exploring their applicability in other cryptographic contexts. This will further enhance cryptographic security against evolving threats and contribute to the advancement of cybersecurity.

APPENDIX A
NOTATIONS AND SYMBOLS

Table III presents the key notations and symbols used throughout this article to help readers better understand the technical terms and their respective roles in the proposed algorithms and analysis methods.

REFERENCES

[1] P. C. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in *Proc. 16th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, 1996, pp. 104–113.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. 19th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, 1999, pp. 388–397.

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.

[3] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Proc. 3rd Int. Workshop Cryptogr. Hardw. Embedd. Syst. (CHES)*, Paris, France, 2001, pp. 251–261.
[4] J. Daemen and V. Rijmen, *The Design of Rijndael*, vol. 2. Berlin, Germany: Springer, 2002.
[5] C. Clavier et al., "Practical improvements of side-channel attacks on AES: Feedback from the 2nd DPA contest," *J. Cryptogr. Eng.*, vol. 4, pp. 259–274, Nov. 2014.
[6] S. Mangard, N. Pramstaller, and E. Oswald, "Successfully attacking masked AES hardware implementations," in *Proc. Int. Workshop Cryptogr. Hardw. Embed. Syst.*, 2005, pp. 157–171.
[7] S. B. Ors, F. Gurkaynak, E. Oswald, and B. Preneel, "Power-analysis attack on an ASIC AES implementation," in *Proc. Int. Conf. Inf. Technol., Coding Comput. (ITCC)*, 2004, pp. 546–552.
[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[9] M. Ren, H. Q. Vu, G. Li, and R. Law, "Large-scale comparative analyses of hotel photo content posted by managers and customers to review platforms based on deep learning: Implications for hospitality marketers," *J. Hosp. Mark. Manag.*, vol. 30, no. 1, pp. 96–119, 2021.
[10] S. Bonthu, A. Dayal, M. S. Lakshmi, and S. Rama Sree, "Effective text augmentation strategy for NLP models," in *Proc. 3rd Int. Conf. Sustain. Comput., SUSCOM*, 2022, pp. 521–531.
[11] Y. Ou and L. Li, "Side-channel analysis attacks based on deep learning network," *Front. Comput. Sci.*, vol. 16, no. 2, 2022, Art. no. 162303.
[12] Y.-S. Won, D.-G. Han, D. Jap, S. Bhasin, and J.-Y. Park, "Non-profiled side-channel attack based on deep learning using picture trace," *IEEE Access*, vol. 9, pp. 22480–22492, 2021.
[13] G. Zaid, L. Bossuet, F. Dassance, A. Habrard, and A. Venelli, "Ranking loss: Maximizing the success rate in deep learning side-channel analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 1, pp. 25–55, 2021.
[14] Z. Martinasek and V. Zeman, "Innovative method of the power analysis," *Radioengineering*, vol. 22, no. 2, pp. 586–594, 2013.
[15] L. Wu, G. Perin, and S. Picek, "I choose you: Automated hyperparameter tuning for deep learning-based side-channel analysis," *IEEE Trans. Emerg. Topics Comput.*, vol. 12, no. 2, pp. 546–557, Apr.–Jun. 2024.
[16] H. Wang, M. Brisfors, S. Forsmark, and E. Dubrova, "How diversity affects deep-learning side-channel attacks," in *Proc. IEEE Nordic Circuits Syst. Conf. (NORCAS), NORCHIP Int. Symp. Syst. Chip (SoC)*, 2019, pp. 1–7.
[17] A. Sayakkara, N.-A. Le-Khac, and M. Scanlon, "Leveraging electromagnetic side-channel analysis for the investigation of IoT devices," *Digit. Investigat.*, vol. 29, pp. S94–S103, Jul. 2019.
[18] F. Hu, J. Shen, and P. Vijayakumar, "Side-channel attacks based on multi-loss regularized denoising autoencoder," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 2051–2065, 2024.
[19] S. Fu et al., "Multi-byte power analysis: A generic approach based on linear regression," *IEEE Access*, vol. 6, pp. 67511–67518, 2018.
[20] F. Hu, H. Wang, and J. Wang, "Cross subkey side channel analysis based on small samples," *Sci. Rep.*, vol. 12, no. 1, p. 6254, 2022.
[21] F. Kong, X. Wang, K. Pu, J. Zhang, and H. Dang, "A practical non-profiled deep-learning-based power analysis with hybrid-supervised neural networks," *Electronics*, vol. 12, no. 15, p. 3361, 2023.
[22] N. Cai, Z. Yang, S. Wang, Y. Jiang, M. Liu, and G. Li, "Efficient side-channel attack through balanced labels compression and variational autoencoder," *EasyChair*, 2024, to be published.
[23] M. Huang, F. Gong, G. Li, T. R. Gadekallu, and N. Victor, "Secure transmission via precoding for satellite-terrestrial downlink NOMA networks," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, 2023, pp. 2172–2177.
[24] Z. Muhammad, Z. Anwar, A. R. Javed, B. Saleem, S. Abbas, and T. R. Gadekallu, "Smartphone security and privacy: A survey on APTs, sensor-based attacks, side-channel attacks, Google play attacks, and defenses," *Technologies*, vol. 11, no. 3, p. 76, 2023.
[25] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Proc. 6th Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng. (SPACE)*, 2016, pp. 3–26.
[26] L. Zhang, X. Xing, J. Fan, Z. Wang, and S. Wang, "Multi-label deep learning-based side channel attack," in *Proc. Asian Hardw. Orient. Security Trust Symp. (AsianHOST)*, Piscataway, NJ, USA, 2019, pp. 16–17.
[27] F. Hu, H. Wang, and J. Wang, "Multi-leak deep-learning side-channel analysis," *IEEE Access*, vol. 10, pp. 22610–22621, 2022.
[28] F. Durvaux and F.-X. Standaert, "From improved leakage detection to the detection of points of interests in leakage traces," in *Proc. 35th Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2016, pp. 240–262.
[29] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, "Deep learning for side-channel analysis and introduction to ASCAD database," *J. Cryptogr. Eng.*, vol. 10, no. 2, pp. 163–188, 2020.
[30] P. Arpaia, F. Caputo, A. Cioffi, A. Esposito, and F. Isgrò, "Uncertainty analysis in cryptographic key recovery for machine learning-based power measurements attacks," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–8, Jun. 2023.
[31] I. Negabi, S. Ait El Asri, S. El Adib, and N. Raissouni, "Deep learning-based power analysis attack for extracting AES keys on ATmega328P microcontroller," *Arabian J. Sci. Eng.*, vol. 49, no. 3, pp. 4197–4208, 2024.

Zhiyuan Xiao received the B.S. degree from the Gannan University of Science and Technology, Ganzhou, China, in 2021. He is currently pursuing the master's degree with Zhejiang Sci-tech University, Hangzhou, China.

His research interests include hardware encryption and side channel analysis.

Chen Wang (Member, IEEE) received the Ph.D. degree from the Nanjing University of Information Science and Technology, Nanjing, China, in 2022.

He was a Foreign Research Fellow with the University of Aizu, Fukushima, Japan. He is an Associate Professor with the School of Information Science and Engineering (School of Cyber Science and Technology), Zhejiang Sci-tech University, Hangzhou, China. His research interests include secure data aggregation, edge computing security, and AI-enabled security.

Jian Shen (Member, IEEE) received the M.E. and Ph.D. degrees in computer science from Chosun University, Gwangju, South Korea, in 2009 and 2012, respectively.

From 2012 to 2022, he was a Professor with the School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, China. He is currently working as a Professor with the School of Information Science and Engineering (School of Cyber Science and Technology), Zhejiang Sci-tech University, Hangzhou, China. His research interests include public cryptography, cloud computing and security, data auditing and sharing, and information security systems.

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.

Q. M. Jonathan Wu (Senior Member, IEEE) received the Ph.D. degree in computer vision and intelligent systems from the University of Wales, Cardiff, U.K, in 1990.

He is a Distinguished Professor of Electrical and Computer Engineering and has been a Tier 1 Canada Research Chair of Automotive Sensors and Information Systems since 2005. He is the Founding Director of the Computer Vision and Sensing Systems Laboratory, University of Windsor, Windsor, ON, Canada. Prior to joining the university, he was a Senior Research Officer with the National Research Council of Canada. He has published one book in the area of 3-D computer vision and more than 350 peer-reviewed papers, including 200 journal articles, in the areas of computer vision, machine learning, and intelligent systems.

Dr. Wu is/was an Associate Editor for IEEE TRANSACTIONS ON CYBENECTICS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He is an Elected Fellow of the Canadian Academy of Engineering.

Debiao He (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2009.

He is currently a Professor of the School of Cyber Science and Engineering, Wuhan University. He has authored or co-authored more than 100 research papers in refereed international journals and conferences, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and Usenix Security Symposium. His work has been cited more than 10 000 times at Google Scholar. His main research interests include cryptography and information security, in particular, cryptographic protocols.

Prof. He was the recipient of the 2018 IEEE Sysems Journal Best Paper Award and the 2019 IET Information Security Best Paper Award. He is in the editorial board of several international journals, such as ACM Distributed Ledger Technologies: Research and Practice, FRONTIERS OF COMPUTER SCIENCE, and IEEE TRANSACTIONS ON COMPUTERS.

Authorized licensed use limited to: Patuakhali Science and Technology Univ. Downloaded on January 18,2026 at 17:00:06 UTC from IEEE Xplore. Restrictions apply.