IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 25, 2022, accepted May 30, 2022, date of publication June 8, 2022, date of current version June 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3180742

Single-Trace Attack Using One-Shot Learning With Siamese Network in Non-Profiled Setting

NAYEON LEE[1], SEOKHIE HONG[1], (Member, IEEE), AND HEESEOK KIM[2], (Member, IEEE)
[1]Institute of Cyber Security and Privacy (ICSP), Korea University, Seoul 02841, Republic of Korea
[2]Department of AI Cyber Security, College of Science and Technology, Korea University, Sejong 30019, Republic of Korea

Corresponding author: Heeseok Kim (80khs@korea.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grant NRF-2019R1A2C2088960.

ABSTRACT Recently, many studies have shown that using deep learning for side-channel attacks offers several advantages, including simplification of the attack phase and target breaking, even in protected implementations, while presenting outstanding attack performance. Power and electromagnetic analysis, which is known as the most robust attack, can be classified into profiling and non-profiling attacks. In the real world, a non-profiling attack is more ideal than a profiling attack. In particular, studies on non-profiling attacks using deep learning for asymmetric cryptosystems are rare and have shortcomings, such as a long analysis time. In this paper, we propose a novel non-profiling attack method for asymmetric cryptosystems that requires only a single trace and a reasonably short attack time to recover a full private key, overcoming the limitations of previous studies. The proposed method applies one-shot learning with a convolutional Siamese network, which is used for the first time in side-channel attacks. Thus, our proposed method can leak private keys used in a protected public-key cryptosystem with up to 100% accuracy with only one single trace in a non-profiled setting.

INDEX TERMS Deep learning, ECC, Montgomery ladder, non-profiling attack, one-shot learning, side channel attack, Siamese network, similarity score.

I. INTRODUCTION

A side-channel analysis is an attack technique that uses additional information about a computer system usage, such as timing information, power consumption [3], [15], and electromagnetic emissions [1], to obtain the secret information inside the computer system. Power and electromagnetic analysis, which is known as the most robust side-channel attack, is classified into profiling [1], [21] and non-profiling attacks [3], [9], [15]. In a profiling attack, a profile is created using devices controlled by an attacker. Subsequently, the attacker matches the profile with the measurements from a device of the victim, to reveal secret information. However, in a non-profiling attack, an attacker can only access a closed target device and collect several side-channel traces. Subsequently, the attacker uses statistical knowledge, such as Pearson's correlation, to obtain secret data. Profiling attacks are more robust than non-profiling attacks; however, latter are more practical because performing profiling attacks requires access to a profiling device, a demand that is not typically satisfied.

To protect against non-profiling attacks, many countermeasures are being studied and installed in embedded devices. For public-key cryptosystems, regular scalar multiplications, such as the Montgomery ladder [13], double-and-add, and atomicity-based implementations [10], and randomization techniques [18] can be considered. To conduct such non-profiling attacks, an attacker requires considerable analysis time and sufficient expertise for the used algorithms. For example, to conduct collision-based attacks (e.g., [4]), an attacker must analyze the details of the algorithms, including where a collision can occur. Moreover, finding points where a collision occurs in traces is quite demanding and takes a long time.

To overcome such difficulties for general side-channel attacks and combine the advantages of deep learning, deep-learning-based side-channel attacks are actively being studied. Many studies have shown that using deep learning for side-channel attacks offers several advantages, including simplification of the attack phase and breaking of the target, even

The associate editor coordinating the review of this manuscript and approving it for publication was Janmenjoy Nayak.

in protected implementations, while presenting outstanding attack performance. However, most deep learning-based side-channel attacks are profiling attacks [5], [6], [17].

Deep learning-based profiling attacks are being actively studied for enhancing the attack performance on asymmetric cryptography. However, from the perspective of deep learning-based non-profiling attacks on asymmetric cryptography, thus far, [19] is the only study to our best knowledge. In their study, a Montgomery ladder, in which countermeasures were adopted, was attacked with their proposed framework based on a convolutional neural network (CNN). The framework used 63,750 data as the training data, and a combination of incorrect and correct labels as the training labels (i.e., noisy bits). Their proposed iterative framework repeats the training, prediction, re-labeling, and shuffling training dataset phases. While repeating these phases, the framework gradually corrects the incorrect labels. Following many iterations, their proposed attack improves the scalar bit recovery accuracy by up to 100% from noisy bits with 52% accuracy. Briefly, they attacked protected implementations and achieved successful results in a Non-profiled setting. However, the proposed framework requires considerably long training time and a large training dataset. Moreover, the performance of the proposed attack framework depends on the presence of characteristic information that can distinguish between two groups of initial training labels. For example, if the initial training labels have numerous incorrect bits, the framework cannot correct them because it cannot train the characteristics that can classify the two groups (bits "0" or "1").

As mentioned earlier, most related studies using deep learning focus on profiled side-channel attacks and symmetric cryptosystems. Moreover, the only reported framework for deep learning-based non-profiling attacks on asymmetric cryptosystems needs a long training time and numerous traces. Therefore, in this paper, we propose a new method for non-profiling attacks on asymmetric cryptosystems that requires only a single trace and a reasonably short training time, overcoming the limitations of existing attacks.

The main contributions of this paper are as follows:

- The proposed method applies one-shot learning with a convolutional Siamese network, leading to high classification accuracy even with one training data point for each class. Therefore, the method overcomes the limitations of existing methods, which need many single traces, because it can also be used even if an attacker has only one single trace.[1]
- As the method trains the network for a few epochs and uses only one data (i.e., one subtrace) per class as the training data, it takes a shorter time than previous studies using a CNN to recover fully private key bits, even considering failure.[2] For example, one of our experiments requires < 2 min. Nevertheless, our method shows a strong performance of up to 100% on protected implementations.
- The intermediate outputs during the framework process enable checking the effectiveness of the attacks. For example, one of the outputs refers to the points of the subtraces that can distinguish between the two groups (bits "0" and "1"). If an attacker overlays these points of the subtraces and the output of the Siamese network or the results of the proposed attacks, visual confirmation can be obtained regarding the learning of valid features by the network[3] or success of the attack.[4] Moreover, the attack can easily be extended to other single traces because an attacker can omit the training network phase when reusing these points.
- The performance of the proposed method does not depend on any initial value, such as the initial labels [19]. A previous study that used a large training dataset required initial labels to train the network. Therefore, in their method, to achieve good performance, the initial labels must not be biased to one class, and simultaneously, they must contain some correct labels that can provide information to classify the two classes. In contrast, the proposed method only requires the assumption that the two randomly selected subtraces belong to different classes, and the possibility of meeting this assumption is 50%.
- Since the proposed method simply applicable as long as the attack target is the algorithms that computes through examining each individual bit of the scalar(exponent in the case of RSA), the proposed attack framework is generic.

Structure of paper:

The remainder of this paper is organized as follows. Section 2 summarizes the background information regarding the aim of the study. Subsequently, it describes side-channel analysis with deep learning and one-shot learning with Siamese neural networks, including related studies. Section 3 details our proposed attack framework, and Section 4 presents the attack results on three datasets. Finally, in section 5, we draw the conclusions and propose future studies.

II. BACKGROUND

This section provides the relevant background knowledge regarding the objective of this study, deep learning-based side-channel attacks, one-shot learning, and Siamese neural networks.

[1] A single trace is split into subtraces that represent the processing of single private key bits "0" and "1," and the proposed method uses two subtraces for the training data.

[2] Because the proposed method works in a Non-profiled setting, we need to randomly select two subtraces from different classes for one-shot learning. In this phase, there is a possibility that subtraces belonging to the same class are chosen, which could lead to an attack failure.

[3] This confirms whether the two chosen subtraces are in the same class.

[4] If the attack is successful, the two groups are notably distinguished visually.

A. MONTGOMERY LADDER

Algorithm 1 Montgomery Ladder

Input: P (point on elliptic curve) and n-bit scalar $k = (k_{n-1}, k_{n-2}, \cdots, k_0)_2$
Output: $Q = kP$
1: $R_0 \leftarrow P$; $R_1 \leftarrow 2P$;
2: **for** $idx \leftarrow n-2$ to 0 **do**
3: $\quad R_{\neg k_{idx}} \leftarrow R_0 + R_1$
4: $\quad R_{k_{idx}} \leftarrow \mathbf{2}R_{k_{idx}}$
5: **end for**
6: **return** R_0

A Montgomery powering ladder [13] is an algorithmic-level countermeasure that is implemented for a fixed time. Because a Montgomery ladder is implemented regularly and there is no redundant operation, Algorithm 1 resists ordinary simple power analysis (SPA) [15] and C safe-error analysis [24]. However, it is still vulnerable to M safe-error attacks. Therefore, Joye and Yen [13] proposed a method to prevent an M safe-error attack (Algorithm 2) by simply changing $R_0 + R_1$ to $R_{k_{idx}} + R_{\neg k_{idx}}$, i.e., line 3 in Algorithms 1 and 2.

Algorithm 2 Modified Montgomery Ladder

Input: P (point on elliptic curve) and n-bit scalar $k = (k_{n-1}, k_{n-2}, \cdots, k_0)_2$
Output: $Q = kP$
1: $R_0 \leftarrow \mathbf{0}$; $R_1 \leftarrow P$;
2: **for** $idx \leftarrow n-1$ to 0 **do**
3: $\quad R_{\neg k_{idx}} \leftarrow R_{k_{idx}} + R_{\neg k_{idx}}$
4: $\quad R_{k_{idx}} \leftarrow \mathbf{2}R_{k_{idx}}$
5: **end for**
6: **return** R_0

In the implementation of Algorithm 3, the conditional branch is replaced by the cswap function to achieve constant time. Algorithm 3 computes on coordinates P_1 and P_2, which are represented as *work_state*. The algorithm is already protected from side-channel attacks, owing to regular and constant-time implementation. However, a blinding countermeasure is added at the beginning of every iteration to blind coordinates P_1 and P_2 to random values for more security (i.e., Re_Randomize_Coordinates in Algorithm 3). For more details on Algorithm 3, we recommend referring to [19].

This study targets various types of Montgomery ladders, as described in Algorithms 2 and 3.

Algorithm 3 Montgomery Ladder With Cswap and Coordinate Re-randomization [19]

1: // . . . initialization omitted . . .
2: $bprev = 0$
3: **for** $i \leftarrow 254$ to 0 **do**
4: $\quad$ Re_Randomize_Coordinates(*work_state*);
5: $\quad b \leftarrow$ bit i of scalar;
6: $\quad s \leftarrow b \oplus bprev$;
7: $\quad bprev \leftarrow b$;
8: $\quad$ cswap(*work_state*, s);
9: $\quad$ Ladderstep(*work_state*);
10: **end for**
11: . . . return omitted . . .

B. SIDE-CHANNEL ANALYSIS WITH DEEP LEARNING

A side-channel attack is a type of attack that can leak secret information used for encryption using the power consumption and electromagnetic emissions generated by hardware during encryption. Side-channel attacks can be classified into profiling (e.g., template attacks [7]) and non-profiling (e.g., SPA, differential power analysis (DPA) [15], and correlation power analysis (CPA) [3]) attacks. In a profiling attack, a profile is created by operations on sensitive data in an environment where an attacker can control the device. Subsequently, the attacker matches the profile with the measurements from the device of a victim to obtain secret information. In contrast, in a non-profiling attack, an attacker can access only a closed target device. Because a profiling attack is under a stronger assumption than a non-profiling attack, the former requires significantly fewer traces than the latter. If side-channel attacks leak secret information, they can be used maliciously in several ways. To deal with this problem, many countermeasures such as a Montgomery powering ladder [13] and blinding [18] are being studied and installed in embedded devices to protect against side-channel attacks. Blinding techniques can incapacitate attacks including DPA and CPA [3]. Furthermore, a Montgomery powering ladder is immune to SPA and fault attacks [13], [24], whereas, profiling attacks can occur. Because in profiling attacks, the profiles do not typically perfectly match the measurements from the device of the victim, non-profiling attacks are more suitable in a practical environment.

Deep learning is a commonly used technology in many academic and industrial fields such as pattern recognition, autonomous driving, and prediction of various phenomena. Recently, deep learning is being used for various objectives in side-channel analysis, including profiling [6], [11], [17] and non-profiling [19], [23] side-channel attacks. There are many advantages to using deep learning in side-channel attacks:

- It is very powerful and can break targets even against protected implementations [6], [19], [23].
- It can omit essential phases of a side-channel such as preprocessing of traces, alignment, and points of interest (PoIs) selection.
- It is also possible to use information that is not employed in existing side-channel analysis.

Most side-channel attacks based on deep learning are profiling attacks. In contrast, few studies have been conducted on non-profiling attacks using deep learning. The first application of non-profiling attacks using deep learning was proposed by Timon [23]. However, this application attacks symmetric cryptography. To our best knowledge, thus

far, [19] is the only study in the field of non-profiling attacks using deep learning on asymmetric cryptography. The authors attacked a Montgomery ladder, in which countermeasures were applied, by their proposed framework based on a CNN. They used 63,750 (255 × 250) traces as training data and improved the accuracy to an average of 90% and up to 100% in the results of clustering-based horizontal attacks with 52% accuracy using the proposed method. However, this framework needs a considerably long time and numerous traces. Moreover, the input labels used for the training networks, i.e., the results of clustering-based horizontal attacks, must have some characteristic information to classify the traces into two classes. Summarizing, the performance of the proposed attack framework depends on the presence of characteristic information in the training labels.

Presumably, the method proposed in this paper is secondary to target public-key encryption in non-profiling attacks using deep learning. The proposed method does not depend on any initial value, and it takes a very short time to recover the entire private key, and simultaneously, it shows a powerful performance of up to 100% on protected implementations.

C. ONE-SHOT LEARNING WITH SIAMESE NEURAL NETWORKS

In general, the features of each data are learned to achieve high performance in deep learning. Learning good features with machine learning can be computationally expensive, and it can be challenging to achieve good performance if there is little available data. In this case, one-shot learning with convolutional Siamese networks can be effective.

One-shot learning is a method for training networks with one data for each class. It can be used when a small dataset is available and a severe overfitting problem needs to be avoided. In [8], by one-shot learning on several objects future images could be predicted using a trained model with a small data sample. The central concept of Siamese networks is supervised learning by sharing the same network. For image classification, a Siamese network is designed to use two image datasets as input to the shared network and output probability values of whether two images are in the same class. In [2], for the first time, Siamese networks were used for signature verification. Using the same network to share two input data, the study extracted feature vectors and obtained the similarity of the two input data as output. Recently, one-shot learning with a convolutional Siamese network has been commonly used to classify various targets. In [12], the authors used malware images as targets and conducted two-way one-shot learning on the dataset. In addition, in [16], the authors conducted one-shot classification on the Omniglot dataset using Siamese neural networks. They proposed a method for one-shot classification by learning deep convolutional Siamese neural networks and showed higher performance than the general Siamese neural networks.

Most deep learning-based side-channel attacks use a multilayer perceptron (MLP) and a convolutional neural network(CNN) [20]. However, a general MLP and CNN require many data samples to train deep learning networks and high computing power to obtain good results. Side-channel attacks, as mentioned in Section 2.B, also require considerable training data and long training times to achieve good performance. In this study, for the first time, we performed one-shot learning using a convolutional Siamese network to recover secret information in a side-channel attack. As we used only two datasets to train for a small number of epochs, a fully scalar bit was recovered in a very short time using the proposed method compared with previous studies using a CNN and a MLP in the side-channel field. Nevertheless, it showed a strong performance of up to 100% on protected implementations.

III. PROPOSED ATTACK FRAMEWORK BASED ON SIAMESE DEEP NEURAL NETWORK

This section proposes a Siamese deep neural network-based attack framework that can recover secret information. The proposed method retains no knowledge of the secret information (i.e., k and s). A single attack framework implements the following six steps, and an attacker can perform additional training steps on PoI subtraces if needed:

1. **Two subtraces $D0$ and $D1$ are randomly chosen**. We do not have any information on secret bits. However, each subtrace represents the processing of a single bit "1" or "0." Therefore, we can select two subtraces that are in different classes with a 50% probability.
2. **Training is performed on $D0$ and $D1$, and the dissimilarity scores between $D0$ and the remainder subtraces are predicted (Figure 1)**. The Siamese deep neural network is trained with a pair of $D0$ and $D1$, with label 1, which represents the dissimilarity score. With the trained Siamese deep neural network, the dissimilarity scores between $D0$ and the other data are predicted. If the network is well-trained, the dissimilarity scores are not biased to one value. If the dissimilarity scores are one-sided, then step III is repeated.
3. **The mean value of the dissimilarity scores is calculated and classified into two classes**. The mean value of all the dissimilarity scores is the distinguisher in this step. If one subtrace dissimilarity score exceeds the distinguisher, it is labeled "1" (i.e., having a different class than $D0$).
4. **Leakage assessment is performed on the datasets using step 3 results**. Leakage assessment is performed using the t-test. The t-test is used to determine whether the difference in the means of the two groups is statistically significant. Therefore, the t-test helps determine whether a dataset is well-classified and what is the difference between the two classified groups. If the result of the t-test does not have any peaks, it implies that the two groups are not classified reasonably. In this case, it is ambiguous if the two selected subtraces belong to the same class. Hence, step 1 is repeated or re-training of the Siamese network is considered.

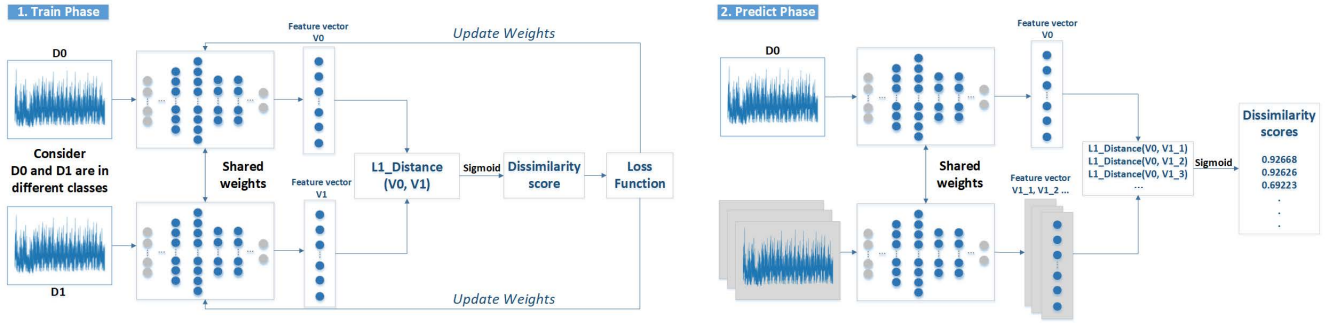

FIGURE 1. Train on D0 and D1 pair and predict dissimilarity scores (step III).

5. **PoIs are extracted from the leakage assessment and PoIs subtraces are generated**. The peaks of the t-test results are points of interest(PoIs). Using these PoIs, we generate PoIs subtraces by extracting values corresponding to the PoIs from the subtraces.
6. **The PoIs subtraces are classified into two classes using a distinguisher**. In this step, we categorize the PoI subtraces into two classes based on the whether the average value of a PoIs subtrace is less than or more than 0.

IV. EXPERIMENTAL RESULTS

This section presents several results of the proposed attack framework on three datasets. Three datasets were used in this study.

TABLE 1. Secp192r1 parameters; p, n, and b are provided in decimal, basepoint, and hexadecimal forms. [14].

	Parameters
Prime p	6277101735386680763835789423207666416083908700390324961279
Order n	6277101735386680763835789423176059013767194773182842284081
Coefficient b	64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1
Basepoint G_x	188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012
Basepoint G_y	07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811

Dataset 1:

Dataset 1 was collected by applying Algorithm 2. In this study, the experiment on Dataset 1 used the secp192r1 parameters recommended by the National Institute of Standards and Technology (NIST). The elliptic curve is defined by the equation, $E : y^2 = x^3 – 3x + b \bmod p$, with the parameters listed in Table 1. For more details, please refer to the NIST document in [14]. The target software was operated on a Cortex-M4-based STM32F4 microcontroller chip on a board. We acquired the power consumption trace of Algorithm 2 using a LeCroy HDO6104A oscilloscope with a sampling rate of 250 MS/s operating at 5 MHz. After the acquisition, we applied a 5 MHz low-pass filter to every trace. The dataset is a single trace and can be divided into 191 subtraces. Each subtrace frames the processing of one scalar bit, and we aimed to recover the scalar bits, k, in Algorithm 2.

Datasets 2 and 3:

Datasets 2 and 3 were obtained by the implementation of Algorithm 3.[5] The traces were acquired from the implementation that added coordinate re-randomization to the μNaCl library for ARM Cortex-M.[6] Also, the traces were collected with a random scalar for each execution of the algorithm. Both datasets consist of 300 single traces each, and can be divided into $255 \times 300 = 76{,}500$ subtraces. Datasets 2 and 3 differ depending on how the cswap operation was performed. For Dataset 2, the operation was by pointer swapping (cswap-pointer), and for Dataset 3, the operation is by arithmetic means swapping (cswap-arith). The bit, s, in Algorithm 3 is the target of recovery in this experiment. Details of the dataset are presented in [19].

First, we preprocessed all datasets by z-score normalization. Z-score normalization is a technique that averages all datasets to zero and the standard deviation to one. Using it, a dataset can be standardized, which can result in better performance in deep learning. The attack framework starts by selecting two subtraces for all three datasets. Following this, the Siamese neural network is trained on the datasets and the dissimilarity scores are obtained. The distribution of the dissimilarity scores is represented by a scatter plot and divided into two different classes. Subsequently, a leakage assessment was conducted on the datasets with the results of the previous step obtained using the t-test. Using the t-test peak, we extracted PoIs and created PoI subtraces. Finally, we applied a distinguisher to divide the PoI subtraces into two classes. The proposed attack framework is generic. Thus, we applied different variations of the attack framework, e.g., additional training steps on the PoI subtraces and adaptation of regularization techniques.

A. ATTACKING DATASET 1

This section provides the specific results of the application of the attack framework on Dataset 1 along with the details of the experiments. This dataset is based on implementation of Algorithm 2. We attacked 20 single traces,

[5]These datasets can be downloaded from https://github.com/AISyLab/IterativeDLFramework.

[6]http://munacl.cryptojedi.org/curve25519-cortexm0.shtml

and all 20 had 100% scalar bit recovery accuracy. We used two subtraces when training the Siamese network. Thus, we recovered the scalar bits, k, against a single trace. The attack can also be extended to other single traces. Using the trained Siamese network, we recovered the scalar bits, k, against 20×191 subtraces. Consequently, we achieved 100% scalar bit recovery accuracy for all 20 single traces with one attack. Summarizing, with two subtraces, we recovered all secret values, i.e., k. We defined the CNN of the Siamese network with three convolution layers and two fully connected layers. The details of the CNN and convolutional Siamese neural network architectures are presented in Tables 2 and 3, respectively. We used a learning rate of 0.001 and the *Adam* optimizer. On this dataset, regularization was not required to achieve 100% scalar bit recovery accuracy on all 20 single traces. *Again, note that only 2 subtraces of a single trace were used in training, and of the remainder 19 single traces were excluded.*

TABLE 2. CNN architecture in Siamese network considered for Dataset 1.

Layer	Dataset 1
Input	input_size = 10293
Conv1D_1	filters = 8, kernel_size = 40, stride = 4, activation = $ReLu$
Conv1D_2	filters = 16, kernel_size = 40, stride = 4, activation = $ReLu$
Conv1D_3	filters = 32, kernel_size = 40, stride = 4, activation = $ReLu$
Dense_1	100 neurons
Dense_2	100 neurons
$Sigmoid$	2 neurons

TABLE 3. Siamese neural network architectures considered for Datasets 1–3.

Layer	Dataset 1	Dataset 2	Dataset 3
Input_1	input_size = 10293	input_size = 1000	input_size = 8000
Input_2	input_size = 10293	input_size = 1000	input_size = 8000
CNN	Table 2	Table 4	Table 5
L1_layer	L_1 distance	L_1 distance	L_1 distance
$Sigmoid$	1 neuron	1 neuron	1 neuron

The details of the experiments and the specific results are presented below.

Step III Two subtraces $D0$ and $D1$ are randomly chosen.

Step III Train is performed on $D0$ and $D1$, and the dissimilarity scores are predicted. For Dataset 1, the Siamese neural network is trained for 30 or more epochs. We monitor the loss values during the training and determine whether more training is required. After training the network, the dissimilarity scores between one of the training data and the remainder subtraces are predicted. The results of this step are represented by a scatter plot, as shown in Figure 2. Figure 2 shows that the dissimilarity scores are not biased to one value. Hence, we can consider that the network is well-trained. Note that, the point with the highest dissimilarity in Figure 2 is $D1$.

Step III The mean value of the dissimilarity scores is calculated, and it is classified into two classes. With this step, 92.67% scalar bit recovery accuracy is achieved, i.e., 177 of 191 bits are recovered. However, higher accuracy can be obtained by performing the remaining steps.

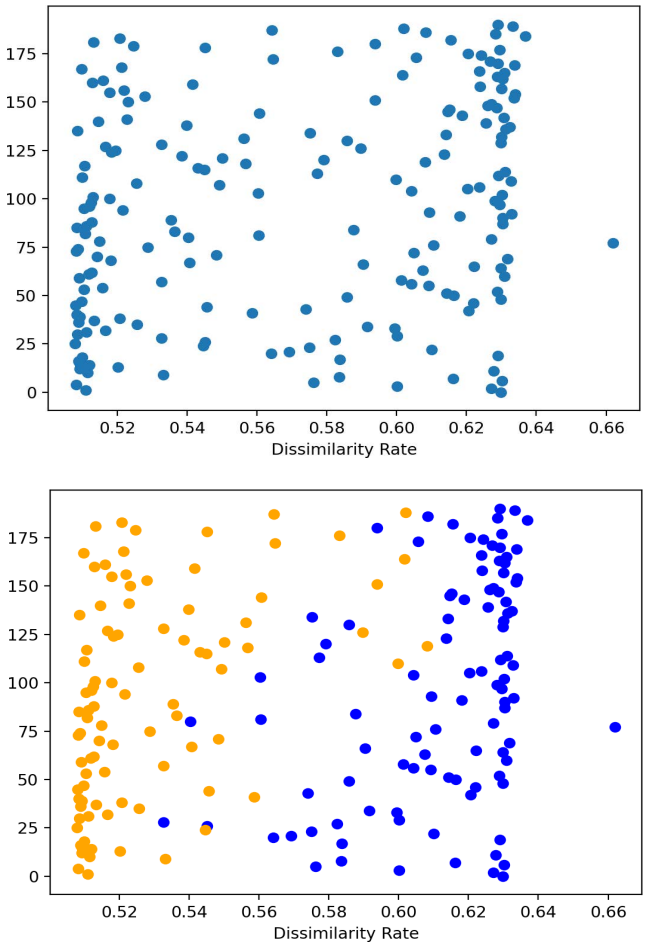

FIGURE 2. Scatter plot of dissimilarity scores; blue and yellow spots represent different classes (Y-Axis: index of the subtrace).

Step III Leakage assessment is performed on the datasets using the step 3 results. In this step, the t-test is performed using the results of step 3 in the leakage assessment. The results of this step are shown in Figure 3. Many peaks can be seen in Figure 3, and the point at which the peak rises can be considered the difference between the two classes. In contrast, Figure 4 shows the results of learning with data pairs corresponding to the same class, and it is difficult to identify a relevant peak. In the absence of a peak in the t-test results, as shown in Figure 4, it is ambiguous if the Siamese network is trained well or that the data pair used as the training data is of the same class. In this case, step 1 is repeated or the network is retrained.

Step III PoIs are extracted, and PoIs subtraces are generated. The peaks in Figure 3 are PoIs. We selected the highest 100 values from the t-test results as the PoIs. From the PoIs, we generated PoIs sutraces. Figure 5 shows that all PoIs subtraces are overlapping. Yellow and green graphs represent PoIs subtraces from different classes. Remarkable differences are apparent between the two classes. As shown in Figure 5, the PoIs can be regarded as features that can distinguish between the two classes.

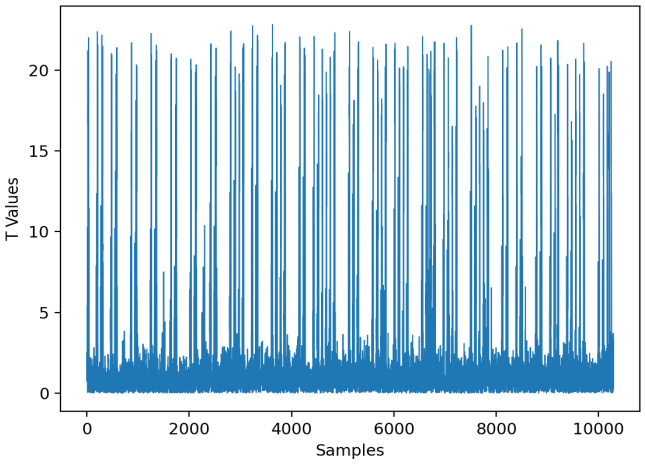

FIGURE 3. t-test on Dataset 1 using results of step III.

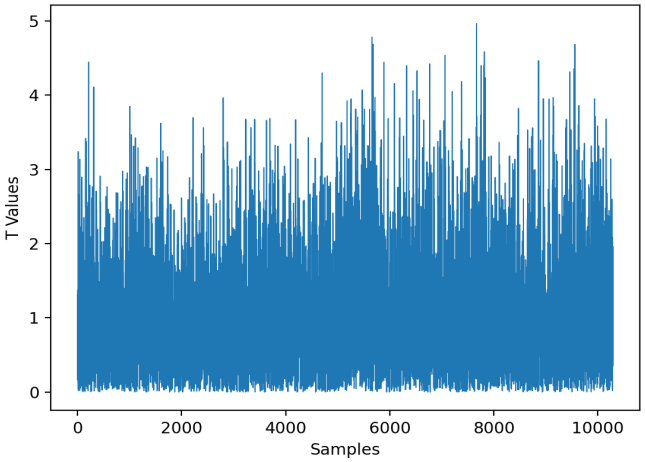

FIGURE 4. t-test on Dataset 1 using results of step 3 when attacker is trained with two same-class data.

Step III The PoIs subtraces are classified into two classes using a distinguisher. We achieve 100% scalar bit recovery accuracy for a single trace.

The results are checked: Even if two selected subtraces are of the same class, occasionally they are not filtered as belonging to the same class in all framework steps. However, we still realize that they are in the same class. From the step 6 results, we can overlap the two classified subtraces. In Figures 6 and 7, the different color plots indicate that the subtraces belong to different classes. If both color plots are symmetric around zero (e.g., the black line in Figures 6 and 7) and the characteristics of each group are noticeable, as shown in Figure 6, then the extracted features are valid for classifying the subtraces. However, Figure 7 shows that the two group plots are asymmetric, and one classified group obscures the features of the other classified group. Thus, the two selected subtraces in step 1 are of the same class, i.e., the features of both groups are not chosen well.

The attacks are extended: Using the PoIs from step III, the attack can also be extended to the other single traces.

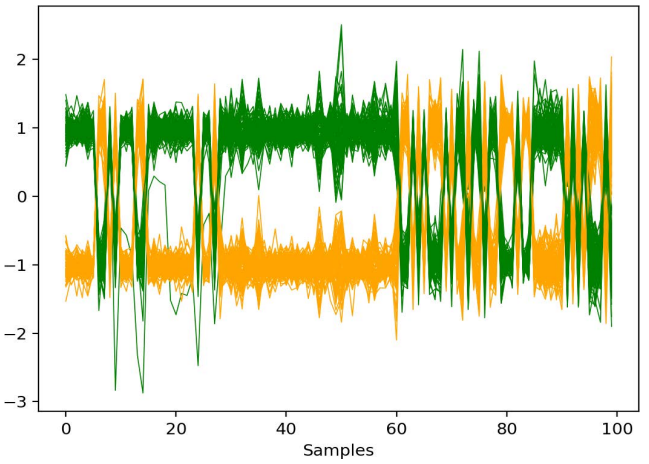

FIGURE 5. Overlapping PoIs subtraces from step 5; yellow and green plots are different classes (Y-Axis: z-score).

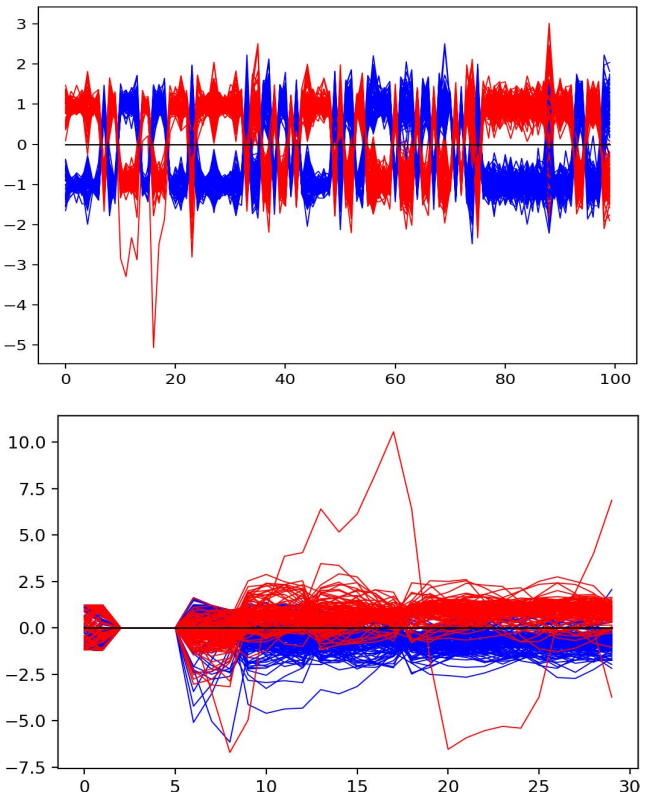

FIGURE 6. Overlapping PoIs subtraces when selected two subtraces are from different classes (X-Axis: Samples, Y-Axis: z-score, upper: Dataset 1, lower: Dataset 2).

This can be done simply by taking the PoIs of step III and performing steps III and III on another single trace. Figure 8 shows the overlapping of all PoIs subtraces when step 5 is applied to another single trace. Figure 8 also shows the prominent points between the two classes. This method is applied to the other 19 single traces. In conclusion, we achieve 100% scalar bit recovery accuracy for all 19 single traces.

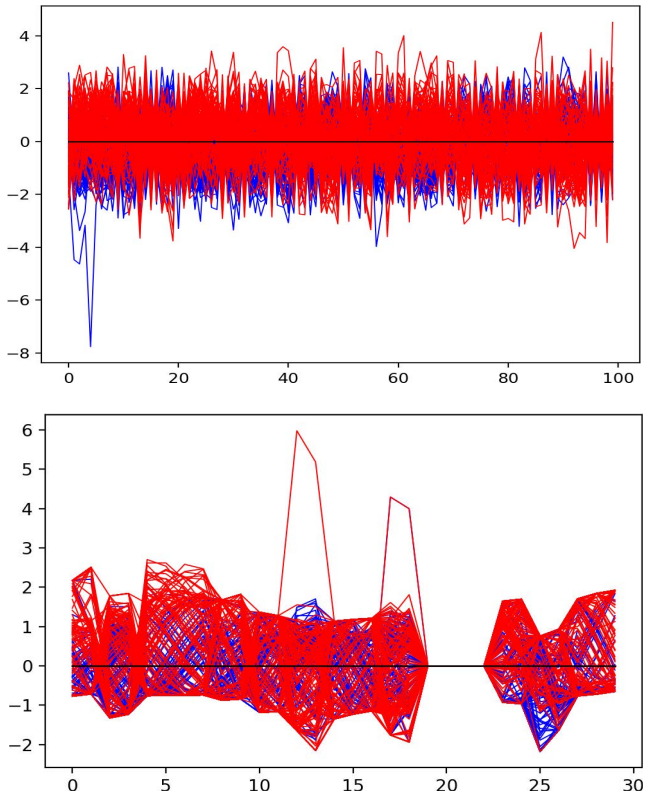

FIGURE 7. Overlapping PoIs subtraces when selected two subtraces are from same class (X-Axis: Samples, Y-Axis: z-score, upper: Dataset 1, lower: Dataset 2).

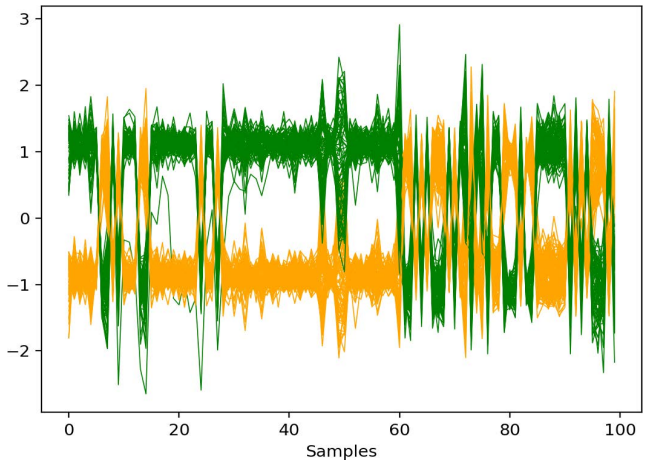

FIGURE 8. Overlapping PoIs Subtraces from another single trace: PoIs from step 5 (Y-Axis: z-score).

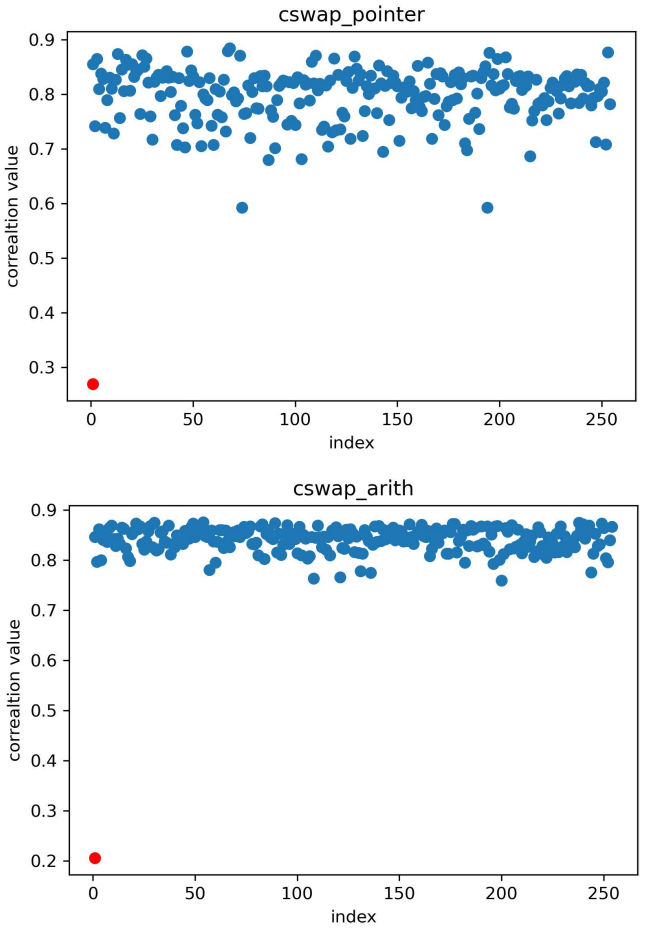

FIGURE 9. Correlation coefficient value between mean trace of all subtraces and each subtrace.

TABLE 4. CNN architectures in Siamese network considered for Dataset 2.

Layer	Dataset 2	Dataset 2_Dropout
Input	input_size = 1000	input_size = 1000
Conv1D_1	f = 8, ks = 40, s = 4	f = 8, ks = 40, s = 4
Conv1D_2	f = 16, ks = 40, s = 4	f = 16, k = 40, s = 4
Conv1D_3	f = 32, ks = 40, s = 4	f = 32, ks = 40, s = 4
Dropout_1	-	dropout_rate = 0.5
Dense_1	100 neurons	100 neurons
Dropout_2	-	dropout_rate = 0.5
Dense_2	100 neurons	100 neurons
Sigmoid	2 neurons	2 neurons

B. ATTACKING DATASETS 2 AND 3: CSWAP-POINTER AND CSWAP-ARITH

This section presents the results of the application of the attack framework on Datasets 2 and 3. Figure 9 shows the correlation coefficient value between the mean trace of all subtraces and each subtrace. In Figure 9, the red points are relatively low. A subtrace with such a low correlation coefficient typically appears in the first subtrace of all 300 single traces. Therefore, we consider the first subtraces of all single traces as outliers. We modified the proposed method slightly to apply it to these datasets. The first subtrace of a single trace in both datasets is classified as a dissimilarity score (e.g., the result of the Siamese network). For Dataset 3, we performed additional training steps on the PoIs subtraces to achieve better results. This section also discusses regularization techniques commonly used in machine learning. We expect a regularization method to reduce the overfitting that may occur during training. The results of each case are presented in the following subsections.

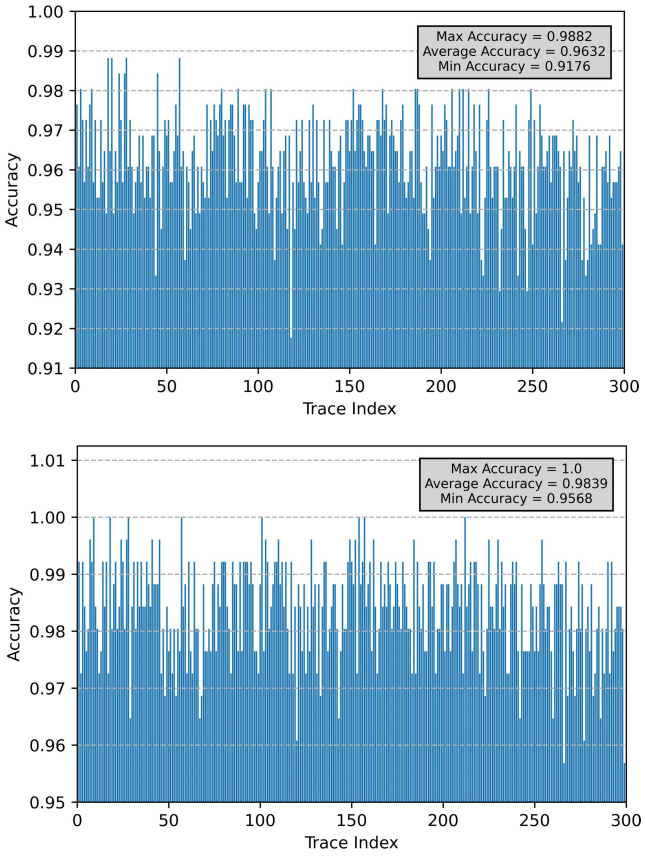

FIGURE 10. Accuracies of scalar bit recovery results for all single traces on Dataset 2.

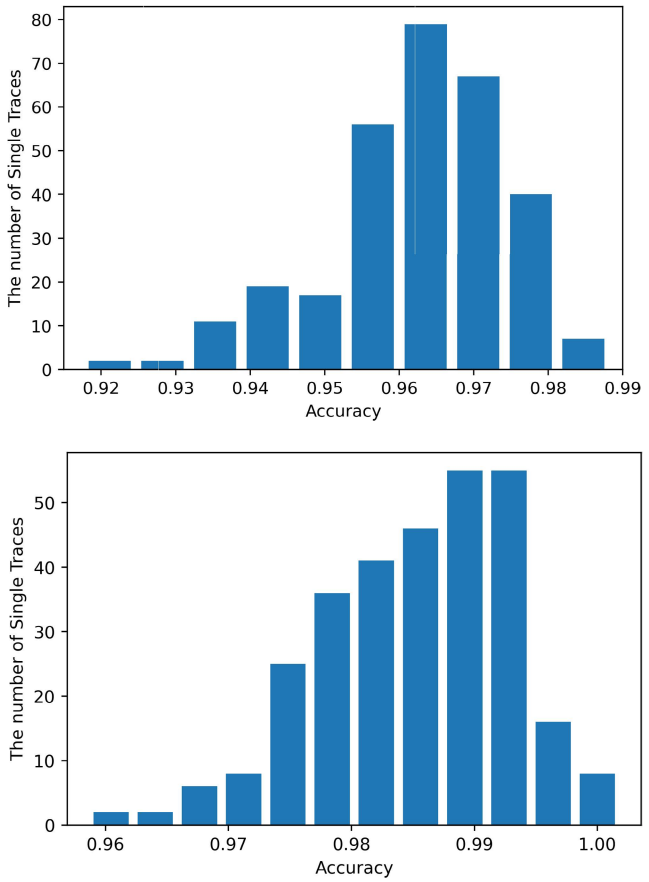

FIGURE 11. Scalar bit recovery accuracy distribution for 300 single traces on Dataset 2 (upper: without dropout, lower: with dropout).

TABLE 5. CNN architectures without *dropout* layers for Siamese network considered for Dataset 3 and related Dataset 3; f denotes filter, ks denotes kernel_size, s denotes stride, and *relu* is activation function in all cases.

Layer	Dataset 3	PoIs Subtraces 1	PoIs Subtraces2
Input	input_size = 8000	input_size = 800	input_size = 80
Conv1D_1	f=8, ks=20, s=4	f=8,ks=10,s=4	f=4, ks=10, s=4
Conv1D_2	f=16, ks=20, s=4	f=8, ks=10, s=4	f=4, ks=10, s=4
Conv1D_3	f=32, ks=20,s=4	f=8, ks=10, s=4	-
Dense_1	100 neurons	100 neurons	100 neurons
Dense_2	100 neurons	100 neurons	100 neurons
Sigmoid	2 neurons	2 neurons	2 neurons

1) ATTACKS ON CSWAP-POINTER

In this part, we applied the proposed attack framework similarly to Dataset 1. Table 4 defines the CNN architecture applied to Dataset 2. Figure 10 shows the accuracy of the scalar bit recovery results for all single traces and the maximum, average, and minimum accuracies of a total of 300 single traces for the two considered cases. We achieved 100% accuracy when dropout was considered. Without regularization, we achieved a maximum accuracy of 98.92%, an average accuracy of 96.32%, and a minimum accuracy of 91.76%. Figure 11 shows the scalar bit recovery accuracy distribution for the 300 single traces.

2) ATTACKS ON CSWAP-ARITH

We transformed the proposed attack framework a bit to achieve better results, as represented in the following with details. Table 5 defines the CNN architecture applied to Dataset 3. A single application of the Siamese network could not extract sufficient differences between the two groups on this dataset. Figure 12 shows the t-test results for a single application of the Siamese network.

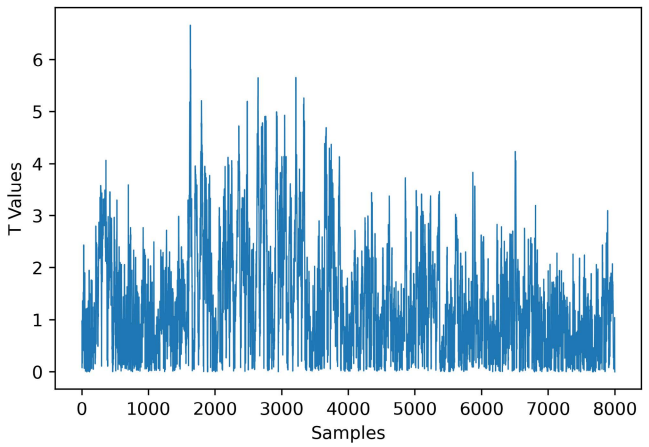

FIGURE 12. t-test results of single application of Siamese network.

A single application of the Siamese network could produce some relevant features; however, the key recovery accuracy

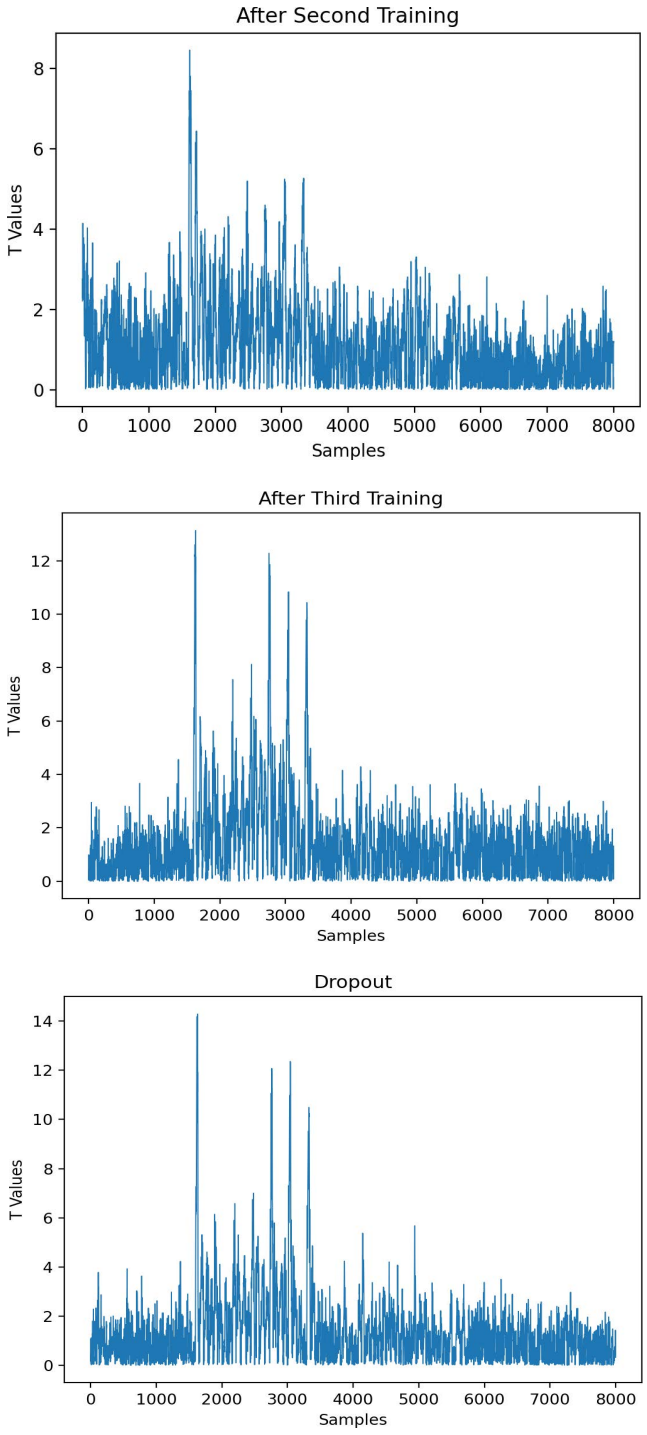

FIGURE 13. t-test results: after second training, after third training, and after third training with dropout.

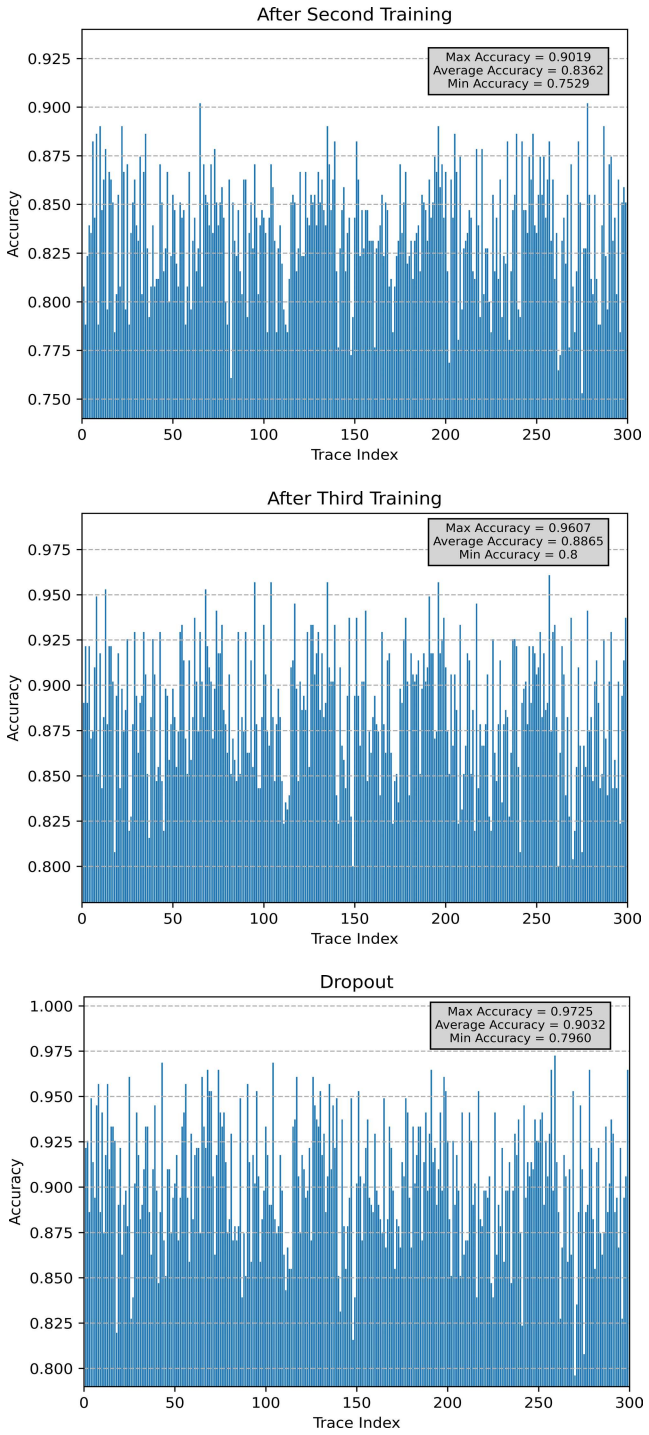

FIGURE 14. Accuracies of scalar bit recovery results for all single traces on Dataset 3.

was quite low and the features were also noisy. Therefore, we selected 800 PoIs and generated the PoIs subtraces. Similarly, steps 2–5 were performed on the generated data. Thus, the peaks became larger and more pronounced than those from the existing t-test results. However, to achieve better results, 80 PoIs were extracted from the new t-test results to create new subtraces. Subsequently, the Siamese network was applied once more. As shown in Figure 13, application of the Siamese network thrice improves the results noticeably compared to its application twice.

As illustrated in Figure 14, the results of the two-time application have a maximum accuracy of 90.19%, an average accuracy of 83.62%, and a minimum accuracy of 75.29%. It also presents much more advanced results with the

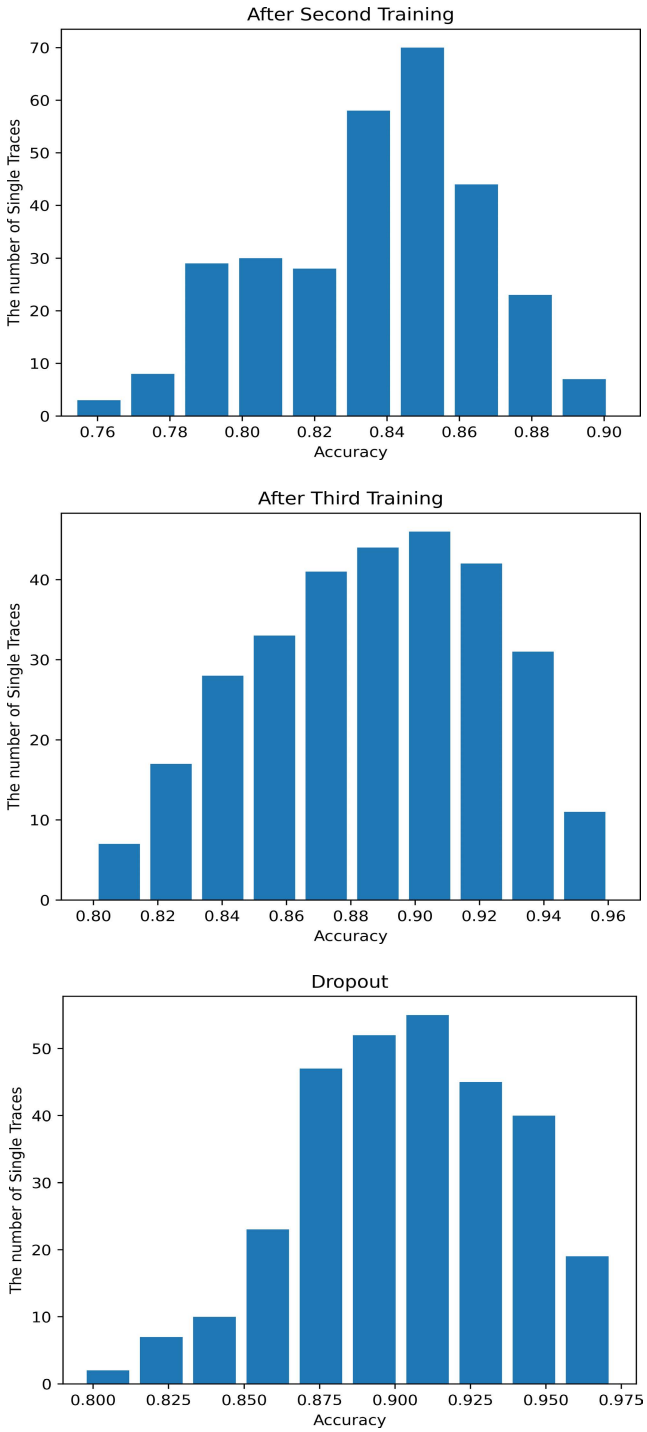

FIGURE 15. Scalar bit recovery accuracy distribution for 300 single traces on Dataset 3.

three-time application, with a maximum accuracy of 96.07%, an average accuracy of 88.65%, and a minimum accuracy of 80%. Moreover, the accuracy distribution for the 300 single traces is shown in Figure 15; after the three-time application, most single traces have over 83.62%, which is the average accuracy from the two-time application. Even when the data that cannot accurately learn the characteristics using a single Siamese network, better results can be obtained if the proposed method is repeatedly applied while reducing the learning area. The one-shot learning with the Siamese network proposed in this paper has a very short learning time. Therefore, even if the proposed attack framework is applied repeatedly as above, it yields good results in a short time. As before, dropout was also applied to this dataset. The details of the network used are listed in Table 6. Figures 13–15 present the results of three-time Siamese network application along with dropout. Comparing with the t-test results obtained without and with dropout, the noise peak is reduced and the t-value is also increased with dropout. Moreover, the maximum, average, and minimum accuracies increase to 97.25%, 90.32%, and 79.60%, respectively, indicating an overall accuracy increase [Figure 14]. Thus, for application to to this dataset, the method proposed in this paper can be appropriately modified to achieve better results.

TABLE 6. CNN architectures with *dropout* layers for Siamese network considered for Dataset 3 and related Dataset 3; f denotes filter, ks denotes kernel_size, s denotes stride, and *relu* is activation functions in all cases.

Layer	Dataset 3	PoIs-SubTraces1	PoIs-SubTraces2
Input	input_size = 8000	input_size = 800	input_size = 80
Conv1D_1	f=8, ks=20, s=4	f=8, ks=10, s=4	f=4, ks=10, s=4
Conv1D_2	f=16, ks=20,s=4	f=8, ks=10, s=4	f=4, ks=10, s=4
Conv1D_3	f=32, ks=20, s=4	f=8, ks=10, s=4	-
Dropout_1	dropout_rate = 0.5	dropout_rate = 0.5	dropout_rate = 0.5
Dense_1	100 neurons	100 neurons	100 neurons
Dropout_2	dropout_rate = 0.5	dropout_rate = 0.5	dropout_rate = 0.5
Dense_2	100 neurons	100 neurons	100 neurons
Sigmoid	2 neurons	2 neurons	2 neurons

V. CONCLUSION AND FUTURE WORKS

This study introduced an attack framework using one-shot learning with a convolutional Siamese network. This is the first such application in the use of deep learning for side-channel attacks in a non-profiled setting. Because the proposed method is based on one-shot learning, it can be applied even in an environment where a small dataset is available to an attacker. Because learning is performed with only two subtraces for a few epochs, a very short time is required to reveal the entire secret information. Nevertheless, it achieved 100% key recovery accuracy within the target where the countermeasure was applied (Datasets 1 and 2). This paper presents the specific results of each step of the framework. We showed a method to determine whether a network learns valid features and whether an attack is successful using the intermediate output during the framework process.

In future studies, we will omit the first step of the proposed framework by zero-shot learning based on a Siamese network [22]. Furthermore, we intend to implement an asymmetric key encryption algorithm as an attack target by appropriately transforming it to utilize the advantages of the proposed method.

REFERENCES

[1] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2002, pp. 13–28.

[2] J. Bromley, "Signature verification using a 'Siamese' time delay neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 6, 1993, pp. 1–8.
[3] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2004, pp. 16–29.
[4] A. Bauer, E. Jaulmes, E. Prouff, J. R. Reinhard, and J. Wild, "Horizontal collision correlation attack on elliptic curves," *Cryptogr. Commun.*, vol. 7, no. 1, pp. 91–119, Mar. 2015.
[5] M. Carbone, "Deep learning to evaluate secure RSA implementations," *TCHES*, vol. 2019, no. 2, pp. 132–161, Feb. 2019.
[6] E. Cagli, C. Dumas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures," in *Proc. Int. Conf. Cryptograph. Hardw. Embedded Syst.* Cham, Switzerland: Springer, 2017, pp. 45–68.
[7] O. Choudary and M. G. Kuhn, "Efficient template attacks," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.* Cham, Switzerland: Springer, 2013, pp. 253–270.
[8] L. Fe-Fei, R. Fergus, and P. Perona, "A Bayesian approach to unsupervised one-shot learning of object categories," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1134–1141.
[9] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2008, pp. 426–442.
[10] C. Giraud and V. Verneuil, "Atomicity improvement for elliptic curve scalar multiplication," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.* Berlin, Germany: Springer, 2010, pp. 80–101.
[11] B. Hettwer, S. Gehrer, and T. Güneysu, "Profiled power analysis attacks using convolutional neural networks with domain knowledge," in *Proc. Int. Conf. Sel. Areas Cryptogr.* Cham, Switzerland: Springer, 2018, pp. 479–498.
[12] S.-C. Hsiao, D.-Y. Kao, Z.-Y. Liu, and R. Tso, "Malware image classification using one-shot learning with Siamese networks," *Proc. Comput. Sci.*, vol. 159, pp. 1863–1871, Jan. 2019.
[13] M. Joye and S.-M. Yen, "The Montgomery powering ladder," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2002, pp. 291–302.
[14] C. F. Kerry and P. D. Gallagher, *Digital Signature Standard (DSS)*, Standard FIPS PUB 186-4, 2013.
[15] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 1999, pp. 388–397.
[16] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, vol. 2, 2015, pp. 1–30.
[17] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Proc. Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng.* Cham, Switzerland: Springer, 2016, pp. 3–26.
[18] E. Oswald and M. Aigner, "Randomized addition-subtraction chains as a countermeasure against power attacks," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2001, pp. 39–50.
[19] G. Perin, Ł. Chmielewski, L. Batina, and S. Picek, "Keep it unsupervised: Horizontal attacks meet deep learning," *TCHES*, vol. 2021, no. 1, pp. 343–372, Dec. 2020.
[20] S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, "SoK: Deep learning-based physical side-channel analysis," *Cryptol. ePrint Arch.*, 2021. [Online]. Available: https://eprint.iacr.org/2021/1092
[21] W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side channel cryptanalysis," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2005, pp. 30–46.
[22] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.
[23] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," *TCHES*, vol. 2019, no. 2, pp. 107–131, Feb. 2019.
[24] S.-M. Yen and M. Joye, "Checking before output may not be enough against fault-based cryptanalysis," *IEEE Trans. Comput.*, vol. 49, no. 9, pp. 967–970, Sep. 2000.

NAYEON LEE received the B.S. degree in AI cyber security from Korea University, South Korea, in 2022. She is currently pursuing the M.S. degree in information security with the Graduate School of Cyber Security, Korea University. Her research interests include side-channel attacks and machine-learning-based cryptanalysis.

SEOKHIE HONG (Member, IEEE) received the M.S. and Ph.D. degrees in mathematics from Korea University, in 1997 and 2001, respectively. From 2000 to 2004, he was with Security Technologies Inc. From 2004 to 2005, he conducted postdoctoral research with COSIC, KU Leuven, Belgium. He joined the Graduate School of Cyber Security, Korea University. His research interests include cryptography, public and symmetric cryptosystems, hash functions, and MACs.

HEESEOK KIM (Member, IEEE) received the B.S. degree in mathematics from Yonsei University, Seoul, South Korea, in 2006, and the M.S. and Ph.D. degrees in engineering and information security from Korea University, Seoul, in 2008 and 2011, respectively. He was a Postdoctoral Researcher with the University of Bristol, U.K., from 2011 to 2012. From 2013 to 2016, he was a Senior Researcher with the Korea Institute of Science and Technology Information (KISTI). Since 2016, he has been with Korea University. His research interests include side-channel attacks, cryptography, and network security.