Side-Channel Analysis Against SecOC-Compliant AES-CMAC

Katsumi Ebina, Rei Ueno, *Member, IEEE*, and Naofumi Homma, *Senior Member, IEEE*

***Abstract*—This brief presents a side-channel analysis (SCA) attack for AES-CMAC, which is used in the controller area network (CAN) protocol for in-vehicle networks. It is difficult to apply conventional SCAs that focus on a single round of Sboxes in the AES-CMAC, as is the case in the AES-CMAC/CTR because the AES input values are unknown to the attacker owing to its structure. The proposed method focuses on the Sboxes of the first three rounds of AES continuously and obtains the secret key by sequentially estimating the intermediate values using a first-order SCA. Our method can be applied to all versions of the Secure Onboard Communication (SecOC) standard for securing CAN protocols. We apply a deep-learning-based SCA to implement the proposed attack, in addition to conventional correlation power analysis. We demonstrate the effectiveness of the attack through an experiment using AES-CMAC software that is implemented on the PASTA automotive security evaluation platform, which is compliant with the SecOC standard that is defined in the AUTomotive Open System ARchitecture. The results show that the proposed attack can successfully reveal the secret key of AES-CMAC with at most 400,000 and 150 measurements using conventional non-profiling SCA and deep-learning-based SCA, respectively.**

***Index Terms*—Controller area network, AES, side-channel analysis, automotive security.**

I. Introduction

In RECENT years, the security risk posed by various attacks is increasing as the number of connected cars expands. One objective of such attacks is to tamper with the gateways and electronic control units (ECUs) that are connected to the in-vehicle network. The AUTomotive Open System Architecture (AUTOSAR), which is a global development partnership of the automotive industry, has released an in-vehicle communication standard that is known as Secure Onboard Communication (SecOC) [1] to provide security for data transmission and reception between ECUs.

SecOC specifies authentication using a message authentication code (MAC) to achieve message integrity (data tampering prevention and detection) and a counter known as the freshness value (FV) to prevent replay attacks. In particular, AES-CMAC [2] is recommended for authentication in the control area network (CAN) protocol, which is a common in-vehicle network protocol. AES-CMAC operates in the same manner as AES Counter mode (AES-CTR) by determining the input value using a counter. The security of AES-CMAC/CTR that is implemented on ECUs is essential to ensure automobile security.

Furthermore, side-channel analysis (SCA) is a realistic threat to cryptographic devices [3]. SCA extracts secret information by observing and analyzing side-channel information, such as leaked power/electromagnetic waves. Many advanced SCAs have been reported for IoT devices [4], [5], [6]. For example, an advanced deep learning-based SCA (DL-SCA) on major open-source cryptographic software libraries that are sometimes used for embedded devices was reported in [6]. Under these circumstances, the importance of securing automated cars against SCAs is recently being pointed out as in [7]. However, the threat of SCAs to SecOC-compliant AES-CMACs has not been explicitly discussed.

In this brief, we present an SCA method for AES-CMAC that is used in the SecOC-compliant CAN protocol, in which it is difficult to apply the conventional SCA methods that focus on a single round, such as in [8]. The proposed method focuses on the Sboxes of the first three rounds of AES, and applies the attacks of AES-CTR [8], [9] to AES-CMAC. In the first two rounds, the SCA is used for sequential estimation of the intermediate values that serve as the input for the latter rounds. In the third round, the secret key is derived by the SCA using the derived intermediate values. We apply DL-SCA in addition to the conventional correlation power analysis (CPA). Experiments using AUTOSAR SecOC-compliant AES-CMAC on the automotive security evaluation platform PASTA [10] demonstrate that all keys can be obtained from approximately 400,000 and 150 waveforms using CPA and DL-SCA, respectively.

II. Preliminaries

A. Notations

Let K be the secret key, K_i be the AES round key, X_i be the input of the ith round ($1 \leq i \leq 10$), and $x_{i,j}$ be the jth byte of X_i ($1 \leq j \leq 16$). The output $y_{i,j}$ following AddRoundKey is expressed as $y_{i,j} = x_{i,j} \oplus k_{i,j}$. Let Sbox be Sub($\cdot$). The output $z_{i,j}$ of SubBytes with the input $y_{i,j}$ is as follows: $z_{i,j} = \mathrm{Sub}(y_{i,j})$. Subsequently, the output $U_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,16})$ of ShiftRows is expressed as

$$u_{i,1} = z_{i,1},\ u_{i,5} = z_{i,5},\ u_{i,9} = z_{i,9},\ u_{i,13} = z_{i,13},$$

Manuscript received 29 March 2023; revised 28 May 2023 and 15 June 2023; accepted 16 June 2023. Date of publication 21 June 2023; date of current version 25 September 2023. This work was supported by JST CREST under Grant JPMJCR19K5. This brief was recommended by Associate Editor D. De Caro. *(Corresponding author: Naofumi Homma.)*

The authors are with the Research Institute of Electrical Communication (RIEC), Tohoku University, Miyagi 980–8577, Japan, and also with CREST, JST, Saitama 332-0012, Japan (e-mail: rei.ueno.a8@tohoku.ac.jp; naofumi.homma.c8@tohoku.ac.jp).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSII.2023.3288278.

Digital Object Identifier 10.1109/TCSII.2023.3288278

TABLE I
SPECIFICATIONS OF SECOC STANDARDS

Algorithm	SecOC_00610 CMAC/AES	SecOC_00620 CMAC/AES	SecOC_00630 CMAC/AES
Length of FV	Not specified	0	64 bits
Length of truncated FV	8 bits	0 bits	4 bits
Length of truncated MAC	24 bits	24 bits	28 bits

Algorithm 1 AES-CMAC Used in CAN

```
Require: $K$: key, $D$: message
Ensure: $Tag$: tag
$\kappa$ = GenerateSubkey_CAN($K$)
$M = (D \| 0x8000000000000000) \oplus \kappa$
$\tau = \mathrm{AES}_K(M)$
$Tag$ = truncate($\tau$)
return $Tag$
```

$$u_{i,2} = z_{i,6},\ u_{i,6} = z_{i,10},\ u_{i,10} = z_{i,14},\ u_{i,14} = z_{i,2},$$
$$u_{i,3} = z_{i,11},\ u_{i,7} = z_{i,15},\ u_{i,11} = z_{i,3},\ u_{i,15} = z_{i,7},$$
$$u_{i,4} = z_{i,16},\ u_{i,8} = z_{i,4},\ u_{i,12} = z_{i,8},\ u_{i,16} = z_{i,12}.$$

Finally, the output X_{i+1} of MixColumns is

$$\begin{aligned}
x_{i+1,4k} &= (0x02 \circ u_{i,4k}) \oplus (0x03 \circ u_{i,4k+1}) \\
&\quad \oplus (0x01 \circ u_{i,4k+2}) \oplus (0x01 \circ u_{i,4k+3}), \\
x_{i+1,4k+1} &= (0x01 \circ u_{i,4k}) \oplus (0x02 \circ u_{i,4k+1}) \\
&\quad \oplus (0x03 \circ u_{i,4k+2}) \oplus (0x01 \circ u_{i,4k+3}), \\
x_{i+1,4k+2} &= (0x01 \circ u_{i,4k}) \oplus (0x01 \circ u_{i,4k+1}) \\
&\quad \oplus (0x02 \circ u_{i,4k+2}) \oplus (0x03 \circ u_{i,4k+3}), \\
x_{i+1,4k+3} &= (0x03 \circ u_{i,4k}) \oplus (0x01 \circ u_{i,4k+1}) \\
&\quad \oplus (0x01 \circ u_{i,4k+2}) \oplus (0x02 \circ u_{i,4k+3}),
\end{aligned}$$

where the operator $\circ$ denotes multiplication on $GF(2^8)$.

In the CTR mode that is covered in this brief, a block cipher using the key K is denoted by E, the initial counter value is denoted by C, and the tth block of plaintext to be encrypted is denoted by $X^{(t)}$. The tth ciphertext $Y^{(t)}$ is represented by

$$Y^{(t)} = X^{(t)} \oplus E(C + t; K).$$

Note that we omit K if it is a fixed value.

B. AUTOSAR SecOC

Table I presents the parameter values of SecOC in the CAN protocol. Three variations exist that use the FV and MAC as parameters, which play the roles of a counter to prevent replay attacks and message authentication, respectively. We use AES and CMAC for the encryption and MAC algorithms, respectively, with a key length of 128 bits. SecOC_00620 does not have an FV. As the payload is 64 bits in the CAN protocol, the FV and MAC are truncated and added to the message as appropriate; the FV is used to prevent replay attacks, but the corresponding initialization vector is not used in the normal CMAC (the FV is used instead of the initialization vector and is always set to 0). In the CAN, AES-CMAC is used according to the above SecOC specification.

Algorithm 1 presents an overview of the AES-CMAC algorithm that is used in the CAN. The input is a 128-bit secret key K and 64-bit authenticated data D (including the FV). Figure 1 shows a block diagram of the AES-CMAC, where the subkey κ is first derived from the secret key K. Subsequently, the 64-bit message M is padded with 0x8000000000000000 to make it 128 bits and XORed with κ. Finally, the result is encrypted with AES to generate a tag. The tag is truncated to 2 bytes. Note that κ is a fixed value if K is fixed and can be pre-computed because it does not depend on D.

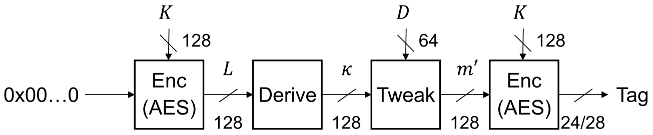


Fig. 1. AES-CMAC used in SecOC for CAN.

In this brief, we focus on AES-CMAC with counters, which is a major authentication implementation in the CAN. In this case, the actual mode of cryptographic use is AES-CMAC, but because the input value is a counter, the operation is equivalent to the AES-CTR mode if we focus on the encryption for each mode call. Therefore, we assume that an attacker can make multiple AES-CMAC calls in accordance with the CAN protocol, and consider an attack on the AES-CTR mode in which the input value is masked by the fixed value κ.

C. DL-SCA

DL-SCA consisting of profiling and attack phases is one of the most powerful attacks among SCAs with profiling.

First, in the profiling phase, side-channel waveforms are used to train a model that predicts the intermediate value of the cryptographic process that is calculated from the key and message. In many cases, the cross-entropy is used as the loss function during model training. The parameters are updated by error backpropagation to minimize the cross-entropy.

Thereafter, in the attack phase, the trained parameters that are obtained in the profiling phase are used to acquire secret information from the target device. The attacker calculates the negative log-likelihood (NLL) for each key candidate: The attacker calculates NLLs for all key candidates and estimates the key candidate $\hat{k}$ with the smallest value as the correct key.

III. PROPOSED METHOD

This section presents our proposed analysis method for SecOC-compliant AES-CMAC in the CAN protocol. The following assumptions are made: the data flowing on the CAN bus are observable, the message length (payload) is 8 bytes, and half of the 128-bit input to AES is a fixed value and cannot be known by an attacker because a subkey is added. Therefore, conventional (first-order) SCAs that focus on a single-round Sbox cannot be applied under these assumptions. Moreover, as the output value of AES is truncated to 2 bytes, the attacker cannot observe the remaining 14 bytes, which makes attacks that target the final round difficult.

The proposed method is an extension of SCAs of AES-CTR in [8]. Instead of the previous 4-round analysis flow, we introduce a 3-round analysis flow dedicated to SecOC-compliant AES-CMAC, which is different in setting the SCA-estimation variables from the previous work. Figure 2

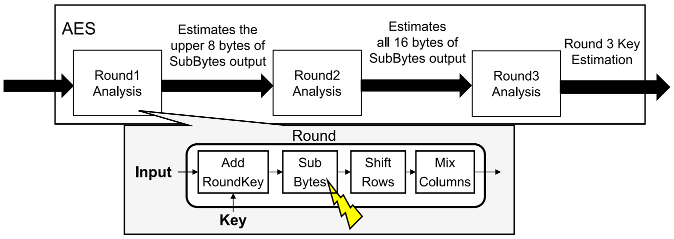


Fig. 2. Proposed analysis flow.

depicts the flow of the proposed method. First, the input of the first-round SubBytes for the attacked AES-CMAC is expressed as $D \oplus K \oplus \kappa$, where D is the 64-bit authenticated data, K is the 128-bit secret key, and κ is the CMAC key. In the first round, the upper 8 bytes of $K \oplus \kappa$ are estimated by SCA for SubBytes using the observed D. Subsequently, the ShiftRows and Mixcolmns operations are applied to the estimated upper 8 bytes to derive the 8 bytes to be input for the second round. In the second round, the round key and a fixed value that is unknown and independent of the plaintext are XORed and estimated by SCA for SubBytes. The estimated value is subjected to ShiftRows and Mixcolumns of the second round to derive all input to the third round. Finally, all round keys for the third round are estimated by SCA, and the secret key is calculated by inversing the key scheduling.

The proposed analysis procedure is demonstrated below. Let K be the secret key, K_i be the ith round key, $k_{i,j}$ be the jth byte of the ith round key, κ be the CMAC subkeys, and κ_j be the jth byte of κ. We describe the case of SecOC_00620 (the version without an FV) as an example.

1) Obtain the input $D^{(l)}$ to AES-CMAC and the corresponding side-channel waveform $W^{(l)}$ $(1 \leq l \leq N)$, where N represents the number of waveforms for the attack. In this case, $D^{(l)}$ represents the upper 8 bytes of the input to AES-CMAC.

2) Conduct SCA (e.g., CPA) targeting the first-round Sbox, and estimate $g_{1,j} = k_{1,j} \oplus \kappa_j (0 \leq j \leq 7)$ using the series of $D^{(l)}$ and $W^{(l)}$. Each element of the output of AddRoundKey in the first round can be classified into attackable and non-attackable elements by XORing the variables and fixed values. Let d_j be the jth byte of the input to AES-CMAC. The output Z_1 of the SubBytes operation in the first round of AES is expressed as

$$\begin{aligned} z_{1,j} &= \mathrm{Sub}(y_{1,j}) \\ &= \begin{cases} \mathrm{Sub}(d_j \oplus \kappa_j \oplus k_{1,j}) \ (0 \leq j \leq 7) \\ \mathrm{Sub}(0x80 \oplus \kappa_j \oplus k_{1,j}) \ (j = 8) \\ \mathrm{Sub}(0x00 \oplus \kappa_j \oplus k_{1,j}) \ (9 \leq j \leq 15). \end{cases} \end{aligned}$$

The input from the 8th to 15th bytes is a fixed value, and the conventional SCA scenario (that estimates the secret key from a single round leakage) cannot be applied. However, the input from the 0th to 7th bytes is a variable and the expected power value $p^{(l)1,j}[\gamma_{1,j}]$ of the Sbox output for the key candidate $\gamma_{1,j}$ of $g_{1,j}$ is calculated as

$$p^{(l)}_{1,j}[\gamma_{1,j}] = HW[\mathrm{Sub}(d_j \oplus \gamma_{1,j})]. \tag{1}$$

Subsequently, we estimate the correct value of $\gamma_{1,j}$, which is equal to $g_{1,j}$, from the highest calculation value (e.g., the highest correlation coefficient value in CPA) using the expected power values and measured waveforms. Using the estimated $\gamma_{1,j}$, we derive the intermediate value U_1 following ShiftRows as follows:

$$\begin{aligned} U_1 = (&z_{1,0} \| z_{1,5} \| \underline{z_{1,10}} \| \underline{z_{1,15}} \| z_{1,4} \| \underline{z_{1,9}} \| z_{1,14} \| z_{1,3} \\ &\| \underline{z_{1,8}} \| \underline{z_{1,13}} \| z_{1,2} \| z_{1,7} \| \underline{z_{1,12}} \| z_{1,1} \| z_{1,6} \| \underline{z_{1,11}}), \end{aligned}$$

where the underlined parts indicate unknown values. Note that the unknown values can also be fixed if the secret key is fixed. Finally, the second-round input X_2 is obtained following MixColumns. For example, the first four bytes $x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3}$ are expressed as

$$\begin{aligned} x_{2,0} &= (0x02 \circ u_{1,0}) \oplus (0x03 \circ u_{1,1}) \\ &\quad \oplus \underline{(0x01 \circ u_{1,2})} \oplus \underline{(0x01 \circ u_{1,3})} \\ x_{2,1} &= (0x01 \circ u_{1,0}) \oplus (0x02 \circ u_{1,1}) \\ &\quad \oplus \underline{(0x03 \circ u_{1,2})} \oplus \underline{(0x01 \circ u_{1,3})} \\ x_{2,2} &= (0x01 \circ u_{1,0}) \oplus (0x01 \circ u_{1,1}) \\ &\quad \oplus \underline{(0x02 \circ u_{1,2})} \oplus \underline{(0x03 \circ u_{1,3})} \\ x_{2,3} &= (0x03 \circ u_{1,0}) \oplus (0x01 \circ u_{1,1}) \\ &\quad \oplus \underline{(0x01 \circ u_{1,2})} \oplus \underline{(0x02 \circ u_{1,3})}. \end{aligned}$$

The other bytes are determined in the same manner. All bytes in the second round is analyzable because MixColumns diffuses known and variable values to all output bytes.

3) Conduct SCA targeting the second-round Sbox output and estimate the unknown elements of $x_{2,j}$ XORed with $k_{2,j}$ $(0 \leq j \leq 7)$ using the series of $D^{(l)}$ and $W^{(l)}$. The second-round input $x_{2,j}$ is given as the XOR of a known variable element $x'_{2,j}$ and an unknown fixed value $x''_{2,j}$, and therefore, can be decomposed as $x_{2,j} = x'_{2,j} \oplus x''_{2,j}$. For example, using the input d_j, the intermediate value $u_{1,j}(0 \leq j \leq 7)$ that is obtained from the estimated value $\gamma_{1,j}$ in Step (2), and the unknown secret information $g_{1,j}$ that could not be estimated in the first round $(8 \leq j \leq 15)$, $x'_{2,0}$ and $x''_{2,0}$ for the 0th byte are represented as

$$\begin{aligned} x'_{2,0} &= (0x02 \circ u_{1,0}) \oplus (0x03 \circ u_{1,1}) \\ &= (0x02 \circ \mathrm{Sub}(d_{1,0} \oplus \gamma_{1,0})) \\ &\quad \oplus (0x03 \circ \mathrm{Sub}(d_{1,5} \oplus \gamma_{1,5})) \\ x''_{2,0} &= (0x01 \circ u_{1,2}) \oplus (0x01 \circ u_{1,3}) \\ &= (\mathrm{Sub}(g_{1,10})) \oplus (\mathrm{Sub}(g_{1,15})). \end{aligned}$$

The other bytes are determined in the same manner. Assuming that $g_{2,j} = x''_{2,j} \oplus k_{2,j}$, the second-round SubBytes output $z_{2,j}$ is represented as

$$z_{2,j} = \mathrm{Sub}(x_{2,j} \oplus k_{2,j}) = \mathrm{Sub}(x'_{2,j} \oplus g_{2,j}),$$

where $x'_{2,j}$ is a variable denoted by d_j. Subsequently, the expected power value $p^{(l)}_{2,j}[\gamma_{2,j}]$ for the key candidate $\gamma_{2,j}$ is expressed as

$$p^{(l)}_{2,j}[\gamma_{2,j}] = HW[\mathrm{Sub}(x'_{2,j} \oplus [\gamma_{2,j})]. \tag{2}$$

As a result, all of $g_{2,j}$ can be estimated by SCA. Using the series of W^l and the expected power values, a $\gamma_{2,j}$ with the maximum calculated value (e.g., the highest correlation value) is considered as the correct value of $g_{2,j}$. Note however that we still cannot directly recover the entire secret key from $g_{2,j}$ that is given as $x''_{2,j} \oplus k_{2,j}$.

4) Using D, the obtained $g_{1,j}$, and the obtained $g_{2,j}$, we derive the full third-round input X_3 following ShiftRows and MixColumns in the same manner as in Step (3). This indicates that SCA on the third round SubBytes using X_3 can recover all third-round keys K_3. Using X_3 and K_3, the third-round SubBytes output $z_{3,j}$ is expressed as

$$z_{3,j} = \mathrm{Sub}(x_{3,j} \oplus k_{3,j}).$$

The expected power value $p^{(l)}_{3,j}[\gamma_{3,j}]$is denoted by

$$p^{(l)}_{3,j}[\gamma_{3,j}] = HW[\mathrm{Sub}(x_{3,j} \oplus k_{3,j})]], \tag{3}$$

and SCA on the third-round SubByte can recover K_3 from the estimated $\gamma_{3,j}$.

5) Finally, derive the secret key using reverse scheduling of the recovered third-round key.

The above analysis flow can be adapted to other versions that include an FV (SecOC_00610/00630) by combining it with the method of [9]. More precisely, if the attacker can observe the counter values, the above analysis is possible without any change except for how to handle the fixed values. If not, we estimate both initial counter and key values from 2^{16} combinations simultaneously. While the number of waveforms would increase to estimate a correct combination from 2^{16} candidates, such an SCA would still be feasible in principle, and the same 3-round analysis flow can be applied. An efficient estimation method is given in [9] in the case that the waveforms are observed with a high SNR.

Furthermore, note that the problem of estimating the initial value can be solved independently of the problem of limited known inputs that are considered in the method mentioned above. There is no major change in the analysis procedure even when it is combined with the estimation method.

In principle, a typical SCA such as CPA can be used for the analysis of each round if no countermeasure is applied. However, considering that long waveforms over multiple rounds are used for the analysis, it is desirable to use an SCA method that is robust against time distortion (i.e., jitter). Moreover, as the analysis will be applied to in-vehicle systems, it is important to evaluate the possibility of the proposed method with SCAs that require minimal waveforms for key recovery. Therefore, the following section presents an experimental evaluation of the proposed method using DL-SCA in addition to CPA.

IV. Experimental Evaluation

A. Application of DL-SCA to Proposed Method

DL-SCA was applied to the proposed analysis method, as described in the following.

In the profiling phase, we first collect intermediate values and the side-channel waveforms corresponding to the first three rounds of AES that are implemented on a reference device. This side-channel information should be proportional to the Hamming weights of the intermediate values (Equations (1) to (3)) that are obtained from the attack target calculation. The measured waveforms are fed into the neural network to train and generate a model using the intermediate values as a label. A convolutional neural network (CNN) is commonly used to eliminate the effect of waveform distortion.

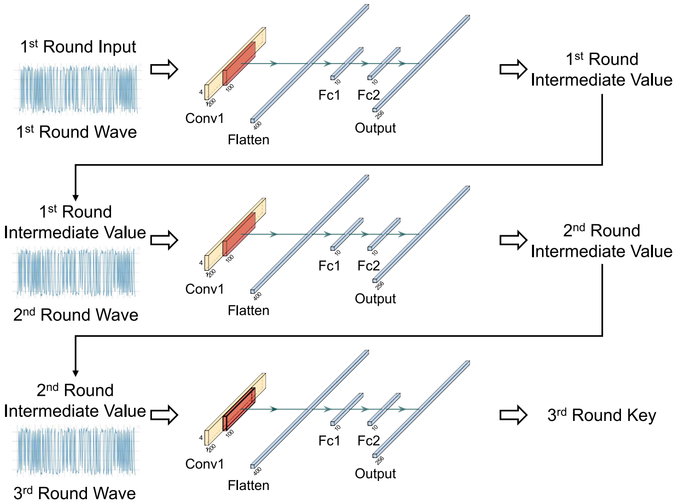


Fig. 3. Overview of DL-SCA.

In the attack phase, as illustrated in Fig. 3, we perform the following steps: (i) We first divide the measured side-channel waveforms into three rounds, input the divided waveforms into the trained model that is obtained in the profiling phase, and obtain the probability distribution of the Hamming weight, as our distinguisher, for the waveform. (ii) We calculate the intermediate values for each key candidate and obtain the Hamming weights of the intermediate values. (iii) We calculate the negative log-likelihood (NLL) of the probability that is obtained by the Hamming weight that is calculated in Step (ii) using the probability distribution that is obtained in Step (i). (iv) We repeat (i) to (iii) for each waveform and select the key candidate with the smallest average NLL as the correct key. We sequentially perform the profiling and attack phases from the first to third rounds, and finally, obtain the secret key by reverse scheduling the derived third-round key.

B. Performance Evaluation

In this experiment, we implemented SecOC_00620 on the Electronic Control Unit: ECU (RX63N) that is used in the automotive security platform PASTA (Fig. 4) and performed the above DL-SCA as a profiling attack in addition to the conventional non-profiling CPA. The PASTA consists of 4 ECUs connected by CAN bus to imitate CAN Protocol. Table II outlines the experimental setup for measuring the electromagnetic (EM) waveforms. Known random plaintexts were input for the AES software running on the ECU, and an EM probe was placed on the chip during the first to third rounds of AES. Figure 4 also shows an overview of the experimental setup.

TABLE II
MEASUREMENT SETUP

EM probe	Langer EMV-Technik RF-R 50-1 (30 MHz – 3 GHz)
Digital oscilloscope	KEYSIGHT MSOX6004A (500 MSamples/s)
Wideband low noise amplifier	LNA270WS (500 kHz – 2.7 GHz)

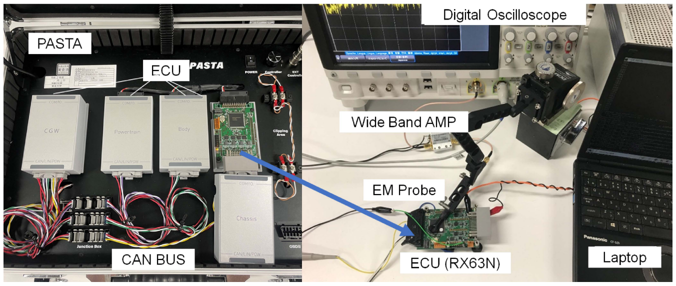


Fig. 4. Overview of PASTA and experimental setup.

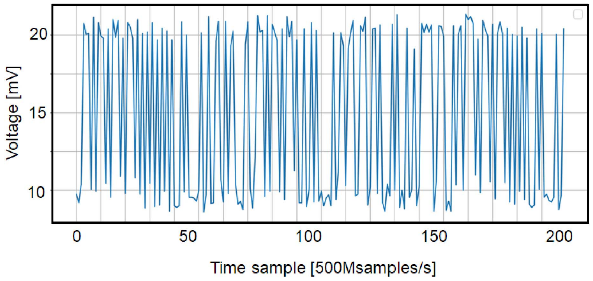
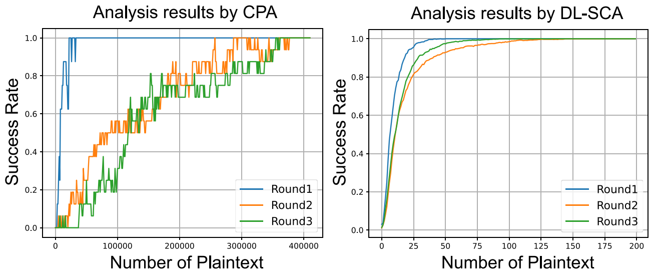


Fig. 5. Example of measured EM waveform for a round.



(a)

(b)

Fig. 6. Results of proposed method: (a) CPA result and (b) DL-SCA result.

A total of 410,000 EM waveforms were obtained. As indicated in Fig. 3, each waveform was divided into three parts, where SubBytes of each round were performed and used in the experiment. Figure 5 shows an example of EM waveform measured and divided for one round. Note that the attacker would need an extra pre-/post-processing to obtain such an EM waveform in a real-world scenario. The CNN model for profiling consists of one convolutional layer, as in [11], which is sometimes considered for comparison. The loss function for training was cross-entropy, the learning rate was 0.0002, the batch size was 64, and the number of epochs was 50. The number of waveforms during training was set to 30,000.

Figure 6 (a) depicts the results of the conventional CPA. In the first round, we could estimate correct values with approximately 30,000 waveforms. In the second and third rounds, we could also estimate the correct intermediate and key values with a maximum of 400,000 waveforms. Furthermore, Fig. 6 (b) shows the results of DL-SCA, where the success rate increased to 1 for the three rounds with approximately 150 waveforms, which indicates that the attack was successful. (Note here that the difference in the attack efficiency between DL-SCA and CPA came from the jitter alignment capability of DL-SCA for the measured EM waveforms.) The results confirm the validity and effectiveness of the proposed method.

V. CONCLUSION

We have presented an SCA method for AES-CMAC that conforms to the SecOC standard defined for CANs in in-vehicle networks and experimentally demonstrated its effectiveness. In particular, we confirmed that DL-SCA can significantly reduce the number of required waveforms compared to non-profiling SCA such as CPA. Future works will include experiments measuring waveforms from a greater distance, assuming actual automobiles, applications of advanced DL-SCA techniques as [12], [13], and the consideration of valid countermeasures.

REFERENCES

[1] "AUTOSAR." Accessed: Dec. 16, 2022. [Online]. Available: https://www.autosar.org

[2] T. Iwata, J. Song, J. Lee, and R. Poovendran, "The AES-CMAC algorithm," IETF, Fremont, CA, USA, RFC 4493, Jun. 2006.

[3] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems—CHES*, Ç. K. Koç, D. Naccache, and C. Paar, Eds. Berlin, Germany: 2001, Springer, pp. 251–261.

[4] D. R. E. Gnad, J. Krautter, and M. B. Tahoori, "Leaky noise: New side-channel attack vectors in mixed-signal IoT devices," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 3, pp. 305–339, May 2019.

[5] 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.

[6] K. Saito, A. Ito, R. Ueno, and N. Homma, "One truth prevails: A deep-learning based single-trance power analysis on RSA–CRT with windowed exponentiation," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2022, no. 4, pp. 490–526, Aug. 2022.

[7] D. Forster, T. Bruckschlogl, J. L. Omer, and T. Schipper, "Challenges and directions for automated driving security," in *Proc. 6th ACM Comput. Sci. Cars Symp.*, Dec. 2022, pp. 1–11.

[8] J. Jaffe, "A first-order DPA attack against AES in counter mode with unknown initial counter," in *Proc. Cryptograph. Hardw. Embedded Syst. (CHES)*, 2007, pp. 1–13.

[9] L. De Meyer, "Recovering the CTR_DRBG state in 256 traces," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2020, no. 1, pp. 37–65, Nov. 2019.

[10] T. Toyama, T. Yoshida, H. Oguma, and T. Matsumoto, "Pasta: Portable automotive security testbed with adaptability, Black Hat Europe." 2018. [Online]. Available: https://i.blackhat.com/eu-18/Wed-Dec-5/eu-18-Toyama-PASTA-Portable-Automotive-Security-Testbed-with-Adaptability-wp.pdf

[11] G. Zaid, L. Bossuet, A. Habrard, and A. Venelli, "Methodology for efficient CNN architectures in profiling attacks," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2020, no. 1, pp. 1–36, Nov. 2019.

[12] A. Ito, K. Saito, R. Ueno, and N. Homma, "Imbalanced data problems in deep learning-based side-channel attacks: Analysis and solution," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3790–3802, Jun. 2021.

[13] A. Ito, R. Ueno, and N. Homma, "Perceived information revisited: New metrics to evaluate success rate of side-channel attacks," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2022, no. 4, pp. 228–254, Aug. 2022.