A Note on the Security of Changing of the Guards

A Security Flaw in a Round-Based AES Implementation

Feng Zhou[1,2,3], Gehui Yang[1,2], Junhuai Yang[1,2], Hua Chen[†2], Limin Fan[2]

[1] University of Chinese Academy of Sciences, Beijing, China
[2] TCA Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing, China
{zhoufeng2021,yanggehui2024,yangjunhuai2023,chenhua,fanlimin}@iscas.ac.cn
[3] Zhongguancun Laboratory, Beijing, China

Abstract. Masking stands as one of the most effective countermeasures against side-channel attacks. The Changing of the Guards (COTG) technique reduces online fresh randomness by reusing masks and borrowing shares across nonlinear computations. We identify a security flaw in the round-based AES Design I of Askeland et al. at CARDIS 2022. When the subsequent linear layer combines outputs from different masked S-boxes, a single glitch-extended probe can observe internal registered terms from several S-boxes. We show that reused random masks can cancel across these observations, leaving a value whose distribution depends on the secret inputs. This note documents the flaw in this particular implementation without proposing a corrected construction.

Keywords: Masking · AES · Hardware · Glitch-extended Probing Security · Changing of the Guards

1 Introduction

Side-channel attacks (SCA) pose a severe threat to cryptographic implementations by exploiting unintended physical leakage such as power consumption or electromagnetic emanations [Koc96, KJJ99]. To counteract such attacks, masking [CJRR99, GP99] has emerged as one of the most widely deployed and provably secure countermeasures, where sensitive intermediate values are split into multiple random shares. By ensuring that any combination of up to d shares remains statistically independent of the secret, d-th order masking schemes effectively thwart SCA adversaries with limited probing capabilities.

The security of conventional masking schemes, however, relies heavily on the availability of fresh randomness to refresh the distribution of shares and maintain uniformity. In hardware implementations, the generation of high-quality randomness necessitates dedicated random number generators (RNGs), which not only introduce additional area and power overhead but also impose stronger assumptions on the underlying hardware platform. Consequently, there has been a growing line of research aimed at minimizing or even eliminating the online fresh randomness required in first-order masking schemes.

Changing of the Guards. One approach is the *Changing of the Guards* technique. First conceptualized in [BDN+13] and later formalized in [Dae17], COTG eliminates the need for online fresh randomness by reusing unrelated state bits—or shares from neighboring S-box computations—as temporary masks. While originally deployed in $td + 1$ Threshold Implementations [WM18, Sug18], the technique has recently been adapted to the more

[†]Hua Chen is the corresponding author.

resource-efficient $d+1$ share setting [ADN+22, DSM22]. By amortizing the cost of randomness over multiple S-boxes and requiring only a small amount of offline initialization entropy, COTG offers a promising route toward low-area, low-latency masking.

Although COTG can eliminate online fresh randomness in nonlinear computations, the security of its use in a complete AES round also depends on the subsequent linear operations. In particular, these operations may combine intermediate values from different S-boxes and expose correlations that are not visible when considering the S-boxes in isolation.

Our Contribution. We identify a security flaw in the round-based AES Design I of Askeland et al. [ADN+22], which uses COTG to reduce the need for online fresh randomness. We show how the subsequent basis transformation and MixColumns can bring internal registered values from different S-boxes into a single glitch-extended observation. The reused random masks then cancel, leaving a secret-dependent quantity and violating first-order probing security. The analysis concerns this particular complete-round implementation, rather than COTG in general.

2 Preliminaries

2.1 Notations

We use bold uppercase letters (e.g., $\mathbf{X}$) to denote sets. A letter with an arrow above it (e.g., $\vec{x}$) denotes a vector, and different elements within the vector are distinguished by subscripts (e.g., x_i denotes the i-th element of $\vec{x}$). A superscript is used to indicate a specific bit of a multi-bit element (e.g., x_i^j denotes the j-th bit of x_i). Unless otherwise specified, all elements are considered to be multi-bit values; the bit-width is either explicitly stated or clear from context.

For a secret value a, its shares are denoted by $a_0, a_1, \ldots, a_d$ such that $a = a_0 + a_1 + \cdots + a_d$ (or $a = \bigoplus_i a_i$ in the Boolean case). To avoid ambiguity with vector indices, the context will always make clear whether a subscript refers to a share index or a vector element index.

2.2 d-Probing Security

Definition 1 (d-Probing Security [ISW03])**.** Let C be a circuit taking a set of secret variables $\mathbf{X}$ as input, where $\mathbf{X} \in \mathbb{F}_2^m$. The circuit C is said to satisfy d-probing security if for any set $\mathbf{Q}$ consisting of d probes, $\mathbf{Q}$ is always statistically independent of the secret information, i.e.,

$$\Pr[\mathbf{Q} \mid \mathbf{X}] = \Pr[\mathbf{Q}] \tag{1}$$

2.3 Formulas for ShiftRows and MixColumns in AES

We denote the 16 S-boxes of the AES algorithm as $\{S_{i,0}, S_{i,1}, S_{i,2}, S_{i,3}\} (0 \le i \le 3)$. The formula for ShiftRows is as follows [AES01]:

$$\begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \rightarrow \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,1} & S_{1,2} & S_{1,3} & S_{1,0} \\ S_{2,2} & S_{2,3} & S_{2,0} & S_{2,1} \\ S_{3,3} & S_{3,0} & S_{3,1} & S_{3,2} \end{bmatrix}. \tag{2}$$

The formula for MixColumns is as follows [AES01]:

$$\begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix}. \quad (3)$$

2.4 Nonlinear and Linear Operations in an AES Round

In the round-based AES implementation considered here, the masked S-box layer is followed by linear operations, including the basis transformation and MixColumns. These operations can combine outputs from different S-boxes. When the relevant computations are not separated by registers, a glitch-extended probe in the linear layer may expose internal signals as far back as the preceding register stage. Consequently, the security of the nonlinear computations alone does not automatically establish the security of the entire round.

2.5 Changing of the Guards With Two Shares

In 2022, Askeland et al. proposed a generic COTG masking scheme for non-linear layers that requires no fresh randomness [ADN+22], allowing the same randomness to be used when masking different S-boxes in different rounds, and presented its application in AES.

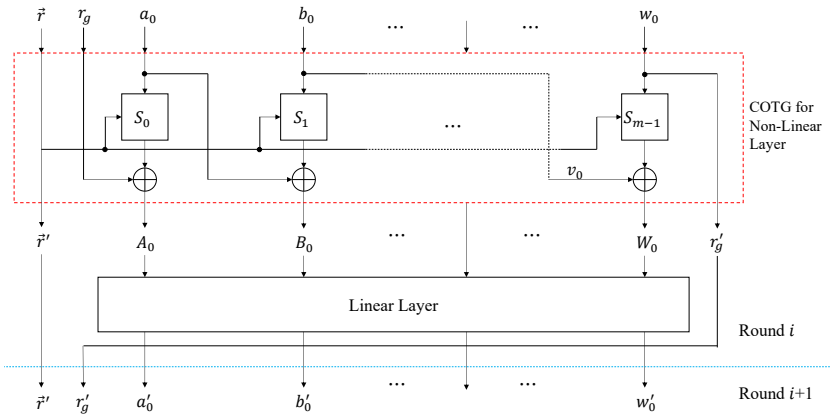

Figure 1: Diagram of COTG Masking Scheme in two Consecutive Rounds (only one share is presented in the figure)

Their concept of COTG is illustrated in Figure 1. There are m S-boxes $\{S_0, S_1, \ldots, S_{m-1}\}$ with secret inputs $\{a, b, \ldots, w\}$ and secret outputs $\{A, B, \ldots, W\}$. The inputs of COTG are input shares $\{(a_0, a_1), (b_0, b_1), \ldots, (w_0, w_1)\}$, the guards r_g and a vector of random numbers $\vec{r}$ which is used in all S-boxes. The output of COTG includes outputs shares $\{(A_0, A_1), (B_0, B_1), \ldots, (W_0, W_1)\}$, the guards r'_g and the vector of random numbers $\vec{r'}$ for the next round. As shown in Table 1, we partition the inputs and outputs of COTG into two categories: the randomness portion and the remaining shares portion.

Given an S-box masking scheme that, when masked with $\vec{r}$, achieves correctness, first-order probing security, and uniformity, Askeland et al. proposed a method for the COTG technique tailored to non-linear layers. It was shown that with this method, both the inputs and outputs preserve the following properties, namely:

Table 1: Input and Output of COTG

	Randomness			Remaining Shares
	Shares	r_g	$\vec{r}$	
Input	$a_0, b_0, \ldots, w_0$	r_g	$\vec{r}$	$a_1, b_1, \ldots, w_1$
Output	$A_0, B_0, \ldots, W_0$	r'_g	$\vec{r'} = \vec{r}$	$A_1, B_1, \ldots, W_1$

1. The randomness portion of the inputs/outputs of COTG follows a joint uniform distribution.
2. The remaining shares portion of the inputs/outputs of COTG ensures correctness of the inputs/outputs.

The formal description of COTG is provided in Definition 3, where the S-box must satisfy the properties in Definition 2. Askeland et al. subsequently stated the security result for the non-linear layer in Theorem 1.

Definition 2 (First-Order 2-Share Threshold Implementation of an S-box)**.** An S-box maps input variable shares $(a_0, a_1) \in \mathbb{F}_2^{2t}$ to output variable shares $(A_0, A_1) \in \mathbb{F}_2^{2t}$. Its input and output are t bits wide, and A^j denotes the j-th bit of the output. The S-box is called a first-order 2-share Threshold Implementation if it satisfies the following properties:

1. Correctness: $\forall 0 \leq j \leq t-1, a_0^j + a_1^j = a^j, A_0^j + A_1^j = A^j$;
2. First-order probing security: Satisfies the properties in Definition 1;
3. Uniformity: $\forall 0 \leq j \leq t-1, i_j \in \{0,1\}, \alpha^j$ being any constant in $\{0,1\}$, we have $\Pr[A_{i_0}^0 = \alpha^0, A_{i_1}^1 = \alpha^1, \ldots, A_{i_{t-1}}^{t-1} = \alpha^{t-1}] = \frac{1}{2^t}$.

Definition 3 (COTG Masking Scheme for Non-linear Layers [ADN+22])**.** COTG is applied to m S-boxes $\{S_0, S_1, \ldots, S_{m-1}\}$ with secret inputs $\{a, b, \ldots, w\}$ and secret outputs $\{A, B, \ldots, W\}$. Each S-box is conforming to Definition 2. The inputs of COTG are input shares $\{(a_0, a_1), (b_0, b_1), \ldots, (w_0, w_1)\}$, the guards r_g and a vector of random numbers $\vec{r}$ which is used in all S-boxes. The output of COTG includes outputs shares $\{(A_0, A_1), (B_0, B_1), \ldots, (W_0, W_1)\}$, the guards r'_g and the vector of random numbers $\vec{r'}$ for the next round. The outputs are computed as follows: (Assume the input shares to S-box S_{m-2} are (v_0, v_1))

$$
\begin{aligned}
&\vec{r'} = \vec{r} \\
&A_0 = S_0^0(a_0, a_1, \vec{r}) + r_g && A_1 = S_0^1(a_0, a_1, \vec{r}) + r_g \\
&B_0 = S_1^0(b_0, b_1, \vec{r}) + a_0 && B_1 = S_1^1(b_0, b_1, \vec{r}) + a_0 \\
&\cdots && \cdots \\
&W_0 = S_{m-1}^0(w_0, w_1, \vec{r}) + v_0 && W_1 = S_{m-1}^1(w_0, w_1, \vec{r}) + v_0 \\
&r'_g = w_0
\end{aligned}
$$

$S_j^i(\cdot)$ means the i-th output share of the j-th S-box.

Theorem 1 ([ADN+22])**.** *The scheme presented in Definition 3 satisfies correctness, first-order probing security, and uniformity.*

This cited result concerns the COTG construction for the non-linear layer. Its use in a complete round-based implementation with subsequent unregistered linear operations requires separate security analysis.

3 A Security Flaw in a Round-Based AES Implementation

As shown in Subsection 2.4, a round function generally comprises a linear layer and a non-linear layer. The scheme proposed by Askeland et al. in CARDIS 2022 [ADN+22] only considered the masking design for the non-linear layer. Therefore, considering the linear layer scheme as well may introduce application flaws. The linear layer may mix the outputs of multiple S-boxes together. Due to the presence of glitches, probing the linear layer could simultaneously reveal masked data from within multiple S-boxes. Since they utilize the same mask, this could potentially leak secret information. For example, during the MixColumns operation of the AES algorithm, as shown in Equation 3, the outputs of S-boxes $S_{2,0}$ and $S_{3,0}$ are added together and mixed.

We provide a detailed flaw analysis below.

In the first-order CMS scheme used in the relevant AES design, the compression process results in expressions of the form "variable share × variable + randomness."

In a first-order CMS scheme, let the input shares be $\{a_0, a_1, b_0, b_1\}$, the random numbers be $\{r_0, r_1, r_2\}$, and the intermediate results be $\{(a_0b_0 + r_0), (a_0b_1 + r_1), (a_1b_0 + r_2), (a_1b_1 + r_0 + r_1 + r_2)\}$. During compression, the first two terms and the last two terms are combined, yielding

$$a_0b + r_0 + r_1, \quad a_1b + r_0 + r_1.$$

In [ADN+22], Askeland et al. presented Design I, a scheme applying CMS and COTG to first-order $\mathbb{F}_2^8$ inverter and full AES, respectively. The output shares of the $\mathbb{F}_2^8$ inverter is precisely the "variable share × variable + randomness" form resulting from the aforementioned compression that leads to security issues in Design I, as explained below.

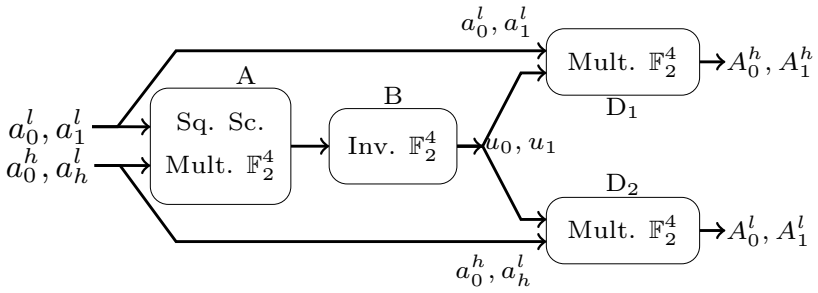

Figure 2: Decomposition of the Canright's Inverter, adopted in Design I of [ADN+22]

In Design I, both input and output are 8 bits wide. During computation, the input and output are split into high and low 4-bit nibbles, denoted by l and h, respectively. The input to the first S-box is denoted as $\{a_0^l, a_0^h, a_1^l, a_1^h\}$ (i.e., $a_0^l \parallel a_0^h = a_0$, where the symbol "$\parallel$" denotes string concatenation; the same applies hereinafter). In the last stage of the AES S-box masking scheme, the two shares (u_0, u_1) of the intermediate variable u (which is calculated from a^l, a^h, as shown in Figure 2) are also inputs to the last two multipliers in $\mathbb{F}_2^4$. The S-box output is $A^l = a^h \cdot u, A^h = a^l \cdot u$. The scheme for A^l is ($\{\cdot\}$ indicates the value is stored in a register, $r_3 = r_0 + r_1 + r_2$):

$$\begin{array}{ll} \{f_0 + r_0\}^l = \{a_0^h \cdot u_0 + r_0^l\} & \rightarrow \\ \{f_1 + r_1\}^l = \{a_0^h \cdot u_1 + r_1^l\} & \rightarrow \{f_0 + r_0\}^l + \{f_1 + r_1\}^l + \{r_g^l\} = A_0^l \\ \hline \{f_2 + r_2\}^l = \{a_1^h \cdot u_0 + r_2^l\} & \rightarrow \\ \{f_3 + r_3\}^l = \{a_1^h \cdot u_1 + r_0^l + r_1^l + r_2^l\} & \rightarrow \{f_2 + r_2\}^l + \{f_3 + r_3\}^l + \{r_g^l\} = A_1^l \end{array} \tag{4}$$

Similarly, the input to the second S-box is $\{b_0^l, b_1^l, b_0^h, b_1^h\}$. Denote the intermediate

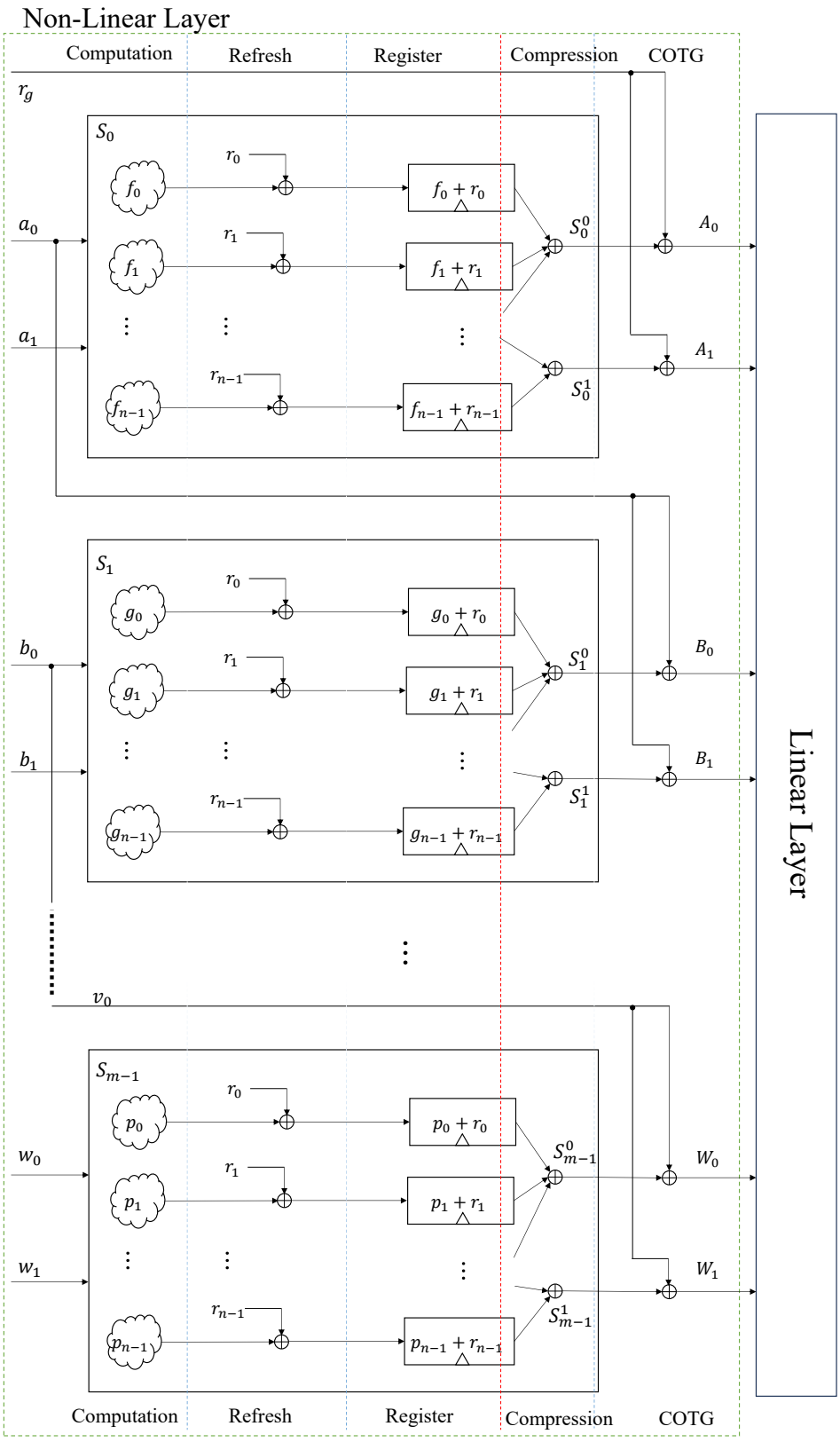

Figure 3: Detailed Diagram of COTG (Only the last register stage of the S-boxes is shown in the figure).

input variable to the last stage by v, and the computation of the low output nibble B^l is:

$$\begin{array}{ll} \{g_0 + r_0\}^l = \{b_0^h \cdot v_0 + r_0^l\} & \rightarrow \\ \{g_1 + r_1\}^l = \{b_0^h \cdot v_1 + r_1^l\} & \rightarrow \{g_0 + r_0\}^l + \{g_1 + r_1\}^l + \{a_0^l\} = B_0^l \\ \hline \{g_2 + r_2\}^l = \{b_1^h \cdot v_0 + r_2^l\} & \rightarrow \\ \{g_3 + r_3\}^l = \{b_1^h \cdot v_1 + r_0^l + r_1^l + r_2^l\} & \rightarrow \{g_2 + r_2\}^l + \{g_3 + r_3\}^l + \{a_0^l\} = B_1^l \end{array} \tag{5}$$

The output of the AES S-box undergoes linear operations: first a basis transformation, followed by MixColumns. During the basis transformation, the matrix AX is left-multiplied [Can05], where

$$AX = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

From the 5th row of matrix AX, the computation of the 5th bit of the basis transformation result involves summing the 1st through 5th bits of the S-box output. After further linear operations, this result undergoes MixColumns.

Therefore, as shown in Figure 3, an adversary probing the 5th bit containing the computed result $A_0 + B_0$ within MixColumns, under the glitch-extended probing model, would trace back along the wires to obtain all values between this point and the register layer within the S-boxes. Through the basis transformation computation, the adversary could obtain the 1st through 5th bits of the S-box outputs, thus acquiring $A_0^l + B_0^l$. Combined with the computation processes of A^l and B^l, the adversary can simultaneously obtain the values $\{\{f_0 + r_0\}^l, \{f_1 + r_1\}^l, \{r_g^l\}, \{g_0 + r_0\}^l, \{g_1 + r_1\}^l, \{a_0^l\}\}$ (where $\{\cdot\}$ indicates the value is stored in a register).

The adversary computes:

$$\begin{aligned} &\{f_0 + r_0\}^l + \{f_1 + r_1\}^l + \{g_0 + r_0\}^l + \{g_1 + r_1\}^l \\ &= \{a_0^h \cdot u_0 + r_0^l\} + \{a_0^h \cdot u_1 + r_1^l\} + \{b_0^h \cdot v_0 + r_0^l\} + \{b_0^h \cdot v_1 + r_1^l\} \\ &= a_0^h \cdot u + b_0^h \cdot v \end{aligned}$$

This expression is not statistically independent of the secret information u, v. To see this explicitly, write $Z = a_0^h \cdot u + b_0^h \cdot v$ and consider independently uniform input sharing. If $u = v = 0$, then $\Pr[Z = 0] = 1$. If $u \neq 0$ and $v = 0$, the input share a_0^h is uniform over $\mathbb{F}_{2^4}$, so $\Pr[Z = 0] = 2^{-4}$. These cases are distinguishable from the distribution of Z. Since u and v are functions of the unshared S-box inputs a and b, this observable dependence violates first-order probing security.

The security issue arises because the same random values $\vec{r}$ are reused in different S-boxes and cancel within an observation permitted by a single glitch-extended probe.

4 Conclusion

This note identifies a security flaw in the round-based AES Design I of Askeland et al. when the COTG-masked S-box layer is followed by linear operations. A glitch-extended probe in the subsequent MixColumns logic can expose registered intermediate values from multiple S-boxes. Because those S-boxes reuse random masks, the masks can cancel, resulting in a secret-dependent observation. Thus, a security argument for the non-linear COTG layer alone does not establish first-order probing security for this complete-round

implementation. The issue concerns the analyzed application of COTG and does not imply that COTG is inherently insecure.

References

[ADN+22] Amund Askeland, Siemen Dhooghe, Svetla Nikova, Vincent Rijmen, and Zhenda Zhang. Guarding the first order: The rise of AES maskings. In Ileana Buhan and Tobias Schneider, editors, *Smart Card Research and Advanced Applications - 21st International Conference, CARDIS 2022, Birmingham, UK, November 7-9, 2022, Revised Selected Papers*, volume 13820 of *Lecture Notes in Computer Science*, pages 103–122. Springer, 2022.

[AES01] Advanced Encryption Standard (AES). National Institute of Standards and Technology, NIST FIPS PUB 197, U.S. Department of Commerce, November 2001.

[BDN+13] Begül Bilgin, Joan Daemen, Ventzislav Nikov, Svetla Nikova, Vincent Rijmen, and Gilles Van Assche. Efficient and first-order DPA resistant implementations of keccak. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *Lecture Notes in Computer Science*, pages 187–199. Springer, 2013.

[Can05] D. Canright. A very compact S-box for AES. In Josyula R. Rao and Berk Sunar, editors, *CHES 2005*, volume 3659 of *LNCS*, pages 441–455. Springer, Berlin, Heidelberg, August / September 2005.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 398–412. Springer, Berlin, Heidelberg, August 1999.

[Dae17] Joan Daemen. Changing of the guards: A simple and efficient method for achieving uniformity in threshold sharing. In Wieland Fischer and Naofumi Homma, editors, *CHES 2017*, volume 10529 of *LNCS*, pages 137–153. Springer, Cham, September 2017.

[DSM22] Siemen Dhooghe, Aein Rezaei Shahmirzadi, and Amir Moradi. Second-order low-randomness d + 1 hardware sharing of the AES. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 815–828. ACM Press, November 2022.

[GMK16] Hannes Gross, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. Cryptology ePrint Archive, Report 2016/486, 2016.

[GMK17] Hannes Groß, Stefan Mangard, and Thomas Korak. An efficient side-channel protected AES implementation with arbitrary protection order. In Helena Handschuh, editor, *CT-RSA 2017*, volume 10159 of *LNCS*, pages 95–112. Springer, Cham, February 2017.

[GP99] Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *CHES'99*, volume 1717 of *LNCS*, pages 158–172. Springer, Berlin, Heidelberg, August 1999.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, Berlin, Heidelberg, August 2003.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer, Berlin, Heidelberg, August 1999.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113. Springer, Berlin, Heidelberg, August 1996.

[OSD26] Artemii Ovchinnikov, Aein Rezaei Shahmirzadi, and Siemen Dhooghe. Uniform sharing in multiple stages: NullFresh for arbitrary functions. Cryptology ePrint Archive, Paper 2026/079, 2026.

[RBN+15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating masking schemes. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 764–783. Springer, Berlin, Heidelberg, August 2015.

[SM21] Aein Rezaei Shahmirzadi and Amir Moradi. Re-consolidating first-order masking schemes. *IACR TCHES*, 2021(1):305–342, 2021.

[Sug18] Takeshi Sugawara. 3-share threshold implementation of AES s-box without fresh randomness. *IACR TCHES*, 2019(1):123–145, 2018.

[WFW+25] Lixuan Wu, Yanhong Fan, Weijia Wang, Bart Preneel, and Meiqin Wang. Extending randomness-free first-order masking schemes and applications to masking-friendly S-boxes. *IACR TCHES*, 2025(1):340–366, 2025.

[WM18] Felix Wegener and Amir Moradi. A first-order SCA resistant AES without fresh randomness. In Junfeng Fan and Benedikt Gierlichs, editors, *COSADE 2018*, volume 10815 of *LNCS*, pages 245–262. Springer, Cham, April 2018.