How to Launch a Powerful Side-Channel Collision Attack?*

Jiangshan Long[1], Changhai Ou[1], Yajun Ma[2], Yifan Fan[1], Hua Chen[3], and Shihui Zheng[4]

[1] School of Cyber Science & Engineering, Wuhan University, Wuhan, China.
ouchanghai@whu.edu.cn
[2] College of Information Engineering, Northwest A&F University, Yangling, Shanxi, China.
[3] Institute of Software, Chinese Academy of Sciences, Beijing, China.
chenhua@tca.iscas.ac.cn
[4] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China.
shihuizh@bupt.edu.cn

Abstract

Benefiting from its independence of leakage model, side-channel collision attack is one of the most common distinguishers and attracts wide attention. Although several improvements have been given, its performance on attacking a single collision value has not been significantly improved. Its optimization and efficiency is still an open problem. To solve this, we theoretically analyze the quantitative relationship between encryptions and collisions in this paper, and propose an efficient side-channel attack named Collision-Paired Correlation Attack (CPCA) for low noise scenarios to guarantee that the side with fewer samples in a collision to be detected is completely paired. This optimizes the inefficient utilization of collision information in the existing collision attacks. Moreover, to further exploit the collision information, we maximize the collision pairing, and this optimization significantly improves CPCA and extends our CPCA to large noise scenarios. Finally, to reduce computation complexity, we further optimize our CPCA to a CPA-like distinguisher. Our further theoretical study fully illustrates that our CPCA provides the upper security bound of CECA, and experimental results fully show its superiority.

Contents

*Corresponding author: Changhai Ou.

1 Introduction

Secret information leaks through side-channels such as time consumption [15], power consumption [27], electromagnetic [11], cache patterns [18] and acoustic [12] when cryptographic algorithms are implemented on hardware devices. These leakages can be collected and exploited to recover the secret information (e.g. the private key). These attacks are named as Side-Channel Attacks (SCAs). They have successfully conquered many cryptographic systems in real world, and attract wide attention. The power side-channel attacks can be divided into two categories according to whether they need to profile a leakage model: profiled attacks and non-profiled attacks.

Compared to non-profiled attacks, profiled attacks exploit the side-channel leakages in the optimal manner, but require the knowledge of the leakage function. This can be done by estimation (i.e., profiling). They then apply a maximum likelihood distinguisher (e.g. Template Attack [8]) to exploit the leakage information. However, the devices may keep their cryptographic implementations as secrets, which are not allowed to profile. Moreover, the variability of fabrication in some particular architectures like deep submicron CMOS technologies can be very large, their leakage model is unpredictable. Finally, assumption error and estimation error of leakage function will make the profiling a hard task [10, 22]. Obviously, it is difficult or even impossible for profiled power side-channel distinguishers like TA and non-profiled power side-channel distinguishers like Correlation Power Analysis (CPA) [6] exploiting classic hypothesis models (e.g. Hamming weight model) to capture leakage information in the above scenarios.

Unlike the above schemes, Collision Attacks (CAs) [16] can be well applied to the above scenarios through exploiting the similarity of leakages without relying on the specific leakage model. However, the existing power side-channel collision attacks only achieve a very low performance when considering only a fixed collision value rather than attacking several subkeys simultaneously, and our goal is to improve them. Related works will be introduced in the next subsection before introducing our contributions.

1.1 Related Works

One advantage of power side-channel collision attacks is that they can circumvent the modeling issue, i.e., they are non-profiled attacks as we have explained before. Benefiting from the repeated operations in the cryptographic implementations, collisions in an encryption can be

judged by only comparing the similarity of leakages. This feature was firstly exploited in [24] on DES and then [23] on AES to identify the same inputs of S-boxes. Bogdanov extended this to different encryptions and generalized the collisions in [3]. They further optimized the threshold decision on similarity and improved the collision detection by binary voting and ternary voting mechanisms in [4].

Unfortunately, the above collision attacks compare the similarity of leakage horizontally, and cannot be applied to masked implementations, since the intermediate values are masked and their similarity of leakages disappears. Moradi et al. estimated leakage moments corresponding to the S-boxes operations and ingeniously abstracted the notation of collision, thus avoiding direct collision detection. They used Pearson correlation coefficient to detect a collision with averaging to de-noise [20], and further exploited it to attack the flawed first-order masking AES-256 implementation provided by DPA *contest v*4.1 [2] in [19]. This Correlation-Enhanced Collision Attacks (CECA) exploits all leakages for its analysis, and each contributes globally. The disadvantage of CECA is that, it only exploits the correlation of a leakage distribution moment of samples. To guarantee the successful key recovery, it requires a large number of samples to provide sufficient collision information.

Bruneau et al. combined the flavours of stochastic and collision attacks, and derived that the stochastic collision attack exploiting the scalar product score was more adapted to multi-collisions [7]. Cezary et al. assumed that the attacker knew the distribution of the leakage function values and had a balanced set-up of traces, and they derived the Optimal Collision Attack (abbreviated as OCA) using maximum likelihood principle [14]. It is noteworthy that they improved performance of key recovery not only through enhancing the collision attack but also through an additional key searching algorithm. Wiemers et al. also exploited a key searching algorithm after the classic CECA to enhance the key recovery in [28]. In other words, the above schemes consider several sub-keys simultaneously, and select the optimal combination that meets the given collision conditions (i.e., discard the combinations that do not meet the collision conditions). This significantly improves the performance, which highlights the advantages of collision attacks.

Existing SCAs can be divided into two general approaches according to the number of sub-keys simultaneously attacked: divide-and-conquer and analytical. They can exploit two types of information: direct leakages and collision leakages. The former considers each independent S-box and conquers them one by one. The latter, analytical attacks such as collision attack, recovers the key through solving a system of equations and exploits more leakage information than divide-and-conquer attacks, but is harder to launch. There were some attacks combining the analytical collision attacks with other divide-and-conquer side-channel attacks (e.g., CECA combined with CPA in [26] and our works in [21]). These combined attacks achieved very good performance in key recovery. However, our goal in this paper is to improve the collision attack against each collision value not only without combining with a divide-and-conquer attack like [21,26], but also without a key searching algorithm after the collision attack like [7,13,26,28]. Although these works are also very interesting, they are orthogonal to our goal, and hence we do not discuss them in the remaining of this paper.

1.2 Our Contributions

This paper is interested in how to improve the performance of collision attack on each single collision value as explained in Section 1.1. Our main contributions are as follows:

(i) We theoretically analyze the relationship between the number of encryptions and the number of collisions, and propose an efficient side-channel attack named Collision-Paired

Correlation Attack (CPCA) for low noise scenarios to guarantee that the side with fewer samples in a collision to be detected is completely paired. This optimizes the inefficient utilization of collision information in the existing collision attacks.

(ii) A part of collision information is still ignored in the above CPCA, and we further maximize the collision pairing. This optimization significantly improves CPCA and works well in large noise scenarios.

(iii) The explosively increasing collision pairs will bring huge computation in the above optimized CPCA. We further extend CPCA to a CPA-like distinguisher. It has strong anti-noise ability and makes the collision pairs increase linearly with the number of encryptions.

(iv) We theoretically prove that the performance of CPCA is the upper bound of CECA.

Experimental results fully illustrate our CPCA's superiority.

1.3 Organization

The rest of this paper is organized as follows: side-channel leakage, collision attack, CECA, stochastic collision attack, the improved CECA and optimal collision attack are introduced in Section 2. Theoretical analysis on relationship between the number of encryptions and the number of collisions, our CPCA and its optimization are detailed in Sections 3 and 4. We then further theoretically prove that our CPCA achieves an upper security bound of classic CECA in Section 5. Experiments on simulated samples, an ATmega328p micro-controller, and an AT89S52 micro-controller are presented in Section 6 to illustrate the superiority of our CPCA. Finally, we conclude this paper in Section 7.

2 Preliminaries

2.1 Side-Channel Leakage

Let n denote the input size of the S-box (e.g. $n = 8$ for AES-128), L denote the number of S-boxes in each round (e.g. $L = 16$ for AES-128), $k^{*(l)}$ denote the l-th sub-key and $k^{(l)}$ denote the corresponding guessing value ($l = 1, 2, \ldots, L$), Q denote the number of plaintexts totally encrypted, $t_q^{(l)}$ denote the l-th block of the q-th encrypted plaintext and $x_q^{(l)}$ denote the corresponding leakage ($q = 1, 2, \ldots, Q$). Here an identical leakage model can be expressed as:

$$x_q^{(l)} = \varphi\left(t_q^{(l)} \oplus k^{*(l)}\right) + \mathbb{N}_q^{(l)}. \tag{1}$$

and simplified as: $x_q^{(l)} = \varphi_{t_q^{(l)}, k^{*(l)}} + \mathbb{N}_q^{(l)}$. Here φ is a deterministic but unknowable leakage function, $\mathbb{N}_q^{(l)}$ is the independent noise component on the q-th trace, and $\mathbb{N}^{(l)}$ follows the normal distribution $\mathcal{N}\left(0, \sigma^2\right)$. $\boldsymbol{x}^{(\cdot)}$ is the matrix with the q-th row corresponding to the L-variate leakage $x_q^{(1)}, x_q^{(2)}, \ldots, x_q^{(L)}$.

2.2 Side-Channel Collision Attack

Let $\mathsf{Sbox}(\cdot)$ denote the look-up table operation, and $z_q^{(l)} = \mathsf{Sbox}\left(t_q^{(l)} \oplus k^{*(l)}\right)$ denote the corresponding output intermediate value. A linear collision happens if two S-boxes in the same AES

encryption or different encryptions receiving the same byte value as their inputs:

$$t_{q_1}^{(l_1)} \oplus k^{*(l_1)} = t_{q_2}^{(l_2)} \oplus k^{*(l_2)}, \tag{2}$$

which means $z_{q_1}^{(l_1)} = z_{q_2}^{(l_2)}$ (see Figure 1). We achieve the collision value $\delta^* = t_{q_1}^{(l_1)} \oplus t_{q_2}^{(l_2)} = k^{*(l_1)} \oplus k^{*(l_2)}$ between them in this case.

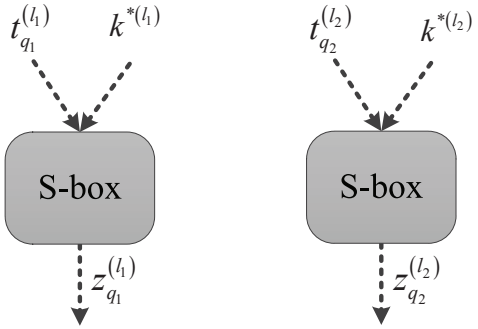

Figure 1: A collision happens if two S-boxes output the same intermediate value $z_{q_1}^{(l_1)} = z_{q_2}^{(l_2)}$.

Collision attack judges collisions by comparing the similarity of leakages, since similar leakage will generate when the hardware performs the S-box operations on the same inputs [5]. The classic side-channel collision attack only exploits the collisions happening in the same encryption (see Equation (11) and its description given in [7]), which can be expressed as:

$$\mathcal{D}_{coll} = \underset{\delta \in \mathbb{F}_2^8}{\arg\min} \frac{\sum_{t_q^{(l_1)} \oplus t_q^{(l_2)} = \delta}^{q=1 \dots Q} \left(x_q^{(l_1)} - x_q^{(l_2)} \right)^2}{\sum_{t_q^{(l_1)} \oplus t_q^{(l_2)} = \delta}^{q=1 \dots Q} 1} \tag{3}$$

under a hypothesis $\delta = t_{q_1}^{(l_1)} \oplus t_{q_2}^{(l_2)} = k^{(l_1)} \oplus k^{(l_2)}$. To successfully recover the key, multiple pairs of collisions are usually required, and the attacker needs to solve a system of δ-s.

2.3 Correlation-Enhanced Collision Attack

Collision attacks are also vulnerable to noise, just like all side-channel distinguishers. The averaging can decrease the amount of electronic noise in measurements as described in [14]. For example, if there are Q_u traces corresponding to a plaintext byte value u ($u \in \mathbb{F}_2^8$) and the noise follows distribution $\mathcal{N}\left(0, \sigma^2\right)$, we obtain "samples" with noise following the new distribution $\mathcal{N}\left(0, \frac{\sigma^2}{Q_u}\right)$ after averaging (see Section 4.6.1 in [17]). Correlation-Enhanced Collision Attack (CECA) [20] averages the leakage according to their input plaintexts. In other words, CECA divides the leakages $x_q^{(l)}$ ($q = 1 \dots Q, l = 1 \dots L$) of each S-box according to plaintext byte value $t_q^{(l)} \in \mathbb{F}_2^8$, then averages them:

$$\tau_u^{(l)} = \frac{\sum_{t_q^{(l)} = u}^{q=1 \dots Q} x_q^{(l)}}{\sum_{t_q^{(l)} = u}^{q=1 \dots Q} 1} \tag{4}$$

$\boldsymbol{x}^{(\cdot)}$ becomes a matrix of real numbers of dimension $2^n \times L$ after performing average, where the q-th row corresponds to the leakage of the plaintext byte value $q-1$. CECA then correlates the l_1-th S-box with the l_2-th S-box under a guessing collision value δ:

$$\mathcal{D}_{ceca} = \underset{\delta \in \mathbb{F}_2^8}{\arg\max}\, \rho\left\{\left(\tau^{(l_1)}_{u\in\mathbb{F}_2^8}, \tau^{(l_2)}_{u\oplus\delta}\right)\right\}. \tag{5}$$

Here $\rho\{\cdot\}$ denotes the Pearson correlation coefficient computation. Obviously, CECA exploits the correlation between two columns of matrix $\boldsymbol{x}^{(\cdot)}$ and tries to find an optimal match (i.e., the δ corresponding to the largest Pearson correlation coefficient).

2.4 Stochastic Collision Attack

Under the assumption that all S-boxes executions are with the same leakage function φ (see Equation 1), which could be approximated with a suitable vector subspace with a relatively "small" basis, the stochastic collision attack given by Bruneau et al . in [7] aimed to maximize the likelihood function from both the full-key $k^* \in \left(\mathbb{F}_2^8\right)^L$ and the leakage function φ. Specifically, they replaced the leakage function values in likelihood function by the estimates of each key k as arithmetic mean over L of the leakages $x_q^{(l)}$ $(t_q^{(l)} \oplus k^{(l)} = u)$, and the stochastic collision attack can be expressed as:

$$\mathcal{D}_{sto.coll} = \underset{k\in\left(\mathbb{F}_2^8\right)^L}{\arg\max} \sum_{u\in\mathbb{F}_2^8} \frac{\left(\sum_{l=1}^{L} \sum_{\substack{q=1\ldots Q \\ t_q^{(l)}\oplus k^{(l)}=u}} x_q^{(l)}\right)^2}{\sum_{l=1}^{L} \sum_{\substack{q=1\ldots Q \\ t_q^{(l)}\oplus k^{(l)}=u}} 1}. \tag{6}$$

This collision attack considers not only collision information between the leakages of two S-boxes like CECA, but also the collision information within each of them (see the example given in Figure 2).

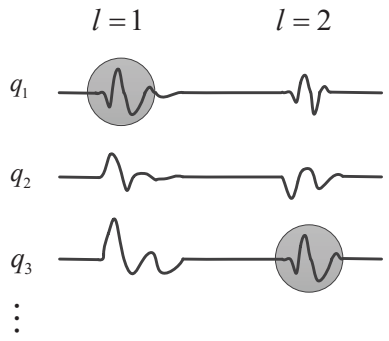

Figure 2: A collision $t_{q_1}^{(1)} \oplus k^{*(1)} = t_{q_3}^{(2)} \oplus k^{*(2)}$ happens on the highlighted parts (grey circles) of two different power traces.

It's noteworthy that due to its complexity, stochastic collision attack is limited to the scenarios that at most 4 sub-keys (with a complexity of $2^{8\cdot4}$) are attacked simultaneously. The

stochastic collision attack can be expressed as:

$$\mathcal{D}_{sto.coll} = \arg\max_{\delta \in \mathbb{F}_2^8} \sum_{u \in \mathbb{F}_2^8} \frac{\left(\sum_{\substack{q=1\ldots Q \\ t_q^{(l_1)}=u}} x_q^{(l_1)} + \sum_{\substack{q=1\ldots Q \\ t_q^{(l_2)}=u\oplus\delta}} x_q^{(l_2)} \right)^2}{\sum_{\substack{q=1\ldots Q \\ t_q^{(l_1)}=u}} 1 + \sum_{\substack{q=1\ldots Q \\ t_q^{(l_2)}=u\oplus\delta}} 1} \tag{7}$$

if we only consider the collisions between the l_1-th and the l_2-th sub-keys.

2.5 Improved CECA and Optimal Collision Attack

CECA given by Bogdanov et al. in [20] and stochastic collision attack given by Bruneau et al . in [7] are based on statistical tools rather than derived from the maximum likelihood principle. Cezary et al. derived an improved side-channel attack named as Optimal Collision Attack (we abbreviate it to OCA):

$$\mathcal{D}_{opt.coll} = \arg\max_{\delta \in \mathbb{F}_2^8} \sum_{u \in \mathbb{F}_2^8} \tau_u^{(l_1)} \times \tau_{u\oplus\delta}^{(l_2)} \tag{8}$$

in [14], which follows the maximum likelihood principle. To facilitate the theoretical analysis and security evaluation, the authors assumed that the attacker knew the distribution of leakage function values, and the traces were balanced (i.e. the same number of traces were sampled for each possible plaintext byte value $u \in \mathbb{F}_2^8$). However, this condition can be more flexible if we only consider attacks.

Bruneau et al. also provided an improved CECA as:

$$\mathcal{D}_{imp.ceca} = \arg\max_{\delta \in \mathbb{F}_2^8} \frac{\sum_{\substack{q=1\ldots Q \\ t_q^{(l_1)}=u}} x_q^{(l_1)}}{\sum_{\substack{q=1\ldots Q \\ t_q^{(l_1)}=u}} 1} \cdot \frac{\sum_{\substack{q=1\ldots Q \\ t_q^{(l_2)}=u\oplus\delta}} x_q^{(l_2)}}{\sum_{\substack{q=1\ldots Q \\ t_q^{(l_2)}=u\oplus\delta}} 1} \tag{9}$$

if only two sub-keys were under consideration (see Eq. 12 in [7]). Obviously, we can derive from Equation 4 that this improved CECA is equal to the distinguisher OCA given in Equation 8 when they are only performed on any two sub-keys. More specifically, the classic CECA in Equation 5 is equivalent to:

$$\mathcal{D}'_{ceca} = \arg\max_{\delta \in \mathbb{F}_2^8} \sum_{u \in \mathbb{F}_2^8} \left(\tau_u^{(l_1)} - \tau^{(l_1)} \right) \left(\tau_{u\oplus\delta}^{(l_2)} - \tau^{(l_2)} \right), \tag{10}$$

and both OCA given in Equation 8 and the improved CECA given in Equation 9 are the unique action item in Equation 5, i.e., other items are constants in an attack. Therefore, we will not distinguish them in the remaining of this paper.

3 Collision-Paired Correlation Attack

3.1 Collision Statistics

If only one S-box is considered, a general definition of collision is that, when encrypting a byte value in the Q-th encryption, it has appeared in the previous $Q-1$ encryptions. Based on the symbols n and Q defined in Section 2.1, here let $\chi_{2^n,Q}$ denote the byte values that have been

taken after de-duplication, and $\phi_{2^n,Q}$ denote the current number of collisions, then we get:

$$\chi_{2^n,Q} + \phi_{2^n,Q} = Q. \tag{11}$$

If a total of χ different plaintext byte values have appeared, the coming byte value will collide with them with a probability of $p_{2^n,\chi} = \frac{\chi}{2^n}$.

There exist only the following two possible cases when considering the $Q+1$ encrypted plaintext byte value:

(i) Collision happens: This occurs with a probability of $p_{2^n,\chi}$, and the number of collisions becomes $\phi_{2^n,Q+1} = \phi_{2^n,Q} + 1$.

(ii) No collision happens: This occurs with a probability of $1 - p_{2^n,\chi}$, and the number of collisions $\phi_{2^n,Q}$ does not change.

Benefiting from the above conclusions, we obtain the following recurrence formula:

$$\begin{aligned} \phi_{2^n,Q+1} &= p_{2^n,\chi} \cdot (\phi_{2^n,Q} + 1) + (1 - p_{2^n,\chi}) \cdot \phi_{2^n,Q} \\ &= \left(1 - \frac{1}{2^n}\right) \phi_{2^n,Q} + \frac{Q}{2^n}. \end{aligned} \tag{12}$$

Here let $\alpha = 1 - \frac{1}{2^n}$ and $\beta = \frac{1}{2^n}$, then $\phi_{2^n,Q+1} = \alpha \cdot \phi_{2^n,Q} + \beta \cdot Q$. We assume a new recurrence formula as:

$$\phi_{2^n,Q+1} + \gamma \cdot (Q+1) + \theta = \alpha \cdot (\phi_{2^n,Q} + \gamma \cdot Q + \theta), \tag{13}$$

then we obtain:

$$\begin{cases} (\alpha - 1)\,\gamma & = \beta, \\ (\alpha - 1)\,\theta - \gamma & = 0, \end{cases} \tag{14}$$

and deduce that $\gamma = \frac{\beta}{\alpha - 1}$ and $\theta = \frac{\beta}{(\alpha-1)^2}$. Thus, we obtain the proportional sequence. Its first term is $\alpha \cdot (\gamma + \theta) = \frac{\alpha \cdot \beta}{(\alpha-1)^2}$, and the corresponding general term is $\alpha \cdot (\phi_{2^n,Q} + \gamma \cdot Q + \theta) = \frac{\alpha \cdot \beta}{(\alpha-1)^2} \cdot \alpha^{(Q-1)}$. We further obtain:

$$\phi_{2^n,Q} = \frac{\beta}{(\alpha-1)^2} \cdot \left\{\alpha^Q - (\alpha - 1) \cdot Q - 1\right\} \tag{15}$$

and finally find that the number of collisions satisfies:

$$\phi_{2^n,Q} = \left\{\left(1 - \frac{1}{2^n}\right)^Q - 1\right\} \cdot 2^n + Q. \tag{16}$$

Cezary et al. compared classic side-channel collision attack, CECA, stochastic collision attack and the optimal collision attack in [14]. The latter two exploit multiple pairs of collisions to select the best candidates and achieve significantly higher performance compared to the first two schemes only considering individual collision values. If only a single collision value is considered, the collision information exploited by CECA, improved CECA and OCA is almost the same as we explained in Section 2. There is no obvious improvement on performance in this case, which will be verified by our experiments in Section 6. Moreover, a collision happens

if two S-boxes in the same AES encryption or different AES encryptions receive the same byte value as their inputs. Thus, a total number of:

$$\phi^{'}_{2^n,Q} = \phi_{2^n,2\cdot Q} \tag{17}$$

collisions are exploited in CECA, stochastic collision attack and OCA in theory if we encrypt Q plaintexts. We can derive from Equation 17 that very a few collisions happen under a small number of encryptions. This results in the waste of a large amount of leaky information. It can also be derived from this formula that at least 500 encryptions are required to guarantee that all 256 plaintext byte values collide with a probability of 1.00.

3.2 Our CPCA

One issue in collision attacks is that, the number of plaintext byte values u and $u \oplus \delta$ satisfying collision does not always match perfectly, i.e., the number of samples corresponding to u and $u \oplus \delta$ does not always equal. CECA computes the first-order moment (i.e., the mean value) for each possible u ($u \oplus \delta$) to make use of collision information. This achieves better match, avoids collision waste, and reduces the noise. However, this also brings a disadvantage: only the first-order moment involved in the Pearson correlation coefficient computation, and the information on the second- and higher-order moments is lost. We can exploit very limited leakage information from a small power trace set, this inefficient information utilization will significantly affect CECA's performance.

To overcome the above disadvantage, we propose a new collision attack named Collision-Paired Correlation Attack (CPCA), which no longer computes the mean values like CECA. All samples independently participate in the Pearson correlation coefficient computation in CPCA, thus maintaining all the leakage information of samples and achieving better performance. Specifically, CPCA extracts a part of samples from the side with more samples, and keeps the number of these extracted samples the same as the samples on the other side. For convenience, we re-express the identical leakage model given in Equation 1 as:

$$x^{(l)}_{u,q} = \varphi\left(u \oplus k^{*(l)}\right) + \mathbb{N}^{(l)}_{u,q} \tag{18}$$

according to the plaintext byte value $u \in \mathbb{F}_2^8$. Here $x^{(l)}_{u,q}$ denotes its q-th ($q = 1, 2, \ldots, Q^{(l)}_u$) encryption and $\mathbb{N}^{(l)}_{u,q}$ denotes the corresponding noise component. Our CPCA can be expressed as:

$$\mathcal{D}_{cpca^1} = \arg\max_{\delta \in \mathbb{F}_2^8} \rho\left\{\left(x^{(l_1)}_{u\in\mathbb{F}_2^8, q=1\ldots \min\left\{Q^{(l_1)}_u, Q^{(l_2)}_{u\oplus\delta}\right\}}, x^{(l_2)}_{u\oplus\delta,q}\right)\right\}. \tag{19}$$

and we mark it as CPCA^1. This means we extract each collision pair $\left(x^{(l_1)}_{u,q}, x^{(l_2)}_{u\oplus\delta,q}\right)$, and a pair of collision in the previous collision attacks like CECA becomes a total of $\min\left\{Q^{(l_1)}_u, Q^{(l_2)}_{u\oplus\delta}\right\}$ pairs. Obviously, different from the previous collision attacks, these collisions mean information gain in small noise scenarios and CPCA^1 exploits more subtle differences between samples.

Unlike the stochastic and optimal collision attacks, CPCA^1 is like CECA and it only considers the collisions between sub-keys, rather than within them. Therefore, the number of collisions

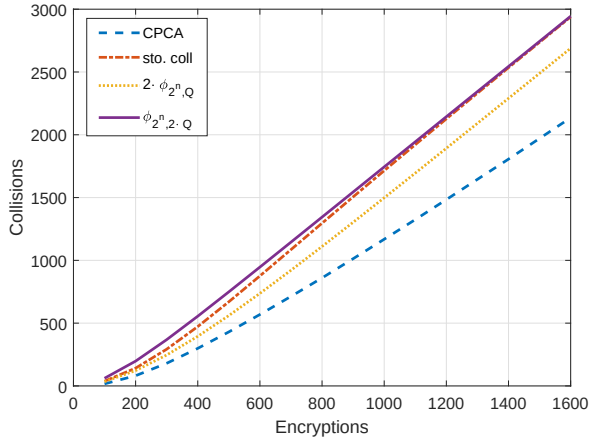

Figure 3: Theoretical and the exploited collisions in our CPCA[1].

exploited by it is:

$$\begin{aligned}\phi'_{2^n,Q} &= 2\cdot\phi_{2^n,Q}+\frac{1}{256}\\ &\approx 2\cdot\phi_{2^n,Q}.\end{aligned} \tag{20}$$

Here $\phi_{2^n,Q}$ denotes the number of collisions for one side after considering all the Q traces, and $\frac{1}{256}$ is the probability of collision between the Q-th encryption, whose impact is slight and could be ignored. Therefore, the expectation of collisions exploited by our CPCA[1] satisfies: $\phi'_{2^n,Q}\approx 2\cdot\phi_{2^n,Q}$ (see Figure 3). CPCA[1] exploits more collision details compared to stochastic and optimal collision attacks in this case.

4 Optimization

4.1 CPCA with Maximum Collision Pairing

CPCA[1] matches the collision pairs according to the side with fewer samples, thus the side with more samples is not completely matched and these side-channel leakages are ignored. To make full use of these informations, we further optimize our CPCA as:

$$\mathcal{D}_{cpca^2}=\arg\max_{\delta\in\mathbb{F}_2^8}\rho\left\{\left(x^{(l_1)}_{u\in\mathbb{F}_2^8,q_1=1\ldots Q_u^{(l_1)}},x^{(l_2)}_{u\oplus\delta,q_2=1\ldots Q_{u\oplus\delta}^{(l_2)}}\right)\right\}, \tag{21}$$

and we mark it as CPCA[2]. This means, we construct all possible pairs $\left(x^{(l_1)}_{u,q_1},x^{(l_2)}_{u\oplus\delta,q_2}\right)$ for each collision $(u,u\oplus\delta)$ $\left(q_1=1,2,\ldots,Q_u^{(l_1)},q_2=1,2,\ldots,Q_{u\oplus\delta}^{(l_2)}\right)$. Thus, for each $u\in\mathbb{F}_2^8$, we exploit a total of $Q_u^{(l_1)}\cdot Q_{u\oplus\delta}^{(l_2)}$ pairs of collisions, i.e., we make full use of the differences of samples.

4.2 Collision Explosion in CPCA^2

The encrypted plaintext bytes $t^{(l_1)}$ and $t^{(l_2)}$ are random and independent. The two sub-keys $k^{*(l_1)}$ and $k^{*(l_2)}$ are fixed, and the XOR and look-up table operations do not change their distributions. Therefore, the intermediate values $z^{(l_1)}$ and $z^{(l_2)}$ are uniformly distributed and independent. In other words, the intermediate values $z^{(l_1)}$ and $z^{(l_2)}$ follow uniform distributions:

$$z^{(l_1)} \sim \mathbb{U}\left(0, 2^n - 1\right), \quad z^{(l_2)} \sim \mathbb{U}\left(0, 2^n - 1\right). \tag{22}$$

with a probability density of $\frac{1}{2^n}$.

Obviously, any intermediate value $z^{(l_1)}$ ($z^{(l_2)}$) will appear with an expectation of $\frac{Q}{2^n}$ times after Q encryptions. This means, any intermediate value will collide about $(\frac{Q}{2^n})^2$ times, and the total number of collisions in CPCA^2 is:

$$\phi_{2^n,Q} \approx \frac{Q^2}{2^n}. \tag{23}$$

The number of collisions will grow explosively, and significantly increase the computational load in this case (as shown in Figure 4), which should be further improved.

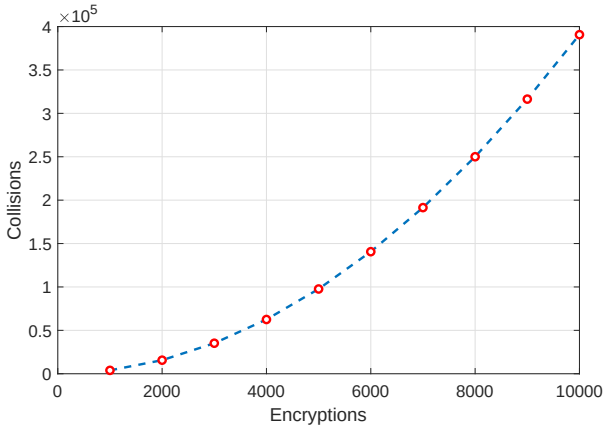

Figure 4: The number of collisions grows explosively in our CPCA^2.

4.3 Collision Explosion Avoidance

To prevent the explosive growth of collision pairs in CPCA^2, we can compute the mean power consumption of the side with more (or fewer) samples for each collision $(u, u \oplus \delta)$ before collision pairing. This strategy guarantees that at least one value in each pair of collision is relatively accurate, just like CPA, thus gaining the ability against noise. Mean value derived from the side with more samples in a collision means higher accuracy and better reference. On the contrary, mean value derived from the side with fewer samples constructs more collision pairs in CPCA. It is noteworthy that these two strategies achieve very close performance, thus we exploit the

first, and our CPCA^2 can be further extended to a CPA-like distinguisher as:

$$\mathcal{D}_{cpca^3} = \arg\max_{\delta \in \mathbb{F}_2^8} \rho \left\{ \left(x^{',(l_1)}_{u \in \mathbb{F}_2^8, q_1 = 1 \ldots Q_u^{(l_1)}}, x^{',(l_2)}_{u \oplus \delta, q_2 = 1 \ldots Q_{u \oplus \delta}^{(l_2)}} \right) \right\}$$
$$s.t. \begin{cases} x^{',(l_1)}_{u,q_1} = \tau_u^{(l_1)}, x^{',(l_2)}_{u \oplus \delta, q_2} = x^{(l_2)}_{u \oplus \delta, q_2} & \text{if } Q_u^{(l_1)} \geq Q_{u \oplus \delta}^{(l_2)} \\ x^{',(l_2)}_{u \oplus \delta, q_2} = \tau_{u \oplus \delta}^{(l_2)}, x^{',(l_1)}_{u,q_1} = x^{(l_1)}_{u,q_1} & \text{else} \end{cases} \tag{24}$$

, and we mark it as CPCA^3. This means, for each collision $(u, u \oplus \delta)$, we achieve $Q_{u \oplus \delta}^{(l_2)}$ pairs of collisions as $\left(\tau_u^{(l_1)}, x^{(l_2)}_{u \oplus \delta, q_2 = 1 \ldots Q_{u \oplus \delta}^{(l_2)}} \right)$ if $Q_u^{(l_1)} \geq Q_{u \oplus \delta}^{(l_2)}$. Otherwise, we achieve $Q_u^{(l_1)}$ pairs of collisions as $\left(x^{(l_1)}_{u, q_1 = 1 \ldots Q_u^{(l_1)}}, \tau_{u \oplus \delta}^{(l_2)} \right)$.

CPCA^3 exploits all collision information, and its complexity can be optimized from $\mathcal{O}\left(Q^2\right)$ to $\mathcal{O}\left(Q\right)$ if a total of Q plaintexts are encrypted, thus achieving explosive growth avoidance of collisions in this case. Our CPCA^3 can be further expressed as:

$$\mathcal{D}_{cpca} = \arg\max_{\delta \in \mathbb{F}_2^8} \sum_{u \in \mathbb{F}_2^8} \varepsilon_u \left(x_u^{'(l_1)} - \tau^{(l_1)} \right) \left(x_{u \oplus \delta}^{'(l_2)} - \tau^{(l_2)} \right). \tag{25}$$

Here the symbol "ε_u" denotes the proportion of samples with intermediate value u over all samples, and $\tau^{(l)}$ is the mean value acquired from the samples after processing. Although the mean values cannot accurately reflect the ideal leakage (i.e., φ in Equation 1), they are more referential compared to original samples, thus achieving significantly better performance in large noise scenarios.

5 Security Evaluations

Let $|\chi| = \chi_{2^n,Q}$ denote the byte values that have been taken after de-duplication as given in Section 3.1. For convenience, we consider a collision happens if $t_{q_1}^{(l_1)} \oplus k^{*(l_1)} = t_{q_2}^{(l_2)} \oplus k^{*(l_2)} = t_{q_1}^{(l_1)} \oplus \delta \oplus k^{*(l_2)} = t_{q_1}^{(l_1)} \oplus k^{(l_2)}$ under a guessing collision value δ. This generalizes the leakages of two S-boxes from the previously given Equation 18 as:

$$\begin{cases} x_{q_1}^{(l_1)} = \varphi\left(t_{q_1}^{(l_1)} \oplus k^{(l_1)} \right) + \mathbb{N}_{q_1}^{(l_1)}, \\ x_{q_2}^{(l_2)} = \varphi\left(t_{q_1}^{(l_1)} \oplus k^{(l_2)} \right) + \mathbb{N}_{q_2}^{(l_2)}, \end{cases} \tag{26}$$

since the attack can only sense the S-box input rather than the plaintext byte and the sub-key. Here $\mathbb{N}_q^{(l)} \sim \mathcal{N}\left(0, \sigma^2\right)$, $k^{(l_1)} = k^{*(l_1)}$, and $k^{(l_2)} = k^{*(l_2)} \oplus \delta$. This enables us to transform the collision into the case where two S-boxes receive the same input, which is convenient for the future processing. We further discuss the relationship between CECA and CPCA^3 in the next Subsections 5.1$\sim$ 5.3.

5.1 Parameters in CECA

CECA averages samples for each possible $u \in \chi$, then computes the Pearson correlation coefficient between two columns of 256 averaged samples. Let $\mathbb{N}_u^{(l)}$ denote the noise after averaging the traces corresponding to the plaintext byte value $t_q^{(l)} = u$ $(q = 1, 2, \ldots, Q_u^{(l)})$. The mean and square for the 256 averaged samples ($|\chi| = 256$) in CECA can be expressed as:

$$\begin{cases} \tau^{(l)} = \frac{1}{|\chi|} \sum_{u \in \chi} \left(\varphi \left(u \oplus k^{(l)} \right) + \mathbb{N}_u^{(l)} \right), \\ r^{(l)} = \frac{1}{|\chi|} \sum_{u \in \chi} \left(\varphi \left(u \oplus k^{(l)} \right) + \mathbb{N}_u^{(l)} \right)^2. \end{cases} \tag{27}$$

The Equation 27 facilitates CECA's Pearson correlation coefficient computation.

5.2 Parameters in Our CPCA^3

Let $\overrightarrow{\mathbf{b}} = (b_{u_1}, \ldots, b_{u_2})$ $(0 \le u_1 \le u_2 \le 255)$ denote $|\chi|$-dimensional Boolean vector, u_1 and u_2 denote the minimum and maximum plaintext byte values appearing, $b_u = 1$ $(=0)$ $(u_1 \le u \le u_2)$ denote that the samples leaking from the l_1-th S-box are more (fewer) than those leaking from the l_2-th S-box. Here we simplify $\varphi \left(u \oplus k^{(l)} \right)$ as $\varphi_{u,k^{(l)}}$ as adopted in Equation 1. This echoes our explanation: the collision attack can only sense the S-box input. For each appearing plaintext byte value u $(u_1 \le u \le u_2)$, the corresponding collisions can be divided into the following two cases:

(i) $b_u = 1$ and we average the samples on the left side leaking from the l_1-th S-box, then match this mean value with all samples on the right side leaking from the l_2-th S-box.

(ii) $b_u = 0$ and we average the samples on the right side, then match this mean value with all samples on the left side.

For the first case, the samples in CPCA^3 become:

$$\begin{cases} x_{u,q_1}^{',(l_1)} & = \varphi_{u,k^{(l_1)}} + \mathbb{N}_u^{(l_1)}, \\ x_{u,q_2}^{',(l_2)} & = \varphi_{u,k^{(l_2)}} + \mathbb{N}_{u,q_2}^{(l_2)}, \end{cases} \tag{28}$$

and a total of $Q_u^{(l_2)}$ pairs of collisions (i.e., $q_1 = q_2 = 1, 2, \ldots, Q_u^{(l_2)}$) generate. For the second case, the samples in CPCA^3 become:

$$\begin{cases} x_{u,q_1}^{',(l_1)} & = \varphi_{u,k^{(l_1)}} + \mathbb{N}_{u,q_1}^{(l_1)}, \\ x_{u,q_2}^{',(l_2)} & = \varphi_{u,k^{(l_2)}} + \mathbb{N}_u^{(l_2)}, \end{cases} \tag{29}$$

and a total of $Q_u^{(l_1)}$ pairs of collisions (i.e., $q_1 = q_2 = 1, 2, \ldots, Q_u^{(l_1)}$) generate.

Since the value of b_u(=0 or =1) is uncertain, we can merge the Equations 28 and 29, and the samples on the left side in our CPCA^3 become:

$$\begin{aligned} x'^{,(l_1)}_{u,q_1} &= b_u \left(\varphi_{u,k^{(l_1)}} + \mathbb{N}_u^{(l_1)}\right) \\ &\quad + (1-b_u)\left(\varphi_{u,k^{(l_1)}} + \mathbb{N}_{u,q_1}^{(l_1)}\right) \\ &= \varphi_{u,k^{(l_1)}} + b_u\mathbb{N}_u^{(l_1)} + (1-b_u)\,\mathbb{N}_{u,q_1}^{(l_1)}. \end{aligned} \tag{30}$$

The samples on the right side in our CPCA^3 become:

$$\begin{aligned} x'^{,(l_2)}_{u,q_2} &= (1-b_u)\left(\varphi_{u,k^{(l_2)}} + \mathbb{N}_u^{(l_2)}\right) \\ &\quad + b_u\left(\varphi_{u,k^{(l_2)}} + \mathbb{N}_{u,q_2}^{(l_2)}\right) \\ &= \varphi_{u,k^{(l_2)}} + b_u\mathbb{N}_{u,q_2}^{(l_2)} + (1-b_u)\,\mathbb{N}_u^{(l_2)} \end{aligned} \tag{31}$$

after alignment, and a total of $b_uQ_u^{(l_2)} + (1-b_u)Q_u^{(l_1)}$ pairs of collisions generate in this case. The total number of samples of CPCA after pairing is:

$$Q = \sum_{u\in\chi} b_uQ_u^{(l_2)} + (1-b_u)Q_u^{(l_1)}. \tag{32}$$

The mean of samples on the left side in our CPCA^3 is:

$$\begin{aligned} \tau^{(l_1)} &= \frac{1}{Q}\sum_{u\in\chi}\left(b_uQ_u^{(l_2)} + (1-b_u)Q_u^{(l_1)}\right)\left(\varphi_{u,k^{(l_1)}} + \mathbb{N}_u^{(l_1)}\right) \\ &= \sum_{u\in\chi}\varepsilon_u\left(\varphi_{u,k^{(l_1)}} + \mathbb{N}_u^{(l_1)}\right) \end{aligned} \tag{33}$$

Here $\varepsilon_u = \frac{b_uQ_u^{(l_2)} + (1-b_u)Q_u^{(l_1)}}{Q}$ is the proportion of u as the input plaintext byte value of the l_1-th S-box after processing. Similarly, we obtain $\tau^{(l_2)} = \sum_{u\in\chi}\varepsilon_u\left(\varphi_{u,k^{(l_2)}} + \mathbb{N}_u^{(l_2)}\right)$ for the mean of the processed samples corresponding to the l_2-th S-box in a collision.

The covariance $\mathsf{Cov}_{x'^{,(l_1)},x'^{,(l_2)}}$ can be separated into four items, i.e., covariances between the two exploitable components $\mathsf{Cov}_{\mathcal{L}_e,\mathcal{R}_e}$, between the two noise components $\mathsf{Cov}_{\mathcal{L}_n,\mathcal{R}_n}$, between the left exploitable component and the right noise component $\mathsf{Cov}_{\mathcal{L}_e,\mathcal{R}_n}$, and between the left noise component and the right exploitable component $\mathsf{Cov}_{\mathcal{L}_n,\mathcal{R}_e}$. Here

$$\begin{aligned} \mathsf{Cov}_{\mathcal{L}_e,\mathcal{R}_e} &= \frac{1}{Q}\sum_{u\in\chi}\left(b_uQ_u^{(l_2)} + (1-b_u)Q_u^{(l_1)}\right) \\ &\quad \left(\varphi_{u,k^{(l_1)}} - \tau^{(l_1)}\right)\left(\varphi_{u,k^{(l_2)}} - \tau^{(l_2)}\right) \\ &= \sum_{u\in\chi}\varepsilon_u\left(\varphi_{u,k^{(l_1)}} - \tau^{(l_1)}\right)\left(\varphi_{u,k^{(l_2)}} - \tau^{(l_2)}\right). \end{aligned} \tag{34}$$

Similarly, we can obtain the remaining three covariances as follows: $\mathsf{Cov}_{\mathcal{L}_e,\mathcal{R}_n} = \sum_{u\in\chi}\varepsilon_u\left(\mathbb{N}_u^{(l_2)} - \mathbb{N}^{(l_2)}\right)\left(\varphi_{u,k^{(l_1)}} - \tau^{(l_1)}\right)$, $\mathsf{Cov}_{\mathcal{L}_n,\mathcal{R}_e} = \sum_{u\in\chi}\varepsilon_u\left(\mathbb{N}_u^{(l_1)} - \mathbb{N}^{(l_1)}\right)\left(\varphi_{u,k^{(l_2)}} - \tau^{(l_2)}\right)$, $\mathsf{Cov}_{\mathcal{L}_n,\mathcal{R}_n}$

$= \sum_{u\in\chi} \varepsilon_u \left(\mathbb{N}_u^{(l_1)} - \mathbb{N}^{(l_1)}\right)\left(\mathbb{N}_u^{(l_2)} - \mathbb{N}^{(l_2)}\right)$.

5.3 CPCA^3: the Upper Security Bound of CECA

In this sub-section, we will prove that CPCA^3 provides the upper security bound of CECA through comparing their computation on Pearson correlation coefficients when considering the widely used evaluation tool success rate [25]. We consider the following two cases:

(i) Small noise. The attack requires very a few traces to guarantee the success, and CPCA^3 achieves significantly higher performance than CECA.

(ii) Large noise. The attack requires a large number of traces to guarantee the success, and CECA achieves performance close to (but still lower than) CPCA^3.

The first case can be verified easily by theory and the experiments given in Section 6, and here we discuss the second case. All possible values of the two plaintext bytes appear when Q is very large, thus $|\chi| = 256$. $Q_0^{(l_1)} \approx Q_1^{(l_1)} \approx \cdots \approx Q_{255}^{(l_1)} \approx \frac{Q}{|\chi|}$ and $Q_0^{(l_2)} \approx Q_1^{(l_2)} \approx \cdots \approx Q_{255}^{(l_2)} \approx \frac{Q}{|\chi|}$ in this case, thus $\varepsilon_0 \approx \varepsilon_1 \approx \cdots \approx \varepsilon_{255} \approx \frac{1}{|\chi|}$. The mean in CPCA^3 can be approximated as:

$$\tau^{(l)} \approx \frac{1}{|\chi|}\sum_{u\in\chi}\left(\varphi_{u,k^{(l)}} + \frac{|\chi|}{Q}\sum_{q=1}^{\frac{Q}{|\chi|}}\mathbb{N}_{u,q}^{(l)}\right), \tag{35}$$

and the noise component $\frac{|\chi|}{Q}\sum_{q=1}^{\frac{Q}{|\chi|}}\mathbb{N}_{u,q}^{(l)} \approx \mathbb{N}_u^{(l)} \approx 0$ since $\frac{Q}{|\chi|}$ is very large. Thus, the mean values of samples in CECA and CPCA given in Equations 27 and 35 are approximate.

For the variance of the samples in CECA and CPCA^3 after processing, we obtain:

$$\frac{|\chi|}{Q}\sum_{q=1}^{\frac{Q}{|\chi|}}\left(\mathbb{N}_{u,q}^{(l)}\right)^2 - \left(\frac{|\chi|}{Q}\sum_{q=1}^{\frac{Q}{|\chi|}}\mathbb{N}_{u,q}^{(l)}\right)^2 \approx \sigma^2. \tag{36}$$

Obviously, this is the noise variance of the normal distribution $\mathcal{N}\left(0,\sigma^2\right)$ under large $\frac{Q}{|\chi|}$. Thus, we obtain the square of the samples on the left side as:

$$\begin{aligned}[r_{\mathcal{L}}]_{cpca} \approx \frac{1}{|\chi|}\sum_{u\in\chi}\left(\varphi_{u,k^{(l_1)}} + \frac{|\chi|}{Q}\sum_{q=1}^{\frac{Q}{|\chi|}}\mathbb{N}_{u,q}^{(l_1)}\right)^2 \\ + \frac{\sum_{u\in\chi}(1-b_u)}{|\chi|}\sigma^2.\end{aligned} \tag{37}$$

It is noteworthy that, for very large power trace set, CPCA^3 averages samples on the left or right side with a probability of about $\frac{1}{2}$. In other words,

$$\sum_{u\in\chi}(1-b_u) \approx \sum_{u\in\chi} b_u \approx \frac{|\chi|}{2}. \tag{38}$$

The above Equation 37 can be further expressed as:

$$\begin{aligned}[r_{\mathcal{L}}]_{cpca} &\approx \frac{1}{|\chi|}\sum_{u\in\chi}\left(\varphi_{u,k^{(l_1)}}+\frac{|\chi|}{Q}\sum_{q=1}^{\frac{Q}{|\chi|}}\mathbb{N}_{u,q}^{(l_1)}\right)^2 \\ &\quad+\frac{\sigma^2}{|\chi|}\sum_{u\in\chi}(1-b_u) \\ &\approx \frac{1}{|\chi|}\sum_{u\in\chi}\left(\varphi_{u,k^{(l_1)}}+\frac{|\chi|}{Q}\sum_{q=1}^{\frac{Q}{|\chi|}}\mathbb{N}_{u,q}^{(l_1)}\right)^2+\frac{\sigma^2}{2}.\end{aligned} \tag{39}$$

The variance of samples on the left side satisfies $\left[\sigma_{\mathcal{L}}^2\right]_{cpca}=[r_{\mathcal{L}}]_{cpca}-\left[\tau^{(l_1)}\right]_{cpca}^2$, and the same conclusion can be drawn from the samples on the right side. The variance of samples in CPCA^3 and CECA satisfies:

$$\begin{cases}\left[\sigma_{\mathcal{L}}^2\right]_{cpca}\approx\left[\sigma_{\mathcal{L}}^2\right]_{ceca}+\frac{\sigma^2}{2} \\ \left[\sigma_{\mathcal{R}}^2\right]_{cpca}\approx\left[\sigma_{\mathcal{R}}^2\right]_{ceca}+\frac{\sigma^2}{2}.\end{cases} \tag{40}$$

according to Equations 27, 35 and 39. Finally, the relationship between denominators of Pearson correlation coefficients in CPCA^3 and CECA satisfies:

$$\begin{aligned}\sqrt{[\sigma_{\mathcal{L}}^2]_{cpca}\cdot[\sigma_{\mathcal{R}}^2]_{cpca}} &\approx \sqrt{[\sigma_{\mathcal{L}}^2]_{ceca}\cdot[\sigma_{\mathcal{R}}^2]_{ceca}}+\frac{\sigma^2}{2} \\ &\approx \left[\sigma^2\right]_{cpca} \\ &\approx \left[\sigma^2\right]_{ceca}+\frac{\sigma^2}{2}.\end{aligned} \tag{41}$$

The above Equation 41 indicates that there are only constant term differences in their denominators, which does not affect the success rate.

Here we further discuss the relationship between the numerators (i.e. covariance) of Pearson correlation coefficients in CECA and CPCA^3. First, the covariance of exploitable components of samples in CPCA^3 is:

$$\begin{aligned}\mathsf{Cov}_{\mathcal{L}_e,\mathcal{R}_e} &= \sum_{u\in\chi}\varepsilon_u\left(\varphi_{u,k^{(l_1)}}-\tau^{(l_1)}\right)\left(\varphi_{u,k^{(l_2)}}-\tau^{(l_2)}\right) \\ &\approx \frac{1}{|\chi|}\sum_{u\in\chi}\left(\varphi_{u,k^{(l_1)}}-\tau^{(l_1)}\right)\left(\varphi_{u,k^{(l_2)}}-\tau^{(l_2)}\right),\end{aligned} \tag{42}$$

which approximates the covariance of exploitable components of samples in CECA. Similarly, we can verify that other three covariances $\mathsf{Cov}_{\mathcal{L}_e,\mathcal{R}_n}$, $\mathsf{Cov}_{\mathcal{L}_n,\mathcal{R}_e}$ and $\mathsf{Cov}_{\mathcal{L}_n,\mathcal{R}_n}$ of CECA and CPCA are also approximate.

Above all, CPCA^3 achieves better performance than CECA in small noise scenarios, where a small number of encryptions can guarantee the success. Moreover, CECA achieves performance closer to CPCA^3 in larger noise scenarios, where a large number of encryptions should be performed to guarantee the success. In other words, our CPCA^3 always achieves higher success rate than CECA, i.e., it is the upper security bound of CECA.

6 Experimental Results

6.1 Simulated Experiments

Our first experiment is performed on the simulated power traces generated from Equation 1, and φ satisfies:

$$\varphi\left(t_q^{(l)} \oplus k^{*(l)}\right) = \mathsf{Hw}\left(\mathsf{Sbox}\left(t_q^{(l)} \oplus k^{*(l)}\right)\right). \tag{43}$$

Here $\mathsf{Hw}(\cdot)$ is the Hamming weight operator. We consider the standard deviation σ of noise from 1 to 10, which corresponds to Signal-to-Noise Ratio (SNR) from 0.02 to 2 since the variance of the Hamming weight distribution of the randomly encrypted plaintexts is about 2. Therefore, our simulated experiments consider both the small noise scenarios and very large noise scenarios. We compare our three CPCA schemes with the existing collision attacks named CECA given by Moradi et al. in [20], OCA given by Cezary et al. in [14] and the improved CECA given by Bruneau et al. in [7] except stochastic collision attack, since it achieves a performance much worse than them when only considering a single collision value. Each experiment is repeated 300 times, and the corresponding experimental results are given in Figure 5.

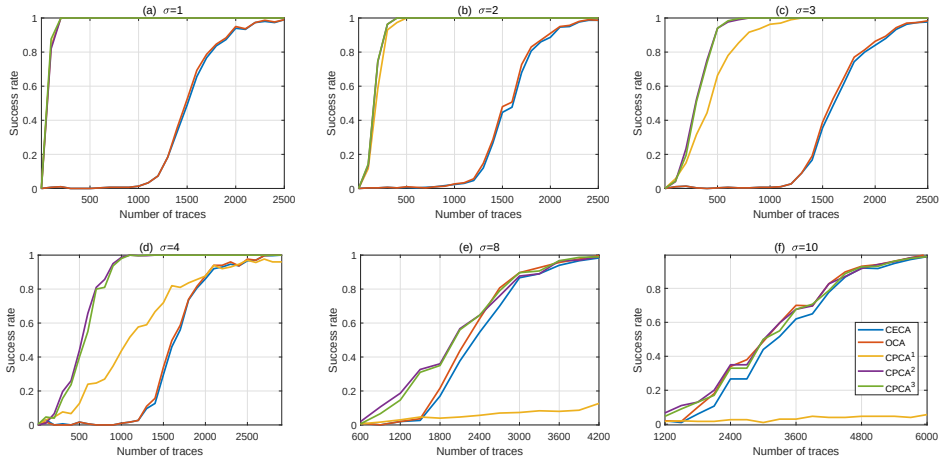

Figure 5: Success rates under different standard deviation σ-s of noise in simulated experiments.

Obviously, our three CPCA schemes obtain much better performance than the above existing schemes in the small noise scenarios. Their success rates are about 1.00 when 200, 500, 1000 traces are exploited under noise level $\sigma = 1, 2, 3$ respectively (as shown in Figure 5(a) $\sim$ Figure 5(c)). Due to the inefficient collision utilization strategies, the success rates of several existing schemes achieve a significant improvement only after 1000 traces. The OCA and the improved CECA achieve the same success rate in our experiments, thus we only give OCA in Figure 5. Its success rate is very close to CECA's. Compared to these existing collision attacks, 1000 traces can guarantee the successful attack in our three CPCAs. This fully illustrates that CPCA is especially suitable for the small noise scenarios.

With the increase of noise level, the success rate of CPCA^1 decreases significantly, and even significantly lower than the existing collision attacks in Figure 5(e) and Figure 5(f). This is because the noise aggravates the deterioration of correlation of the two columns of samples.

Therefore, CPCA^1 is only applicable to small noise scenarios. Taking advantage of the largest number of collision pairs, CPCA^2 achieves the best but not significantly higher performance compared with CPCA^1 and CPCA^3. With the increase of noise level, the performance of the existing collision attacks gradually approaches our CPCA^2 and CPCA^3. This fully illustrates that our CPCA^2 and CPCA^3 are still well suitable for noisy scenarios.

It is worth noting that, with the increasing number of traces, CECA achieves a success rate closer to CPCA^3, which is also reflected in Figure 5(e) and Figure 5(f). In fact, in Section 5.3, we have theoretically proved that the performance of CPCA^3 is the upper bound of CECA's. The OCA given by Cezary et al. in [14] and the improved CECA given by Bruneau et al. in [7] only obtain very similar performance to CECA, which fully shows the superiority of our CPCAs.

6.2 Experiments on an ATmega328p Micro-controller

Our second experiment is performed on an ATmega328p micro-controller with a clock operating frequency of 16 MHz. We implemented the unprotected AES-128 algorithm provided by [1]. We randomly encrypt 100, 000 plaintexts and use a WaveRunner 8104 oscilloscope to sample the power traces. The sampling rate is set to 1 GS/s. The leakages of S-boxes cannot be well aligned in this AES-128 implementation, thus we perform CPA to select the sample named Point-of-interest (POI) [9] with the highest Pearson correlation coefficient for the first two S-boxes to perform the subsequent experiments. Similar to Section 6.1, we compare our three CPCA schemes with the collision attacks named CECA given by Moradi et al. in [20], OCA given by Cezary et al. in [14] and the improved CECA given by Bruneau et al. in [7]. Each experiment is repeated 300 times, and the corresponding results are shown in Figure 6.

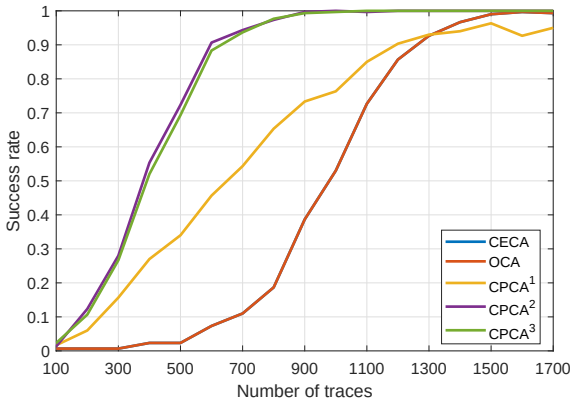

Figure 6: Success rates under different number of traces in experiments performed on an ATmega328p micro-controller.

The classic CECA, OCA and the improved CECA achieve the same success rate in Figure 6. This illustrates that it is difficult to significantly improve the attack performance by only changing the collision measurement method but without improving the utilization of collisions. Our CPCA^1 achieves higher success rates than the existing collision attacks when the sample size is small. However, with the increasing number of samples, this advantage disappears and CPCA^1 is finally defeated at the 1300-th sample. The similar phenomenon can be observed

from Figure 5(d). We draw a conclusion that CPCA^1 is vulnerable to noise interference. It is noteworthy that the improved CPCA^2 still obtains the best performance taking the advantage of high collision utilization. CPCA^3, which aims to reduce the computational load caused by the explosive growth of collision pairs in CPCA^2, performs very well and achieves success rates very close to CPCA^2. Their performance is significantly higher than CPCA^1 and the existing collision attacks.

6.3 Experiments on an AT89S52 Micro-controller

Our third experiment is performed on the AES-128 algorithm implemented on an AT89S52 micro-controller with a clock operating frequency of 12 MHz using assembly language. The shortest instructions take 12 clock cycles. We exploit an Picoscope 3000 to sample the leakage and the sampling rate is set to 125 MS/s. We acquire 120, 000 power traces and perform classic CPA to select the sample named POI with the highest Pearson correlation coefficient for the first S-box, then extract the well aligned sample for the second S-box. The collision attacks named CECA given by Moradi et al. in [20], OCA given by Cezary et al. in [14], the improved CECA given by Bruneau et al. in [7], and our three CPCAs are compared in this section. Each experiment is repeated 300 times, and the corresponding results are shown in Figure 7.

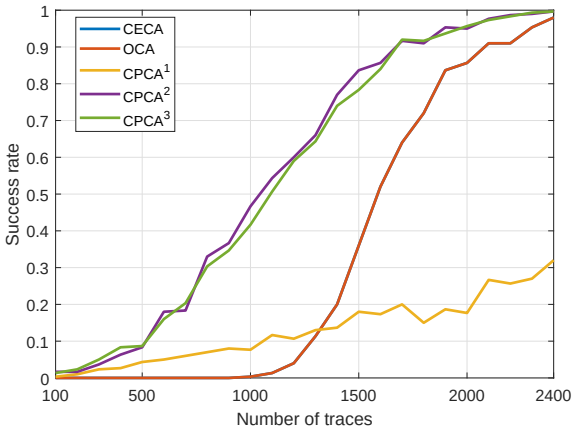

Figure 7: Success rates under different number of traces in experiments performed on an AT89S52 micro-controller.

Compared with the leakage of ATmega328p micro-controller introduced in Section 6.2, the SNR of samples acquired from AT89S52 micro-controller is lower, so that collision attack requires more power traces to guarantee the same success rates. However, we can draw similar conclusions from them. CECA given by Moradi et al., OCA given by Cezary et al. and the improved CECA given by Bruneau et al. achieve exactly the same success rates in the attacks. They achieve success rates lower than CPCA^1 when fewer than 1300 traces are exploited in each attack, and then higher than CPCA^1. This highlights the advantages of CPCA^1 in collision utilization. However, the success rates of the existing collision attacks are close to 1.00 at 2400 power traces, and the success rate of CPCA^1 is only about 0.3. This also illustrates that CPCA^1 is vulnerable to noise interference. Benefiting from the high collision utilization

rate, CPCA^2 achieves the best performance, while CPCA^3 achieves success rate very close to CPCA^2. Their performance is significantly higher than the other collision attacks.

7 Conclusions and Future Works

To improve the performance of the existing collision attacks against a given single collision value, we theoretically analyzed the relationship between the number of encryptions and the number of collisions in this paper, and proposed an efficient collision attack named CPCA for low noise scenarios to guarantee that the side with fewer samples in a pair of collision values was completely collision-paired. To further improve the utilization of collision information, we optimized our CPCA by maximizing the collision pairing and extended our CPCA to large noise scenarios. We finally optimized CPCA to a CPA-like distinguisher to reduce computation complexity. Our theoretical prove fully illustrated that the performance of our CPCA was the upper bound of CECA, and experimental results fully showed its superiority.

It is worth noting that although we have theoretically proved that CPCA provides the upper security bound of CECA, there is no good theoretical success rate estimation method for collision attacks such as CECA. This will be our work in the future. Moreover, CPCAs achieve significantly better performance in attacking each single collision value compared with the existing collision schemes. Their performance against several sub-keys simultaneously is also worthy of our expectation. Finally, as we mentioned in the introduction, several existing schemes combine a collision attack and a divide-and-conquer attack to enhance the key recovery ability. We will also carry out corresponding research and look forward to the "surprises" brought by our CPCAs.

References

[1] AVR-Crypto-Lib. https://github.com/DavyLandman/AESLib.

[2] DPA Contest. http://www.dpacontest.org/home/.

[3] Andrey Bogdanov. Improved Side-Channel Collision Attacks on AES. In *Selected Areas in Cryptography, 14th International Workshop, SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers*, volume 4876 of *Lecture Notes in Computer Science*, pages 84–95. Springer, 2007.

[4] Andrey Bogdanov. Multiple-Differential Side-Channel Collision Attacks on AES. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154, pages 30–44. Springer, 2008.

[5] Andrey Bogdanov and Ilya Kizhvatov. Beyond the Limits of DPA: Combined Side-Channel Collision Attacks. *IEEE Trans. Computers*, 61(8):1153–1164, 2012.

[6] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 16–29, 2004.

[7] Nicolas Bruneau, Claude Carlet, Sylvain Guilley, Annelie Heuser, Emmanuel Prouff, and Olivier Rioul. Stochastic Collision Attack. *IEEE Trans. Inf. Forensics Secur.*, 12(9):2090–2104, 2017.

[8] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, pages 13–28, 2002.

[9] François Durvaux and François-Xavier Standaert. From Improved Leakage Detection to the Detection of Points of Interests in Leakage Traces. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, pages 240–262, 2016.

[10] François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to Certify the Leakage of a Chip? In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 459–476. Springer, 2014.

[11] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-Channel Attacks on BLISS Lattice-Based Signatures: Exploiting Branch Tracing against strongSwan and Electromagnetic Emanations in Microcontrollers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1857–1874, 2017.

[12] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 444–461, 2014.

[13] Benoît Gérard and François-Xavier Standaert. Unified and Optimized Linear Collision Attacks and Their Application in a Non-profiled Setting. In *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, pages 175–192, 2012.

[14] Cezary Glowacz and Vincent Grosso. Optimal Collision Side-Channel Attacks. In *Smart Card Research and Advanced Applications - 18th International Conference, CARDIS 2019, Prague, Czech Republic, November 11-13, 2019, Revised Selected Papers*, volume 11833 of *Lecture Notes in Computer Science*, pages 126–140. Springer, 2019.

[15] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 104–113, 1996.

[16] Hervé Ledig, Frédéric Muller, and Frédéric Valette. Enhancing Collision Attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 176–190, 2004.

[17] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks - Revealing the Secrets of Smart Cards*. Springer, 2007.

[18] Ahmad Moghimi, Gorka Irazoqui, and Thomas Eisenbarth. CacheZoom: How SGX Amplifies the Power of Cache Attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 69–90, 2017.

[19] Amir Moradi, Sylvain Guilley, and Annelie Heuser. Detecting Hidden Leakages. In *Applied Cryptography and Network Security - 12th International Conference, ACNS 2014, Lausanne, Switzerland, June 10-13, 2014. Proceedings*, pages 324–342, 2014.

[20] Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-Enhanced Power Analysis Collision Attack. In *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, pages 125–139, 2010.

[21] Changhai Ou, Chengju Zhou, Siew-Kei Lam, and Guiyuan Jiang. Multiple-Differential Mechanism for Collision-Optimized Divide-and-Conquer Attacks. *IEEE Trans. Inf. Forensics Secur.*, 16:418–430, 2021.

[22] Changhai Ou, Xinping Zhou, Siew-Kei Lam, Chengju Zhou, and Fangxin Ning. Information Entropy-Based Leakage Profiling. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 40(6):1052–1062, 2021.

[23] Kai Schramm, Gregor Leander, Patrick Felke, and Christof Paar. A Collision-Attack on AES: Combining Side Channel- and Differential-Attack. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 163–175, 2004.

[24] Kai Schramm, Thomas J. Wollinger, and Christof Paar. A New Class of Collision Attacks and Its Application to DES. In *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*, pages 206–222. Springer, 2003.

[25] François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

[26] Danhui Wang, An Wang, and Xuexin Zheng. Fault-Tolerant Linear Collision Attack: A Combination with Correlation Power Analysis. In *Information Security Practice and Experience - 10th International Conference, ISPEC 2014, Fuzhou, China, May 5-8, 2014. Proceedings*, pages 232–246, 2014.

[27] Weijia Wang, Yu Yu, François-Xavier Standaert, Junrong Liu, Zheng Guo, and Dawu Gu. Ridge-Based DPA: Improvement of Differential Power Analysis For Nanoscale Chips. *IEEE Trans. Information Forensics and Security*, 13(5):1301–1316, 2018.

[28] Andreas Wiemers and Dominik Klein. Entropy Reduction for the Correlation-Enhanced Power Analysis Collision Attack. In *Advances in Information and Computer Security - 13th International Workshop on Security, IWSEC 2018, Sendai, Japan, September 3-5, 2018, Proceedings*, pages 51–67, 2018.