# Optimal Collision Side-Channel Attacks

Cezary Glowacz[1] and Vincent Grosso[2]

[1] Telekom Security
cezary.glowacz@t-systems.com
[2] CNRS/Laboratoire Hubert Curien, Université de Lyon
vincent.grosso@univ-st-etienne.fr

**Abstract.** Collision side-channel attacks are effective attacks against cryptographic implementations, however, optimality and efficiency of collision side-channel attacks is an open question. In this paper, we show that collision side-channel attacks can be derived using maximum likelihood principle when the distribution of the values of the leakage function is known. This allows us to exhibit the optimal collision side-channel attack and its efficient computation. Finally, we can compute an upper bound for the success rate of the optimal post-processing strategy, and we show that our method and the optimal strategy have success rates close to each other. Attackers can benefit from our method as we present an efficient collision side-channel attack. Evaluators can benefit from our method as we present a tight upper bound for the success rate of the optimal strategy.

## 1 Introduction

Since the late 90's and the first side-channel attacks by Kocher, various techniques of side-channel attacks have been proposed in the literature. Side-channel attacks are attacks against cryptographic implementations, the goal of such attacks is to link a physical property (e.g. power consumption, electromagnetic radiation) of the device to some secret information used in the implementation.

The optimal manner to exploit side-channel leakages is known in general [13]. It requires knowledge of the leakage function (estimated through profiling), then the maximum likelihood distinguisher is applied. However, the profiling step is not always possible, in some context like banking the attacker may not have access to an open device. Moreover, estimation of the leakage function can be a hard task [6] and model errors can be made. For these settings where profiling is difficult or impossible, it is interesting to look at optimal non-profiled attacks.

Among non-profiled attacks, collision attacks [12] are efficient side-channel attacks. The idea of collision side-channel attacks is that the same code processing the same data should have the same impact on monitored physical properties. This allows the attacker to detect when two sensitive values are equal. From this equality, the attacker extracts a relation between two different subkeys. Repeating this strategy for different couples of subkeys she ends with a system of equations that involve all subkeys with a degree of freedom of 1. Thus the set

of potential keys is reduced to a set of computationally enumerable candidates. In a noisy leakage scenario, detecting collisions may be not directly possible. To improve the success rate of collision side-channel attacks, Bogdanov introduces a collision attack across different executions of the AES and uses Euclidean distance as a score [1]. In [2] Bogdanov suggests using binary and ternary voting to improve the detection method. Moradi et al. suggest using the Pearson correlation coefficient to detect a collision with averaging to remove a part of the measurement noise [10]. Thus every trace can be exploited for the attack. In [4] Bruneau et al. derive a collision attack from stochastic side-channel attacks. They show that the scalar product score is more adapted to multi-collisions. However, they did not give any computationally efficient manner to maximize the score for the full key.

Once the scores of each collision between two subkeys are computed the attacker needs to select an independent collision relation to recover the key. Moreover, due to noise in measurements, some equations might be incorrect. Thus performing a key recovery attack can be tricky. Several algorithms have been proposed in the literature in order to extract information from the result of collision side-channel attacks. The proposed methods are based on heuristics, e.g. LDPC decoding for the solution of Gérard and Standaert [7] or branch-and-bound for the solution of Wiemers and Klein [14]. While both approaches improve the success rate of the collision attack the status of the optimality of these methods is not known, thus leaving space for potentially more efficient exploitation of side-channel collision attacks.

Having an efficient attack is interesting for attackers and evaluation labs. Security labs are also interested in computing security margins independent of the adversarial strategy. Thus finding the adversarial best strategy is important and computing a bound for its success rate is essential for a fair evaluation. Due to the dependence among the relations in collision side-channel attacks formulating the best strategy and evaluation of security margins were an open problem.

*Our contributions.* We derive optimal collision side-channel attacks when the attacker knows the distribution of the leakage function values, and the attacker has a balanced set-up of traces. Bruneau et al. [4] use other hypotheses: same leakage function $\varphi$ and a white Gaussian noise with the same variance for each of the measurements. Bogdanov and Moradi et al. solutions are based on statistical tools and are not derived using the maximum likelihood principle. From our derivation of collision attacks using the maximum likelihood principle we extract an evaluation of the distinguisher in an efficient manner. We show that the success rate of this manner is in practice close to that of the optimal evaluation of the distinguisher (the optimal evaluation is computationally unfeasible). This is achieved thanks to bounding the success rate of the optimal evaluation of the distinguishers. To the best of our knowledge, it is the first time an upper bound for the first order success rate of optimal collision side-channel attacks is exhibited. We compare our method to existing techniques and show that our method achieved better performance than previous methods, and that its success

rate is close to the upper bound of the success rate of optimal collision side-channel attack. We use the maximum likelihood principle for the score derivation. The experimental results show that our method for maximization of the sum of the scores is close in term of success rate to the optimal strategy (bounded thanks to the upper bound). In our case, we measure the optimality in terms of achieving the same success rate as the optimal strategy (using the exact ML distinguisher derived according to the knowledge of the probability distribution of the leakage function values).

## 2 Background and model notations

### 2.1 Collision side-channel attacks

Collision side-channel attacks were invented to exploit the similarity between leakages of similar computations over similar data values. Collision side-channel attacks do not require profiling of the leakage or a hypothesis of the leakage model. This is one of the main differences between collision side-channel attacks and other side-channel attacks such as template attacks [5] or correlation attacks [3]. Collision side-channel attacks have been introduced as attacks against block cipher implementations in [12].

As collision attacks aim at detecting repeated code execution with the same data we target in this paper block cipher implementations that reuse the same instance of the S-box, like in several reference implementations of Present or of AES. We denote by $n$ the input size of the S-box (e.g. $n = 8$ for AES). We denote by $L$ the number of S-box calls in one round (e.g. $L = 16$ for AES-128).

For all $l \in \{1, \ldots, L\}$ we denote by $k^{*(l)}$ the $l$-th secret key byte and by $k^{(l)}$ any possible $l$-th key byte hypothesis. We denote by $\boldsymbol{k} = (k^{(1)}, \ldots, k^{(L)})$ the full key. The $l$-th byte of the plaintext corresponding to the $q$-th query is denoted by $t_q^{(l)}$ and the associated leakage is denoted by $x_q^{(l)}$. $\boldsymbol{x}^{(.)}$ is the matrix with $q - th$ row corresponding to the $L$-variate leakage $x_q^{(1)}, ..., x_q^{(L)}$.

We assume an identical, but unknown, leakage model for all $l \in \{1, \ldots, L\}$. I.e.

$$x_q^{(l)} = \varphi(t_q^{(l)} \oplus k^{*(l)}) + N,$$

where the noise $N$ is independent among $l$ and $q$ and $\varphi$ is a deterministic leakage function.

The goal of collision side-channel attacks is to find links among the different key bytes $k^{(l)}$. The main idea is to detect when $\varphi(t_{q_1}^{(l_1)} \oplus k^{*(l_1)}) = \varphi(t_{q_2}^{(l_2)} \oplus k^{*(l_2)})$ for $l_1, l_2 \in \{1, \ldots, L\}, l_1 \neq l_2$ and some know plaintext byte $t_{q_1}^{(l_1)}, t_{q_2}^{(l_2)}$.

In this paper, we consider only the case where we have a number of measurements that is a multiple of $2^n$, and for each S-box calculation, we have observed the same number of traces for each value of the plaintext. This balanced setup allows to remove the bias of the plaintext distribution and it can be easily implemented using shuffling of batches. Hence, after performing averaging over the

traces $x_q^{(l)}$ with the same plaintext values $\boldsymbol{x}^{(\cdot)}$ becomes a matrix of real numbers of dimension $2^n \times L$, where the $i$-th row corresponds to the leakage of the plaintext $i-1$.

In the rest of the paper we consider leakage functions that are partially unknown, i.e. the leakage function values are random variables and follow some plausible probability distribution. Without knowing this distribution, we cannot figure out an optimal distinguisher using maximum likelihood principle. For the experiment part, we also consider a more classical case where the leakage function is the Hamming weight.

### 2.2 Stochastic and correlation enhanced collision attacks

Bruneau et al. combine flavours of collision and of stochastic side-channel attacks [4]. Contrary to previous formulations, Bruneau et al. derive the attack rather than inventing it. The derivation is based on maximizing the likelihood function stated for the full key, given the measured leakages under the assumption of the same leakage function $\varphi$ for each of the executions of the S-box and of the Gaussian noise having the mean 0 and the same variance for each of the measurements.

Stochastic differential side-channel attacks [11] were introduced in order to optimize the efficiency of DPA. The key idea of stochastic DPA is to approximate the leakage function $\varphi$ within a suitable vector subspace with a relatively "small" basis to be efficient.

To use a stochastic approach in the collision context Bruneau et al. consider the unknown leakage function $\varphi$ as an additional part of the secret. Thus the optimization problem, i.e. maximizing the likelihood function, is not only on the key value $\boldsymbol{k}$, but also on the leakage function. The stochastic approach for the representation of the leakage function $\varphi$ can be shown to be equivalent to replacing the leakage function values in the likelihood function by their estimates calculated for each key $\boldsymbol{k}$ as the arithmetic mean $x^{(l)}_{q\oplus k^{(l)}}$ of the measured leakages. Using these estimates maximizes the likelihood function values,[3] as it is also the case when using the stochastic approach utilizing the full basis for the representation of the leakage function $\varphi$. Finally, Bruneau et al. obtain the following distinguisher:

$$\mathcal{D}_{sto.coll} = \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{u\in\mathbb{F}_2^n} \frac{\left(\sum_{l=1}^{L}\sum_{q=1\ldots Q|t_q\oplus k^{(l)}=u} x_q^{(l)}\right)^2}{\sum_{l=1}^{L}\sum_{q=1\ldots Q|t_q\oplus k^{(l)}=u} 1}.$$

[3] To see this we rewrite the $\mathcal{D}_{opt}$ from the equation (2) [4] in the balanced setup as

$$\mathcal{D}_{opt} = \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{q=0}^{2^n-1} \left( -\left(\varphi(t_q^{(l)}) - \frac{1}{L}\sum_{l=1}^{L} x^{(l)}_{q\oplus k^{(l)}}\right)^2 + \frac{2}{L^2}\sum_{l_1=1}^{L}\sum_{l_2=l_1+1}^{L} x^{(l_1)}_{q\oplus k^{(l_1)}} \times x^{(l_2)}_{q\oplus k^{(l_2)}} \right).$$

As the distinguisher is computed over $L$ key bytes, the formula can be maximized over all keys only for small values $L$ (e.g. up to 5).

When the data set is balanced and averaging of traces is performed we can rewrite the distinguisher as a sum of scalar products between rows of the matrix $\boldsymbol{x}^{(.)}$ (re-indexed by the key). As a matter of fact, we have $\forall u \in \mathbb{F}_2^n, \forall l \in \{1,\dots,L\} \sum_{q=0\dots 2^n-1 | q\oplus k^{(l)}=u} 1 = L$, thus:

$$\begin{aligned}
\mathcal{D}_{sto.coll.bal} &= \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{u\in\mathbb{F}_2^n} \left( \sum_{l=1}^{L} \sum_{\substack{q=0\dots 2^n-1 \\ q\oplus k^{(l)}=u}} x_q^{(l)} \right)^2 \\
&= \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{u\in\mathbb{F}_2^n} \left( \sum_{l=1}^{L} x_{u\oplus k^{(l)}}^{(l)} \right)^2 \\
&= \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{u\in\mathbb{F}_2^n} \sum_{l=1}^{L} \left( x_{u\oplus k^{(l)}}^{(l)} \right)^2 + 2 \sum_{l_1=1}^{L} \sum_{l_2=l_1+1}^{L} \left( x_{u\oplus k^{(l_1)}}^{(l_1)} \times x_{u\oplus k^{(l_2)}}^{(l_2)} \right) \\
&= \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{u\in\mathbb{F}_2^n} \sum_{l_1=1}^{L} \sum_{l_2=l_1+1}^{L} \left( x_{u\oplus k^{(l_1)}}^{(l_1)} \times x_{u\oplus k^{(l_2)}}^{(l_2)} \right),
\end{aligned}$$

since $\sum_{u\in\mathbb{F}_2^n} \sum_{l=1}^{L} \left( x_{u\oplus k^{(l)}}^{(l)} \right)^2$ is constant for every key.

We can notice that $\forall i \in \mathbb{F}_2^n$,

$$\begin{aligned}
&\sum_{u\in\mathbb{F}_2^n} \sum_{l=1}^{L} \sum_{l_2=l_1+1}^{L} \left( x_{u\oplus k^{(l_1)}}^{(l_1)} \times x_{u\oplus k^{(l_2)}}^{(l_2)} \right) \\
&= \sum_{u\in\mathbb{F}_2^n} \sum_{l=1}^{L} \sum_{l_2=l_1+1}^{L} \left( x_{u\oplus k^{(l_1)}\oplus i}^{(l_1)} \times x_{u\oplus k^{(l_2)}\oplus i}^{(l_2)} \right),
\end{aligned}$$

thus the keys are equivalent up to a byte $i$ xor on every key byte, i.e.

$$\left(k^{(1)},\dots,k^{(L)}\right) \sim \left(k^{(1)}\oplus i,\dots,k^{(L)}\oplus i\right).$$

Moradi et al. proposed correlation-enhanced collision attack in [10]. They average traces to reduce the impact of randomness (noise in measurement). Then, they use the correlation between every two rows of the matrix $\boldsymbol{x}^{(.)}$ and for every re-indexing of the coefficients due to the differential value of any two sub-keys. To recover the full differential of the sub-keys of the key ad hoc solutions were proposed. E.g. extract a system of independent equations [10], perform a branch-and-bound on the sum of correlation coefficients [14], or use an adapted decoding technique [7].

None of these techniques based on correlation enhanced collision attack address the optimality of the approach, leaving the question about it open.

Actually, for any two key bytes we can link scalar product and correlation coefficient as:

$$\rho_{k^{(l_1)},k^{(l_2)}}\left(x^{(l_1)},x^{(l_2)}\right) = \frac{2^n\sum_{i=0}^{2^n-1}x_{i\oplus k^{(l_2)}}^{(l_1)}\times x_{i\oplus k^{(l_2)}}^{(l_2)}-\sum_{i=0}^{2^n-1}x_i^{(l_1)}\sum_{i=0}^{2^n-1}x_i^{(l_2)}}{\sqrt{2^n\sum_{i=0}^{2^n-1}\left(x_i^{(l_1)}\right)^2-\left(\sum_{i=0}^{2^n-1}x_i^{(l_1)}\right)^2}\sqrt{2^n\sum_{i=0}^{2^n-1}\left(x_i^{(l_2)}\right)^2-\left(\sum_{i=0}^{2^n-1}x_i^{(l_2)}\right)^2}}.$$

It can be seen from the above formula that for the balanced setup the couple $k^{(l_1)}, k^{(l_2)}$ that maximizes the correlation coefficient is the same that maximizes the scalar product. However, maximizing the sum of correlation coefficients or of scalar products might not give the same relation between key bytes. The reason for this is a statistical fluctuations of the factors used as weights (see the denominator in the above formula) when going from the sum of scalar products to the sum of correlation coefficients.

## 3 Optimal distinguishers for random leakage functions

With reference to the equation (4) [4] and to the previously introduced notations the maximum likelihood (ML) distinguisher can be written as:

$$\mathcal{D}_{opt}=\underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}}\prod_{q=0}^{2^n-1}\prod_{l=1}^{L}f_{\sigma^2}\left(x_q^{(l)}-\varphi\left(t_q^{(l)}\oplus k^{(l)}\right)\right),$$

where $f_{\sigma^2}$ denotes Gaussian distribution with the mean value 0 and the standard deviation $\sigma$. Bruneau et al. maximize $D_{opt}$ also over the leakage function values. This approach is not sufficient for obtaining a provably optimal, i.e. one that maximizes the likelihood of the key given the measured leakage, distinguisher for the key. However, in some practical situations the attacker might have some a priori knowledge about the leakage function and using it she may try to derive an optimal distinguisher, e.g. by considering each of the leakage function values $\varphi(x)$ as random variables with some guessed distribution. Using such distinguisher maximizes the average success probability when repeating attacks while each time the leakage function values are selected according to the assumed distribution. In particular it is also expected that the attack succeeds on some actual leakage function with higher success probability than in a case of using the distinguisher $D_{sto.coll}$. This can be explained by the fact that the actual leakage function might be a kind of a typical leakage function with respect to the assumed distribution of the leakage function values and with respect to the success probability of the derived distinguisher.

We verified in case of two 8 bit wide S-boxes the higher success rate of 0.90 when using the distinguisher derived (see below) using the knowledge of the distribution of leakage function values as compared to the success rate of 0.50

when using the $D_{sto.coll.bal}$ distinguisher. The following describes the used distribution. Each leakage function $\varphi$ is created randomly according to the following rule: for each $u \in \{0, \ldots, 255\}$ assign to $\varphi(u)$ a value $v$ selected randomly from a distribution given by the following histogram:

$$H_{ex} = \left\{ \left(0, \frac{246}{256}\right), \left(1, \frac{1}{256}\right), \left(2, \frac{2}{256}\right), \left(3, \frac{3}{256}\right), \left(4, \frac{4}{256}\right) \right\},$$

where $(v, p)$ means that the value $v$ has the probability $p$ of being selected. The higher success rate was also verified for some fixed leakage functions with values drawn from that distribution. Note that the example is given only to show that the $D_{sto.coll.bal}$ or equivalently $D_{sto.coll}$ distinguisher might be not optimal given additional knowledge of the distribution of the leakage function values, and thus to motivate further investigations. No other claims are made at that point.

The optimal distinguisher derived under known distribution $p$ of leakage function values $\varphi$ is given by:[4]

$$D_{opt.fun.p} = \underset{\boldsymbol{k} \in (\mathbb{F}_2^n)^L}{\operatorname{argmax}} \prod_{q=0}^{2^n-1} \int \left( \prod_{l=1}^{L} f_{\sigma^2}\left(x_{q \oplus k^{(l)}}^{(l)} - \varphi\right) \right) dp(\varphi),^{5}$$

where $\int \alpha(\varphi) dp(\varphi)$ means the expectation value of $\alpha(\varphi)$ given the distribution density $dp(\varphi)$ of the leakage function values.

In the example above the integral was just a sum over the values $v \in \{0, \ldots, 4\}$ and $dp(v)$ was set to the probability of the occurrence of each of the value $v$.

Of special practical interest is the case of Hamming weight leakages. Even without knowing the exact leakage model, it is reasonable in many situations to assume a Hamming weight leakage, and therefore the distribution of the leakage function values is binomial, e.g. it is given by the following histogram:

$$H_{bin.4} = \left\{ \left(0, \frac{1}{16}\right), \left(m, \frac{4}{16}\right), \left(m \times 2, \frac{6}{16}\right), \left(m \times 3, \frac{4}{16}\right), \left(m \times 4, \frac{1}{16}\right) \right\},$$

in case of a 4 bit wide S-box, where the $m$ is a parameter of the distribution.

While it is straightforward to write an exact formula for the optimal distinguisher $D_{opt.fun.binomial}$ in that case, the parameters $m$ of the leakage and the standard deviation $\sigma$ of the noise are still unknown. However, later on we will use

[4] The derivation is based on the following equation with statistically independent $K$ and $\phi$

$$P(K = k | X = x) = \sum_{\varphi} \frac{P\left(X = x | (K = k, \phi = \varphi)\right) \times P(K = k) \times P(\phi = \varphi)}{P(X = x)}.$$

Without knowing the distribution $P(\phi)$ of the leakage function values we cannot figure out an optimal distinguisher using the maximum likelihood principle.

[5] Any constant mask which is applied to each S-box and which does not change the distribution $p$ has no effect on the distinguisher $D_{opt.fun.p}$.

such distinguisher derived with known values of $m$ and $\sigma$ as a benchmark when comparing the success rate of the related $D_{opt.fun.gauss}$ distinguisher derived for leakage function values distributed according to Gaussian distribution. We expect similar success rates for both $D_{opt.fun.binomial}$ and $D_{opt.fun.gauss}$ because Gaussian distribution is an approximation of the binomial distribution.

The integration in the formula for $D_{opt.fun.p}$ can be performed with the standard deviation $\sigma$ of the noise and $dp$ taken as a density of Gaussian distribution with the mean $m_\varphi$ and the standard deviation $\sigma_\varphi$. In addition to knowing that the leakage function values are drawn from a Gaussian distribution we also require a balanced set-up of traces.

The result in that case is

$$
\begin{aligned}
&D_{opt.fun.gauss} \\
&= \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \prod_{q=0}^{2^n-1} \left( \int \left( \prod_{l=1}^{L} f_{\sigma^2}(x^{(l)}_{q\oplus k^{(l)}} - \varphi) \right) \times f_{\sigma_\varphi^2}(\varphi - m_\varphi) d\varphi \right) \\
&= \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \prod_{q=0}^{2^n-1} \left( e^{\frac{(\sigma_\varphi^2 \times \sum_{l=1}^{L} x^{(l)}_{q\oplus k^{(l)}} + \sigma^2 \times m_\varphi)^2}{2\times\sigma^2\times\sigma_\varphi^2\times(\sigma^2+L\times\sigma_\varphi^2)} - \frac{m_\varphi^2}{2\times\sigma_\varphi^2} - \frac{\sum_{l=1}^{L}(x^{(l)}_{q\oplus k^{(l)}})^2}{2\times\sigma^2}} \right. \\
&\left. \times \int e^{-\frac{\sigma^2+L\times\sigma_\varphi^2}{2\times\sigma^2\times\sigma_\varphi^2} \times (\varphi - \frac{\sigma_\varphi^2 \times \sum_{l=1}^{L} x^{(l)}_{q\oplus k^{(l)}} + \sigma^2 \times m_\varphi}{\sigma^2+L\times\sigma_\varphi^2})^2} d\varphi \right) \\
&= \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{q=0}^{2^n-1} \left( \sigma_\varphi^2 \times \sum_{l=1}^{L} x^{(l)}_{q\oplus k^{(l)}} + \sigma^2 \times m_\varphi \right)^2 \\
&= \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{q=0}^{2^n-1} \sum_{l_1=1}^{L} \sum_{l_2=l_1+1}^{L} \left( x^{(l_1)}_{q\oplus k^{(l_1)}} \times x^{(l_2)}_{q\oplus k^{(l_2)}} \right).
\end{aligned}
$$

Remarkably, this special result is independent of the parameters, i.e. of the standard deviation $\sigma$ of the noise, of the mean $m_\varphi$ and of the standard deviation $\sigma_\varphi$ of the leakage function values. It also shows the optimality in terms of maximum likelihood of the $D_{sto.coll.bal}$ distinguisher in case of Gaussian distributed leakage function values.

In a similar way an optimal distinguisher for profiled side channel attacks can be derived by jointly applying the maximum likelihood principle to the profiling and attack steps. The result is

$$
\begin{aligned}
&D_{opt.fun.gauss.profiled} = \underset{\boldsymbol{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \\
&\left( \mathcal{N}_x \times \sum_{q=0}^{2^n-1} \sum_{l_1=1}^{L} \sum_{l_2=1}^{L} \left( x^{(l_1)}_{q\oplus k^{(l_1)}} \times x^{(l_2)}_{q\oplus k^{(l_2)}} \right) + \mathcal{N}_y \times \sum_{q=0}^{2^n-1} \sum_{l=1}^{L} \left( x^{(l)}_{q\oplus k^{(l)}} \times y_q \right) \right),
\end{aligned}
$$

where $n$, $L$, and $x_q^{(l)}$ are as defined before, $y_q$ denotes the estimated mean of the leakage function value for the input $q$, and $\mathcal{N}_x$ and $\mathcal{N}_y$ denote the number of

traces [6] used per input data value and S-Box in the attack and per input data value in the profiling phase. The state of the art profiled side channel attacks use the right term in the sum as a distinguisher which is optimal if $\mathcal{N}_y$ is significantly higher than $\mathcal{N}_x$.

## 4 Optimal evaluation of distinguishers

The $D_{opt.fun.p}$ distinguishers require to maximize over all $\boldsymbol{k} \in (\mathbb{F}_2^n)^L$. Unfortunately, the triple sum has no structure (local maximum does not lead to global maximum), and to find the $\boldsymbol{k}$ maximizing the sums all cases need to be computed. Thus the optimal solution to recover the key using an optimal collision side-channel attack requires to compute all of the $2^{(n-1)L}$ (e.g. $2^{120}$ in the case of AES-128) values.[7] We present here an algorithm that aims to find the maximizing $\boldsymbol{k}$ using random space exploration by looking only at a small number of candidate keys $\boldsymbol{k}$.

### 4.1 Random space exploration

The random space exploration algorithm is described in Algorithm 1.[8]

The algorithm returns a key candidate that maximizes the sum of scalar products over the small set of key candidates we explore. Hence, the algorithm tries to find the maximizing key $\boldsymbol{k}$ of the $D_{opt.fun.gauss}$ distinguisher. We note that the term $\sum_{j=1}^{s-1} s(l_j, l_s, \delta \oplus k^{(l_j)})$ in step 14: results in matching the leakage of S-box $l_s$ with a kind of template given by the sum of the leakages of the S-boxes $l_1$ to $l_{(s-1)}$. If the guess for the keys $k^{(l_1)}$ to $k^{(l_{s-1})}$ is correct, that template converges for larger values of $s$ to the true leakage function value and the chance to recover the correct key $k_s$ increases with greater $s$. Actually this observation could already have been the starting point for designing the algorithm. Another

[6] The ratio of the two numbers determines the ratio of the noise variances in the profiling and attack phases, which is actually used in the derivation.

[7] The minus 1 comes from the equivalence of the keys when xor-ing any fixed value with each subkey.

[8] The random space exploration algorithm can be seen as a repeated execution of the Wiemers' and Klein's algorithm variant 1 with $W = 1$, the details of the algorithm are given in [14]. While the algorithm of Wiemers and Klein was designed for entropy reduction of collision attacks, the target of the random space exploration algorithm was to enable the investigation of the limits of success rates for collision attacks. To sum up, the differences between the Wiemers' and Klein's algorithm and the random space exploration algorithm are:

- the repetition of the execution of variant 1 with $W = 1$ instead of one run with $W > 1$,
- randomized order of S-boxes on each run instead of the fixed order,
- the output of only one candidate instead of a list of $W > 1$ candidates,
- the use of $D_{opt.fun.gauss}$ distinguisher instead of a sum of correlation coefficients.

design idea was based on the observation, that when having a set of different pairs of S-boxes there is a good chance to have one pair for which the correct key can be found. This pair then results in a better template for matching the leakage of the third S-box, and so on.

The cost of the proposed algorithm is modest in terms of memory, we just need to store the maximum key. In terms of time, the algorithm is also efficient and it has a running time $O(L \times 2^n \times max_tries)$.

**Algorithm 1** random space exploration

1: **Input**: The $\frac{L(L-1)}{2}$ lists of $2^n$ scalar products $s(l_1, l_2, \delta) = \sum_{q=0}^{2^n-1} x_q^{(l_1)} \times x_{q\oplus\delta}^{(l_2)}$
2: **Output**: A key candidate $\boldsymbol{k}$
3: **Notation**: $\xleftarrow{\$}$ means we pick a value in the set on the right randomly following a uniform distribution
4: $Max = -\infty$
5: **for** $1 \leq try \leq max_tries$ **do**
6: $l_1 \xleftarrow{\$} \{1, \ldots, L\}$
7: $ktmp^{(l_1)} = 0$
8: $l_2 \xleftarrow{\$} \{1, \ldots, L\} \backslash \{l_1\}$
9: $Sum = max_\delta(s(l_1, l_2, \delta))$
10: $ktmp^{(l_2)} = \text{argmax}_\delta(s(l_1, l_2, \delta))$
11: **for** $3 \leq s \leq L$ **do**
12: $l_s \xleftarrow{\$} \{1, \ldots, L\} \backslash \{l_1, \ldots, l_{s-1}\}$
13: **for** $0 \leq \delta \leq 2^n$ **do**
14: $Current(\delta) = Sum + \sum_{j=1}^{s-1} s(l_j, l_s, \delta \oplus ktmp^{(l_j)})$
15: **end for**
16: $Sum = max_\delta(Current(\delta))$
17: $ktmp^{(l_s)} = \text{argmax}_\delta(Current(\delta))$
18: **end for**
19: **if** $Sum > Max$ **then**
20: $\boldsymbol{k} = \boldsymbol{ktmp}$
21: $Max = Sum$
22: **end if**
23: **end for**
24: **return** $\boldsymbol{k}$

We also use a modified version of the Algorithm 1 for the evaluation of the $D_{opt.fun.binomial}$ distinguisher, i.e. for finding its maximizing key $\boldsymbol{k}$. First, the algorithm receives as input the full matrix $\boldsymbol{x}^{(\cdot)}$, and the modification consists also of replacing in line 9:

$$Sum = max_\delta \left(s(l_1, l_2, \delta)\right)$$

by

$$Sum = max_\delta \left( \sum_{q=1}^{2^n-1} log \left( \int f_{\sigma^2} \left( x^{(l_1)}_{q \oplus k^{(l_1)}} - \varphi \right) \times f_{\sigma^2} \left( x^{(l_2)}_{q \oplus \delta} - \varphi \right) dp(\varphi) \right) \right)$$

and in line 14:

$$Current(\delta) = Sum + \sum_{j=1}^{s-1} s \left( l_j, l_s, \delta \oplus k^{(l_j)} \right)$$

by

$$Current(\delta) = \sum_{q=1}^{2^n-1} log \left( \int f_{\sigma^2} \left( x^{(l_s)}_{q \oplus \delta} - \varphi \right) \times \prod_{j=1}^{s-1} f_{\sigma^2} \left( x^{(l_j)}_{q \oplus k^{(l_j)}} - \varphi \right) dp(\varphi) \right),$$

where $\int \alpha(\varphi) dp(\varphi)$ means the expectation value of $\alpha(\varphi)$ given the distribution density $p(\varphi)$ of the variable $\varphi$. Here the distribution density $p(\varphi)$ is the binomial distribution of the $n$-bit Hamming weights and the integral is effectively a sum (see also section 3).

### 4.2 Upper bound for the success rate

Interestingly, in an evaluation setup the random space exploration can also be used to find an upper bound for the success rate of the optimal exploration, i.e. the one that recovers the key by computing the maximum of the distinguisher over all $2^{(n-1)L}$ key candidates. As a matter of fact, in the evaluation setup the correct key $\boldsymbol{k^*}$ is known, thus the score

$$S_{\boldsymbol{k}^*} = \sum_{u \in \mathbb{F}_2^n} \sum_{l=1}^{L} \sum_{l_2=l_1+1}^{L} \left( x^{(l_1)}_{u \oplus k^{*(l_1)}} \times x^{(l_2)}_{u \oplus k^{*(l_2)}} \right)$$

is also known. According to the $Max$ value (see line 21:, Algorithm 1) which we find in random space after reaching line 23: we have two cases:

1. $S_{\boldsymbol{k}^*} < Max$, in that case, we know that the optimal exploration, and our random space exploration, will fail. They both output a candidate key that has a higher score than the actual key.
2. $S_{\boldsymbol{k}^*} \geq Max$, in that case, the optimal exploration might find the right key.

Thus, in the evaluation setup, we can count the number of times the case 2 happens and this way obtain an upper bound for the success rate of the optimal exploration. This upper bound can be computed with almost no overhead, we just need to additionally return the value $Max$ in Algorithm 1. To the best of our knowledge, it is the first time an upper bound for the first-order success rate of optimal collision side-channel attack can be computed.

The value of the parameter $max_tries$ (see line 5:, Algorithm 1) of the attack plays a role in the attack phase and also in the evaluation step. Higher values of $max_tries$ result in higher success rates of the attack and in lower calculated upper bound values.

We will also calculate the upper bound for the success rate of the modified Algorithm 1 for the evaluation of the $D_{opt.fun.binomial}$ distinguisher using a method similar to the method described above.

## 5 Simulation results

We present the upper bounds for success rates of collision side-channel attacks, and we compare our method to these upper bounds and to the previous methods in terms of success rate. We choose to evaluate the method using simulation to highlight the differences between the methods without being blurred by slight modifications of the leakage function according to key byte used [7]. For collision side-channel attacks we compare:

- our method presented in Algorithm 1 (labelled 'Prop.') using the distinguisher $D_{opt.fun.gauss}$ (labelled 'scalar') and using the $D_{opt.fun.binomial}$ distinguisher (labelled 'binomial') with $max_tries = 128$ (labelled '128 tries') and with $max_tries = 2^{13}$ (labelled '$2^{13}$ tries'), and upper bounds (denoted 'UB') computed along the success rates;
- variant 1 of Wiemers' and Klein's algorithm [14] (labelled 'Wiemers') with $W = 128$ [9] and using the sum of correlation coefficients (labelled 'corre.') and its modification using the sum of scalar products (labelled 'scalar'). Among all solutions in $B_{16}$ we kept only the maximum to have only one solution to test as for the other solutions;[10]
- Gérard's and Standaert's solution [7] (labelled 'Best Gérard') with normalized correlation, we use six loops of message passing, that is greater that two times the graph's diameter.[11]

For a reference, we also plot template attacks (labelled 'Template'), which in case of the simulation are optimal profiled attacks.

We consider attacks on 16 key bytes, i.e. $L = 16$ and $n = 8$, similar to the AES case. We assume that the attacker has an access to a balanced set of traces.

[9] Algorithm 1 with $max_tries = 128$ and the variant 1 of Wiemers' and Klein's algorithm with $W = 128$ visit almost the same number of nodes of the search tree/trees. These settings allow meaningful performance comparison of the two algorithms.

[10] In our experiments using only the highest ranked solution or testing of all solutions has a small impact on the success rate of the method.

[11] In our experiments this setting provides the highest success rate compared to the other methods described in the paper of Gérard and Standaert, i.e. Euclidean distance vs. correlation coefficient and normalization vs. Bayesian extension. The Bayesian extension is a boost for score combination, but its derivation uses Fisher transform that is an asymptotic tool. Thus, the Bayesian extension can be counterproductive for attacks which use a small number of traces like $2^8$.

She observes each plaintext byte the same number of times, thanks to averaging she can just use $2^8$ plaintexts per S-box. We utilize the balanced setup, and instead of varying the number of traces, we increase or decrease the variance of the white Gaussian noise in our simulations.

For the leakage function we consider two cases. The first case is the setting corresponding to derivation in Section 3, i.e. the distribution of the leakage function values is 8 bit binomial (labelled 'rand. leak') [12]. As the second case we consider Hamming weight (HW) leakage of the output of the AES S-box (labelled 'HW leak').

We compute all success rates based on 2500 experiments. This results in a value of standard deviation of estimated success rates less then 0.01.

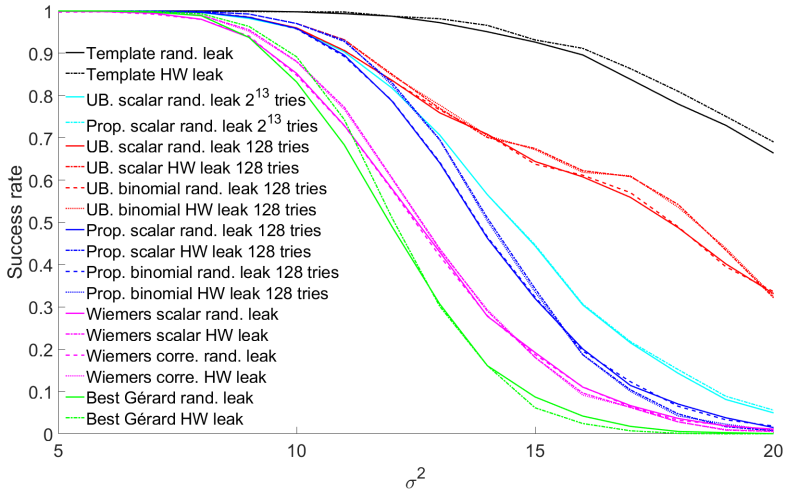



**Fig. 1.** Upper bounds and success rates of different techniques.

In figure 1, we plot results for the proposed method given in Algorithm 1 using the distinguishers $D_{opt.fun.gauss}$ and $D_{opt.fun.binomial}$ and for the previous methods applied to the same set of data. We can make several observations from the figure.

– For success rates greater than 0.90 the upper bound and the success rate of the Algorithm 1 are close to each other for small value of $max_tries$, i.e. 128. For example, with $\sigma^2 = 11$ and random leakage function values, the success rates are 0.9064 for the upper bound and 0.8956 for Algorithm 1 when

[12] In more details, for each experiment we draw a new leakage function $\varphi$ randomly according to the following rule: for each $u \in \{0, ..., 255\}$ assign to $\varphi(u)$ a value selected randomly according to the binomial distribution of 8-bit Hamming weights.

using the $D_{opt.fun.binomial}$ distinguisher, and the success rates are 0.9068 for the upper bound and 0.8924 for Algorithm 1 when using the $D_{opt.fun.gauss}$ distinguisher. In the same scenario for Gérard's and Standaert's solution the success rate is 0.6832, and for Wiemers' and Klein's solution the success rates are: 0.7284 using the sum of correlation coefficients and 0.7292 using the sum of scalar products.[13] [14]

- For larger values of the parameter $max_tries$, i.e. $2^{13}$, the distance between the upper bound and the success rate is small for all noise levels. In our experiments performed using the $D_{opt.fun.gauss}$ distinguisher and Algorithm 1 we obtained a maximum distance of 0.0088 between the upper bound and the actual success rate for $\sigma^2 = 18$.
- The use of the distinguisher $D_{opt.fun.gauss}$ instead of the optimal distinguisher $D_{opt.fun.binomial}$ has only a very small impact on the upper bound and on the success rate of the collision side-channel attacks performed using Algorithm 1.
- The Hamming weight of the output of the AES S-box seems to lead only to a bit higher success rates than the average success rate over random leakage function values with binomial distribution. This indicates that the AES S-box Hamming weight leakage can be considered as a kind of typical leakage function for the set of random leakage function values with binomial distribution.
- There exist a gap between success rates of template attacks and the upper bounds for success rate of collision side-channel attacks. This gap cannot be closed.

## 6 Summary

Our results provide new insights on collision side-channel attacks. We derive optimal distinguishers for collision side-channel attacks and a computationally efficient algorithm for the evaluation of these distinguishers. The developed evaluation algorithm can also be applied to Bruneau et al. [4] to make their attack computationally feasible for large values of $L$. The proposed solution offers better results than all previous solutions in terms of success rate. Our approach provides an upper bound for the success rate of collision side-channel attacks.

[13] When testing all elements in $B_{16}$ we obtain respectively success rates 0.7808 and 0.7824.

[14] Wiemers and Klein give in [14] an approximate lower bound value of 1.2 for $\tau = \frac{b-a}{\sigma_c}$ for the variant 2 of their algorithm in the special case of the remaining entropy value of 0. This bound is also valid when the distinguisher $D_{opt.fun.gauss}$ is used. We calculated the means $a$ and $b$ and the variance $\sigma_c^2$ of the scalar products $c_{l_1,l_2}(k^{(l_1)}, k^{(l_2)}) = \sum_{q=0}^{255}(x^{(l_1)}_{q\oplus k^{(l_1)}} \times x^{(l_2)}_{q\oplus k^{(l_2)}})$ for AES-128, Hamming weight leakage and noise variance $\sigma^2$. Using $\delta = k^{(l_1)} \oplus k^{(l_2)}$, $a(\delta) \in [3978, 4192]$ for all $\delta \neq 0$, $b = a(0) = 4608$, $\sigma_c^2 = \sigma^2(2b + 256\sigma^2)$, and $\tau = 1.2$ we obtained for the variance $\sigma^2$ values from 10.2 for $a = 4192$ to 19.4 for $a = 3978$. Already the smaller of these approximate values does not agree with our upper bound.

We show experimentally that we are able to reach this upper bound for the optimal distinguishers. This result demonstrates the optimality of our approach to collision side-channel attacks.

As a future work one may try to look at higher-order success rate of collision side-channel attacks. To improve the post-processing of collision side-channel attacks in that case, it might be worth to describe the problem as a dependent knapsack problem, as it was proposed for divide and conquer strategy [9]. Another direction is to look at collision side-channel attacks for higher-order leakage. The correlation collision side-channel attack exploits only first order leakages.

*Acknowledgments:* The authors thank Wolfgang Thumser, Telekom Security for fruitful discussions on the notion of optimality of collision side-channel attacks.

## References

1. Bogdanov, A.: Improved side-channel collision attacks on AES. In: Adams, C.M., Miri, A., Wiener, M.J. (eds.) Selected Areas in Cryptography, 14th International Workshop, SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers. Lecture Notes in Computer Science, vol. 4876, pp. 84–95. Springer (2007), https://doi.org/10.1007/978-3-540-77360-3_6
2. Bogdanov, A.: Multiple-differential side-channel collision attacks on AES. In: Oswald, E., Rohatgi, P. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5154, pp. 30–44. Springer (2008), https://doi.org/10.1007/978-3-540-85053-3_3
3. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye and Quisquater [8], pp. 16–29, https://doi.org/10.1007/978-3-540-28632-5_2
4. Bruneau, N., Carlet, C., Guilley, S., Heuser, A., Prouff, E., Rioul, O.: Stochastic collision attack. IEEE Trans. Information Forensics and Security 12(9), 2090–2104 (2017), https://doi.org/10.1109/TIFS.2017.2697401
5. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Jr., B.S.K., Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2523, pp. 13–28. Springer (2002), https://doi.org/10.1007/3-540-36400-5_3
6. Durvaux, F., Standaert, F., Veyrat-Charvillon, N.: How to certify the leakage of a chip? In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 459–476. Springer (2014), https://doi.org/10.1007/978-3-642-55220-5_26
7. Gérard, B., Standaert, F.: Unified and optimized linear collision attacks and their application in a non-profiled setting: extended version. J. Cryptographic Engineering 3(1), 45–58 (2013), https://doi.org/10.1007/s13389-013-0051-9
8. Joye, M., Quisquater, J. (eds.): Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings, Lecture Notes in Computer Science, vol. 3156. Springer (2004), https://doi.org/10.1007/b99451

9. Martin, D.P., O'Connell, J.F., Oswald, E., Stam, M.: Counting keys in parallel after a side channel attack. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 313–337. Springer (2015), https://doi.org/10.1007/978-3-662-48800-3_13
10. Moradi, A., Mischke, O., Eisenbarth, T.: Correlation-enhanced power analysis collision attack. In: Mangard, S., Standaert, F. (eds.) Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6225, pp. 125–139. Springer (2010), https://doi.org/10.1007/978-3-642-15031-9_9
11. Schindler, W., Lemke, K., Paar, C.: A stochastic model for differential side channel cryptanalysis. In: Rao, J.R., Sunar, B. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3659, pp. 30–46. Springer (2005), https://doi.org/10.1007/11545262_3
12. Schramm, K., Leander, G., Felke, P., Paar, C.: A collision-attack on AES: combining side channel- and differential-attack. In: Joye and Quisquater [8], pp. 163–175, https://doi.org/10.1007/978-3-540-28632-5_12
13. Standaert, F., Malkin, T., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Joux, A. (ed.) Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5479, pp. 443–461. Springer (2009), https://doi.org/10.1007/978-3-642-01001-9_26
14. Wiemers, A., Klein, D.: Entropy reduction for the correlation-enhanced power analysis collision attack. In: Inomata, A., Yasuda, K. (eds.) Advances in Information and Computer Security - 13th International Workshop on Security, IWSEC 2018, Sendai, Japan, September 3-5, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11049, pp. 51–67. Springer (2018), https://doi.org/10.1007/978-3-319-97916-8_4

## A NP-hardness of optimal collision side-channel attacks

Computing the distinguisher $D_{opt.fun.gauss}$ in an unrestricted setting (i.e. without any assumptions on the leakages $x_q^{(l)}$) is NP-hard. We show this below via a reduction from the MAX-CUT problem. It follows that in this setting also key enumeration and key ranking are NP-hard.

Let $G := ([L], E)$, be a graph, $E \subset \{\{i, j\} | i, j \in [L], i \neq j\}$, $|E| > 1$, then let $n := \lceil log_2(2|E|) \rceil$, $p := 2^n$.

The NP-hard MAX-CUT problem is to find $\text{argmax}_{s \in \{+1,-1\}^L}$ of a cut size $C(s) := \frac{1}{4} \sum_{e \in E} (\sum_{l=1}^{L} b_e^{(l)} s^{(l)})^2$ with $b_e^{(l)} := \delta_{min(e),l} - \delta_{max(e),l}$.

That problem will be reduced in deterministic poly$(L)$ time to finding $\text{argmax}_{k \in (\mathbb{F}_2^n)^L}$ of a key score $S(k) := \sum_{q \in \mathbb{F}_2^n} (\sum_{l=1}^{L} x_{q \oplus k^{(l)}}^{(l)})^2$ with $x_q^{(l)} \in \mathbb{Z}$.

For $f, g, h \in \mathbb{F}_2^n$ let $\chi_f(g) := (-1)^{f \cdot g}$, then $\sum_{h \in \mathbb{F}_2^n} \chi_f(h) \chi_g(h) = p \delta_{f,g}$, and $\chi_f(g \oplus h) = \chi_f(g) \chi_f(h)$.

For $x \in \mathbb{Z}^p$ and $f \in \mathbb{F}_2^n$ let $\hat{x}_f := \sum_{q \in \mathbb{F}_2^n} x_q \chi_f(q)$, then $\sum_{q \in \mathbb{F}_2^n} x_q^2 = \frac{1}{p} \sum_{f \in \mathbb{F}_2^n} \hat{x}_f^2$.

Let $F_1 := \{e_i | i \in [n], i > 1\}$ and $F_{-1} := \left\{ f \middle| f \in \mathbb{F}_2^n, \chi_f(e_1) = -1 \right\}$ where $e_i$ is the $i$-th canonical basis vector of $\mathbb{F}_2^n$, then $F_1 \cap F_2 = \emptyset$ and $|F_{-1}| = p/2 \geq |E|$.

Set $x_q^{(l)} := 4|E| \sum_{f \in F_1} \chi_f(q) + \sum_{e \in E} b_e^{(l)} \chi_{\alpha(e)}(q)$ for all $l \in [L]$ and $q \in \mathbb{F}_2^n$ with an injective map $\alpha \colon E \to F_{-1}$.

Then $S(k) = \frac{1}{p} \sum_{f \in \mathbb{F}_2^n} (\sum_{l=1}^{L} \hat{x}_f^{(l)} \chi_f(k^{(l)}))^2 = S_1(k) + S_2(k)$ with $S_1(k) := p(4|E|)^2$ $\sum_{f \in F_1} (\sum_{l=1}^{L} \chi_f(k^{(l)})^2$ and $S_2(k) := p \sum_{e \in E} (\sum_{l=1}^{L} b_e^{(l)} \chi_{\alpha(e)}(k^{(l)}))^2$.

$S_1(k) = p(4|E|)^2 (n-1) L^2$ exactly if $k^{(i)} \oplus k^{(j)} \in \{0, e_1\}$ for all $i, j \in [L]$, and $S_1(k) \leq p(4|E|)^2 (n-1) L^2 - p(4|E|)^2 (2L-1)$ otherwise.

$S_2(k) \leq 4p|E| < p(4|E|)^2 (2L-1)$, so $k^{(l)} \oplus k^{(1)} \in \{0, e_1\}$ for all $l \in [L]$ if $S(k)$ is maximal.

$S_2(k) = p \sum_{e \in E} (\chi_{\alpha(e)}(k^{(1)}))^2 (\sum_{l=1}^{L} b_e^{(l)} s^{(l)})^2$ with $s^{(l)} := \chi_{e_1}(k^{(l)} \oplus k^{(1)})$ and $k^{(l)} \oplus k^{(1)} \in \{0, e_1\}$.

So, $\text{argmax}_{k \in (\mathbb{F}_2^n)^L} S(k) = p(4|E|)^2 (n-1) L^2 + \text{argmax}_{s \in \{+1,-1\}^L} 4pC(s)$. ■